MLIDS22- IDS Design by Applying Hybrid CNN-LSTM Model on Mixed-Datasets

Inam Abdullah Abdulmajeed*[1] and Idress Mohammed Husien[2]

E-mail: stcha002@uokirkuk.edu.iq[1], enaam.alsanee@gmail.com[1], idress@uokirkuk.edu.iq[2]

*Inam Abdullah Abdulmajeed

[1,2]Department of Computer Science, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq

Keywords: Machine Learning, Intrusion Detection System, Inter-Dataset, CNN, LSTM, Accuracy, and ROC curve

Received: August 21, 2022

The intrusion detection system (IDS) is an essential part of cyber security which captures and investigates traffic to distinguish between legitimate and malicious activities and determines the type of attack. The selection of the dataset used in training the machine learning-based IDS is crucial in ensuring that IDS performs accurately in cyber-attack classification. When utilizing multiple datasets in the training process, the metrics will relate numerically between the ML algorithm and a particular dataset. Previous research concluded a major decline in metrics when using inter-datasets evaluation. This research thoroughly investigates the use of the most modern and comprehensive IDS datasets, CIC-IDS2017 and CSE-CIC-IDS2018, to design and evaluate machine learning-based IDS systems using hybrid CNN-LSTM architecture. The new approach followed is to generate a new dataset which is the output of mixing both datasets. The experimental testing showed superior metrics values yielded when training with the mixture dataset against the use of individual datasets, especially when performing the inter-datasets evaluation, which overcomes the generalization problem.

Povzetek: S kombiniranjem učnih množic postopek doseže boljše zaznavanje kibernetskega napada.

1 Introduction

The intrusion detection process refers to the monitoring and analysis of a computer system and network's activities for signs of prospective incident breaches or imminent risks of violations of computer security legislation [1]. Software that automates the intrusion detection process is known as an intrusion detection system (IDS) [2]. Jim Anderson first put forth the concept of IDS in 1980. Since then, a wide variety of IDS technologies have been created and improved to meet the demands of network security [3]. The modern IDS logs the transaction and takes appropriate action when it detects a potential malicious incident. The action could be as simple as continuing to log, alerting security administrators, rerouting the attack, or stopping harmful activity by itself, all through interaction with other security systems [4].

There are two types of IDS deployment scenarios; network-based IDS and host-based IDS. Network-based IDS sniffs network packets to discover the attack. While host-based IDSs monitor a single server or endpoint [5]. Also, there are two IDS detection approaches, signature-based, where attack signatures are listed in a repository, and IDS compares that to the real-time traffic. When signature-based IDS finds a match, it will generate an alert to network/system administrators. IDS based on signatures is quick and precise, but it must be updated periodically; otherwise, the intruder could perform a successful attack without being stopped or recorded if the IDS signatures are not updated in time. The second approach is anomaly-based or behavioral-based. In this approach, the IDS will build a normal network profile in the setup stage, and then in the operation stage, it detects unexpected traffic or behavior and records it as an intrusion. This approach will be able to handle known and new attacks, but it suffers from a high false positive rate [6]. Hybrid-based approach mixes both above approaches, yielding a more powerful operation., but it may consume more IDS resources [7]. The more recent detection approach is to use Machine Learning Algorithms to build the IDS model able to learn attacks with high accuracy. In [8], An inter-dataset assessment technique is proposed for assessing the generalization power of ML models and comparing them to the usual intra-dataset evaluation. This paper discusses this ML-based IDS lack of generalization and contributes to a new overcome for this problem.

2 Literature review

The protection of businesses from cyberattacks is a crucial issue nowadays, and a difficult subject since it

impacts them financially and affects their market image. New and sophisticated assaults that target businesses worldwide emerge every day. For that reason, the scientific community has been interested in the development and optimization of IDS [9]. A group of earlier works on this topic are summarized in this section.

The researchers in [10], developed deep learning models for DDoS attack detection and validated them using CICIDS2017 datasets. The proposed models are compared with other machine learning algorithms. They also discussed the problems in deploying deep learning solutions for the Internet of Things (IoT) security. They found that the hybrid CNN+LSTM model performs better than the rest of the deep learning models and machine learning algorithms, with an accuracy of 97.16%.

In [11], the CSE-CIC-IDS2018 dataset was used to train an IDS using CNN deep-learning methods, which included two convolution layers and two max-pooling layers. They found a performance rise after the researchers compared its performance with the RNN model. The authors suggested adjusting the ratio of benign and attack data to enhance system performance.

In order to improve intrusion detection, the DL-IDS (deep learning-based intrusion detection system) model combined a CNN and LSTM hybrid network [12]. The authors employed a category weight optimization strategy to increase resilience by processing an imbalanced number of samples of various attack types in the CICIDS2017 training dataset. As a consequence, DL-IDS provided another proof that the hybrid CNN-LSTM model yields a superior overall accuracy of 98.67% for the multi-classification test over the individual CNN-only and LSTM-only models.

Another experiment that used a hybrid CNN-LSTM deep learning model to classify attack types using the NSL-KDD dataset was documented in [13]. In the beginning, they experimented using the LSTM-only model, and then CNN layers were added to the architecture, which increased accuracy in a significant manner.

In [14], a hybrid convolutional recurrent neural network (CRNN) named HCRNNIDS is used to create a DL-based IDS. In this model, CNN uses convolution to collect local features, whereas RNN captures temporal features to increase IDS accuracy. After training the model using CSE-CIC-DS2018 dataset, the resultant accuracy was up to 97.75%.

As outlined in the Introduction Section above, the study in [8] was intended to assess the generalization potential of promising IDS ML algorithms. A unique inter-dataset evaluation approach was deployed by making the model trained using the first dataset and then tested using the second. The generalization capabilities of four unsupervised machine learning algorithms trained on two current datasets are investigated in this approach. The findings demonstrate that while all models are capable of producing excellent classification scores on a single dataset, they are unable to do so on a second, unrelated dataset that is not employed in the training process. The average decrease in AUROC and accuracy scores caused by this challenging inter-dataset experimental setup were 30.45% and 25.63%, respectively.

Another hybrid CNN with LSTM optimized custom model named RC-NN-IDS was introduced in [15]. In this model. The meta-heuristic ALO algorithm was deployed for training optimization, thus providing less error rate and better classification accuracy.

