Progress in Artificial Intelligence
https://doi.org/10.1007/s13748-025-00398-5

REGULAR PAPER

A transformer-based hierarchical learning model for the detection of cybersecurity threats

J. M. Martínez-Ramírez[1] · M. Lucena[1] · O. Cordón[2] · C. J. Carmona[1,3,4]

Received: 12 March 2025 / Accepted: 19 August 2025

Abstract

Cybersecurity is often stagnant, fighting a silent war against new attacks while developing much more slowly than other technologies. Due to the wide variety of attacks that we can find in technology, several branches of cybersecurity have also appeared. Deep learning has recently emerged as the machine learning technology best suited to predicting these attacks. This contribution presents a new hierarchical model based on deep learning able to handle accurately two different cybersecurity threat detection tasks. First, it will determine whether a given connection is an attack or not, thus dealing with a binary classification problem. Then, it will classify the malicious connections within a family of attacks. The proposed model offers accurate results when compared with those of state-of-the-art proposals, especially in the second task tackled. This study has been tested on three different datasets of real attack data, obtaining predictions with an accuracy of 99.92% on dataset CIC-IDS2017, 98.39% on CIC-CSE-IDS2018 and 93.74% on CIC-DDoS2019.

Keywords Cybersecurity · Intrusion detection · Deep learning · Transformers · Hierarchical models

1 Introduction

As technologies develop, so do attacks that are based on these technologies or that are specifically designed to target them. As such, cybersecurity is of utmost importance in detecting attackers and defending against them. However, cybersecurity is often overlooked to focus on improving efficiency or efficacy [1]. As a consequence of the quick development of the discipline, many new attacks cannot be detected by classic defense mechanisms. Efforts should focus on improving these mechanisms so that they can overcome modern threats. For this purpose, paradigms such as classical machine learning and more specifically deep learning can be used to analyze large amounts of data and design models to resolve these issues [2].

A wide variety of disciplines are encompassed under cybersecurity. In [3], the main branches of knowledge are defined as user access authentication, network situation awareness, abnormal traffic identification and dangerous behavior monitoring. Furthermore, dangerous behavior monitoring is defined as the task typically performed by intrusion detection systems (IDS), which will be the main focus of this article.

As such, it is key to analyze the behavior of state-of-the-art algorithms for intrusion detection that are supported by classical machine learning and deep learning. Furthermore, it is important to analyze if algorithms can discern not only whether traffic is considered benign or malicious, but also discriminate between different families of attacks based on

M. Lucena and O. Cordón contributed equally to this work.

✉ J. M. Martínez-Ramírez
jmmr0049@red.ujaen.es

M. Lucena
mlucena@ujaen.es

O. Cordón
ocordon@decsai.ugr.es

C. J. Carmona
ccarmona@ujaen.es

[1] Department of Computer Science, University of Jaen, Campus Las Lagunillas, 23071 Jaén, Spain

[2] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, Avenida del Conocimiento, 37, 18016 Granada, Spain

[3] Andalusian Research Institute in Data Science and Computational Intelligence, University of Jaén, Campus Las Lagunillas, 23071 Jaén, Spain

[4] Leicester School of Pharmacy, DeMontfort University, Gateway House, LE1 7RH Leicester, United Kingdom

Published online: 27 August 2025

connection data, such as the number of packets sent, flags active, etc.

Deep learning is being used in a number of activities, such as the generation of content or the detection of certain patterns on data [4, 5]. These new technologies may cause the amount and complexity of cyberattacks to increase, but can also be applied to mechanisms such as IDS, particularly to increase their ability to detect new attacks (known as 0-day attacks) [6]. By using deep learning applied to cybersecurity, IDS may be able to keep up with these attacks, ensuring an effective deployment of these systems that allows users to manage the high number of alerts, ideally offering accurate predictions that reduce the amount of false positives, which are the greatest issue when it comes to IDS, as they need to be manually reviewed by humans, consuming a vast amount of time. This study contributes to the research on these topics by experimenting with several machine learning and deep learning algorithms in an attempt to determine which of them are most effective when tackling intrusion detection, as well as attempting to implement a hierarchical model that is able to take advantage of the specific capabilities provided by the models that offer the most accurate results.

With this purpose, a new Hierarchical Learning Model for IDS (HiLMIDS) is proposed. This model is based on both classical machine learning methods and deep learning algorithms fine-tuned to efficiently and accurately detect whether a given connection is benign or malicious. As a first stage, the system solves a binary problem in order to discriminate between benign and malicious traffic. Furthermore, if a connection is detected as malicious, it is forwarded onto a second stage, where the model can discriminate between several families of attacks in order to predict which one the connection belongs to. A full experimental study with real attack data from the years 2017 to 2019, including more than 25 kinds of attack and over 5.5 million different instances of connections throughout 3 different datasets, is included. These datasets are commonly used to train state-of-the-art technologies, and as such allow for an easy comparison between our model and those existing in the specialized literature.

The HiLMIDS model was compared to Random Forest (RF), Multilayer Perceptron, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Transformer (TRANSF) models, improving their results by a large margin (up to 30% in the case of the MLP when classifying traffic as malicious or benign, and up to 70% in more complicated tasks, such as discriminating between different families of attacks). It will also be compared to more complex models in the specialized literature.

The remainder of the manuscript is structured as follows: Section 2 includes information about the state-of-the-art of deep learning applied to cybersecurity, Section 3 offers additional information about the proposed hierarchical model for dangerous behavior monitoring, Section 4 presents the experimental framework upon which all models have been built, Section 5 presents the experimental results and Section 6 concludes this paper.

2 Background

The state-of-the-art of artificial intelligence (AI) models used in cybersecurity and their limitations will be presented in Section 2.1. Afterwards, the four branches of knowledge of cybersecurity will be presented in Section 2.2, as well as several applications for each of them.

2.1 Advanced models for cybersecurity

Many of the state-of-the-art models used in cybersecurity are based on AI. As such, they present some limitations that are related to this paradigm. In order for AI to be trustworthy, it must be both robust and easily explainable; however, among the main issues lies the fact that most models return completely different results when the data they were trained with are modified, even if very slightly [7]. As a consequence, generative adversarial networks (GAN) can be used to create examples of malicious traffic artificially which are not detected as such [8, 9]. The most common models will be analyzed hereunder:

- The **Random Forest** model [10] consists of an ensemble (a model made out of several, simpler models) based on tree-type models, such as C4.5 [11], to obtain a final result.
- The **Multilayer Perceptron** [12] is the simplest neural network. The basic structure of a perceptron is insufficient for solving particularly complex predictions. To improve the results, several hidden layers are added, thus forming a MLP.
- **Convolutional Neural Networks** [13] are one of the most commonly used deep learning models, mainly when working with multidimensional data. A first layer groups all data in a given data structure - a tensor - which will be forwarded to the convolution layer. There, a series of mathematical transformations will take place, obtaining a combination of input data as output.
- **Recurrent Neural Networks** [14] are able to remember information of every element used during the training ("tokens") as they receive it, and use this extra information when attempting to predict future tokens. They often use techniques such as keeping a system state [15], the usage of backpropagation through time [16] or long short-term memory (LSTM) [17].
- **Transformers** are often used in complex tasks [18] as an attention network that can be trained and used simulta-

neously. Each element is placed upon an n-dimensional space in which elements with a given correlation occupy nearby places (embedding), which allows vector logic to be applied regardless of data. The most common kind of embedding is ELMo [19], which uses a structure similar to two neural networks, one of which has forward propagation while the other one has backward propagation, creating a behavior akin to that of a bidirectional LSTM.

2.2 Main approaches in cybersecurity

In this section, the four branches of knowledge in cybersecurity will be briefly explained. These branches are user access authentication, network situation awareness, abnormal traffic identification and dangerous behavior monitoring. Moreover, several applications will be revised for each branch.

User access authentication aims to develop ways to detect camouflage maneuvers or users that attempt to impersonate others, whilst also providing new methods for users to prove that they are who they claim to be.

Developments include the use of neural networks to gather additional information on passwords used for login [20], or the use of RF for multiauthentication [21]. In [22], a deep learning model that verifies users in two stages is proposed. Some biometry-based methods are also proposed in [23].

Network situation awareness attempts to gather information about the complete data and packet flows of a given network, discerning a set of key parameters that define the different possible network states.

Many models that attempt to solve this task apply AI-based technologies, as seen in [24]. Some specific models include CNN and Gated Recurrent Units (GRU), as seen in [25].

Abnormal traffic identification encompasses all research that attempts to detect attacks based around the injection of large amounts of data onto a network causing a denial of service (DoS).

Throughout time, several other attempts to detect abnormal traffic appeared. Most of them applied CNN [26, 27], oftentimes combined with other techniques such as wavelet decomposition and RNN [28].

