[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

A Hybrid Deep Learning Approach for Intrusion Detection in IoT Networks

Murat EMEÇ, Mehmet Hilal ÖZCANHAN
Department of Computer Engineering, Dokuz Eylul University, Izmir, 35001, Turkey
murat.emec@deu.edu.tr

***Abstract*—Internet of Things (IoT) devices have flocked the whole world through the Internet. With increasing mission-critical IoT data traffic, attacks on IoT networks have also increased. Many newly crafted attacks on IoT communication require equally intelligent intrusion detection methods to form the first step of countering the attacks. Our work contributes to intrusion detection in IoT networks, by putting state-of-the-art Deep learning methods into service. A BLSTM-GRU Hybrid (BGH) model has been designed to detect eight known IoT network attacks, based on two well-accepted CIC-IDS-2018 and BoT-IoT IoT network traffic datasets. The results of our BGH model in IoT network traffic intrusion detection have been auspicious. The accuracies of prediction on the two datasets are 98.78% and 99.99%. The f1-scores are 98.64% and 99.99%, respectively. The comparison of our results with similar previous studies showed that our BGH model has the best performance ratio (time/accuracy, time/f1-score), where time is the training time of the model. The performance of our proposed model is proof that hybrid Deep Learning methods can prove to be an innovative perspective on Intrusion Detection in IoT networks.**

***Index Terms*—hybrid intelligent systems, Internet of Things, intrusion detection, learning systems, prediction methods.**

I. INTRODUCTION

The Internet of Things (IoT) provides a smart network system between addressable machines to exchange data and interact with each other through various communication protocols. The smart systems include network infrastructure, sensors, software, control panels, servers, and more [1]. IoT technology makes passive objects smart, forming a digital interlink layer for everyday real-life problems. IoT is now actively used in smart homes and offices [2], online educations, economics, marketing, smart cities, and many other ubiquitous global systems [3-4]. By 2025, it is estimated that about 75 billion IoT devices will be connected to the Internet globally [5], and the number is predicted to reach one trillion by 2035 [6].

Given the increasing number of Internet-connected devices, it is natural to expect large amounts of data to be transmitted on the network, producing information-rich network traffic. Of course, some of the transmitted information is both confidential and critical. Experience shows that sensitive information inevitably attracts the attention of cybercriminals. Any intrusion (capture or distortion) in the shared data can be very risky for its owners. Therefore, secrecy, integrity, privacy, and usability should be undoubtedly in place. In other words, eliminating security threats is essential for IoT device manufacturers and users. However, due to their low resource capacity and variety, it is not easy to design a single security solution for every IoT device on the network. Therefore, IoT devices are increasingly targeted by cybercriminals, or "hackers". Hackers use and design software and hardware to launch attacks on the IoT systems by placing malicious code, installing viruses, and ultimately infiltrating the IoT network [7]. Hackers have reportedly created unique strains of malware that can circumvent security measures and disrupt large parts of the Internet. In 2016, the Mirai Botnet brought down the Internet in the first reported wave of IoT attacks [8]. The malware embedded into the IoT infrastructure transformed devices such as gateways, routers, and Internet Protocol (IP) cameras into botnets (network of hijacked computerized devices). The centrally controlled IoT botnets flooded a Domain Name Services (DNS) provider and caused a disruptive bottleneck disrupting Internet access of millions of worldwide users. Events show that problems caused by IoT network attacks are real, increasing, becoming more complex, and need to be countered by equally clever measures.

This study proposes IoT security based on anomaly detection in IoT network traffic, through a binary and multi-label classification Deep Learning approach. The CIC-IDS-2018 [9] and BoT-IoT [10] datasets containing normal and attack network traffic data are trained and then tested. Our BGH Deep Learning approach outperforms previous works, especially in large data sets [11]. Our solution is non-invasive and allows IoT-based devices and applications to interact without intervention [12].

The remainder of this paper is organized as follows: Section 2 presents related works of studies in the related field. In Section 3, we explain our materials and methodology. In Section 4, our proposed model is presented. In Section 5, we present the performance evaluation of our model and its comparison with previous models. Finally, Section 6 concludes and describes future work.

II. RELATED WORK

The attacks on the IoT systems are numerous and diverse. In November 2016, a German researcher discovered a vulnerability in a sanitizing device for medical surgical instruments. The vulnerability allowed remote attackers to access the file directories of the device. Thus, highly sensitive health-related data was put at risk. For three months, authorities were open to information theft. Later, the IoT attacks between October 2019 and June 2020 were higher than the combined attacks of the previous two years [13].

The number of attacks exploded during the COVID-19

Digital Object Identifier 10.4316/AECE.2022.01001

1582-7445 © 2022 AECE

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

coronavirus outbreak. A leading cyber security firm's July 2020 report revealed a significant increase in various attacks on IoT devices and mobile devices. According to the report, 375 cyber threats per minute were detected by Intrusion Detection Systems (IDS) in the first quarter of 2020, not counting the undetected [14]. Intrusion is an 'Unauthorized Entry' achieved by transmitting malicious packets to steal or modify important information. IDS are critical in detecting intrusions and work as a second line of defense after the firewall [15]. IDS monitors the network traffic for suspicious activity or an anomaly. An anomaly is an outlier and happens when network traffic deviates from expected behavior [16]. Anomalies are detected by comparing the present network traffic with the previously known normal traffic. Unfortunately, the reports show that traditional IDS are insufficient for today's new attacks.

The IoT communication traffic resembles computer network traffic, except for some critical differences. Although classifying IoT traffic flow as either normal or abnormal is similar to computer-traffic binary classification, traditional works miss the differences in the traffic data. Yet, most experts apply the same classification techniques to IoT traffic, as well. As a result, many researchers have attempted to identify IoT attacks using the same Machine Learning (ML) and Deep Learning methods used for computer network traffic analysis. Our related literature review focused on computer-network traffic anomaly detection using Machine Learning [15-17] and Deep Learning methods [18-23]. The Machine Learning algorithms classify existing traffic data as normal flow or anomaly, by learning a dataset using the classification methods [17]. Recently, Deep Learning methods have also been used to analyze IoT network traffic. Deep Learning related works use only Bidirectional Long Short Term Memory (BLSTM) [18], only Gated Recurrent Unit (GRU) [20], and hybrid models [23] in IDS. Hybrid models have also been used in other research areas [34-35]. For example, work [34] uses a hybrid BLSTM-GRU model for monthly rainfall prediction. However, the studies [15], [21-22], [27-33] that used the CSE-CIC-IDS2018 network traffic dataset and the studies [36-39] that used the IoT-specific BoT-IoT data were examined thoroughly. From the above-related literature review, it is evident that Deep Learning models have not been used extensively in IoT network attack detection, yet.

