DNNET-Ensemble approach to detecting and identifying attacks in IoT environments

Cristiano A. de Souza[1], **Carlos B. Westphall**[1], **Jean D. G. Valencio**[2]
Renato B. Machado[2], **Wesley dos R. Bezerra**[1]

[1]Departamento de Informática – Universidade Federal de Santa Catarina (UFSC)

[2]Universidade Estadual do Oeste do Paraná (UNIOESTE)

cristiano.souza.c@posgrad.ufsc.br, westphal@inf.ufsc.br

renato.machado@unioeste.br, jean.valencio@unioeste.br

wesley.bezerra@ifc.edu.br

Abstract. *Special security techniques like intrusion detection mechanisms are indispensable in modern computer systems. It is important to detect and identify the attack in a category so that specific countermeasures for the threat category are solved. However, most existing multiclass detection approaches have some weaknesses, mainly related to detecting specific categories of attacks and problems with false positives. This article addresses this research problem and advances state-of-the-art, bringing contributions to a two-stage detection architecture called DNNET-Ensemble, combining binary and multiclass detection. The results obtained in experiments with renowned intrusion datasets demonstrate that the approach can achieve superior detection rates and false positives performance compared to other state-of-the-art approaches.*

1. Introduction

Internet of Things (IoT) devices have limited resources thereby is a need to transfer, through the Internet, the data generated by these devices to process and store it in a computational environment of major capacity. Regarding that, once Cloud Computing has latency problems caused by the data center distance, Fog Computing provides services closer to the end devices (Edge) with less latency [Bonomi et al. 2012]. This way, it stores and processes information close to IoT devices, reducing the traffic sent to the cloud [Bonomi et al. 2012]. These environments are not free from threats and security vulnerabilities. Furthermore, computational growth increases the likelihood of vulnerabilities, which malicious entities can use to cause damage. As a consequence, special security techniques are indispensable in modern computer systems.

Intrusion Detection mechanisms are critical points of security, aiming to identify attempted attacks by unauthorized users [Souza et al. 2022]. Hence, methods that only perform the detection that an intrusion is occurring (i.e., binary detection) are insufficient to provide efficient security wherein the approach must be able to mitigate the invasion not to succeed. Therefore, it is essential identify and categorize the attack selecting its specific countermeasures to mitigate the related vulnerability. Also, the classification of the type or category of the attack is paramount for the decision network

administrator which, based on category identification of a recurrent attack, can decide to implement actions, correcting the vulnerability exploited by the attack. The identification or categorization of attacks is a difficult and more complex task than the simple detection of abnormal activities. Thus, multiclass detection approaches are generally more complex, with higher computational costs and lower accuracy rates than binary methods [Prabavathy et al. 2018, Nguyen et al. 2019]. This is justified by the difficulties in identifying specific types of attacks [Prabavathy et al. 2018, Diro and Chilamkurti 2018, Almiani et al. 2020]. Furthermore, existing multiclass approaches present problems related to normal traffic identification rates [Ieracitano et al. 2020, Dat-Thinh et al. 2022]. A high rate of false positives is a big problem and, in some cases, degrading performance of the network. Furthermore, the IoT and Fog computing environments limit the design of robust approaches due to the constraint resources present in such environments.

This article addresses this research problem and advances state-of-art, bringing contributions to a two-stage detection architecture combining binary and multiclass detection. While the benign traffic can be quickly released on the first detection, the intrusive traffic can be subjected to a robust analysis without causing delay issues. So, we propose the DNNET binary approach for the binary detection level, which can provide faster binary detection than the DNNKNN [Souza et al. 2020] approach. The proposed Hybrid Attribute Selection strategy can find an optimal subset of attributes through a wrapper method having a lower training cost due to pre-selection with a filter method. Furthermore, the proposed Soft-SMOTE improvement allows operating with a balanced dataset without generating a relevant increase in training time, even in scenarios with a large number of classes and a large imbalance among them.