Dangerous behavior monitoring: the objective of this branch, which will be the main focus of our research, is the detection of attacks targeting a specific system and the defense of its “lethal points”, defined as particularly vulnerable elements of the system that are easier to exploit. Given the amount of plausible attacks, there are two main paths of research:

- Global models attempting to detect attacks regardless of their family. These often involve deep learning techniques combined with other models, such as SVM [29], fuzzy logic [30], gradient boosting trees [31] and blockchain technologies [32]. Specifically for malicious network connections, the nearest-neighbor algorithm is combined with SMOTE and hybrid neural networks in [33] (SMOTE-ENN). Auto-Encoders are also often used in anomaly-based IDS [34] (AE-2). Other methods, such as Recurrent DL [35] or Deep Belief Networks [36] are also common. In || [37], a Light Gradient Boosting Machine (LGBM) is used to predict malicious connections in IoT environments. Lastly, the most predominant algorithms make use of LSTM-based models, such as a CNN that is supported by a LSTM (CNN-LSTM) [38] or a LSTM trained by using GAN [39] (LSTM-GAN).
- Models for specific attacks. These models are based around large amounts of stored data, and are often supported by platforms such as Hadoop or Spark. Therefore, data must be analyzed and studied in order to tackle issues correctly [40]. Several proposals, such as [41], include full frameworks spanning several stages whilst some simpler models make use of deep learning models, such as Transformers [42]. Some proposals focus on DDoS attacks and make use of either DL techniques such as Deep Neural Networks (DNN) or CNN-AE [43] or contractive AUE [44].

3 A Hierarchical Learning Model for Intrusion Detection Systems (HiLMIDS)

As discussed previously, cybersecurity mechanisms must be kept up to date in order to keep up with the amount of new attacks and specific exploits that keep appearing as technology develops. Since classic paradigms have been proven to offer worse results when it comes to the detection of newer attacks, making use of newer technologies (such as deep learning) to improve the scalability and functionality of previous mechanisms will be key to defend from attacks.

In order to detect attacks in a more efficient manner, a hierarchical learning model is proposed. The structure of the proposed model is depicted graphically in figure 1. As described, the model is able to accurately detect both if a given system is under attack, and the specific family to which the attack belongs to.

The HiLMIDS model first receives data regarding different traffic which was used to train both of the models used in subsequent stages. This data is first processed to remove noise and improve the training of the model. In order to minimize bias and properly assess each model, they were trained and evaluated applying stratified cross validation, so that for every iteration of the loop, the entire dataset was divided into five subsets, referred to as folds, which contain a distribution of classes that is as similar as possible to the original

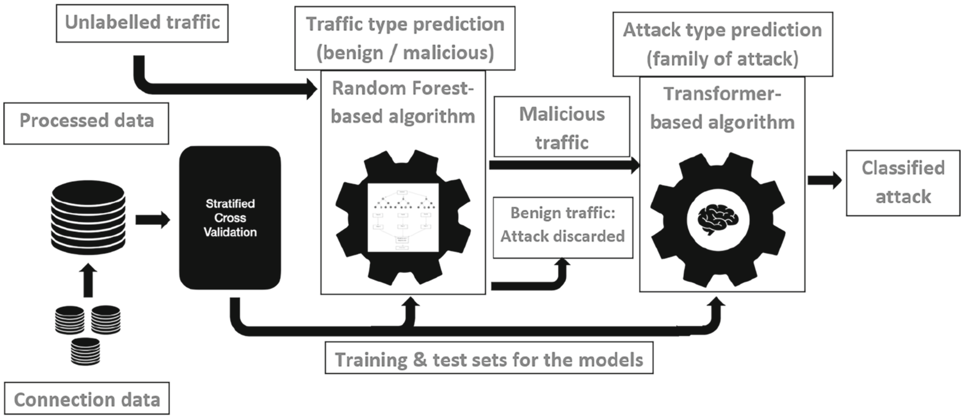
Unlabelled traffic
Traffic type prediction (benign / malicious)
Attack type prediction (family of attack)
Random Forest-based algorithm
Malicious traffic
Transformer-based algorithm
Processed data
Stratified Cross Validation
Benign traffic: Attack discarded
Classified attack
Training & test sets for the models
Connection data

Fig. 1 Structure of the proposed hierarchical model

data. Four of these folds were used as the training set and the remaining one was used as the test set.

This training process is used for both sub-models. The first stage makes use of a RF-based algorithm which can filter all connections discriminating between benign and malicious ones regardless of the different kinds of attack. If all traffic is classified as benign, the model will stop its execution. Only malicious predictions are stored and then forwarded to the second stage of analysis, where a TRANSF-based model will attempt to differentiate between different kinds of malicious connections, attempting to determine the general family of the attack (i.e DDoS) that the traffic belongs to. Both of these models were chosen after testing them alongside all other models mentioned in section 2.1 on all datasets that will be relevant to this study, as they were the ones that provided the highest accuracy for that specific task. Further details of the experimental study and results can be found in sections 4 and 5.1, respectively.

Using such a model offers several advantages: First, it allows us to make good use of the stronger points of each model used. Furthermore, if a model is particularly suited for a given task in the workflow, it can be relegated to doing that task specifically, leaving the rest to another model.

Although models are used for either general attacks or specific families, the two kinds are seldom seen working together. Combining this strategy with newer classification techniques, such as ensemble models or deep learning, the objective is to find a new model that can match the results of state-of-the-art models for each individual task whilst also being able to perform both as a general model and a specific one.

The detection based on two stages is also more efficient. Binary problems are easier to solve, and as such, take less time. As the first stage consists of detecting whether a connection is benign or malicious, the model only needs to continue running if the traffic is classified as malicious, which makes it more efficient and less energy-consuming.

The two main models used for the proposed hierarchical model are type of traffic and type of attack. Their specific operation is described in sections 3.1 and 3.2, respectively. Afterwards, the algorithm designed will be shown in section 3.3.

3.1 Type of traffic

A RF-based model [10] will be used during the first stage of the hierarchy in order to discriminate between benign and malicious traffic.

For each tree generated, a subset of all attributes is considered instead of the full set. After considering all trees, the results are unified by using a weighted sum in which every tree has an associated weight based on its relevancy to predictions during the training of the model, thus returning the class deemed most appropriate. This allows for an accurate first assessment that may detect malicious traffic even if the attack is unknown, which is particularly important when it comes to the detection of 0-day attacks, one of the main foci of our system.

Since every tree considers different subsets of attributes and the information is later combined, the results tend to be more robust and the ill effects of both overfitting and issues related to the large amount of attributes being worked with are mitigated. The Random Forest model that was used consisted in 100 J48 trees and used the gini index as the main criterion considered in order to build each tree. Additionally, for each node, up to nine features were considered (as it is the square root of the total number of features). No maximum depth was established for the trees. These hyperparameters were selected by thorough experimentation and fine-tuning in order to achieve a higher accuracy.

Furthermore, since in this stage the specific family of attacks the connection might belong to is not considered, RF is able to detect even 0-day attacks due to them sharing certain characteristics with known attacks or being significantly different to what is known about benign connections.

Thus, a whitelist-like behavior is shown for the first stage: any connection that shares little to no similarities to those known as benign is considered malicious, which allows the user to attempt to protect their system.

Lastly, this also makes it so that the different stages of the model can be trained more easily. If a new attack is discovered as such but its effects are not fully known, it can be given to this first RF-based model exclusively, so that similar attacks can be detected at this first stage.

3.2 Type of attack

For the second stage, a TRANSF-based algorithm will be used in order to discriminate between the different families of attack.

Large language models (LLM) such as TRANSF are relatively recent, having just been proposed in 2017 [45]. As such, they have not been thoroughly tested in any of the previously-discussed branches of knowledge related to cybersecurity, since their original purpose was a different one thus making them a novelty in this field.

However, it has been proven that TRANSF can be used for a variety of tasks that are not limited to natural language processing [18], including the monitoring of dangerous behavior. For the purpose of that specific task and HiLMIDS, the best results were achieved with a modified BERT [46] (BERT-base-cased) from HuggingFace [47], which was afterwards trained using K-fold validation as described in section 3. This model was chosen due to it having a large bibliography from which to gather information, as well as its simplicity of use due to HuggingFace methods. It additionally allows for an easier way to process data, as users can simply paste the string associated to one connection and need not substantially change it.

For training, the model was fed a string that included all relevant information of one singular connection, and was asked to predict the family of the attack that was associated to that information. The learning rate of the model was set to $lr = 4 * 10^{-4}$, the model ran for only one epoch in order to avoid overfitting towards the majority classes, used GELU as an activation function, included 12 hidden layers and 768 cells per hidden layer. Additionally, each weight was initialized with a standard deviation of 0.2.

The TRANSF-based model used is extremely accurate when it comes to discerning different families of attack, and can furthermore make predictions at high speeds. However, the training process is slow. Nonetheless, since the training must be carried out only when there is new data to extract information from, this long training time is not a major drawback.

Through experimentation it has been proven that a TRANSF-based model offers higher accuracy than RandomForest specifically when dealing when the classification of attacks into several families, though it offers much worse results when classifying connections as benign or attacks. Therefore, since the RF-based model forwards only malicious connections to the TRANSF, it can easily discern the families of attacks while ignoring its main drawback.

3.3 Pseudocode

This section includes the pseudocode for the hierarchical model proposed, as well as a detailed explanation of all the functions it includes.