A. Motivation

Literature shows that Machine Learning and, lately, Deep Learning methods have great prediction potential in many intensive data-related research. Our motivation for analyzing the IoT traffic using Deep Learning models arises from:

- Lack of research on intrusion detection in IoT networks, based on hybrid Deep Learning methods;
- The decreasing success of traditional security solutions in protecting the increasing IoT network traffic;
- The success of Recurrent Neural Networks (RNN) in diverse flow types research;
- Although complex, the efficient classification capability of RNN architectures in data-intensive solutions;
- Increasing success stories of hybrid Deep Learning models in different big data analyses.

B. Contribution

Our main goal is to make IoT data transmission safer through intrusion detection. To achieve our goal, we contribute the following:

- Detection of attacks on IoT network traffic by analyzing different IoT network traffic datasets using novel hybrid Deep Learning models;
- Perfection of IoT traffic feature selection through comparison of seven feature selector methods;
- More efficient detection of intrusion in IoT network traffic than previous methods;
- Performance comparison of previous various models with our proposed model;
- Higher detection accuracy of rarely launched attacks.

III. MATERIALS AND METHODS

This section explains the materials (datasets), hardware and software tools, and the applied classification methods used in our proposed hybrid Deep Learning model.

A. Materials

1) Description of Datasets Used in Our Work

The first dataset chosen in our work was the CIC-IDS2018 dataset, as it is one of the latest network datasets containing the newest known attack types. The dataset was created by collecting traffic data on the Amazon AWS LAN network by the Canadian Cyber Security Institute (CIC) and the Communications Security Authority (CSE) [9]. BruteForce (Web, XSS, FTP, SSH), Botnet, DoS (Hulk, SlowHTTPTest, GoldenEye, Slowloris), DDoS (HOIC, LOIC-UDP, LOIC-HTTP), Web SQL Injection and Infiltration to Network attacks are present in the dataset. There are 6,546,654 records of normal network flow data and 2,746,934 records of six known attacks with fourteen subcategories. The packets obtained using the CICFlowMeter-V3 [25] are transformed into network traffic flows with 80 features. Fig. 1 shows the distribution of the network traffic and the types of attacks.

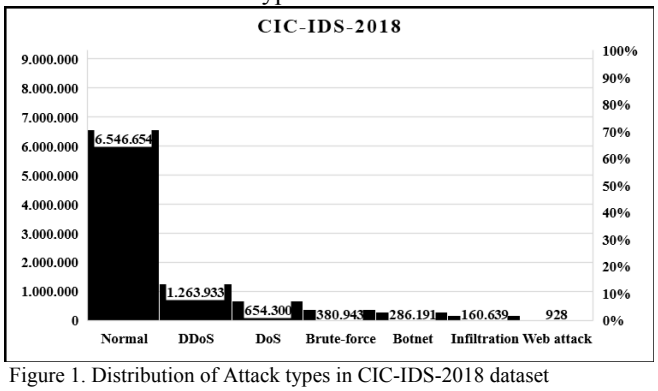

Figure 1. Distribution of Attack types in CIC-IDS-2018 dataset

From Fig. 1, it is evident that the distribution of the attack types is imbalanced, in the dataset. Unlike other types of attacks, the infiltration attack within the dataset is very similar to normal traffic, as it follows an infiltration path into the network. It is mentioned in the literature that most of the infiltration attacks cannot be classified into a specific category. Therefore, distinguishing infiltration attacks is reported as difficult for neural networks [21], [26].

The second dataset used in our model is the latest Bot-IoT dataset created using a real IoT systems network

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

environment, by the Cyber Range Lab of the UNSW Canberra Cyber Centre [10]. The Bot-IoT dataset contains more than 72 million records, including DDoS, DoS, OS and Service Scan, Keylogging, and Data theft attacks [36]. The authors share only 5% (approximately 3.6 million records) of their total data. The dataset contains the lightweight Message Queue Telemetry Transport (MQTT) communication protocol used in machine-to-machine (M2M) communications. Similar to work [10] strategy, we also used a "Full-feature" (all 43 features) and "Best-10 features" analyses technique for fair comparison of our works. Fig. 2 shows the distribution of normal network traffic and types of attacks in the BoT-IoT dataset.

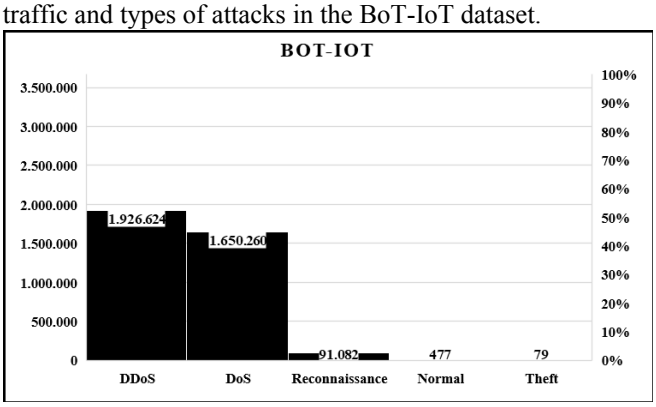

Figure 2. Distribution of Attack types in BoT-IoT dataset

As shown in Fig. 2, the BoT-IoT dataset is also prone to imbalanced distribution. A more detailed search of the CIC-IDS-2018 and BoT-IoT dataset literature with the keyword "imbalance" resulted in 70 and 83 articles, respectively. The applied date range for the literature search is from January 2018 to July 2021. The results prove the imbalanced distribution in the datasets. In works [40] and [44], the authors explain the accuracy results of the proposed Machine Learning techniques, by referring to the imbalanced distribution of the data. Fortunately, problems of imbalanced data distribution can be resolved by pre-processing data, applying feature selection, and eliminating missing or corrupted records, in Deep Learning.

2) Tools Used in Our Deep Learning Analyses

The following hardware and the latest versions of software tools have been used in our analyses:

- Server with Intel Xeon E5-2124 CPU, four cores.
- NVidia Quadra Graphic accelerator card P620 GPU.
- 32 GB main memory.
- Windows 10™ operating system.
- Anaconda platform (open access).
- Python programming language.
- Pandas library.
- NumPy library.
- Keras library.
- TensorFlow library.
- Scikit-learn (Sklearn) library.