Ref.	Method	MLIDS22 Strong Points						
		NSL-KDD	CIC-IDS 2017	CIC-IDS 2018	Dataset Mix	Balancing Datasets	Hybrid CNN+LSTM	Inter-datasets Evaluation
[8]	PCA, Isolation Forest, Autoencoder, One-Class SVM		X	X		X		X
[10]	Hybrid CNN+LSTM		X				X	
[11]	CNN-Only, RNN-Only			X				
[12]	Hybrid CNN+LSTM CCN-Only, LSTM-Only		X			X	X	
[13]	Hybrid CNN+LSTM LSTM-Only	X					X	
[14]	HCRNNIDS Hybrid CNN+RNN			X				
[15]	RC-NN-IDS Hybrid CNN+LSTM			X		X	X	
MLIDS22	**Hybrid CNN+LSTM**		X	X	X	X	X	X

Table 1: Justification of MLIDS22 Strong Points compared to literature

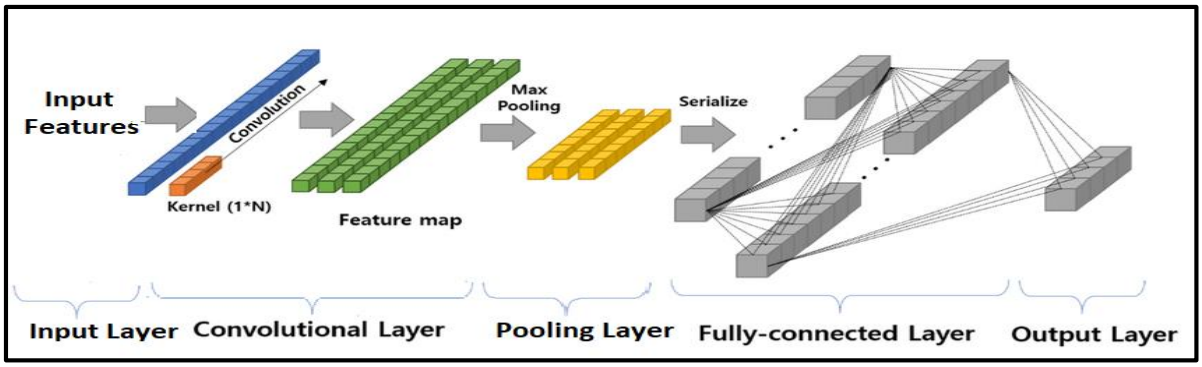

Figure 1: Typical 1D CNN Architecture [18]

The training and evaluation setup was using the DARPA and CSE-CIC-IDS2018 datasets, resulting in an enhanced system accuracy of 94.28%. **Error! Reference source not found.** indicates why this model MLIDS22 covers the lack areas that exist in the other articles explained in the literature.

3 Background

3.1 Deep learning and artificial neural networks

Artificial Neural Network (ANN) was invented from the analogy with the structure of human brain cells. The term "Deep Learning" (DL) refers to the most recent technological advancement and current research focus on the Machine Learning domain. DL becomes pervasive in our everyday lives, providing answers that were the stuff of science fiction just a decade ago. This new age was created, with the publication of Hinton and Salakhutdinov's [16]. In essence, it demonstrates that ANNs with a high number of hidden layers may exhibit impressive learning potential. In a conclusion, DL hence described this subfield of ML that is capable of dealing with enormous datasets including complex patterns and objects.

The most modern type of DL in NN is Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The standard CNNs can only process Two-Dimensional data, such as photos or movies. The term "2D CNN" describes their typical output format. On the other hand, One-Dimensional Convolutional Neural Networks (1D CNNs) have been created as a streamlined alternative to 2D CNNs. Recent researches have shown the benefits of using 1D CNNs for handling 1D signals over their 2D counterparts in a number of scenarios. [17], [18], and [19].

1D CNNs have the potential to outperform their 2D CNN counterpart in a number of ways.

1D CNNs have far less demanding processing requirements than their 2D counterparts. When it comes to the training of small 1D CNNs with few hidden layers and neurons, any typical modern computer is a practical platform. In comparison, specialized hardware is needed for training 2D CNNs (e.g. Cloud computing or GPU farms) [19]. Figure 1 shows a typical 1D CNN Architecture with two kinds of layers. The first kind is the so-called "CNN layers," which perform 1D convolutions and sub-sampling (pooling), and the second type is the Fully-connected layers. The following hyper-parameters make up a 1D CNN configuration.

First: The sum of concealed layers/neurons in the CNN and fully-connected layers.

Second: The size of the kernel filters used by each CNN layer.

Third: The subsampling factor for each CNN layer.

Fourth: Deciding on a pooling method and activation function.

Another popular model of deep learning is the Recurrent Neural Network (RNN). The widely used RNN is the Long-Short Term Memory (LSTM). Instead of a linear progression like conventional neural networks, LSTM contains input connections. Data streams in their whole (not just individual data points) may be processed by it. Non-segmented handwriting recognition, voice recognition, and the detection of anomalies in network traffic are all examples of applications of LSTM. To summarize, a typical LSTM unit comprises a cell, an input door, an output door, and a forgetting door. The cell stores values for undetermined amounts of time, and the three gates control the entry and exit of information. Because there may be delays of indeterminate duration in a time series between major occurrences, LSTM networks are particularly well-suited for categorizing, evaluating, and predicting based on time series data [20].

3.2 Datasets overview

The dataset is essential for guiding ML on how to detect abnormal threats. The network infrastructure has

been effectively modified by the introduction of new technologies such as cloud computing, social media, and the IoT. New forms of threats will emerge as a result of these changes, that impose ongoing development in the dataset quality and scope. Below paragraphs introduces the datasets used in the research:

3.2.1 CIC-IDS-2017 dataset

Communications Security Establishments (CSE) and the Canadian Institute for Cybersecurity (CIC) invented this dataset in 2017. In order to produce this dataset, two networks for attackers and victims were established in a setup Lab. A mix of Linux and Windows hosts environment were used on both sides. In order to record the traffic, the uplink port of the router was sniffed to the dataset server. This dataset contains both benign and important attacks. The Flow Meter output with labelled flows using time stamps and other protocol parameters was embedded in the dataset [21] [22].

3.2.2 CSE-CIC-IDS2018 dataset

It depicts real-time network traffic derived from Amazon's AWS platform by Communications Security Corporation (CSE) and the Canadian Cybersecurity Institute (CIC). It is one of the most reliable sources of data for evaluating intrusion detection systems based on network abnormalities.
The final dataset consists of seven distinct attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web assaults, and network penetration from inside. The attacking infrastructure consists of 50 computers, whereas the infrastructure of the victim firm consists of 420 machines and 30 servers across five departments. Each machine's network traffic and system logs are included in the dataset, together with 80 characteristics derived from the collected traffic using CICFlowMeter-V3. [web link] and [19].