Algorithm 1: Workflow of the proposed hierarchical model.

```
Input: <Dataset> traffic Result: Classified input traffic
rawData ← readVariables(traffic);
data ← preprocessing(rawData);
maliciousPredictions[];
attackPredictions[];
allPredictions[] ;
for element in data do
    prediction ← trafficTypeModel(element);
    allPredictions ← allPredictions.add(element + prediction);
    if prediction is malicious then
        maliciousPredictions ←
        maliciousPredictions.add(element);

end
if maliciousPredictions is not empty then
    for element in maliciousPredictions do
        attackPrediction ← attackTypeModel(element);
        attackPredictions ← attackPredictions.add(element +
        attackPrediction);
    end

show(allPredictions);
show(attackPredictions);
```

As seen in algorithm 1, upon receiving data corresponding to web traffic, it is pre-processed. Most of the datasets are highly imbalanced, with minority classes containing as few as 11 examples whilst majority classes had over 1 million. As such, in order to reduce the differences, undersampling techniques were applied to majority classes in the multiclass problem, reducing the amount of examples each had by applying a distribution-based balance so that every class was reduced based on the number of examples it originally had. The majority class for each dataset was reduced by 50%, whilst all other classes were reduced less according to the number of examples they presented.

While oversampling techniques were considered for minority classes they were not applied since it would have been necessary to create too many artificial examples, thus introducing too much noise and reducing the overall quality of the results. No further pre-processing techniques were applied, as feature scaling may cause a loss of information due to the ample variety of attributes, and any single attribute may be relevant for distinguishing specific kinds of attack though not useful for others. Each dataset may have undergone specific changes, which can be found in subsection 4.1.

The model can be trained afterwards. The hierarchical model first uses the RF-based algorithm to discriminate whether it is benign traffic or malicious traffic. Once the model has analyzed all traffic, if there are no malicious examples the program ends. Otherwise, the TRANSF-based model, which is trained with data from several families of attack, attempts to discriminate the specific kind of attack. The predictions of the model are afterwards shown for the users to see.

For the training of the models, data were split between training and test subsets. The first of them was used to train the model, whilst the second was used to check the accuracy of its predictions. For certain models, the training process took place throughout several epochs so that the model was properly fitted to the data and its predictions improved. When applicable, an inner loop that specifically included the training of the model was inserted.[1]

4 Experimental study

This section includes all relevant information regarding the framework used during the experimentation. It additionally includes the experimental results and the analysis and discussion of the results obtained.

4.1 Datasets

In this section, some of the datasets with relevant information which will be used to train the models used will be presented. These datasets have been widely used in state-of-the-art researches.

The three datasets used are CIC-IDS2017, CIC-CSE-IDS2018 and CIC-DDoS2019. They were created by the Canadian Institute of Cybersecurity (CIC) using CICFlowMeter [48] to extract specific data from a network, including detailed flows, their duration, protocol used, etc. For each connection, a label that marks it as either benign or an attack (and which type of attack, if so) is also provided. Since both CIC-IDS2017 and CIC-CSE-IDS2018 had several known annotation issues, the datasets were manually corrected according to known issues.

It should be noted that, as all the datasets were gathered by using the same tool, they all provide roughly the same information, which consists of a series of real numbers each representing a parameter such as information on data flows, connection protocols, length of packages, flag counts, statistical calculations on previous parameters, etc. The CIC-CSE-IDS2018 additionally included a timestamp for every connection, and data was organized based on timestamp. This meant that all attacks of a similar family shared a similar timestamp, and as such, that attribute was removed to ensure unbiased predictions. All other characteristics of all the datasets were deemed potentially relevant in order to make predictions and, as such, no other characteristics were removed, leaving 78 characteristics as well as the class of each example. Some elements in the dataset are the following:

- 88, 773, 9, 4, 612, 2944, 306, 0, 68, 134.9333169, 1472, 0, 736, 849.8595962, 4600258.732, 16817.59379, 64.41666667, 148.698, 531, 1, 773, 96.625, 196.665, 580, 1, 675, 225, 348.901, 627, 1, 0, 0, 0, 0, 204, 104, 11642.94955, 5174.644243, 0, 1472, 254, 527.5207615, 278278.1538, 0, 0, 0, 1, 0, 0, 0, 0, 0, 273.5384615, 68, 736, 204, 0, 0, 0, 0, 0, 0, 9, 612, 4, 2944, 8192, 2053, 2, 20, 0, 0, 0, 0, 0, 0, 0, 0, BENIGN.
- 22, 13652185, 22, 33, 2008, 2745, 640, 0, 91.27272727, 138.182, 976, 0, 83.181, 217.2857356, 348.1493988, 4.0286, 252818.2407, 633086.5034, 2178689, 3, 11600000, 550887.1429, 869278.0704, 2232461, 839, 13700000, 426630.4063, 782897.5727, 2178689, 3, 0, 0, 0, 0, 712, 1064, 1.611463659, 2.417195489, 0, 976, 84.875, 186.8396554, 34909.05682, 0, 0, 0, 1, 0, 0, 0, 0, 1, 86.41818182, 91.27272727, 83.18181818, 712, 0, 0, 0, 0, 0, 0, 22, 2008, 33, 2745, 29200, 247, 16, 32, 0, 0, 0, 0,0, 0, 0, 0, SSH-Patator.

The CIC-IDS2017 has around 1.75 million examples of benign traffic and around 500.000 attacks, CIC-CSE-IDS2018 has over 2 million benign examples and around 1 million attacks, and CIC-DDoS2019 has about 100.000 benign examples and over 300.000 attacks.

Moreover, given fields of the data throughout all three datasets were values that could not be worked with, as some attributes had values of either NaN or infinity. These two values always appeared together in the same two attributes, usually in standard deviations or average flow speeds. As these values may only be positive, infinity values were replaced with the highest possible value that Python allowed, while NaN values were replaced with -1. Removing examples with these data was considered, particularly if they were prevalent in majority classes in order to further balance the datasets, but since they appeared in both majority and minority classes, they were kept.

Lastly, all datasets underwent specific changes. For instance, CIC-IDS2017 had its data presented throughout

[1] The source code for the algorithm can be found in the following link: https://github.com/jmmartinezr/HiMIDS.

Table 1 Specifications of the computer used for the experiments

Component	Model
Operative System	Windows 10 Home
CPU	Intel Core i7-10875h CPU @ 2.30GH
RAM	32 GB
Storage	SSD 1 TB
Graphics card	NVIDIA GeForce RTX 2070

different files. One of these files included exclusively benign data, which was not considered during experimentation to further balance that specific dataset. CIC-DDoS2019 had two classes for a family of attack, UDP-Lag and UDP Lag. Since both classes behaved very similarly, they were combined, as the different naming was considered a human mistake made while preparing the dataset.

4.2 Relevant tools

All models were implemented in Python 3.13. The Random Forest model was built by using the Scikit-Learn framework for python [49] then trained with the datasets described in section 4.1 as described in section 3.1. The Transformer model was built using the BERT-case-based transformer provided by HuggingFace [47] as a starting point, then trained with the datasets as described in section 3.2.

All experiments were made on the same computer to ensure a fair comparison of the results for each model. The details on the hardware are present in table 1.

4.3 Experimentation

The training and evaluation process of a given model requires both a given dataset and an execution mode. The dataset can be any file in CSV format that contains information structured like any of the previously mentioned datasets. Execution modes are defined by a string that determines how data will be processed during training and evaluations that can take three values:

- **BinaryClassificationProblem**: Specific kinds of attack are ignored, and all types of attack are grouped under the "Malicious" label. This execution mode is designed so that models can differentiate between malicious and benign traffic.
- **MaliciousOnly**: All benign traffic is ignored, and only attacks are considered. This execution mode is designed so that models can differentiate between different types of attack.
- **HiLMIDS Study**: This execution mode attempts to differentiate between benign traffic and all possible types of attack in the previously defined two-stage process. It was used to test the accuracy of HiLMIDS against current state-of-the-art algorithms and models.

Furthermore, when deciding which model was the most appropriate for each execution mode, two main metrics were considered: accuracy and recall [50], which are the metrics most used in classification tasks such as this. Moreover, an additional performance metric, execution time, was considered. All metrics used are described here:

- **Precision**: accuracy can be defined as the percentage of cases in which the model is able to correctly predict the label of a given example. Since it is of utmost importance that the model can differentiate between benign and malicious traffic and the type of attack when relevant, this is the main metric considered regardless of the execution mode.
- **Recall**: due to the specific details of the data, it is imperative to be able to discern whether a given connection is or is not an attack. It is much more harmful to wrongly classify an attack as a benign connection than vice versa. As such, recall will also be considered when analyzing the results of a given model, as this represents the ratio of true positives (TP).
- **F1 Score**: The F1 score is the harmonic mean of precision and recall, thus symmetrically representing both in one metric.
- **Execution time**: if a system was is attacked, it is of utmost interest that the attack can be detected as such as soon as possible. Since most of the models presented are deep learning models, the time training process is often quite slow. However, once trained they are able to make predictions in an acceptable amount of time. Due to this, the time required to train a model will be considered, but it is not deemed critical. However, the amount of time taken for predictions will be of utmost importance when it comes to deciding which is the best model.

5 Results and analysis

In this section, the results for each combination of execution mode and model will be shown along with further discussion of the experimental results obtained.

5.1 Results

Due to the nature of the proposal, the results will be divided into different parts. First, the results for each individual model for both BinaryClassificationProblem and MaliciousOnly will be shown. Models used include RF, MLP, CNN,

Table 2 Execution times for each model and dataset

	RF		MLP		CNN		RNN		TRANSF	
	Training	Test	Training	Test	Training	Test	Training	Test	Training	Test
CIC-IDS2017	37s	5s	40s	6s	70s	6s	81s	9s	384s	22s
CIC-CSE-IDS2018	57s	9s	61s	11s	118s	10s	131s	15s	561s	31s
CIC-DDoS2019	29s	3s	32s	5s	59s	4s	74s	8s	315s	18s

RNN and TRANSF. Afterwards, the hierarchical model will be compared to every other individual model using the HiLMIDS analysis.