B. Methods

Our approach consists of binary and multi-labeling classifications using a hybrid Deep Learning model. The sections below present the data pre-processing and feature selection steps of the binary and multi-label classification and the classification models.

1) Data Pre-processing

Data pre-processing is the first preparation phase before classification. After the dataset is stored in the computer memory, it is checked for missing, misprinted, or out-of-limit values. There are no missing, misprinted, or out-of-limit values in either of the datasets. Therefore, no pre-conditioning was necessary. However, since our tool Phyton does not support heterogeneous data types, the non-numeric entries have been converted to numeric values. The next phase is data normalization. As in previous works, a standard scaler normalization method has been used to put the data into the [0, 1] range [20], [35]. Normalization is followed by converting the features and labels into a TensorFlow data structure. In the last step, input and output features are determined, and the data set is separated into training and test subsets. Briefly, our data pre-processing consisted of:

- Storing and checking the dataset in computer memory;
- Converting nominal data including IPv4 and IPv6 addresses into numeric data;
- Normalizing data using the standard scaler;
- For CIC-IDS-2018 dataset, randomly dividing data into:
  - Training set, 7.434.870 records (%80)
  - Testing set, 1.858.718 records (%20)
- For BoT-IoT dataset, randomly dividing data into:
  - Training set, 2.934.818 records (%80)
  - Testing set, 733.704 records (%20)

2) Feature Selection

After data pre-processing, feature selection is applied for increasing prediction accuracy and decreasing model training time. The CIC-IDS-2018 dataset contains 79 features. In our work, seven commonly used feature selection methods (XGBoost, Decision Tree, F_classif, CHI2, PCA, Extra tree, and Correlation matrix) were applied to select the Best-20 features. The determined 20 features of each method were compared and only the best 10 features (init_fwd_win_byts, dst_port, fwd_seg_size_min, flow_iat_min, flow_duration, fwd_pkts_s, bwd_pkts_s, fwd_act_data_pkts, bwd_pkt_len_std, totlen_fwd_pkts) common to XGBoost and Decision Tree were selected. Hence, the features used in our classification models were determined. Table I shows the significance-level scores of the Best-10 features selected.

TABLE I. XGBOOST AND DECISION TREE TOP 10 FEATURE SCORES FOR CIC-IDS-2018 DATASET

Features	XGBoost Score	Decision Tree Score
init_fwd_win_byts(1)	722	3094
dst_port(2)	290	2183
fwd_seg_size_min(3)	141	937
flow_iat_min(4)	116	194
flow_duration(5)	83	43
fwd_pkts_s(6)	60	164
bwd_pkts_s(7)	59	577
fwd_act_data_pkts(8)	50	11
bwd_pkt_len_std(9)	40	26
totlen_fwd_pkts(10)	33	1377

The next step covers the feature selection of the BoT-IoT dataset. We used the same Full-feature and Best-10 features approach to equate our analysis with work [36]. Features like packet sequence ID (pkSeqID) unrelated to attacks are

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

disregarded. Then, attack-related features are matched according to significance scores, as in the previous step. The Best-10 features obtained by using XGBoost and Decision Tree are shown in Table II. For example, the highest significance scorer 'TnP_PerProto' feature in XGBoost, subscripted as (1), matches the ninth significant feature (score 52), in the Decision Tree method. Thus, TnP_PerProto feature is selected as the first Best-10 feature. The rest of the highest significance scorers are listed in order, in the table.

TABLE II. XGBOOST AND DECISION TREE TOP 10 FEATURE SCORES FOR BOT-IOT DATASET

Features	XGBoost Score	Decision Tree Score
TnP_PerProto(1)	32	52
ltime(2)	22	660
daddr(3)	20	69
dbytes(4)	20	257
proto(5)	17	904
dport(6)	16	204
rate(7)	15	102
AR_P_Proto_P_Sport(8)	12	27
proto_number(9)	9	44
N_IN_Conn_P_DstIP(10)	5	309

3) Binary Classification

In the first phase, the BLSTM, GRU, and our BGH model have been trained for binary classification to distinguish between normal traffic flow and attacks.

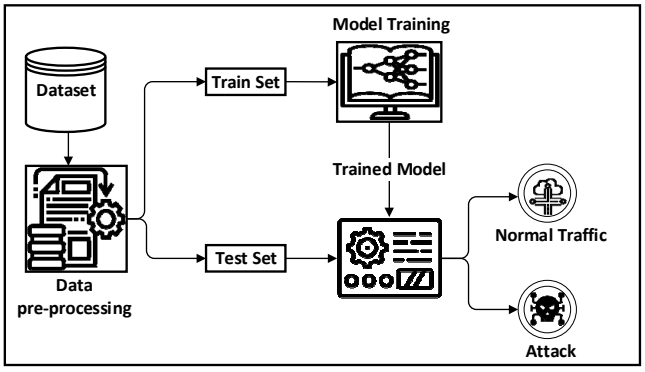

Figure 3. Binary classification architecture

The method architecture is shown in Fig. 3. Based on the architecture, experiments were performed using different hyperparameters (learning rate, epoch, and number of hidden layers). The hyperparameters have been adjusted for best results, as explained in Section IV.C.

4) Multi-label Classification

In the second method, BLSTM, GRU, and our BGH model are trained by multi-labeling both datasets. The method architecture is shown in Fig. 4.

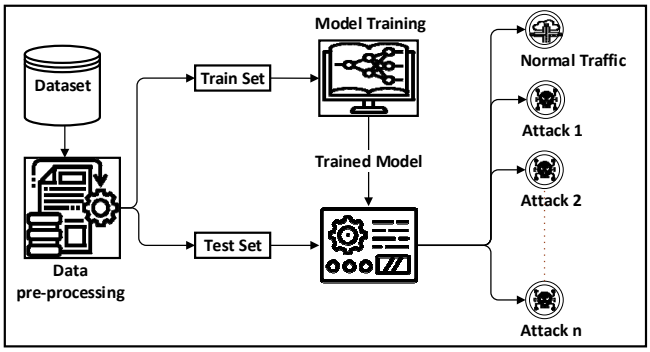

Figure 4. Multi-labeling classification architecture

After the data pre-processing and feature selection steps, each dataset model is individually trained and tested. The multi-labeling method distinguishes normal traffic and different attack types. The different classification algorithms used in the multi-labeling architecture of Fig. 4 are BLSTM, GRU, and our hybrid model BGH. The models based on the algorithms are explained in detail, in the next section.