3.2.3 The mixture of the CIC-IDS2017 dataset and CSE-CIC-IDS2018 dataset

The new approach followed in this research is to generate a new dataset made by aggregating the above datasets and generate this new dataset. In order to be able to unite the two datasets, special pre-processing procedures must be conducted on each dataset individually to generate common features.

3.3 Metrics

In multi-classification, to assess the detection effectiveness of the model on the unbalanced dataset in a more reasonable manner, each index is computed using a weighted average technique based on the number of samples in each category. Each indicator's formula is presented in equations (1) to (7). [21] [23] [24]:

$$\text{True Positive Rate} = \frac{\text{TP}}{(\text{TP+FN})} \quad (1)$$

$$\text{False Positive Rate} = \frac{\text{FP}}{(\text{TN+ FP})} \quad (2)$$

$$\text{True Negative Rate} = \frac{\text{TN}}{(\text{FP+ TN})} \quad (3)$$

$$\text{False Negative Rate} = \frac{\text{FN}}{(\text{TP+ FN})} \quad (4)$$

$$\text{Accuracy} = \frac{(\text{TP+TN})}{(\text{TP+TN+FP+FN})} \quad (5)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{FP + TP})} \quad (6)$$

$$\text{F1 score} = \frac{2\text{TP}}{(2\text{TP+FN+ FP})} \quad (7)$$

The curves for the ROC (Receiver Operating Characteristic) and their associated AUC (Areas Under the Curve) are used to quantify the output quality of machine learning; hence, they measure how well a classifier has been trained. Typically, the ROC curve is a compromise between the classifier's sensitivity (TPR) and specificity (TNR). They provide a decent indication of the classifier's performance. For the ROC curve, sensitivity increases and specificity decreases as the curve moves toward the right. The ROC curve along a 45◦ angle is equivalent to a random classifier. On the other hand, the closer the AUC is to 1, the machine learning system approaches perfect behaviour [25].
The micro-average measure is a weighted average that accounts for each risk class's contribution. It estimates one performance statistic as opposed to several performance metrics. For the multiclass ML algorithm, the micro-average is used to plot a single ROC curve and calculate a single AUC value representing the performance of the algorithm for all classes. In order to tackle the fact that the micro-average tends to bring the overall metrics more toward the majority class, the class imbalance problem was taken care of in the pre-processing stage, even before fitting the classification models [25].

4 Methodology

This section will explain the procedures followed to train and evaluate the MLIDS22 employing the inter-datasets evaluation strategy. It is divided into subsections for a further detailed explanation.
As illustrated in [8], A minimum of two datasets are required to apply the inter-dataset evaluation technique. In addition to that, the benign traffic should be sampled from the same distribution. It is crucial that these datasets are related and have the same properties.

This presumption enables the evaluation of a model trained on a first dataset to be graded with the second dataset that is linked to it because the normal and malicious behavior that was learnt should theoretically still be transferable and relevant during the checking. In a perfect model, a well-generalized system would successfully categorize the benign and attack samples in both datasets. The crucial aspect is that all data pass through the same preparation pipeline, maintaining the same set of features [8].

Table 2: The dictionary for numericalizing (CIC-IDS2017 and CSE-CIC-IDS2018) datasets

Label In Cicids 2017dataset	**Label In Cicids 2018 Dataset**	**Grouping**	**Numerical Class**
BENIGN	BENIGN		**0**
DDOS		Dos Group	**1**
	DDOS ATTACK-HOIC		
	DDOS ATTACK-LOIC-UDP		
	DDOS ATTACKS-LOIC-HTTP		
DOS GOLDENEYE	DOS ATTACKS-GOLDENEYE		
DOS HULK	DOS ATTACKS-HULK		
DOS SLOWLORIS	DOS ATTACKS-SLOWLORIS		
HEARTBLEED		Web Attack Group	**2**
DOS SLOWHTTPTEST			
WEB ATTACK – SQL INJECTION	SQL INJECTION		
WEB ATTACK – BRUTE FORCE	BRUTE FORCE - WEB		
WEB ATTACK – XSS	BRUTE FORCE - XSS		
	DOS ATTACKS-SLOWHTTPTEST	FTP Group	**3**
FTP-PATATOR	FTP-BRUTEFORCE		
SSH-PATATOR	SSH-BRUTEFORCE		**4**
BOT	BOT		**5**
INFILTRATION	INFILTRATION		**6**
PORTSCAN			**7**

The datasets that satisfy these requirements are the datasets created by Canadian Institute for Cybersecurity (CIC), the Intrusion Detection Evaluation Dataset (CIC-IDS2017) and the IPS/IDS dataset on AWS (CSE-CIC-IDS2018) that are explained in the Background Section above.

4.1 Pre-processing stage

Before we can pass the datasets to Neural Network Training Algorithm, they need to be pre-processed to change their format without destructing their intrinsic features, maintaining the scientific approach in designing the required IDS model.

In the Pre-processing Stage, the following have been done:

- Network Flow files were aggregated into a single file.
- The empty and duplicated rows were removed from the datasets.
- The columns (features) which represent time and flow IDs like ("Flow ID" and "Timestamp") have been removed.
- Some of the characters in the dataset header and instances have been changed or removed.
- Infinity values have been replaced by the maximum of the columns' data.
- The NA and NAN values have been replaced by zero in columns of data
- The ‘Label’ column data have been numericalized, as per Table 1.
- The whole datasets have been normalized using pre-processing.MinMaxScaler() function from sci-kit learn library [26]. The normalization has been applied to features only, but the 'Label' column, which has a qualitative description of each flow has been left with integer representation since this will facilitate more clear prediction process for the target classes.

After the pre-processing stage has finished, the two dataset files have been aggregated to 850 MB for CIC-IDS2017 and 1.5 GB for CSE-CIC-IDS2018. Both resultant datasets have the same attributes. This enables inter-dataset processing and dataset mixing, as will be illustrated in the next stages.

4.2 Sampling stage

Either feature set reduction or sample set reduction can be used to reduce the size of a dataset before the dataset can be deployed in ML training. In this work, both approaches have been followed. Hence, in order to reduce the effect of unbalanced data in the datasets, they were subjected to sampling which is necessary to increase system prediction accuracy as shown in [23, 24, 25, 26, 27]. Furthermore, when the balanced dataset was created via under-sampling/over-sampling, such that all classes had the same size. It contributes to greater fairness, representation, and imbalance reduction. The amount of anomalous samples in the dataset for intrusion detection is intrinsically tiny;

hence, under-sampling alone is insufficient. However, adopting merely oversampling would add too much duplication in data and raise space and time expenses [23]. The strategy used in this research is a hybrid sampling algorithm combining Adaptive Synthetic Sampling (ADASYN) for over-sampling and then Random Under Sampling (RUS).