In each table, the amount of instances of each dataset will also be reported. This information is labelled as N_s. Additionally, the execution time for each model and dataset can be found in table 2.

As it can be seen, the proposed model takes longer for both the training and test sets. While this may be a disadvantage, the main difference in runtime is caused during the training, which does not need to be done frequently and, as such, it can be accepted. The time to predict whether a given connection is an attack is roughly twice as much as other models, though it should be noted that this includes both stages of the hierarchical model. Discerning between a benign connection and a malicious one takes roughly the same amount of time as the other models, while specifically predicting a type of attack makes up for the remainder of the runtime.

On the other hand, when considering the amount of memory required by each model, it should be noted that most models required very high resources during the training stage, and up to 99% of available memory for RNN, the transformer model and the HiLMIDS model. However, the testing stage for all models is not particularly memory-consuming, and is oftentimes below 50% of the available memory even in the most complex models, such as transformers and HiLMIDS. This is partly due to the hierarchical nature of the model, which allows it to only make use of only either Random Forest or the TRANSF-based model based on the task that is being handled.

5.1.1 BinaryClassificationProblem

The results of the binary problem executed on every model appear in table 3.

As can be seen, for this first execution mode, RF obtains far more accurate results than any other model. Models such as the MLP and the RNN can reach around 90% accuracy on certain datasets, but are still far from the results obtained by RF. CNN and TRANSF can achieve 100% accuracy on certain occasions, but it is because they are overfitted and only predict the majority label.

Among the most important elements that Random Forest detected as relevant to discern whether a connection is benign or malicious are the following characteristics:

- **Flag count**. Generally speaking, whenever a flag has a high count, the connection is considered malicious by the RandomForest algorithm.
- **Forwarded packets per second**. If the number of packets exceeds a certain threshold, roughly 5000 packets, the connection is also deemed malicious.
- **Flow duration**. Longer data flows are most commonly associated with malicious connections, as it may represent not finishing a connection to saturate a server or sending a massive amount of packets.
- **Flow size**. A flow that handles large amounts of data per second (roughly 250 MB / s) is often associated with benign connections. Flows with less data, or flows with roughly that data that consist on smaller packets (about 350 B / packet) are often associated with malicious connections.

If the data that was to be processed lacked some of these elements, the model would likely offer results of a lower accuracy, as they are oftentimes associated with attacks. However, since all these values can be easily gathered by using CICFlowMeter, it should not be a hindrance in most practical cases as a correct network data capture is ensured.

5.1.2 MaliciousOnly

This section includes the detailed results of the MaliciousOnly execution mode. In order to simplify the information presented, it is divided into three tables, one for each dataset. Results for **CIC-IDS2017** can be found in table 4, results for **CIC-CSE-IDS2018** can be found in table 5 and results for **CIC-DDoS2019** can be found in table 6.

In this specific case, it can be seen that while RF continues to offer a remarkable degree of accuracy, the TRANSF can surpass its results in both the CIC-CSE-IDS2018 dataset and the CIC-DDoS2019 dataset. While the difference is of less than 2%, this means correctly detecting 3000 more attacks on the first dataset and around 1600 on the second one.

Table 3 Results of all models using BinaryProblem

			RF			MLP			CNN			RNN			TRANSF		
	N_s		Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
CIC-IDS2017	348636	Benign	0.999	**0.999**	**0.999**	0.889	0.747	0.812	**1.000**	0.758	0.862	0.979	0.797	0.879	**1.000**	0.758	0.862
	111529	Malicious	**0.998**	**0.999**	**0.998**	0.058	0.143	0.083	0.000	-	0.00	0.221	0.773	0.344	0.000	-	0.00
CIC-CSE-IDS2018	422071	Benign	0.999	**0.999**	**0.999**	0.704	0.744	0.723	**1.000**	0.681	0.810	0.9208	0.783	0.846	**1.000**	0.681	0.810
	197359	Malicious	**0.999**	**0.999**	**0.999**	0.481	0.432	0.455	0.000	-	0.00	0.454	0.725	0.558	0.000	-	0.00
CIC-DDoS2019	19566	Benign	**0.999**	**0.999**	**0.999**	0.746	0.956	0.838	0.000	-	0.00	0.600	0.836	0.699	0.000	-	0.00
	66709	Malicious	0.999	**0.999**	**0.999**	0.990	0.930	0.959	**1.000**	0.773	0.872	0.966	0.892	0.928	**1.000**	0.773	0.872

Table 4 Results for dataset CIC-IDS2017 using MaliciousOnly mode

		RF			MLP			CNN			RNN			TRANSF		
	N_s	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
DDoS	25606	**1.000**	**1.000**	**1.000**	**1.000**	0.230	0.374	**1.000**	0.230	0.374	0.455	0.907	0.606	0.999	**1.000**	0.999
Portscan	31789	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.997	0.951	0.973	0.999	0.999	0.999
Bot	393	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	0.000	-	**1.000**	**1.000**	**1.000**
Infiltration	7	**0.857**	**1.000**	0.923	0.000	-	-	0.000	-	-	0.000	-	-	0.000	-	-
WebAttack Bruteforce	299	0.784	**0.724**	0.753	0.000	-	-	0.000	-	-	0.000	-	-	**0.983**	0.688	0.809
WebAttack XSS	130	**0.331**	**0.402**	0.363	0.000	-	-	0.000	-	-	0.000	0.000	-	0.000	0.000	-
WebAttack SQL Injection	4	**0.750**	**1.000**	0.857	0.000	-	-	0.000	-	-	0.000	-	-	0.000	-	-
FTP Patator	1587	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	-	-	0.999	**1.000**	0.999
SSH Patator	1180	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	-	-	0.999	**1.000**	0.999
Slowloris	1159	**0.994**	0.991	0.992	0.000	-	-	0.000	-	-	0.000	-	-	0.991	**0.992**	0.991
SlowHttpTest	1099	0.987	**0.996**	0.991	0.000	-	-	0.000	-	-	0.000	0.000	-	**0.988**	0.993	0.990
DoSHulk	46215	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.955	0.690	0.801	**1.000**	0.999	0.999
GoldenEye	2059	**0.997**	**0.996**	0.996	0.000	-	-	0.000	-	-	0.000	0.000	-	0.992	0.994	0.993
Heartbleed	2	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	0.000	-	0.000	-	-

Table 5 Results for dataset CIC-CSE-IDS2018 using MaliciousOnly mode

		RF			MLP			CNN			RNN			TRANSF		
	N_s	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
FTP Bruteforce	38672	0.915	0.784	0.844	**1.000**	0.196	0.328	**1.000**	0.196	0.328	**1.000**	0.511	0.676	**1.000**	**0.792**	**0.884**
SSH Bruteforce	37518	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.499	0.707	0.585	**1.000**	**1.000**	**1.000**
Goldeneye	8302	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	-	-	0.998	**1.000**	0.999
Slowloris	2198	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.000	-	-	0.988	**1.000**	0.994
SlowHttp	18287	**0.466**	0.722	0.566	0.000	-	-	0.000	-	-	0.000	-	-	**0.446**	**0.998**	**0.616**
DoSHulk	92382	**1.000**	**1.000**	**1.000**	0.000	-	-	0.000	-	-	0.978	0.978	0.978	**1.000**	**1.000**	**1.000**

Table 6 Results for dataset CIC-DDoS2019 using MaliciousOnly mode

		RF			MLP			CNN			RNN			TRANSF		
	N_s	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
UDP	3618	0.692	**0.623**	0.656	0.000	-	-	0.000	-	-	0.000	0.000	-	**0.979**	0.617	**0.757**
MS SQL	1705	0.464	0.449	0.456	0.000	-	-	0.000	-	-	0.000	-	-	**0.967**	**0.631**	**0.764**
Portmap	137	**0.365**	**0.376**	**0.370**	0.000	0.000	-	0.000	0.000	-	0.000	-	-	0.000	0.000	-
Syn	9875	**0.992**	**0.993**	**0.992**	0.000	-	-	0.000	-	-	0.774	0.861		0.982	0.991	0.986
Netbios	129	0.341	0.232	0.276	0.000	-	-	0.000	-	-	0.000	0.000	-	**0.961**	**0.403**	**0.568**
UDPLag	1785	**0.762**	**0.824**	**0.792**	0.989	0.027	0.053	0.989	0.027	0.053	0.001	0.009	0.002	0.740	0.822	0.779
LDAP	381	0.223	0.228	0.225	0.000	-	-	0.000	-	-	0.000	0.000	-	**0.438**	**0.356**	**0.393**
DrDoS DNS	734	**0.454**	0.531	0.489	0.000	0.000	-	0.000	0.000	-	0.000	-	-	0.435	**0.573**	**0.495**
DrDoS WebDDoS	10	**0.000**	**0.000**	-	**0.000**	-	-	**0.000**	-	-	**0.000**	**0.000**	-	**0.000**	**0.000**	-
DrDoS TFTP	19783	**0.997**	**0.998**	**0.997**	0.000	-	-	0.000	-	-	0.993	0.744	0.851	0.996	0.997	0.996
DrDoS UDP	2084	**0.369**	**0.424**	**0.395**	0.000	0.000	-	0.000	0.000	-	0.000	0.000	-	0.000	0.000	-
DrDoS SNMP	543	0.622	0.625	0.623	0.000	-	-	0.000	-	-	0.000	0.000	-	**0.745**	**0.696**	**0.720**
DrDoS NetBIOS	120	**0.050**	**0.078**	**0.061**	0.000	-	-	0.000	-	-	0.000	0.000	-	0.000	0.000	-
DrDoS LDAP	288	**0.156**	**0.157**	**0.156**	0.000	-	-	0.000	-	-	0.000	0.000	-	0.000	0.000	-
DrDoS MSSQL	1242	0.281	0.269	0.275	0.000	-	-	0.000	-	-	0.000	-	-	**0.282**	**0.681**	**0.399**
DrDoS NTP	24274	**0.999**	**0.998**	**0.998**	0.000	0.000	-	0.000	-	-	0.983	0.766	0.861	0.997	**0.998**	0.997

With the objective of gathering more information about every model tested, a breakdown of the results will be shown afterwards. This breakdown includes more information about the performance of the model when used in each dataset. Further information, such as the confusion matrices for each model and dataset, can be found in A.