IV. SINGULAR AND OUR PROPOSED BGH MODELS

A. Only BLSTM Model

LSTM is one of the most popular models in time series analysis because it gives access to long-term context by using three gates in, out, and forget. LSTM has many variants, such as one-way LSTM, BLSTM etc. [41]. The BLSTM model processes the sequential data forward and backward with two separate hidden layers (Fig. 5) to capture past and future information, respectively. The success of bidirectional RNN over unidirectional networks is the reason for their use in multi-labeling classifications [42].

B. Only GRU Model

GRU's natural language processing, speech signal, and music modeling performances are similar to the LSTM model. GRU outperforms LSTM, when dealing with small datasets due to its fewer gates. To solve the vanishing gradient problem of standard RNN, the GRU consists of an update and reset gate, but unlike LSTM it lacks an output gate. Therefore, a GRU needs fewer input parameters and less training time than LSTM. Thus, naturally GRU has the capacity to have better time performance in IDS. Strikingly, GRU was used in the second stage of a hybrid classification architecture in work [43].

C. Our Proposed Hybrid Model BGH

Before deciding on model architecture, Deep Learning models in the attack detection literature were examined in detail. Our choice of a two-stage hybrid model is based on the superior performance results of previous hybrid models [23], [34-35], compared to single algorithm models [18], [20]. BGH consists of seven layers, as shown in Fig. 5: Input, BLSTM, GRU, Normalization, Dense, and Output layers. Linearly connected BLSTM units are used for feature extraction. Similarly, linear connected GRU units were used for classification. Linear units have been defined by the "rectified linear unit (ReLU) activation" in the Keras library. All of the BLSTM and GRU layer neurons are connected to the neurons of the following Dropout layers. First Dropout and Batch Normalization combine the BLSTM and GRU layers. The second Dropout and Batch Normalization layer combines the GRU classification layer to the Dense layer. Dense layer is used in the final stage to change the preceding layer's output dimensionality and define the relationship between the data values worked on by the model. Hence, the Dense layer decides whether the network flow is normal or an attack.

Briefly, the stages of our proposed BGH model are:

1. 79 inputs in the CIC-IDS-2018 dataset and 43 in the BoT-IoT dataset at the input layer (X_0, .. X_n);
2. Feature selection at the BLSTM layer;
3. Feature selection and classification layer combining by the first dropout and batch normalization layer;
4. Classification at the GRU layer;

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

5. Classification and Dense layer combining by the second dropout and batch normalization layer;
6. Model decision forwarding by the Dense Layer to the output layer (Y_0, .. Y_n).

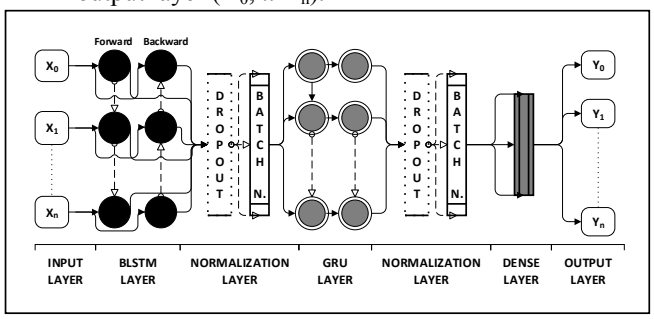

Figure 5. Proposed model architecture

In binary classification, both datasets have two output classes. In multi-label classification, there are seven output classes for the CIC-IDS-2018 dataset and five for the BoT-IoT dataset.

In the BGH model, hyperparameter tuning was accomplished through a random grid search and programmer's heuristics, as well as previous experiments and literature reports [20], [38].

Our best performing hyperparameters are as follows:

- The number of BLSTM hidden layers: 1
- The number of neurons in each BLSTM layer: 64
- The number of GRU hidden layers: 1
- The number of neurons in each GRU layer: 64
- The number of neurons in each Dense layer: 32
- Dropout rate: 0.1
- Learning rate: 0.001
- Batch size: 64
- Epoch size: 10
- Optimizer: Adam
- Cost function: cross-entropy

V. Results and Comparison of Previous Work

In this section, the metrics used for measuring the proposed models' performance will be first defined. Secondly, the results of our binary classification and multi-label classification models will be presented.

Then, the performances of the models will be compared. Finally, our results will be compared to previous IoT intrusion detection works using different methods.

A. Evaluation metrics

Four popular metrics have been used to evaluate the performances, as in previous works [38]. These are accuracy (1), precision (2), recall (3), and f1-score (4). The statistical parameters used in the performance calculations are:

- True Positive (TP) - the number of attacks correctly classified;
- True Negative (TN) - the number of normal traffic correctly classified;
- False Positive (FP) - the number of attacks wrongly classified as normal;
- False Negative (FN) - the number of normal traffic wrongly classified as attacks;

The following equations are used in calculating the performance metrics:

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$f1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

Another crucial metric is the time taken by a method to run on a computer. In our work, the measured metric is the time to train a model, called "time" for simplicity [21-22], [29], [36-39]. Once time is considered, the significance of scores alone diminishes because the time to score ratio can give a better idea about the efficiency of a proposed model [29]. Therefore, we defined two new metrics, "Efficiency1" as time divided by accuracy and "Efficiency2" as time divided by the f1-score. Hence in comparisons, the best performer method can be identified by the smallest Efficiency1 or Efficiency2 value.

B. Results of binary classification

The first classification method is binary classification. The results of testing the models using the CIC-IDS-2018 dataset are given in Table III. The table is divided into two sections according to the number of features used, while training the model. The upper sections show the results when Full-feature (all 79 features) in the dataset are used, and the lower section shows when only the Best-10 features are used (shown in Table I).

Table III shows the results of only BLSTM, only GRU, and our BGH model. In the Full-feature test, the best accuracy and f1-score are achieved by our BGH model at 98.12% and 98.10%, respectively. By examining the Efficiency1 metric, it is evident that the best efficiency performance is again obtained in our model with a 3.62 value. In the Best-10 features test, the performance superiority of Our BGH model becomes more evident with 98.67% accuracy and 98.66% f1-score. The Efficiency1 value of our design is the best with 3.00, compared to the Efficiency1 values of the other models.

Table IV shows the results obtained when using the Bot-IoT dataset. However, the Best-10 features method was obtained from work [36] for comparison equality. The table shows that our BGH model outperforms the singular models as in the first dataset, with 1.22 and 1.02 Efficiency1 values.

C. Results of multi-label classification

The second Deep Learning method used is multi-label classification. The results of testing the three models with the CIC-IDS-2018 dataset have been divided into two tables. Table V shows the results when Full-feature (all 79 features) are used.