The results obtained from experiments with the NSL-KDD and IoTID20 intrusion datasets demonstrated that the approach achieved superior performance over other classical Machine Learning (ML) techniques and state-of-art approaches. The proposed approach obtained superior average balanced accuracy, precision, and recall rates than classical machine learning and state-of-art approaches. As a result, itself proved superior to other approaches regarding identifying benign traffic, indicating a low rate of false positives and requiring a fewer computational cost.

Contributions. The main contributions of this work are as follows:

- Proposal of a two-level approach called DNNET-Ensemble for intrusion detection and identification;
- Proposal of the soft-SMOTE strategy for class balancing with resource constraints.
- Proposal of the Hybrid Attribute Selection strategy to reduce the cost of wrapper attribute selection approaches;
- Detection and identification results superior to classical machine learning methods and state-of-art approaches;

The remainder of this paper is organized as follows. Section 2 presents recent works. Section 3 presents a description of the proposed approach. The experimental evaluation results are presented in Section 4. Finally, Section 5 concludes our paper.

2. Related works

Several works have proposed approaches focused on single methods. Some with based methods and neural models like DNN [Diro and Chilamkurti 2018, Liang et al. 2022],

Deep Recurrent Neural Network (DRNN) [Almiani et al. 2020], AutoEncoder (AE) [Ieracitano et al. 2020] and Convolutional Neural Network (CNN) [Blanco et al. 2018]. Some of these approaches presented drawbacks regarding false positives. In these situations, normal traffic identification rates were lower than expected, mistakenly blocking much benign traffic.

Table 1. Related works.

Work	Detection method	Observations
[Prabavathy et al. 2018]	OS-ELM	Low accuracy for some types of attacks
[Diro and Chilamkurti 2018]	DNN	Difficulties related to false positives
[Blanco et al. 2018]	CNN+GA	Difficulties related to false positives
[Almiani et al. 2020]	RNN	Lower accuracy than binary methods
[Ieracitano et al. 2020]	AE	Difficulties related to false positives
[Qaddoura et al. 2021]	DNN+LSTM	Balancing cost can become high
[Liang et al. 2022]	DNN	Preserve data privacy
[Zhao et al. 2022]	Ensemble stacking	Low accuracy for some types of attacks
[Dat-Thinh et al. 2022]	DT	Difficulties related to false positives
[Albulayhi et al. 2022]	Ensemble voting	Feature selection filter method
[Sarwar et al. 2022]	PSO+XGB+RF	Low accuracy for some types of attacks
Our work	DNNET-Ensemble	No countermeasures

Furthermore, approaches that work with single classifiers can suffer from instabilities. However, with Ensemble Learning, better classification performance than any single classifier can be achieved. The authors [Prabavathy et al. 2018] have proposed a new multiclass anomaly intrusion detection technique based on the ensemble learning and Online Sequential-Extreme Learning Machine (OS-ELM). [Zhao et al. 2022] proposes a hybrid weighted ensemble stacking intrusion detection system. Random Forest (RF), XGBoost, and KNN methods are used as basic classifiers, and Logistic Regression (LR) is selected as meta classifier. [Albulayhi et al. 2022] proposes an approach with an ensemble method by a majority vote of four basic classifiers: KNN, Decision Tree (DT), a neural model, and a bagging method. Although ensemble methods provide greater robustness, they generally require greater computational capabilities as they work with multiple classifiers.

Another important problem in state-of-art is the difficulties related to false positives, generally existing in anomaly-based approaches [Ieracitano et al. 2020, Dat-Thinh et al. 2022]. The approach proposed in [Dat-Thinh et al. 2022] presents these difficulties, as it blocks all traffic previously identified as intrusive by the first level of binary detection with the DT method. Nevertheless, DTs are susceptible to overfitting issues. In addition, the second level only classifies the type of attack, not being able to correct misclassifications.