The RF model offers great accuracy in all cases, although it has some issues properly classifying minority classes, though it is capable of properly predicting their label in some cases. Unlike the RF model, the MLP offers poor results. This is due to the fact that most of the time it selects a majority class and always predicts that class, applying a ZeroR strategy that is unacceptable for this problem. Much like the MLP, the CNN also tends to overfit, always predicting a given majority class. As such, this model is discarded following the same reasoning as the previous one.

The RNN model offers better results than every other model shown until now except RF. Unlike the other models, although it tends to overfit towards the majority classes, it continues to be able to predict the minority ones, though with lesser accuracy. In spite of the poor results that the TRANSF offered in the BinaryClassificationProblem mode, it offers some of the best result in the MaliciousOnly mode. Its results are better than those offered by RF, if only marginally, and considerably better than any other model.

5.1.3 HiLMIDS Study

This section includes the results of using HiLMIDS, as well as both RF and TRANSF individually (i.e. an ablation study for our proposal), in order to analyze a given connection and determine whether it is benign or malicious, and the family of attack it may be encompassed in.

The study was executed as follows: HiLMIDS was used as explained in section 3. Both TRANSF and RF were used to directly discriminate between benign traffic and all individual families of attack, as well as attempting to differentiate between benign and malicious traffic directly, in order to obtain easily comparable results. Furthermore, several models from specialized literature where also compared to HiLMIDS. Results from models are directly extracted from the results provided by the authors. All comparisons appear in tables 7, 8 and 9. The highest accuracy achieved in each specific task is bolded.

As can be seen, for every single dataset, the hierarchical model offers greater accuracy than both the RF model and the TRANSF model individually. Although in some datasets, namely CIC-DDoS2019, this increase is extremely subtle, due to the sheer amount of traffic that is analyzed, it makes a difference of several hundred more attacks detected. In the other datasets, the difference is much more noticeable. The hierarchical model achieves results almost 2% higher than RF, which is the most accurate for both CIC-IDS2017 and CIC-CSE-IDS2018, and over 20% more than the TRANSF model which, as seen in previous sections, only offers satisfying results when detecting malicious traffic without considering benign connections.

Furthermore, when compared with other models from the literature, it can be seen that HiLMIDS offers results that are slightly more accurate than the best results from prior models. However, the prior models with the worst results are improved by up to 7% in CIC-IDS2017 and about 2-5% in the other datasets.

It should be noted that most advanced state-of-art models provide only the overall results for the model. As such, the results for the BinaryProblem configuration, that is, attempting to discern between benign and malicious predictions, is not provided and cannot be compared directly. The same can be said about the MaliciousOnly configuration. The only exception is the recurrent deep learning model [35], which was tested on CIC-IDS2017 providing a precision of 0.990 for the first configuration and a precision of 0.980 for the second configuration. Both of these are beaten by HiLMIDS, which achieves a precision of 0.999 and 0.998, respectively.

The deep bayesian network proposed in [36] achieves an accuracy only slightly worse than our own at a 0.998. In spite of this, the model lacks information for specific tasks (such as the BinaryProblem configuration), and HiLMIDS manages to outperform it by 0.001%.

The contractive autoencoder proposed in [44] provides results for both CIC-IDS2017 and CIC-DDoS2019. In the first dataset, the results provided are of an accuracy of 0.925, somewhat lower than the proposed model. However, in the CIC-DDoS2019 dataset, the results it provides are of a higher accuracy than our own with an accuracy of 0.961. Nevertheless, it should also be noted that, in spite of achieving a higher accuracy than HiLMIDS, the contractive AE proposed in [44] makes use of only a fragment of the dataset, as it considers only 6 kinds of attack instead of the full 16 that the dataset includes. No information was provided by the authors regarding the results using the 16 classes.

The fuzzy artificial neural network proposed in [30] achieves an accuracy of 0.887 in CIC-DDoS2019, which is considerably lower than that of HiLMIDS, achieving a 0.937 accuracy in the same task. The same can be said about the deep neural network proposed in [43], which achieves an accuracy of 0.870, and the CNN-AE model proposed in the same paper, which achieves an accuracy of 0.919, both lower than that of HiLMIDS.

The results of all advanced techniques that were presented are shown as provided by the authors. Though there is an algorithm that provides higher accuracy than our own, it should be noted that our model is the only one that can ideally achieve a high accuracy in all tasks: discerning between malicious and benign data and discerning between different attacks individually, as well as classifying connections as

Table 7 Comparison between HiLMIDS and other models in specialized literature for CIC-IDS2017

	Reference	BinaryProblem	MaliciousOnly	Overall results
RF	[10]	**0.999**	**0.998**	0.977
TRANSF	[18]	0.797	**0.998**	0.788
Recurrent DL	[35]	0.990	0.980	-
DBN	[36]	-	-	0.998
Contractive AE	[44]	-	-	0.925
HiLMIDS	This manuscript.	**0.999**	**0.998**	**0.999**

Table 8 Comparison between HiLMIDS and other models in specialized literature for CIC-CSE-IDS2018

	Reference	BinaryProblem	MaliciousOnly	Overall results
RF	[10]	**0.999**	0.935	0.974
TRANSF	[18]	0.681	**0.948**	0.681
HiLMIDS	This manuscript.	**0.999**	**0.948**	**0.984**

Table 9 Comparison between HiLMIDS and other models in specialized literature for CIC-DDoS2019

	Reference	BinaryProblem	MaliciousOnly	Overall results
RF	[10]	**0.993**	0.906	0.927
TRANSF	[18]	0.775	**0.916**	0.934
Fuzzy ANN	[30]	-	-	0.887
DNN	[43]	-	-	0.870
CNN-AE	[43]	-	-	0.919
Contractive AE	[44]	-	-	**0.961**
HiLMIDS	This manuscript.	**0.993**	**0.916**	0.937

either benign or a specific attack, while most other models are designed to achieve a high accuracy only in one of the aforementioned tasks.

Due to this, HiLMIDS has proved to be a robust model that can be used to detect cybersecurity threats in many circumstances, as it offers a high accuracy both when detecting whether a given connection is an attack and discerning the specific attack. As such, it could be used in any context that requires intrusion detection systems and that already makes use of a tool such as CICFLowMeter to gather network information that can be steadily provided to HiLMIDS in order to detect whether a given network is under attack.

5.1.4 Statistical tests

In order to discriminate whether one model actually offers statistically significantly better accuracy than another, a series of statistical tests was carried out. For every single test, $p = 0.05$ will be considered. The accuracy results that were taken into consideration for each test were the ones provided in sections 5.1.1 and 5.1.2, respectively, and thus considered the three datasets that were used during the experimental study. The statistical tests and their results were as follows:

- The first comparison was be a pairwise comparison. For this purpose, a Wilcoxon test was carried out between every possible combination of two models among all the models used during the prediction. Results showed that there was a statistically significant difference in accuracy between HiLMIDS and the Random Forest, MLP, CNN, RNN and Transformer models.
- The second and third comparison will be multiple comparisons between all models used, simultaneously. For this purpose, two tests will be carried out: Friedman test and post-hoc Holm test. These tests showed that there was a statistically significant difference in accuracy between HiLMIDS and RNN, CNN and MLP in all datasets used, and it also showed that HiLMIDS had statistically significantly better accuracy than both Random Forest and Transformers in at least two datasets for each model.

6 Conclusions

A general revision of the state-of-the-art regarding deep learning methods applied to cybersecurity has been presented in this study. Several deep learning techniques such as RNN and ANN, among others, are widely used in different cybersecurity problems, such as user identification, network situation awareness, abnormal traffic identification and dangerous behavior monitoring. This contribution focuses on dangerous behavior monitoring, and specifically malicious traffic detection and IDS.

In order to improve the classic IDS, which nowadays are often insufficient to defend systems from attackers, a novel Hierarchical Learning Model for IDS (HiLMIDS) is proposed. This model makes use of a RF-based algorithm in the first stage, which can efficiently filter out benign traffic. If any malicious traffic is detected, it is forwarded onto the second stage, where a TRANSF-based algorithm discriminates the family of attack the connection belongs to. HiLMIDS makes use of the strong points of both RF and TRANSF models, allowing for an efficient analysis that also offers highly accurate results when detecting attacks.