The results show that our BGH model achieves the best accuracy and f1-score with 97.11656% and 96.29762%, respectively. When Efficiency1 is calculated, it is observed that our model has the lowest value 3.71. But, it is worth paying attention to a significant result in Table V.

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

TABLE III. CIC-IDS-2018 TESTING DATA MODEL SCORES FOR BINARY CLASSIFICATION

Type	Model	Category	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)	Efficiency1
Full-features	BLSTM	Model	98.12139	98.15290	98.12139	98.10565	429	4.37
		Normal	97.39988	97.56860	99.81792	98.68044		
		Attack	94.01838	99.54083	94.07176	96.72905		
	GRU	Model	98.11348	98.14396	98.11348	98.09784	378	3.85
		Normal	97.39177	97.58069	99.79616	98.67599		
		Attack	94.00036	99.48639	94.10325	96.71998		
	Our BGH	Model	98.11951	98.15153	98.11951	98.10366	356	3.62
		Normal	97.40242	97.57605	99.81265	98.68168		
		Attack	94.02401	99.52770	94.09069	96.73286		
Best-10 features	BLSTM	Model	98.53371	98.53333	98.53371	98.52932	328	3.32
		Normal	97.96403	98.55239	99.38151	98.96521		
		Attack	95.25680	98.48766	96.50248	97.48496		
	GRU	Model	98.58493	98.58351	98.58493	98.58158	324	3.28
		Normal	98.03375	98.67225	99.33093	99.00049		
		Attack	95.41489	98.37092	96.79758	97.57791		
	Our BGH	Model	98.67445	98.67798	98.67445	98.66936	296	3.00
		Normal	98.15585	98.54051	99.59633	99.06561		
		Attack	95.69247	99.00735	96.46573	97.72002		

TABLE IV. BOT-IOT TESTING DATA MODEL SCORES FOR BINARY CLASSIFICATION

Type	Model	Category	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)	Efficiency1
Full-feature	BLSTM	Model	99.99945	99.99945	99.99945	99.99945	173	1.73
		Normal	95.95960	97.89474	97.89474	97.89474		
		Attack	99.99945	99.99973	99.99973	99.99973		
	GRU	Model	99.99863	99.99862	99.99863	99.99862	169	1.69
		Normal	90.47619	95.69892	93.68421	94.68085		
		Attack	99.99864	99.99918	99.99945	99.99932		
	Our BGH	Model	99.99965	99.99966	99.99965	99.99965	122	1.22
		Normal	95.95960	96.90722	98.94737	97.91667		
		Attack	99.99945	99.99986	99.99959	99.99973		
Best-10 features	BLSTM	Model	99.99809	99.99807	99.99809	99.99802	147	1.47
		Normal	87.15596	98.79518	86.31579	92.13483		
		Attack	99.99809	99.99823	99.99986	99.99905		
	GRU	Model	99.99809	99.99809	99.99809	99.99801	132	1.32
		Normal	87.15596	99.99999	85.26316	92.04545		
		Attack	99.99809	99.99809	99.99999	99.99905		
	Our BGH	Model	99.99918	99.99923	99.99918	99.99919	102	1.02
		Normal	94.05941	94.05941	99.99999	96.93878		
		Attack	99.99918	99.99999	99.99918	99.99959		

TABLE V. MULTI-LABEL CLASSIFICATION SCORES FOR MODELS USING FULL-FEATURES CIC-IDS-2018 DATASET

Model	Category	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)	Efficiency1
BLSTM	Model	97.08992	96.39641	97.08992	96.27774	419	4.35
	Botnet	96.74464	97.16074	99.54401	98.33794		
	Brute-force	80.02794	83.13916	94.13432	88.29576		
	DDos-attack	99.79078	99.88451	99.90585	99.89518		
	Dos-attack	87.26560	96.23926	88.88048	92.41361		
	Infiltration	50.01946	52.41779	00.84350	01.66028		
	Normal	94.80469	96.20949	98.39666	97.29079		
	Web-attack	63.13993	99.99999	41.62162	58.77863		
GRU	Model	97.08503	97.18401	97.08503	96.24368	412	4.27
	Botnet	96.66948	96.91511	99.72920	98.30202		
	Brute-force	92.10246	98.09700	93.23393	95.60366		
	DDos-attack	97.67092	97.78836	99.87420	98.82027		
	Dos-attack	90.64461	99.97785	89.69891	94.55987		
	Infiltration	50.00078	99.99999	00.00311	00.00622		
	Normal	95.61786	96.21806	99.32093	97.74488		
	Web-attack	60.26059	97.01493	35.13514	51.58730		

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

Our BGH	Model	97.11656	96.40489	97.11656	96.29762	357	3.71
	Botnet	96.74955	97.52719	99.15441	98.33407		
	Brute-force	80.62584	83.61774	94.48083	88.71799		
	DDos-attack	99.72385	99.80323	99.92009	99.86162		
	Dos-attack	87.43644	96.13488	89.21825	92.54751		
	Infiltration	50.00934	51.57895	00.61006	01.20586		
	Normal	94.85936	96.24148	98.42456	97.32078		
	Web-attack	64.01384	99.99999	43.78378	60.90226		

The difficulty of detecting infiltration attacks in the CIC-IDS-2018 dataset is mentioned in work [26]. Table V proves this assertion by reporting the recall and f1-scores of the infiltration attacks as approximately 1.00%. The observation verifies the difficulty of detecting infiltration attacks using the Full-feature method. Another observation of Table V is the success of our hybrid model in detecting web-attacks with a 61.00% f1-score, even though web-attacks constitute only 0.01% of all data (Fig. 1).