State-of-the-art multiclass detection approaches have lower accuracy rates than binary methods [Prabavathy et al. 2018, Sarwar et al. 2022, Zhao et al. 2022]. There are difficulties in identifying specific types of attacks. Furthermore, some approaches present problems related to normal traffic identification [Ieracitano et al. 2020, Dat-Thinh et al. 2022]. Attack detection difficulties are often related to the imbalance of existing training data. Some works used the Synthetic Minority Oversampling Technique (SMOTE) technique to balance the data [Qaddoura et al. 2021]. However, applying the

full SMOTE strategy in extremely unbalanced scenarios with many classes will create a very large number of synthetic registers, which increases the cost of training and can downgrade the machine learning model's performance. Some works have tried to filter the best traffic characteristics with wrapper attribute selection techniques, where classification methods are embedded in the selector. Comparatively, wrapper methods get higher quality attribute sets for detection than filter methods. However, wrapper approaches demand more processing and generate higher computational costs, which can be prohibitive when dealing large amounts of data. Therefore, in many cases, the techniques used in the detection, attribute selection, and class balancing approaches can make the approaches cost high to operate in the Fog-IoT environment.

This article addresses these research problems and advances state-of-art, bringing contributions to a two-stage detection architecture, wherein benign traffic can be quickly released on the first detection. Thus, intrusive traffic can be subjected to a robust analysis without causing delay issues.

3. Proposed approach

Resource limitations in the IoT environment make it difficult to perform complex anomaly analyses. Hence, we propose a two-level detection approach designed to operate in the fog and the cloud, as proposed in [Souza et al. 2020]. Although the cloud has devices with more significant computing resources than the IoT [Ni et al. 2018] and the Fog, it suffer from latency problems caused by the large distance between the IoT network and the datacenters. Moreover, Fog is closer to the IoT network and can provide processing and storage mechanisms at the edge of the network [Bonomi et al. 2012]. Therefore, this makes it possible to detect threats faster.

Initially, the information captured from the monitored network is pre-processed and sent to the first detection level, where a binary detection analysis is performed. The proposed DNNET classifier is responsible for operating on the first level and assigning the intrusive or non-intrusive label to each event. If it is benign traffic, it is automatically allowed. Otherwise, if the traffic is classified as intrusive, it is sent to the second level of detection in the cloud layer, where a multiclass classifier identifies the attack category. The multiclass classifier was adapted from the approach proposed in [Souza et al. 2022]. This method will classify the event as an attack type or benign. If it is identified as a type of attack, this information will be sent to the mitigation module, which will implement appropriate countermeasures for each intrusion. However, the second level approach can classify an event as benign, in this case, identifying an event wrongly classified as intrusive by the first level. In consequence, the approach allows for the recovery of first-level false positives. In the next sections, details about each detection level are provided.

Ensemble methods provide greater robustness to classification. However, they have higher processing and training costs than single classifier models. The proposed approach proposes to circumvent this limitation by executing the ensemble method in the cloud layer, thus having greater computational power. Furthermore, the ensemble approach will only be applied to classify previously detected intrusive events into a specific attack category. As normal events have already been detected by the first stage and have already been released, there is no need for an urgent response to the events sent to this second stage of classification, as there is a high chance that they will be intrusive. As a result,

the delay will be minimal in the flow of legitimate traffic caused by the communication latency with the cloud.

3.1. First Level - DNNET Binary detection

The hybrid method for binary detection, DNNKNN [Souza et al. 2020], has a highly detection rate. However, it has a high computational cost in prediction time, as it performs several comparisons with the instance base during the analysis. As the binary method is designed to operate in the fog and will act on the first level of analysis used on all IoT network traffic, it must be light so as not to cause a significant delay in the traffic. We optimize the DNNKNN [Souza et al. 2020] method and propose the new DNNET method, proposing some improvements.

In the DNNET method, the traffic is initially submitted to the neural model, as illustrated in Figure 1. The model generates outputs on two neurons, one corresponding to the intrusive class and the other to the non-intrusive class. Each neuron generates an output between 0 and 1. This value corresponds to the probability that the flow belongs to the class to which the neuron corresponds.