HiLMIDS was afterwards tested with three datasets that contained real traffic data, including attacks. The predictions generated by this hierarchical model are more accurate than both RF and TRANSF individually, and also than all other models tested. The accuracy is between 93.74% and 99.92%, based on the dataset used, which also improves the result of state-of-the-art technology for general attack detection and the detection of specific families of attack by up to roughly 2-3%. In some specific cases, the improvement is much better (roughly 18%).

However, models such as HiLMIDS are not without drawbacks. Even though the results provided are accurate when it comes to the detection of dangerous behavior, the main issue lies in the fact that neither RF or TRANSF are easily explainable. While they are able to properly associate input data to predict if a connection is an attack and the family of attacks it belongs to, it is not easy to extrapolate what parameters the model has used to make this prediction. Another issue is the fact that the datasets are severely unbalanced, making the prediction of specific types of attack considerably harder simply due to a lack of specific information on them, which makes it harder for the model to correctly predict them. Therefore, the main line of future work is to apply post-hoc techniques in order to improve its ability to explain the reasoning behind its results so that we can properly extract knowledge from the predictions it makes, which can be used to make detecting attacks a simpler task and further improve this model and its predictive capabilities.

Appendix A Confusion matrices

This section includes confusion matrices for all executed models and datasets detecting only malicious traffic Tables 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24.

Table 10 Confusion matrix of RandomForest on dataset CIC-IDS2017 using MaliciousOnly mode

	DDoS	Portscan	Bot	Infiltration	Webattack Brute-force	Webattack XSS	Webattack SQL Injection	FTP Patator	SSH Patator	Slowloris	Slowhttptest	DoSHulk	Goldeneye	Heartbleed
DDoS	25606	0	0	0	0	0	0	0	0	0	0	0	0	0
Portscan	0	31777	0	0	1	1	0	0	0	0	0	6	1	0
Bot	0	0	393	0	0	0	0	0	0	0	0	0	0	0
Infiltration	0	0	0	6	0	0	0	0	0	0	0	1	0	0
Webattack Bruteforce	0	2	0	0	236	62	0	0	0	1	0	0	0	0
Webattack XSS	0	0	0	0	86	43	0	0	0	0	0	1	0	0
Webattack SQL Injection	0	0	0	0	1	0	3	0	0	0	0	0	0	0
FTP Patator	0	0	0	0	0	0	0	1587	0	0	0	0	0	0
SSH Patator	0	0	0	0	0	0	0	0	1180	0	0	0	0	0
Slowloris	0	1	0	0	2	0	0	0	0	1152	3	0	1	0
Slowhttptest	0	0	0	0	0	1	0	0	0	9	1086	1	3	0
DoSHulk	0	1	0	0	0	0	0	0	0	0	0	46211	3	0
Goldeneye	0	0	0	0	0	0	0	0	0	0	1	6	2052	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	0	0	2

Table 11 Confusion matrix of RandomForest on dataset CIC-CSE-IDS2018 using MaliciousOnly mode

	FTP-Bruteforce	SSH-Bruteforce	GoldenEye	Slowloris	SlowHttp	DoS Hulk
FTP-Bruteforce	35395	0	0	0	3277	0
SSH-Bruteforce	2	37514	0	0	2	0
GoldenEye	0	0	8301	0	0	1
Slowloris	0	0	0	2197	0	1
SlowHttp	9762	0	0	0	8525	0
DosHulk	0	0	1	0	0	92381

Table 12 Confusion matrix of RandomForest on dataset CIC-DDoS2019 using MaliciousOnly mode

	UDP	MS SQL	Portmap	Syn	Netbios	UDPlag	LDAP	DrDoS DNS	DrDoS WebD-DoS	DrDoS TFTP	DrDoS UDP	DrDoS SNMP	DrDoS NetBIOS	DrDoS LDAP	DrDoS MSSQL	DrDoS NTP
UDP	2504	41	2	3	0	133	0	4	0	1	909	0	2	0	19	0
MS SQL	23	791	1	1	1	2	6	24	0	1	12	22	12	3	799	7
Portmap	0	0	50	8	59	1	0	1	0	0	0	0	13	0	4	1
Syn	3	3	7	9793	0	55	0	1	0	6	1	0	1	0	0	5
Netbios	1	0	44	1	44	0	0	2	0	0	0	7	30	0	0	0
UDPlag	246	4	1	45	0	1360	0	3	0	19	98	4	0	2	0	2
LDAP	0	5	1	0	0	2	85	101	0	0	1	58	0	122	6	0
DrDoS DNS	6	67	4	0	5	1	92	333	0	2	7	52	2	75	73	15
DrDoS WebDDoS	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	8
DrDoS TFTP	0	0	1	2	0	34	0	0	0	19732	2	0	0	0	2	10
DrDoS UDP	1222	15	1	2	0	60	1	4	0	1	768	0	1	0	8	1
DrDoS SNMP	0	21	1	0	18	0	60	52	0	0	0	344	4	36	17	0
DrDoS Netbios	1	14	17	0	63	0	0	1	0	0	1	1	6	0	16	0
DrDoS LDAP	0	2	0	0	0	0	118	72	0	0	0	44	0	45	4	3
DrDoS MSSQL	13	795	3	2	0	0	10	28	0	1	10	17	6	4	349	4
DrDoS NTP	0	3	0	5	0	2	0	1	5	6	1	1	0	0	2	24248

Table 13 Confusion matrix of the Multilayer Perceptron on dataset CIC-IDS2017 using MaliciousOnly mode

	DDoS	Portscan	Bot	Infiltration	Webattack Brute-force	Webattack XSS	Webattack SQL Injection	FTP Patator	SSH Patator	Slowloris	Slowhttptest	DoSHulk	Goldeneye	Heartbleed
DDoS	25606	0	0	0	0	0	0	0	0	0	0	0	0	0
Portscan	31789	0	0	0	0	0	0	0	0	0	0	0	0	0
Bot	393	0	0	0	0	0	0	0	0	0	0	0	0	0
Infiltration	299	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack Bruteforce	299	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack XSS	130	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack SQL Injection	4	0	0	0	0	0	0	0	0	0	0	0	0	0
FTP Patator	1587	0	0	0	0	0	0	0	0	0	0	0	0	0
SSH Patator	1180	0	0	0	0	0	0	0	0	0	0	0	0	0
Slowloris	1159	0	0	0	0	0	0	0	0	0	0	0	0	0
Slowhttptest	1099	0	0	0	0	0	0	0	0	0	0	0	0	0
DoSHulk	46215	0	0	0	0	0	0	0	0	0	0	0	0	0
Goldeneye	2059	0	0	0	0	0	0	0	0	0	0	0	0	0
Heartbleed	2	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14 Confusion matrix of the Multilayer Perceptron on dataset CIC-CSE-IDS2018 using MaliciousOnly mode

	FTP-Bruteforce	SSH-Bruteforce	GoldenEye	Slowloris	SlowHttp	DoS Hulk
FTP-Bruteforce	38672	0	0	0	0	0
SSH-Bruteforce	37518	0	0	0	0	0
GoldenEye	8302	0	0	0	0	0
Slowloris	2198	0	0	0	0	0
SlowHttp	18287	0	0	0	0	0
DosHulk	92382	0	0	0	0	0

Table 15 Confusion matrix of the Multilayer Perceptron on dataset CIC-DDoS2019 using MaliciousOnly mode

	UDP	MS SQL	Portmap	Syn	Netbios	UDPlag	LDAP	DrDoS DNS	DrDoS WebD-DoS	DrDoS TFTP	DrDoS UDP	DrDoS SNMP	DrDoS NetBIOS	DrDoS LDAP	DrDoS MSSQL	DrDoS NTP
UDP	0	0	0	0	0	3618	0	0	0	0	0	0	0	0	0	0
MS SQL	0	0	0	0	0	1704	0	0	0	0	1	0	0	0	0	0
Portmap	0	0	0	0	0	136	0	0	0	0	1	0	0	0	0	0
Syn	0	0	0	0	0	9851	0	15	0	0	9	0	0	0	0	0
Netbios	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0
UDPlag	0	0	0	0	0	1766	0	17	0	0	2	0	0	0	0	0
LDAP	0	0	0	0	0	381	0	0	0	0	0	0	0	0	0	0
DrDoS DNS	0	0	0	0	0	734	0	0	0	0	0	0	0	0	0	0
DrDoS WebDDoS	0	0	2	0	0	3	0	1	0	0	4	0	0	0	0	0
DrDoS TFTP	0	0	1	0	0	19770	0	7	0	0	5	0	0	0	0	0
DrDoS UDP	0	0	0	0	0	2084	0	0	0	0	0	0	0	0	0	0
DrDoS SNMP	0	0	0	0	0	543	0	0	0	0	0	0	0	0	0	0
DrDoS Netbios	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0
DrDoS LDAP	0	0	0	0	0	288	0	0	0	0	0	0	0	0	0	0
DrDoS MSSQL	0	0	0	0	0	1242	0	0	0	0	0	0	0	0	0	0
DrDoS NTP	0	0	6	0	0	24239	0	5	0	0	24	0	0	0	0	0

Table 16 Confusion matrix of the CNN on dataset CIC-IDS2017 using MaliciousOnly mode