TABLE VI. MULTI-LABEL CLASSIFICATION SCORES FOR MODELS USING BEST-10 FEATURE CIC-IDS-2018 DATASET

Model	**Category**	**Accuracy (%)**	**F1-score (%)**	**Time (s)**	**Eff.1**
BLSTM	Model	98.50789	98.36837	405	4.13
	Botnet	99.77861	99.88903		
	Brute-force	99.98688	99.99344		
	DDos-attack	99.96757	99.98378		
	Dos-attack	99.90762	99.95379		
	Infiltration	54.16430	49.52206		
	Normal	97.92840	98.94650		
	Web-attack	62.62626	59.04059		
GRU	Model	98.42554	97.83961	408	4.14
	Botnet	99.80297	99.90124		
	Brute-force	99.98425	99.99213		
	DDos-attack	99.92055	99.96023		
	Dos-attack	99.91525	99.95760		
	Infiltration	52.93377	21.40342		
	Normal	97.81719	98.89615		
	Web-attack	95.69247	38.55422		
Our BGH	Model	98.78214	98.64596	345	3.49
	Botnet	99.82734	99.91347		
	Brute-force	99.98819	99.99409		
	DDos-attack	99.93003	99.96498		
	Dos-attack	99.94730	99.97364		
	Infiltration	59.25824	57.67293		
	Normal	98.30408	99.14147		
	Web-attack	64.13793	64.13793		

The results of testing the three models using the Best-10 features are shown in Table VI. Similar to the Full-feature test, the best accuracy and f1-scores are achieved by our BGH model at 98.78214% and 98.64596%, respectively. Examining the Efficiency1 results reveals that the best efficiency performance is obtained in our model, with the lowest 3.49 value.

However, infiltration attack detection has gone up to 57.67293% f1-score, from that of 01.20586% in Table V. This proves the success of feature selection in Deep Learning analysis to a great extent. Contrarily, the detection of web-attacks has deteriorated. This outcome can be the negative effect of feature reduction on distinguishing the low presence of web-attack data (Fig. 1: 0.01%), in the dataset. The model simply cannot learn the web-attack with the redacted feature set. A similar account is given in previous works [17], [21], [26], [28].

The results of multi-label classification obtained using the Bot-IoT dataset are compiled in Table VII. However, the Best-10 features were not obtained using our extraction method for comparison equality, but they were derived from work [36]. Examining the table shows that our BGH model outperforms the singular models again with 1.35 and 1.19 Efficiency1 values.

TABLE VII. BOT-IOT TESTING DATA MODEL SCORES FOR MULTI-LABEL CLASSIFICATION

Type	**Model**	**Category**	**Accuracy (%)**	**F1-score (%)**	**Time (s)**	**Eff. 1**
Full-feature	BLSTM	Model	99.99938	99.99935	161	1.61
		DDOS	99.99983	99.99992		
		DoS	99.99667	99.99833		
		Normal	95.95960	97.87234		
		Recon.	99.97805	99.98902		
		Theft	80.00000	86.66667		
	GRU	Model	99.99931	99.99930	153	1.53
		DDOS	99.99844	99.99922		
		DoS	99.99727	99.99864		
		Normal	94.05941	96.77419		
		Recon.	99.97256	99.98628		
		Theft	76.19048	82.75862		
	Our BGH	Model	99.99972	99.99972	135	1.35
		DDOS	99.99922	99.99961		
		DoS	99.99818	99.99909		
		Normal	96.93878	98.41270		
		Recon.	99.98353	99.99177		
		Theft	84.21053	90.32258		
10-best features	BLSTM	Model	99.98950	99.98944	152	1.52
		DDOS	99.99870	99.99935		
		DoS	99.98031	99.99015		
		Normal	87.15596	92.13483		
		Recon.	99.98902	99.99451		
		Theft	88.88889	93.33333		
	GRU	Model	99.99672	99.99664	146	1.46
		DDOS	99.99844	99.99922		
		DoS	99.99697	99.99849		
		Normal	85.58559	90.90909		
		Recon.	99.97805	99.98902		
		Theft	84.21053	89.65517		
	Our BGH	Model	99.99754	99.99752	119	1.19
		DDOS	99.99922	99.99961		
		DoS	99.99697	99.99849		
		Normal	96.93878	98.41270		
		Recon.	99.98353	99.99177		
		Theft	94.11765	96.77419		

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

To find the combined performances of the models, a study has been carried out on the combination of multiple performances. Work [24] proposes a multiplicative model for education, experience and perceptual skills to decide the joint contributions to the performances. We also adopted the multiplicative model, respecting the well-defended approach, in that work. By multiplying the efficiency of each model for each feature choice, the Efficiency1 Products were obtained. Table VIII summarizes the Efficiency1 Product results obtained in the classifications for the two datasets used. The table shows that our BGH model has the best efficiency in every CIC-IDS-2018 dataset classification. As a result, the Efficiency1 Product of our BGH model is also the best, at a value of 140.61.

TABLE VIII. THE TOTAL EFFICIENCY SCORE OF OUR STUDY

Dataset	Model	Full-feature		Best-10		Efficiency1 Product
		Binary	Multi-label	Binary	Multi-label	
CIC-IDS-2018	BLSTM	4.37	4.35	3.32	4.13	260.65
	GRU	3.85	4.27	3.28	4.14	223.24
	Our BGH	3.62	3.71	3.00	3.49	140.61
BoT-IoT	BLSTM	1.73	1.61	1.47	1.52	6.22
	GRU	1.69	1.53	1.32	1.46	4.98
	Our BGH	1.22	1.35	1.02	1.19	2.00

In fact, BGH's efficiency is 58.75% better than the closest competitor GRU efficiency (223.24). In the BoT-IoT dataset classification, our proposed model's efficiencies were again the best, in every test. Naturally, BGH's Efficiency1 Product was also the best at a value of 2.00. In fact, the BGH's Efficiency1 Product is 149.26% better than the closest competitor GRU (4.98).

The overall result is that our proposed model achieves intrusion detection with the best time-score ratio, in both datasets. Therefore, our proposed method can be declared the winner of IoT Intrusion Detection among the three methods. This assertion is further verified by the result of the multiplication of Efficiency1 Products of the two datasets 140.61×2.00, which is the best product of Efficiency1 Products.

D. Comparison of Our BGH Model Results with Previous Works

This section compares our BGH model with previous IoT Intrusion Detection works using Deep Learning methods. Accuracy, precision, recall, f1-score results, and time to train the proposed model have been used in the comparison. The training time has been included because it is accepted as a performance metric in work [29]. Two new time-dependent performance indicators have been defined, similar to the argument in the previous section. The training time has been divided by the accuracy and the f1-scores. The new performance indicators have been named as Efficiency1 and Efficiency2, respectively. Table IX summarizes the performances of the models for the CIC-IDS-2018 dataset. Works that did not report any timing have been omitted from the Efficiency1 and Efficiency2 comparisons.