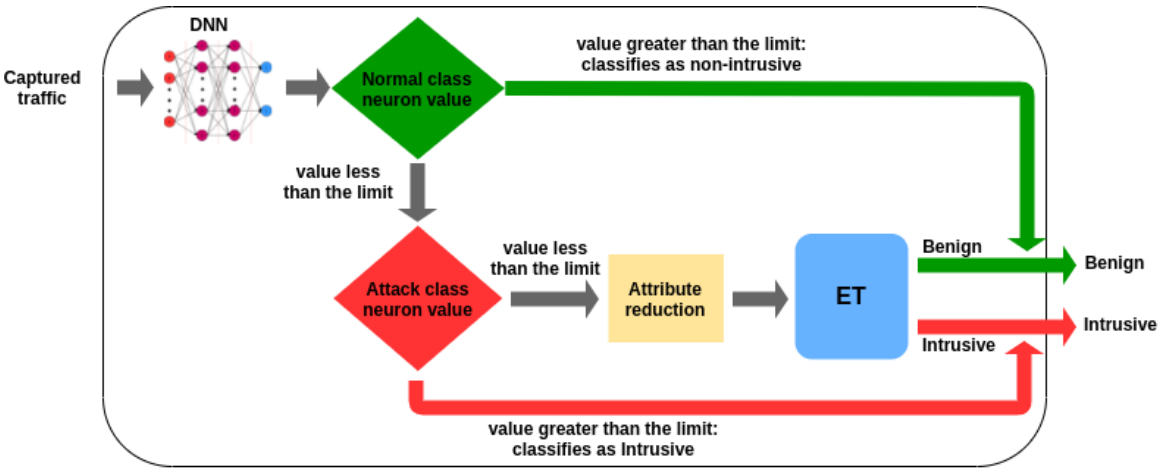

Figure 1. Illustration of the DNNET binary method.

The approach has a predefined threshold for each neuron. If the neuron of the intrusive class has an output bigger than its predefined threshold, the analyzed traffic instance will be classified as intrusive. The same procedure is applied to the output of the non-intrusive class neuron. Instances that obtained values below the output limits on both neurons are considered situations where the neural model did not obtain clarity or precision. Hence, traffic is sent to a binary Extra Tree. So, before being analyzed by the ET method, the instance has its attributes reduced. This step is to reduce instance complexity and reduce ET processing. The attributes are reduced according to the attributes selected during the DNNET training stage. The dataset with reduced attributes is then submitted to the ET, and the binary classification generated by the ET is accepted as final.

ET is a Decision Tree ensemble method. DTs are lightweight and simple methods with good classification performance. However, they are susceptible to overfitting issues. Through the ensemble strategy, the ET technique builds a more robust classification model and reduces overfitting.

3.1.1. Deep Neural Network (DNN)

The DNN used is Feed-forward of the Multilayer Perceptron type, motivated by the ability to solve non-linearly separable problems. The number of neurons assigned to each hidden layer of the network was defined, according to [Souza et al. 2020], as equal to the input size. The hyperbolic tangent function was used as an activation function of hidden layer neurons. It can be defined from the ratio of hyperbolic sine to hyperbolic cosine.

The output layer was fixed in two neurons with a softmax activation function. Each neuron generates a rank value between 0 and 1. The softmax function transforms the outputs of each neuron into values between 0 and 1. It divides them by the sum of the outputs, generating the probability that the input is in a specific class.

3.1.2. Extra Tree (ET)

Extra Tree is an ensemble method aggregating the results of several decorrelated DTs accumulated within a "forest" to produce the classification results. DTs classify using a set of hierarchical resource decisions. Decisions made at internal nodes are the division criteria. The ET focuses on strongly randomizing both the attributes' choice and the attributes' cutoff point while dividing a node in the tree. In the extreme case, it builds totally random trees, whose structures are independent of the output values of the learning sample [Geurts et al. 2006]. As in RF a random subset of candidate attributes is used. However, in ET, instead of looking for the most discriminating cut points, cut points are drawn randomly for each candidate attribute, and the best of these generated cut points is randomly chosen as the division rule. The logic behind the method is that the explicit cutoff and attribute randomization combined with the ensemble mean should be able to reduce the variance more strongly than the weaker randomization schemes used by other methods. Using original and complete training data, rather than bootstrap replicas, is motivated to minimize bias [Geurts et al. 2006].