To reduce the dataset to a realistic size, the Benign class was under-sampled to 500,000 samples; then, as explained in [23], a threshold S_{TH} was calculated based on equation (8) to determine which class to over-sample or under-sample

$$S_{TH} = int\left(\frac{N}{C}\right) \quad (8)$$

Where N is the size of the dataset in "number of samples" and C is the class's number. If the class size was found to be higher than S_{TH}, it was under-sampled, while over-sampling classes which are found to be less than threshold S_{TH}.

4.3 Feature reduction stage

Before these datasets can be passed to the Neural Network training algorithm, the number of features of the datasets needs to be reduced so the Neural Network Algorithm evaluation time will be more realistic. Feature reduction has a wide range of advantages, including bettering prediction performance by overcoming the curse of dimensionality, lowering measurement and storage needs, shortening training durations, and facilitating data visualization and understanding [28].

In reality, building Machine Learning models from datasets with a high number of features needs more computational resources. Feature selection is the process of selecting features from the original feature set, keeping the interpretation, and ensuring that they are appropriate for the analytic objective [29].

First of all, the columns (features), which represent arbitrary host address information that changes from setup to setup like ("Source IP", "Source Port", and "Destination IP"), have been removed. The feature ("Destination Port") was kept since it is highly correlated with Traffic Type. The important port numbers have been emphasized with higher weights to reflect the importance of the corresponding traffic types.

Secondly, and according to [8], there are a number of features in both datasets found to be with no variance, which in conclusion has no effect on the training process, so those features have been eliminated. The output of this stage is a vector of 68 attributes that will be used in the next stage in constructing the Neural Network algorithm model.

4.4 Model construction

In order to achieve the best performance, The MLIDS22 model is designed as a mix of CNN and LSTM, as can be shown in Table 2, which provides a model summary.

We developed a 1D-CNN model with 12 layers that is optimized for NIDS. The network topology is shown in Figure 2.

The first layer is the input layer which receives features' values from the dataset. Then, the CNN stage consists of four convolutional layers, with a Max-Pooling layer placed after every pair of convolutional layers to enhance accuracy. LSTM layer was deployed in order to add RNN functionality to the proposed model. Following that, three fully connected layers have been deployed to enhance system stability.

The last layer is the output layer, typically used in the context of classifying and forecasting attack classes. It includes one neuron for each class, giving the probability of that class. Thus their sum should add up to 1. The softmax activation function has been used in the output layer

The number of hidden layers, along with neurons in each layer, are the main parameters used in ML architecture deployment. The model summary as implemented by Intel Python v3 is illustrated in Table 2. Figure 2 shows the MLIDS22 architecture.

4.5 Inter-datasets strategy

Following the approach used in [8], it is crucial that datasets used in the experiment are connected and have the same properties. In a perfect model, a well-generalized system would successfully categorize the benign and attack samples in both datasets [8].

The new approach used in this experiment is to create a mixture dataset made by aggregating the train part of the CIC-IDS2017 dataset with the train part of the CSE-CIC-IDS2018 dataset and aggregating the test pars accordingly, as shown in Figure 2. As a consequence of this strategy, MLIDS22 trains three machine learning models; each model has been constructed from the corresponding train part of that dataset.

On the other hand, the new evaluation strategy extends the one used in [8] to three inter-datasets evaluations, so the test part of the CIC-IDS2017 dataset, for example, is used to evaluate the three trained models and follow the same procedures with other test parts, ending up with nine evaluation metrics groups as shown in Figure 3

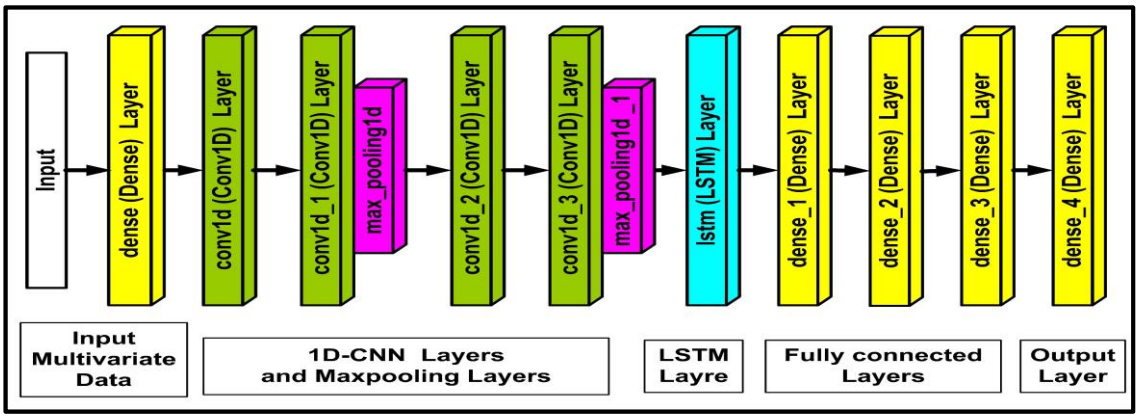

Figure 1: The MLIDS22 Model Architecture.

Table 3: The MLIDS22 Model Summary.

Layer (type)	**Output Shape**	**Param #**
dense (Dense)	(None, 68, 8)	16
conv1d (Conv1D)	(None, 68, 8)	200
conv1d_1 (Conv1D)	(None, 66, 16)	400
max_pooling1d (MaxPooling1D)	(None, 33, 16)	0
conv1d_2 (Conv1D)	(None, 31, 16)	784
conv1d_3 (Conv1D)	(None, 29, 32)	1568
max_pooling1d_1(MaxPooling1D)	(None, 14, 32)	0
lstm (LSTM)	(None, 50)	16600
dense_1 (Dense)	(None, 32)	1632
dense_2 (Dense)	(None, 32)	1056
dense_3 (Dense)	(None, 32)	1056
dense_4 (Dense)	(None, 8)	264