TABLE IX. CIC-IDS-2018: EFFICIENCY AND TEST SCORES OF THE PROPOSED MODELS

Authors	Proposed model	Accuracy (%)	F1-Score (%)	Time (s)	Efficiency1	Efficiency2
Kim et al. [15]	CNN	91.50	84.00	n/a	-	-
Lin et al. [21]	LSTM	96.20	93.00	500	5.20	5.38
Ferrag et. al. [22]	RNN, Deep AE	97.38	n/a	390	4.00	-
Zhao et al. [27]	Deep AE	97.90	97.90	n/a	-	-
Li et al. [28]	Deep AE	n/a	n/a	n/a	-	-
Gamage et al. [29]	Deep NN	98.40	97.82	630	6.40	6.44
Catillo et al. [30]	Deep AE	99.20	n/a	n/a	-	-
Farhan et al. [31]	DNN	90.25	66.00	n/a	-	-
Nwakanma et. al. [32]	ANN	76.47	72.00	n/a	-	-
Wanjau et al. [33]	CNN	94.30	91.80	n/a	-	-
Our Model	BGH	98.78	98.64	296	3.00	3.00

Comparing the Efficiency1 performances, the best score belongs to our proposed BGH model with a 3.00 value. The nearest Efficiency1 performance is 4.00 [22].

The comparison indicates 25% performance superiority of our BGH model from its nearest contender. Examining Efficiency2 reveals that our proposed model's performance is 79.3% better than its nearest contender, with a 3.00 value.

Table X shows the comparison for the models using the BoT-IoT dataset. Since the previous works supplied only their accuracy values, only Efficiency1 is present in the table. In brief, the Efficiency1 of our proposed BGH model is 1.02.

Among the declared performances, the nearest Efficiency1 performance is 2.05. The comparison indicates that our BGH model is 102.00% more efficient than its closest contender. As a conclusion of comparisons, the BGH model is the clear winner of the compared models.

TABLE X. BOT-IOT: EFFICIENCY AND TEST SCORES OF THE PROPOSED MODELS

Author(s)	Proposed model	Acc. (%)	F1-Score (%)	Time (s)	Eff.1
Koroniotis et al. [36]	SVM	99.98	-	663.69	6.63
Idriss et al. [37]	CNN	99.94	-	1395.0	13.96
Ge et al. [38]	FNN	82.00	-	346.0	4.22
Ferrag et al. [39]	BPTT	98.20	-	201.29	2.05
Our Model	BGH	99.99	99.99	102.00	1.02

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

E. The Strong and Weak Points of Our Proposed Model

The strengths of our proposed model compared to other architectures are the best efficiency performances (Tables IX and X) and the ability to detect rare attacks better than others (Table VI). The weaknesses of our proposed model are our supervised learning method and the inability to retrain a network flow for correcting failed classifications. Therefore, our work cannot be compared with unsupervised learning [23] or re-trainable methods [18].

VI. CONCLUSION

Attacks on IoT devices are well known. This study proposes a BLSTM-GRU Hybrid (BGH) Deep Learning model for attack detection in IoT networks. The proposed model is based on binary and multi-label classification of IoT network traffic data. Two different datasets, the CIC-IDS-2018 and BoT-IoT containing normal and attack traffic were used, for training the classification models. Our BGH model competed against two basic Deep Learning models BLSTM and GRU. A comprehensive feature extraction process was carried out using seven feature extraction algorithms. Two feature variations (Full-feature and Best-10 features) were used in the classifications. The work demonstrated the importance of feature extraction in IoT Intrusion Detection. Time/accuracy and Time/f1-score ratios were used for better performance comparison. The individual performances were aggregated multiplicatively, yielding a performance product. Our BGH model emerged as the best performer among the models, with the best performance results in all binary and multi-label classifications.

Our BGH model was also compared with other IoT Intrusion Detection models, using various methods. The comparison showed that our proposed model's time/accuracy performance doubled the nearest competitor's performance. Therefore, it can be concluded that our hybrid Deep Learning method is very promising for IoT Intrusion Detection. As future work, it would be appropriate to apply our proposed intrusion detection approach to other types of traffic. Also, labeled logs or datasets of enterprise network traffic data can be used with our proposed model to detect specific activities.

REFERENCES

[1] B. Javed, M. W. Iqbal and H. Abbas, "Internet of Things (IoT) design considerations for developers and manufacturers," 2017 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2017, pp. 834–839. doi:10.1109/ICCW.2017.7962762

[2] K. Xu, X. Wang, W. Wei, H. Song and B. Mao, "Toward software defined smart home," IEEE Communications Magazine, vol. 54, no. 5, May 2016, pp. 116–122. doi:10.1109/MCOM.2016.7470945

[3] A. Q. Mobark and A. Sidorova, "Consumer acceptance of Internet of Things (IoT): Smart home context," Journal of Computer Information Systems, vol. 60, no. 6, Nov. 2020, pp. 507–517. doi:10.1080/08874417.2018.1543000

[4] M. Sendhil, and J. Spiess, "Machine learning: An applied econometric approach," Journal of Economic Perspectives, vol. 31, no. 2, May 2017, pp. 87–106. doi:10.1257/jep.31.2.87

[5] "Number of IoT Devices 2015-2025," Statista, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. Accessed 13 Dec. 2021

[6] ITU. "ARM Predicts 1 Trillion IoT Devices by 2035 with New End-to-End Platform," ITU News, 6 Aug. 2018, https://news.itu.int/arm-pelion-iot-end-to-end-platform/

[7] J. Wurm, K. Hoang, O. Arias, A. Sadeghi and Y. Jin , "Security analysis on consumer and industrial IoT devices," 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2016, pp. 519–524. doi:10.1109/ASPDAC.2016.7428064

[8] What You Need to Know about the Mirai Botnet behind Recent Major DDoS Attacks. https://securitybrief.com.au/story/what-you-need-know-about-mirai-botnet-behind-recent-major-ddos-attacks. Accessed 13 Dec. 2021

[9] IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/ids-2018.html. Accessed 13 Dec. 2021

[10] Moustafa, Nour. The Bot-IoT Dataset. Oct. 2019. ieee-dataport.org, https://ieee-dataport.org/documents/bot-iot-dataset

[11] Y. LeCun, et al., "Deep Learning," Nature, vol. 521, no. 7553, May 2015, pp. 436–444. doi:org/10.1038/nature14539

[12] M. T. Mahmud et al., "Using machine learning to secure IOT systems," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, 2020, pp. 1–7. doi:10.1109/ISMSIT50672.2020.9254304

[13] "A New Botnet Attack Just Mozied Into Town," Security Intelligence, https://securityintelligence.com/posts/botnet-attack-mozi-mozied-into-town/. Accessed 13 Dec. 2021

[14] McAfee Labs Threats Report | November 2020. https://www.mcafee.com/enterprise/en-us/lp/threats-reports/nov-2020.html. Accessed 13 Dec. 2021