	DDoS	Portscan	Bot	Infiltration	Webattack Brute-force	Webattack XSS	Webattack SQL Injection	FTP Patator	SSH Patator	Slowloris	Slowhttptest	DoSHulk	Goldeneye	Heartbleed
DDoS	25606	0	0	0	0	0	0	0	0	0	0	0	0	0
Portscan	31789	0	0	0	0	0	0	0	0	0	0	0	0	0
Bot	393	0	0	0	0	0	0	0	0	0	0	0	0	0
Infiltration	299	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack Bruteforce	299	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack XSS	130	0	0	0	0	0	0	0	0	0	0	0	0	0
Webattack SQL Injection	4	0	0	0	0	0	0	0	0	0	0	0	0	0
FTP Patator	1587	0	0	0	0	0	0	0	0	0	0	0	0	0
SSH Patator	1180	0	0	0	0	0	0	0	0	0	0	0	0	0
Slowloris	1159	0	0	0	0	0	0	0	0	0	0	0	0	0
Slowhttptest	1099	0	0	0	0	0	0	0	0	0	0	0	0	0
DoSHulk	46215	0	0	0	0	0	0	0	0	0	0	0	0	0
Goldeneye	2059	0	0	0	0	0	0	0	0	0	0	0	0	0
Heartbleed	2	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 17 Confusion matrix of the CNN on dataset CIC-CSE-IDS2018 using MaliciousOnly mode

	FTP-Bruteforce	SSH-Bruteforce	GoldenEye	Slowloris	SlowHttp	DoS Hulk
FTP-Bruteforce	38672	0	0	0	0	0
SSH-Bruteforce	37518	0	0	0	0	0
GoldenEye	8302	0	0	0	0	0
Slowloris	2198	0	0	0	0	0
SlowHttp	18287	0	0	0	0	0
DosHulk	92382	0	0	0	0	0

Table 18 Confusion matrix of the CNN on dataset CIC-DDoS2019 using MaliciousOnly mode

	UDP	MS SQL	Portmap	Syn	Netbios	UDPlag	LDAP	DrDoS DNS	DrDoS WebD-DoS	DrDoS TFTP	DrDoS UDP	DrDoS SNMP	DrDoS NetBIOS	DrDoS LDAP	DrDoS MSSQL	DrDoS NTP
UDP	3618	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MS SQL	1705	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Portmap	137	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Syn	9875	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Netbios	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UDPlag	1785	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LDAP	381	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS DNS	734	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS WebDDoS	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS TFTP	19783	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS UDP	2084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS SNMP	543	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS Netbios	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS LDAP	288	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS MSSQL	1242	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DrDoS NTP	24274	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 19 Confusion matrix of the RNN on dataset CIC-IDS2017 using MaliciousOnly mode

	DDoS	Portscan	Bot	Infiltration	Webattack Brute-force	Webattack XSS	Webattack SQL Injection	FTP Patator	SSH Patator	Slowloris	Slowhttptest	DoSHulk	Goldeneye	Heartbleed
DDoS	11663	26	0	0	0	0	0	0	0	0	0	13917	0	0
Portscan	0	31682	8	0	0	3	0	0	0	0	0	88	5	0
Bot	9	142	0	0	0	0	0	0	0	0	0	242	0	0
Infiltration	1	1	0	0	0	0	0	0	0	0	0	5	0	0
Webattack Bruteforce	0	0	0	0	0	0	0	0	0	0	0	301	0	0
Webattack XSS	0	0	0	0	0	0	0	0	0	0	0	130	0	0
Webattack SQL Injection	0	0	0	0	0	0	0	0	0	0	0	4	0	0
FTP Patator	0	794	0	0	0	0	0	0	0	0	0	794	0	0
SSH Patator	0	0	0	0	0	0	0	0	0	0	0	1179	1	0
Slowloris	349	114	0	0	0	0	0	0	0	0	0	696	0	0
Slowhttptest	605	73	0	0	0	0	0	0	0	0	0	422	0	0
DoSHulk	229	488	5	0	0	0	0	0	0	0	1	44130	7	1335
Goldeneye	6	0	0	0	0	0	0	0	0	0	0	2053	0	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	2	0	0

Table 20 Confusion matrix of the RNN on dataset CIC-CSE-IDS2018 using MaliciousOnly mode

	FTP-Bruteforce	SSH-Bruteforce	GoldenEye	Slowloris	SlowHttp	DoS Hulk
FTP-Bruteforce	38672	0	0	0	0	0
SSH-Bruteforce	18765	18729	0	0	0	24
GoldenEye	0	4756	0	0	0	3546
Slowloris	0	957	0	0	0	1241
SlowHttp	18287	0	0	0	0	0
DosHulk	0	2059	0	0	0	90323

Table 21 Confusion matrix of the RNN on dataset CIC-DDoS2019 using MaliciousOnly mode

	UDP	MS SQL	Portmap	Syn	Netbios	UDPlag	LDAP	DrDoS DNS	DrDoS WebD-DoS	DrDoS TFTP	DrDoS UDP	DrDoS SNMP	DrDoS NetBIOS	DrDoS LDAP	DrDoS MSSQL	DrDoS NTP
UDP	0	0	0	3	0	27	0	0	0	3244	0	0	0	1	0	343
MS SQL	1	0	0	2	0	43	0	0	0	53	0	0	0	0	0	1606
Portmap	0	0	0	1	0	4	0	0	0	17	0	0	0	0	0	115
Syn	6	0	0	7640	0	0	0	0	1	635	0	0	0	2	0	1591
Netbios	0	0	0	2	0	1	0	0	0	5	0	0	0	0	0	121
UDPlag	0	0	0	1131	0	2	0	0	0	583	0	0	0	0	0	69
LDAP	0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	377
DrDoS DNS	0	0	0	2	0	6	0	0	0	18	0	0	0	0	0	708
DrDoS WebDDoS	0	0	0	2	0	0	0	0	0	8	0	0	0	0	0	0
DrDoS TFTP	1	0	0	29	1	23	0	0	0	19651	0	0	0	0	0	78
DrDoS UDP	0	0	0	1	0	34	0	0	0	1861	0	0	0	0	0	188
DrDoS SNMP	0	0	0	0	0	1	0	0	0	5	0	0	0	0	0	537
DrDoS Netbios	0	0	0	0	0	4	0	0	0	7	0	0	0	0	0	109
DrDoS LDAP	0	0	0	2	0	2	0	0	0	5	0	0	0	0	0	279
DrDoS MSSQL	0	0	0	1	0	38	0	0	0	26	0	0	0	0	0	1177
DrDoS NTP	3	0	0	60	2	34	33	0	0	278	6	1	2	0	0	23849

Table 22 Confusion matrix of Transformers on dataset CIC-IDS2017 using MaliciousOnly mode

	DDoS	Portscan	Bot	Infiltration	Webattack Brute-force	Webattack XSS	Webattack SQL Injection	FTP Patator	SSH Patator	Slowloris	Slowhttptest	DoSHulk	Goldeneye	Heartbleed
DDoS	25588	8	0	0	0	0	0	0	0	0	0	10	0	0
Portscan	6	31757	0	0	1	1	0	0	0	0	0	18	3	0
Bot	0	0	393	0	0	0	0	0	0	0	0	0	0	0
Infiltration	0	1	0	0	0	0	0	0	0	0	0	6	0	0
Webattack Bruteforce	0	2	0	0	296	2	0	0	0	1	0	0	0	0
Webattack XSS	0	1	0	0	126	0	0	0	0	0	0	3	0	0
Webattack SQL Injection	0	0	0	0	4	0	0	0	0	0	0	0	0	0
FTP Patator	0	0	0	0	0	0	0	1586	0	0	0	1	0	0
SSH Patator	0	0	0	0	0	0	0	0	1179	0	0	1	0	0
Slowloris	0	1	0	0	3	0	0	0	0	1149	5	0	1	0
Slowhttptest	0	0	0	0	0	1	0	0	0	8	1087	1	3	0
DoSHulk	4	4	0	0	0	0	0	0	0	0	0	46201	6	0
Goldeneye	0	0	0	0	0	0	0	0	0	0	3	13	2041	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	2	0	0

Table 23 Confusion matrix of Transformers on dataset CIC-CSE-IDS2018 using MaliciousOnly mode

	FTP-Bruteforce	SSH-Bruteforce	GoldenEye	Slowloris	SlowHttp	DoS Hulk
FTP-Bruteforce	38626	0	0	0	6	0
SSH-Bruteforce	7	37506	0	0	5	0
GoldenEye	0	0	8282	0	2	18
Slowloris	0	0	0	2171	6	21
SlowHttp	10149	0	0	0	8138	0
DosHulk	0	0	0	0	0	92382

Table 24 Confusion matrix of Transformers on dataset CIC-DDoS2019 using MaliciousOnly mode

	UDP	MS SQL	Portmap	Syn	Netbios	UDPlag	LDAP	DrDoS DNS	DrDoS WebD-DoS	DrDoS TFTP	DrDoS UDP	DrDoS SNMP	DrDoS NetBIOS	DrDoS LDAP	DrDoS MSSQL	DrDoS NTP
UDP	3543	3	2	3	0	10	0	3	0	1	49	0	2	0	2	0
MS SQL	1	1649	1	1	1	2	4	4	0	1	2	4	2	1	31	1
Portmap	0	0	0	16	94	1	0	1	0	0	0	0	20	0	4	1
Syn	11	13	27	9702	1	98	0	4	0	9	2	0	1	0	0	7
Netbios	0	0	4	0	124	0	0	0	0	0	0	0	1	0	0	0
UDPlag	276	4	1	45	1	1321	1	3	0	20	104	4	0	2	0	2
LDAP	0	5	1	0	0	2	167	66	0	0	1	46	0	87	6	0
DrDoS DNS	6	67	4	0	5	1	96	319	0	2	7	57	2	78	75	15
DrDoS WebDDoS	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	8
DrDoS TFTP	0	0	1	2	0	49	0	0	0	19710	2	0	0	0	3	16
DrDoS UDP	1890	35	2	4	0	120	2	10	0	2	0	0	2	0	14	2
DrDoS SNMP	0	14	0	0	14	0	40	34	0	0	0	412	2	26	11	0
DrDoS Netbios	1	14	17	0	68	0	0	1	0	0	1	1	0	0	10	0
DrDoS LDAP	0	2	0	0	0	0	148	82	0	0	0	49	0	0	4	3
DrDoS MSSQL	13	798	3	2	0	0	10	28	0	1	10	17	6	0	350	4
DrDoS NTP	0	9	0	15	0	4	1	2	11	28	2	2	0	0	4	24196

Acknowledgements The research carried out in this study is part of the project "Advances in the development of trustworthy AI models to contribute to the adoption and use of responsible AI in healthcare (TAIH)" financed by the Ministry of Science, Innovation and Universities with code PID2023-149511OB-I00.