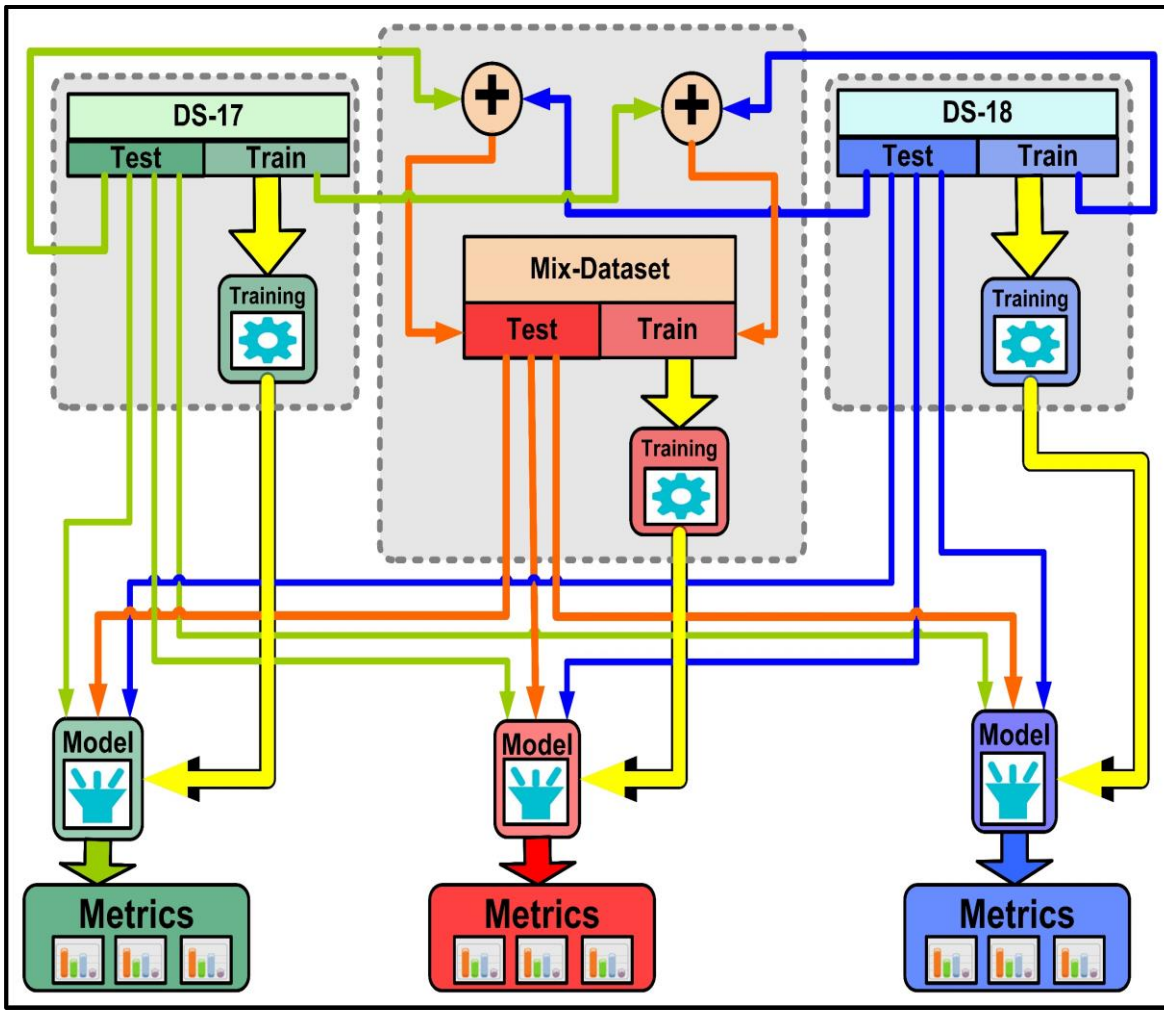

Figure 2: The MLIDS22 Model Evaluation Strategy

4.6 Implementation tools

Hardware:

- CPU: Intel Core i7, 1.80GHz, 4 Core(s),
- Installed Physical Memory (RAM) 64.0 GB
- GPU 1: Intel(R) UHD Graphics with 1 GB RAM

Software:

- OS: Microsoft Windows 10 Pro
- IDE: Microsoft Visual Code
- Intel® Distribution for Python: For multicore acceleration.
- Python version: 3.9.7 64-bit
- Python Libraries: Numpy, Scikit-learn, TensorFlow, Keras, Pandas, and Matplotlib

5 Results and discussion

As in the previous research [8], discussions are focused on intra-dataset results before shifting to the analysis of the inter-dataset evaluation procedures.

To further understand the MLIDS22 model performance, two graphs were made during the training process: one shows training and validation accuracy, and the other one shows training and validation loss over all epochs. Figure 4 shows these graphs for the CIC-IDS2017 Dataset, CSE-CIC-IDS2018 Dataset, and Mixture Dataset. The accuracy graphs show that model MLIDS22 converges after a few epochs and fluctuates over the saturation level for the rest of the training, which indicates an excellent learning rate. This behaviour generalizes on the three datasets listed earlier with relative differences. The accuracy for DS-17 is superior to other datasets. The new achievement in this research is the good accuracy obtained for the Mix-DS when tested with all three datasets. The loss graphs show the same behaviour, except that the convergence,