[15] J. Kim, et al., "CNN-Based network intrusion detection against denial-of-service attacks," Electronics, vol. 9, no. 6, June 2020, p. 916. doi:10.3390/electronics9060916

[16] S. Aljawarneh M. Aldwairi, M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, Mar. 2018, pp. 152–160. doi:10.1016/j.jocs.2017.03.006

[17] M. Esmalifalak, Nam Tuan Nguyen, R. Zheng and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," 2013 IEEE Global Communications Conference (GLOBECOM), IEEE, 2013, pp. 808–813. doi:10.1109/GLOCOM.2013.6831172

[18] B. Yan, and H. Guodong, "LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network," Security and Communication Networks, vol. 2018, Aug. 2018, pp. 1-13. doi:10.1155/2018/6026878

[19] M. A. Altuncu, F. K. Gülağiz, H. Özcan, Ö. F. Bayir, A. Gezgın, A. Niyazov, M. A. Çavuşlu, S. Şahin, "Deep learning based DNS tunneling detection and blocking system," Advances in Electrical and Computer Engineering, vol. 21, no. 3, 2021, pp. 39–48. doi:10.4316/AECE.2021.03005

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

[21] P. Lin, K. Ye, C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," Cloud Computing – CLOUD 2019, edited by Dilma Da Silva et al., vol. 11513, Springer International Publishing, 2019, pp. 161–176. doi:10.1007/978-3-030-23502-4_12

[22] M. A. Ferrag, M. Leandros, M. Sotiris, J. Helge, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, Feb. 2020, p. 102419. doi:10.1016/j.jisa.2019.102419

[23] M. Tao, et al., "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," Sensors, vol. 16, no. 10, Oct. 2016, p. 1701. doi:10.3390/s16101701

[24] P. Liu, and J. Liu, "Combined effect of multiple performance shaping factors on human reliability: Multiplicative or additive?," International Journal of Human–Computer Interaction, vol. 36, no. 9, May 2020, pp. 828–838. Taylor and Francis+NEJM, doi:10.1080/10447318.2019.1695461

[25] Y. Li, eJ. Huang, H. Chen, "Time series prediction of wireless network traffic flow based on wavelet analysis and BP neural network," Journal of Physics: Conference Series, vol. 1533, no. 3, Apr. 2020, p. 032098. doi:10.1088/1742-6596/1533/3/032098

[26] Q. R. S. Fitni, and R. Kalamullah, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), IEEE, 2020, pp. 118–124. doi:10.1109/IAICT50021.2020.9172014

[Downloaded from www.aece.ro on Wednesday, June 18, 2025 at 02:22:27 (UTC) by 172.69.58.64. Redistribution subject to AECE license or copyright.]

[27] F. Zhao, H. Zhang, J. Peng, et al., "A semi-self-taught network intrusion detection system," Neural Computing and Applications, vol. 32, no. 23, Dec. 2020, pp. 17169–79. doi:10.1007/s00521-020-04914-7

[28] X. Li, W. Chen, Q. Zhang, L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection," Computers & Security, vol. 95, Aug. 2020, p. 101851. doi:10.1016/j.cose.2020.101851

[29] G. Sunanda, and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," Journal of Network and Computer Applications, vol. 169, Nov. 2020, p. 102767.,doi:10.1016/j.jnca.2020.102767

[30] M. Catillo, M. Rak, U. Villano, "2L-ZED-IDS: A two-level anomaly detector for multiple attack classes," Web, Artificial Intelligence and Network Applications, edited by Leonard Barolli et al., vol. 1150, Springer International Publishing, 2020, pp. 687–696. doi:10.1007/978-3-030-44038-1_63

[31] R. I. Farhan A. T. Maolood, N. Hassan, "Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning," Indonesian Journal of Electrical Engineering and Computer Science, vol. 20, no. 3, Dec. 2020, p. 1413-1418. doi:10.11591/ijeecs.v20.i3

[32] G. C. Amaizu, C. I. Nwakanma, J. -M. Lee and D. -S. Kim, "Investigating network intrusion detection datasets using machine learning," 2020 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2020, pp. 1325–1328. doi:10.1109/ICTC49870.2020.9289329

[33] S. K. Wanjau, G. M. Wambugu, G. N. Kamau,,"SSH-Brute force attack detection model based on deep learning," International Journal of Computer Applications Technology and Research, vol. 10, no. 01, 2021, pp. 42–50

[34] M. Chhetri, S. Kumar, P. P. Roy, B.-G. Kim, "Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan," Remote Sensing, vol. 12, no. 19, Sept. 2020, p. 3174. doi:10.3390/rs12193174

[35] P. Kaushik, A. Gupta, P. P. Roy and D. P. Dogra, "EEG-Based age and gender prediction using deep BLSTM-LSTM network model," IEEE Sensors Journal, vol. 19, no. 7, Apr. 2019, pp. 2634–2641. doi:10.1109/JSEN.2018.2885582

[36] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull,, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, Nov. 2019, pp. 779–796. doi:10.1016/j.future.2019.05.041

[37] I. Idrissi, et al., "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, no. 1, Mar. 2021, p. 110-120. doi:10.11591/ijai.v10.i1

[38] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo and A. Robles-Kelly, "Deep learning-based intrusion detection for IoT networks," IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE, 2019, pp. 256–25609. doi:10.1109/PRDC47002.2019.00056

[39] M. A. Ferrag and L. Maglaras, "DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," IEEE Transactions on Engineering Management, vol. 67, no. 4, Nov. 2020, pp. 1285–1297.doi:10.1109/TEM.2019.2922936

[40] M. Buda, A. Maki, M. A. Mazurowschi, "A systematic study of the class imbalance problem in convolutional neural networks," Neural Networks, vol. 106, Oct. 2018, pp. 249–259. doi:10.1016/j.neunet.2018.07.011

[41] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion," IEEE Sensors Journal, vol. 19, no. 16, Aug. 2019, pp. 7056–7063. doi:10.1109/JSEN.2019.2909837

[42] M. Khan, H. Wang, A. Ngueilbaye, et al., "End-to-end multivariate time series classification via hybrid deep learning architectures," Personal and Ubiquitous Computing, Sept. 2020. doi:10.1007/s00779-020-01447-7

[43] K. Cho, et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1724–1734. doi:10.3115/v1/D14-1179

[44] D. Chicco, and J. Giuseppe, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, Dec. 2020, p. 6. doi:10.1186/s12864-019-6413-7