References

1. Gupta, M., Akiri, C.K., Aryal, K., Parker, E.: and Lopamudra Praharaj. Impact of generative ai in cybersecurity and privacy, From chatgpt to threatgpt (2023)
2. Miranda-García, A., Rego, A.Z., Pastor-López, I., Sanz, B., Tellaeche, A., Gaviria, J., Bringas, P.G.: Deep learning applications on cybersecurity: A practical approach. Neurocomputing **563**, 126904 (2024)
3. Zhang, Z., Ning, H., Shi, F., Farha, F., Yang, X., Jiabo, X., Zhang, F., Raymond Choo, K.-K.: Artifcial intelligence in cyber security: research advances, challenges, and opportunities. Artif. Intell. Rev. **55**, 1029–1053 (2022)
4. Kalota, F.: A primer on generative artificial intelligence. *Education Sciences*, 14(2), (2024)
5. Serey, J., Alfaro, M., Fuertes, G., Vargas, M., Durán, C., Ternero, R., Rivera, R., Sabattin, J.: Pattern recognition and deep learning technologies, enablers of industry 4.0, and their role in engineering research. *Symmetry*, 15(2) (2023)
6. Bilge, L., Dumitraş, T.: Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 833-844, New York, NY, USA, Association for Computing Machinery (2012)
7. Wang, P., Li, Q., Li, D., Meng, S., Bilal, M., Mukherjee, A.: Security in defect detection: A new one-pixel attack for fooling dnns. Journal of King Saud University - Computer and Information Sciences **35**(8), 101689 (2023)
8. Hu, W., Tan, Y.: Generating adversarial malware examples for black-box attacks based on gan. In *International Conference on Data Mining and Big Data*, pages 409–423. Springer, (2022)
9. Macas, M., Chunming, W., Fuertes, W.: Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. Expert Syst. Appl. **238**, 122223 (2024)
10. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
11. Quinlan, JR.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (1993)
12. Ivakhnenko, AG (1971) Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4):364–378
13. Homma, T., Atlas, L.E., Marks, R.J.: An artificial neural network for spatiotemporal: application to phoneme classification. In *Proceedings of the 1987 International Conference on Neural Information Processing Systems*, NIPS'87, page 31-40, Cambridge, MA, USA, MIT Press (1987)
14. Rumelhart, D.E., McClelland, J.L.: *Learning Internal Representations by Error Propagation*, pages 318–362. Mit Pr, (1987)
15. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, (2016). http://www.deeplearningbook.org
16. Rezende, DJ., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models, (2014)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., Pedrycz, W.: A comprehensive survey on applications of transformers for deep learning tasks. Expert Syst. Appl. **241**, 122666 (2024)
19. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, (2007)
20. Pal, B., Daniel, T., Chatterjee, R., Ristenpart. T.: Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 417–434, (2019)
21. Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., Koucheryavy, Y.: Multi-factor authentication: A survey. Cryptography **2**, 01 (2018)
22. Xin, D., Chen, S., Liu, Z., Wang, J.: Multiple userids identification with deep learning. Expert Syst. Appl. **207**, 117924 (2022)
23. Amberkar, A., Awasarmol, P., Deshmukh, G., Dave, P.: Speech recognition using recurrent neural networks. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–4, (2018)
24. Chen, J., Seng, K., Smith, J., Ang, L.: Situation awareness in ai-based technologies and multimodal systems: Architectures, challenges and applications. *IEEE Access*, PP:1–1, 01 (2024)
25. Feng, Y., Zhao, H., Zhang, J., Cai, Z., Zhu, L., Zhang, R.: Prediction of network security situation based on attention mechanism and convolutional neural network-gated recurrent unit. *Applied Sciences*, 14(15), (2024)
26. Li, D., Dong, X., Gao, J., Hu, K.: Abnormal traffic detection based on attention and big step convolution. *IEEE Access*, PP:1–1, 01 (2023)
27. Zhao, L., Yuan, H., Kangyuan, X., Bi, J., Li, B.H.: Hybrid network attack prediction with savitzky-golay filter-assisted informer. Expert Syst. Appl. **235**, 121126 (2024)
28. Kun Wang, Y.F., Duan, X., Liu, T., Jianqiao, X.: Abnormal traffic detection system in sdn based on deep learning hybrid models. Comput. Commun. **216**, 183–194 (2024)
29. Dwivedi, D., Bhushan, A., Singh, AK., Snehlata.: Detection ofÂ malicious network traffic attacks using support vector machine. In Anshul Verma, Pradeepika Verma, Kiran Kumar Pattanaik, Sanjay Kumar Dhurandher, and Isaac Woungang, editors, *Advanced Network Technologies and Intelligent Computing*, pages 54–68, Cham, (2024) Springer Nature Switzerland
30. Javaheri, D., Gorgin, S., Lee, J.A., Masdari, M.: Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives. Inf. Sci. **626**, 05 (2023)
31. Gibert, D., Planes, J., Mateu, C., Le, Q.: Fusing feature engineering and deep learning: A case study for malware classification. Expert Syst. Appl. **207**, 117957 (2022)
32. Mansour, R.F.: Blockchain assisted clustering with intrusion detection system for industrial internet of things environment. Expert Syst. Appl. **207**, 117995 (2022)
33. Jain, U., Srivastava, Y., Malik, A., Dhingra, D., Kumar, A., Nagrath, P.: Malicious dns detection and prediction using smote-enn and hybrid artificial neural network. In *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 138–144, (2022)
34. Xing, X., Jin, X., Elahi, H., Jiang, H., Wang, G.: A malware detection approach using autoencoder in deep learning. IEEE Access **10**, 25696–25706 (2022)
35. Ravi, V., Chaganti, R., Alazab, M.: Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Comput. Electr. Eng.*, 102(C), sep (2022)
36. Belarbi, O., Khan, A., Carnelli, P., Spyridopoulos, T.: *An Intrusion Detection System Based on Deep Belief Networks*, page 377-392. Springer International Publishing, (2022)
37. Mishra, D., Naik, B., Nayak, J., Souri, A., Dash, P.B., Vimal, S.: Light gradient boosting machine with optimized hyperparameters for identification of malicious access in iot network. Digital Communications and Networks **9**(1), 125–137 (2023)

38. Bensaoud, A., Kalita, J.: Cnn-lstm and transfer learning models for malware classification based on opcodes and api calls. Knowl.-Based Syst. **290**, 111543 (2024)
39. Gupta, I., Kumari, S., Jha, P., Ghosh, M.: Leveraging lstm and gan for modern malware detection, (2024)
40. Ullah, F., Babar, M.A., Aleti, A.: Design and evaluation of adaptive system for big data cyber security analytics. Expert Syst. Appl. **207**, 117948 (2022)
41. Ouhssini, M., Afdel, K., Agherrabi, E., Akouhar, M., Abarda, A.: Deepdefend: A comprehensive framework for ddos attack detection and prevention in cloud computing. Journal of King Saud University - Computer and Information Sciences **36**(2), 101938 (2024)
42. Zihan, W., Zhang, H., Wang, P., Sun, Z.: Rtids: A robust transformer-based approach for intrusion detection system. IEEE Access **10**, 64375–64387 (2022)
43. Boonchai, J., Kitchat, K., Nonsiri, S.: The classification of ddos attacks using deep learning techniques. In *2022 7th International Conference on Business and Industrial Research (ICBIR)*, pages 544–550. IEEE, (2022)
44. Sharmin Aktar and Abdullah Yasin Nur: Towards ddos attack detection using deep learning approach. Computers & Security **129**, 103251 (2023)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, AN., Kaiser, Ł u, Polosukhin, I.: Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., (2017)
46. Devlin, J., Chang, M-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, (2018)
47. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T.: RÃ©mi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. State-of-the-art natural language processing, Rush. Huggingface's transformers (2020)
48. Lashkari, AH.: Cicflowmeter-v4.0 (formerly known as iscxflowmeter) is a network traffic bi-flow generator and analyser for anomaly detection. https://github.com/iscx/cicflowmeter, (08 2018)
49. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
50. Hossin, M.: Sulaiman MN. A review on evaluation metrics for data classification evaluations International Journal of Data Mining & Knowledge Management Process **5**, 01–11 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.