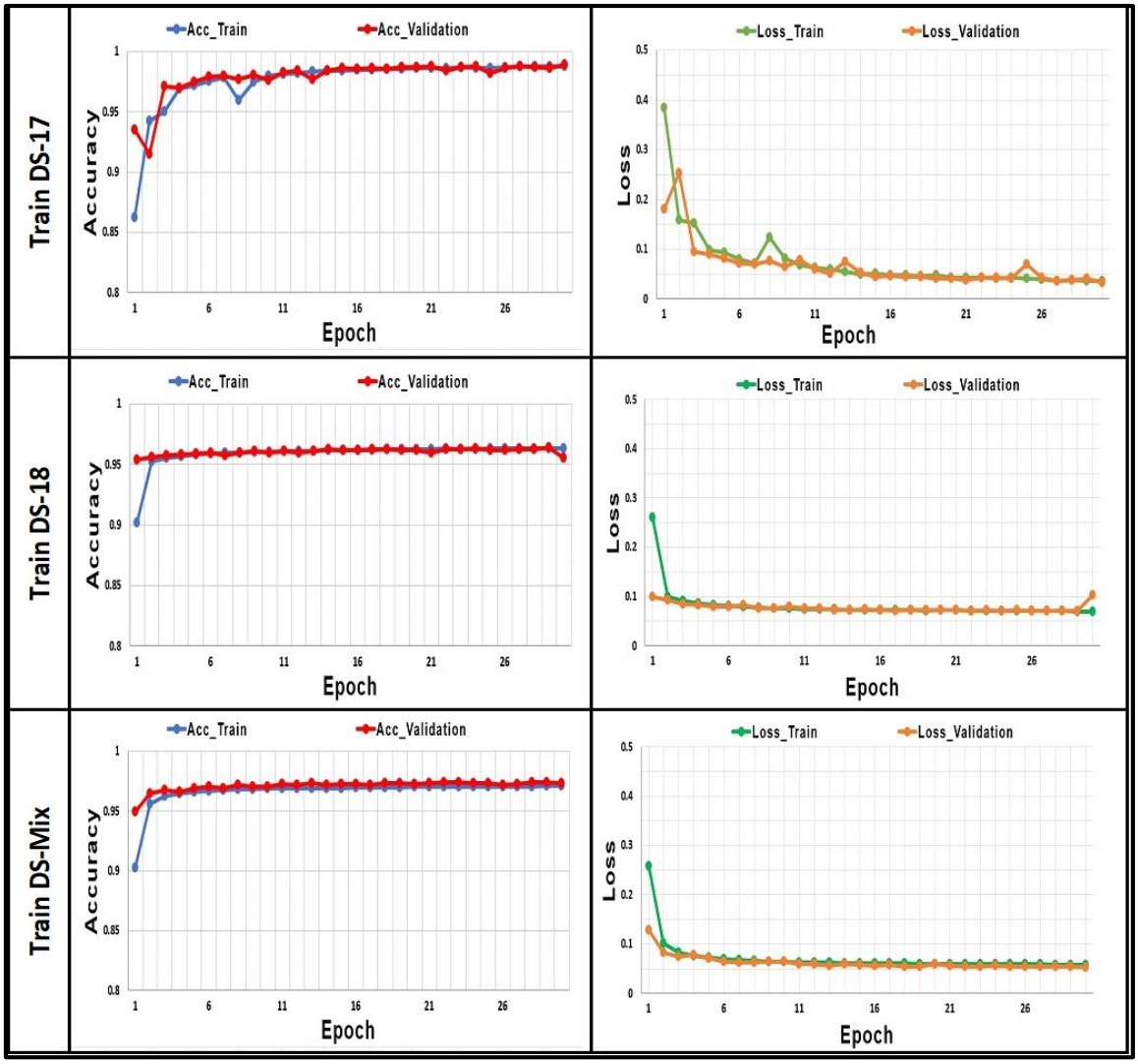

Figure 3: The Accuracy and Loss function for Three Dataset

in this case, approaches the minimum value. Accuracy, F1 score, and its accompanying precision and recall are also included for completeness and to facilitate simple comparison with relevant work in the literature. Their calculation equations have been explained in Metrics Section. Table 5 provides a summary of these ratings. Both the training and testing datasets are included in the dataset column. These datasets will be different for the inter-datasets assessment but the same for the intra-dataset evaluation. The same results have been shown graphically in Figure 5. The intra-dataset evaluation shows excellent metrics. However, the inter-datasets evaluation metrics drop dramatically, which indicates the lake of generalization in each model while trained using individual datasets. The good news came from the mixture dataset, which proved that it could train the model to produce excellent metrics when tested with all three datasets. This concludes that mixing datasets can produce a fair, accurate model in defending against cyber security attacks. Similarly, the accuracy consistently dropped with a noticeable difference between the intra- and inter-dataset evaluation strategy. Table 4 provides a comparison between a subset of MLIDS22 results and corresponding results in the literature articles. When comparing intra-dataset results MLIDS22 shows comparable accuracy, while it gives lower results than [8] with inter-datasets evaluation.

Further analysis of the model is done with the ROC curve. Figure 6 shows the ROC curve for the nine scenarios. AUC score is used as the main evaluation metric during the analysis. The intra-dataset ROCs and AUCs indicate perfect classification. On the other side, looking at the model trained on DS-137 and evaluated on DS-18 was still able to achieve an AUC of 0.7327, while the other inter-datasets with the model trained on DS-18 and evaluated on DS-17 achieved the lowest AUC of all nine scenarios of 0.6103. On the other hand, the model trained with DS-Mix gives the same high AUC score as the two intra-datasets evaluation scenarios, which again demonstrates a superior detection rate even when the model is evaluated with

individual datasets. [8] identifies two underlying factors for the drop in categorization performance between the CIC-IDS2017 and CSE-CIC-IDS2018 Datasets. First, the port scan attack class, which was readily observable in the 2017 sample, completely vanished from the 2018 dataset. Secondly, the findings reveal that the relatively simple detectable infiltration assault becomes more difficult in the 2018 dataset, and due to its bigger proportion, it also has a greater impact on the final outcome. Collectively, they are responsible for the decline in AUC.

Table 5: The MLIDS22 Measured metrics

Train DS	Test DS	Recall	Precision	F1	Accuracy	AUC
17	**17**	0.9889	0.989	0.9889	0.9889	0.9999
17	**18**	0.3583	0.5699	0.3748	0.3583	0.7327
17	**Mix**	0.5682	0.8025	0.6024	0.5682	0.8396
18	**17**	0.3164	0.2033	0.2338	0.3164	0.6103
18	**18**	0.9656	0.9656	0.9654	0.9656	0.9996
18	**Mix**	0.7495	0.8041	0.7501	0.7495	0.8832
Mix	**17**	0.9893	0.9895	0.9893	0.9893	0.9999
Mix	**18**	0.9649	0.9646	0.9647	0.9649	0.9996
Mix	**Mix**	0.9730	0.9728	0.9729	0.9730	0.9997

Table 4: Comparison Of MLIDS22 Accuracy with Literature

Train Dataset	Test Dataset	[8]	[10]	[12]	[14]	[15]	MLIDS22
17	**17**	**0.9426**	**0.9716**	**0.9950**	-	-	**0.9889**
17	**18**	**0.6748**	-	-	-	-	**0.3583**
17	**Mix**	-	-	-	-	-	**0.5682**
18	**17**	**0.7363**	-	-	-	-	**0.3164**
18	**18**	**0.8898**	-	-	**0.9775**	**0.9428**	**0.9656**
18	**Mix**	-	-	-	-	-	**0.7495**
Mix	**17**	-	-	-	-	-	**0.9893**
Mix	**18**	-	-	-	-	-	**0.9649**
Mix	**Mix**	-	-	-	-	-	**0.9730**

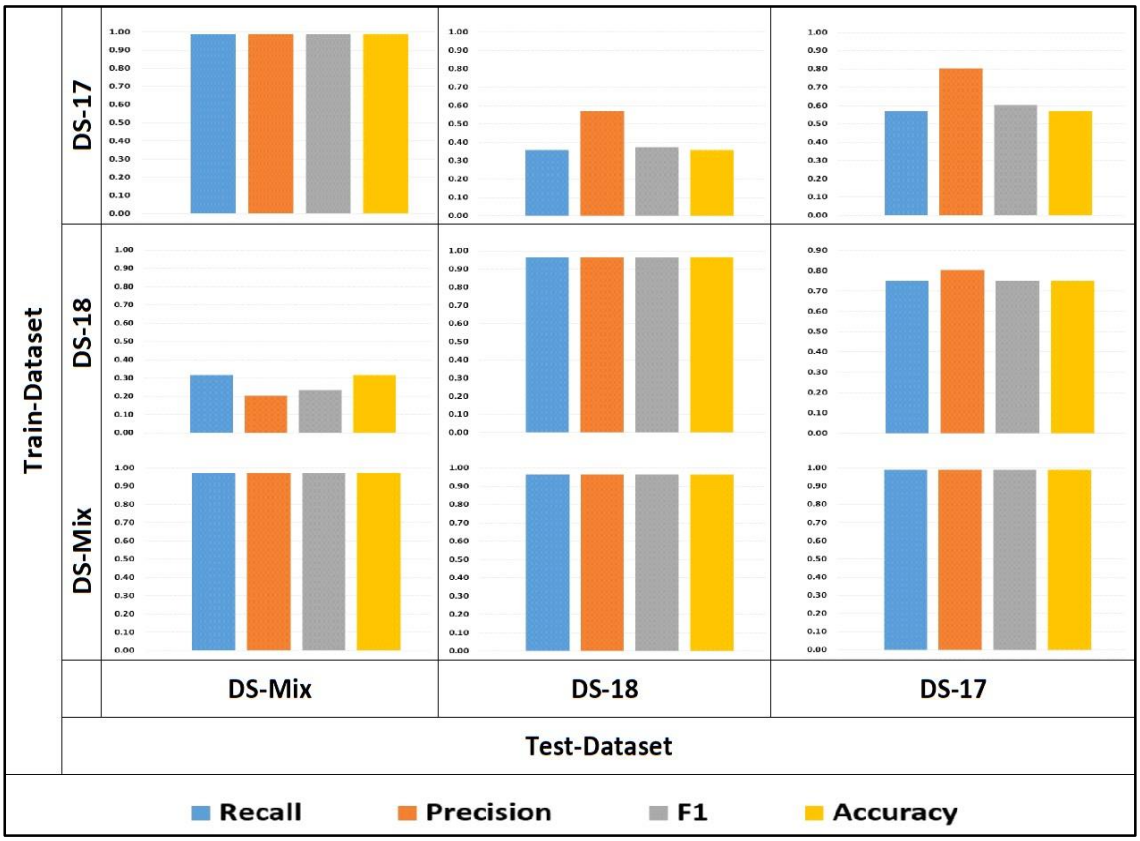

Figure 5: Performance metrics for the nine scenarios

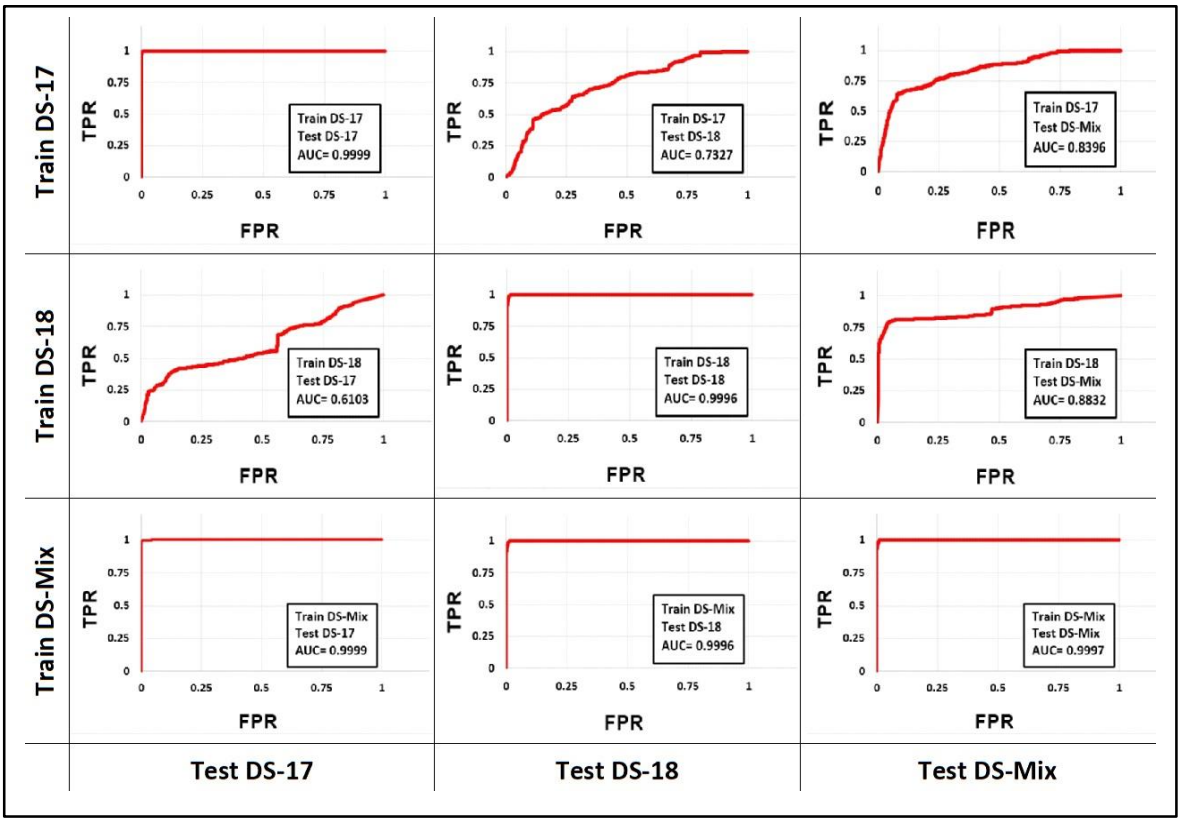

Figure 6: ROC curve for the nine scenarios

6 Conclusion

One definition of an IDS is a "device or software program that monitors system or network operations for malicious actions or policy breaches." [25]. According to [8], when presented with an unseen but related dataset, existing state-of-the-art algorithms for unsupervised anomaly-based NIDS lose, in the best scenario, around 25% of their classification performance. To satisfy the demand of improving the generalization performance of the IDS systems, the new approach followed in this article is to create a mixture dataset and evaluate the model performance trained with this new dataset. The hybrid CNN-LSTM neural network model (MlIDS22) was created to examine this concept.

The first observation on the model MLIDS22 is that it converges after a few epochs, which indicates an excellent learning rate.

The intra-dataset evaluations show excellent metrics, but when evaluating the inter-datasets, the performance metrics drop dramatically, which indicates the lack of generalization in each individual dataset, as confirmed in [8]. The new mixture dataset proves that it can train the model to produce excellent metrics when tested with all three datasets. This

concludes that mixing datasets can produce a fair, accurate model in defending against cyber security attacks.

Further analysis of the model ROC curve and AUC score for the nine scenarios show that the intra-datasets ROCs and AUCs indicate perfect classification. On the other side, the inter-datasets evaluation shows poor classification abilities. The model trained with DS-Mix gives the same high AUC score as the two intra-datasets evaluation scenarios,

which again demonstrates a superior detection rate even when the model is evaluated with individual datasets.

Finally, it has been shown that the mix-dataset assessment technique presented in this work is an excellent option for adaptation in future research to emphasize on the generalization power of newly created IDS models.

References

[1] K. Scarfone, P. Mell and others, "Guide to intrusion detection and prevention systems (idps)," *NIST special publication*, vol. 800, p. 94, 2007. https://doi.org/10.6028/nist.sp.800-94

[2] B. I. A. Barry and H. A. Chan, "Intrusion detection systems," in *Handbook of information and communication security*, Springer, 2010, p. 193–205. https://doi.org/10.1007/978-3-642-04117-4_10

[3] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, p. e4150, 2021. https://doi.org/10.1002/ett.4150

[4] M. Rhodes-Ousley (2013). *Information Security The Complete Reference, Second Edition*. McGraw-Hill Osborne Media.

[5] A. A. Ghorbani, W. Lu and M. Tavallaee, Network intrusion detection and prevention: concepts and techniques, vol. 47, Springer Science & Business Media, 2009. https://doi.org/10.1007/978-0-387-88771-5

[6] H. El-Taj, F. Najjar, H. Alsenawi and M. Najjar, "Intrusion detection and prevention response based on signature-based and anomaly-based: Investigation study," *International Journal of Computer Science and Information Security*, vol. 10, p. 50, 2012.

[7] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan and S. Anwar, "Intrusion response systems: Foundations, design, and challenges," *Journal of Network and Computer Applications*, vol. 62, p. 53–74, 2016. https://doi.org/10.1016/j.jnca.2015.12.006

[8] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, p. 1–25, 2022. https://doi.org/10.1007/s10922-021-09615-7

[9] D. Chamou, P. Toupas, E. Ketzaki, S. Papadopoulos, K. M. Giannoutakis, A. Drosou and D. Tzovaras, "Intrusion detection system based on network traffic using deep neural networks," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019. https://doi.org/10.1109/camad.2019.8858475

[10] M. Roopak, G. Y. Tian and J. Chambers, "Deep learning models for cyber security in IoT networks," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, 2019. https://doi.org/10.1109/ccwc.2019.8666588

[11] J. Kim, Y. Shin and E. Choi, "An intrusion detection model based on a convolutional neural network," *Journal of Multimedia Information System*, vol. 6, p. 165–172, 2019. https://doi.org/10.33851/jmis.2019.6.4.165

[12] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao and J. Chen, "DL-IDS: Extracting features using

CNN-LSTM hybrid network for intrusion detection system," *Security and communication networks*, vol. 2020, 2020. https://doi.org/10.1155/2020/8890306

[13] C.-M. Hsu, M. Z. Azhari, H.-Y. Hsieh, S. W. Prakosa and J.-S. Leu, "Robust network intrusion detection scheme using long-short term memory based convolutional neural networks," *Mobile Networks and Applications*, vol. 26, p. 1137–1144, 2021. https://doi.org/10.1007/s11036-020-01623-2

[14] M. A. Khan, "HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, p. 834, 2021. https://doi.org/10.3390/pr9050834

[15] T. Thilagam and R. Aruna, "Intrusion detection for network based cloud computing by custom RC-NN and optimization," *ICT Express*, vol. 7, pp. 512-520, 2021. https://doi.org/10.1016/j.icte.2021.04.006

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, p. 504–507, 2006. https://doi.org/10.1126/science.1127647

[17] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, p. 154–170, 2017. https://doi.org/10.1016/j.jsv.2016.10.043

[18] O. Abdeljaber, O. Avci, M. S. Kiranyaz, B. Boashash, H. Sodano and D. J. Inman, "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data," *Neurocomputing*, vol. 275, p. 1308–1317, 2018. https://doi.org/10.1016/j.neucom.2017.09.069

[19] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021. https://doi.org/10.1016/j.ymssp.2020.107398

[20] G. Van Houdt, C. Mosquera and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, p. 5929–5955, 2020. https://doi.org/10.1007/s10462-020-09838-1

[21] I. A. Abdulmajeed and I. M. Husien, "Machine Learning Algorithms and Datasets for Modern IDS Design," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 2022. http://doi.org/10.1109/CyberneticsCom55287.2022.9865255

[22] R. Yao, N. Wang, Z. Liu, P. Chen and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, p. 626, 2021. https://doi.org/10.3390/s21020626

[23] Zhang, Hongpo and Huang, Lulu and Wu, Chase Q and Li, Zhanbo, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177, p. 107315, 2020. https://doi.org/10.1016/j.comnet.2020.107315

[24] L. Wang, M. Han, X. Li, N. Zhang and H. Cheng, "Review of classification methods on unbalanced data sets," *IEEE Access*, vol. 9, p. 64606–64628, 2021. https://doi.org/10.1109/access.2021.3074243

[25] A. Ampountolas, T. Nyarko Nde, P. Date and C. Constantinescu, "A Machine Learning Approach for Micro-Credit Scoring," *Risks*, vol. 9, 2021. https://doi.org/10.3390/risks9030050

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and others, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, p. 2825–2830, 2011. https://doi.org/10.48550/arXiv.1201.0490.

[27] B. Cao, C. Li, Y. Song, Y. Qin and C. Chen, "Network Intrusion Detection Model Based on CNN and GRU," *Applied Sciences*, vol. 12, p. 4184, 2022. https://doi.org/10.3390/app12094184

[28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, p. 1157–1182, 2003.

[29] A. Jović, K. Brkić and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, 2015. https://doi.org/10.1109/mipro.2015.7160458