The ET has some important parameters. The number of decision trees (a) present in the structure was set to 10 ($a = 10$), the minimum size of the training set to split a node is 2 (n_min = 2), and the number of attributes considered for better division is the root of the number of existing attributes ($K = sqrt(N)$). The Gini Index is used as a criterion. The parameter p indicates the number of depth levels each tree can grow. In the design of the ET structure of the DNNET method, the depth limit was set at ten levels ($p = 10$). This parameter controls the size of trees. Failure to define this structure generates unpruned fully grown trees that can potentially be very large in some datasets [Geurts et al. 2006].

3.1.3. DNNET training

The training process of the DNNET approach includes the training of the neural model, the selection of binary attributes, the training of the binary ET, and the adjustment process of the limits of the output neurons of the neural model, as can be seen in the Algorithm 1. DNNET method training has two main parameters, the acceptable false positive rate ($acceptable\_FP\_rate$) and the acceptable false-negative rate ($acceptable\_FN\_rate$). The

Algorithm 1: Training of the DNNET method.

Input: x, y, $acceptable\_FP\_rate$, $acceptable\_FN\_rate$
$model \leftarrow$ TrainingModelDNN (x, y);
$reduced\_x \leftarrow$ HybridAttributeSelection (x, y)
$et \leftarrow$ TrainingBinaryET ($reduced\_x$, y);
$benign\_neuron\_limit \leftarrow 0.5$;
$attack\_neuron\_limit \leftarrow 0.5$;
while $i <= 10$ **do**
  $pred\_y, neurons\_output \leftarrow$ DNNET (x)
  $fp\_rate, fn\_rate \leftarrow$ CalculateMetrics ($pred\_y, y$);
  if ($fp\_rate > acceptable\_FP\_rate$) **then**
    $attack\_neuron\_limit \leftarrow$ percentile($neurons\_output.attack$, i*10);
  end
  if ($fn\_rate > acceptable\_FN\_rate$) **then**
    $benign\_neuron\_limit \leftarrow$ percentile($neurons\_output.benign$, i*10);
  end
  if ($fp\_rate <= acceptable\_FP\_rate$) ***and***
   ($fn\_rate <= acceptable\_FN\_rate$) **then**
    break;
  end
  $i \leftarrow i + 1$;
end

process of defining the limits is iterative. It is carried out until the false positive (FP) and false-negative (FN) rates obtained are less than or equal to the acceptable rates or until the limits are equal to 1, in the latter, in which case all traffic instances would be sent to be classified by ET.

Initially, the training of the DNN model is carried out. After this step, a strategy is applied to select the attributes that can best contribute to the binary analysis. Hence, we propose a Hybrid Attribute Selection strategy with two main steps. In Step 1, a filter-type attribute selection algorithm is applied, the Information Gain (IG). Also, the set of attributes selected by Step 1 is then submitted to Step 2, composed of wrapper algorithms, Recursive Feature Elimination (RFE), and Sequential Forward Feature Selection (SFFS). They use the ET classifier during the selection step to assess the importance of attributes. Finally, the set of best attributes generated by the second step is taken as a result. Wrapper methods tend to obtain higher quality attribute sets for the detection process. However, wrapper approaches demand more processing and generate higher computational costs, which can be prohibitive when dealing large amounts of data. Thus, using a filter method in Step 1 is of great importance, as it allows submitting a partially reduced data set to Step 2, making the process less costly.

The dataset with the attributes selected by the hybrid attribute selection strategy is then used to build the binary ET structure. Thereafter, the benign neuron and attack neuron thresholds are initially set at 0.5. In this case, all instances classified by DNN are accepted, and none are sent to ET.

From this, an iterative process of up to 10 steps begins. In each iteration, the DNNET method is applied to the training data to generate a list of predictions and two sets of values. These values correspond to the output generated by the neurons for each of the training data. A set of neuron output values corresponding to the benign class and a set of neuron output values for the intrusive class. Subsequently, the metrics fp_rate and fn_rate are generated from the predictions. The fp_rate metric, corresponding to the false positive rate, is compared with the $acceptable\_FP\_rate$ parameter. A new threshold is defined for the attack neuron if it is higher than acceptable. This new limit will be higher so that a smaller number of instances are classified only by the DNN, and a larger one is sent to the ET. The same procedure occurs with the false-negative rate (fn_rate) and the parameter $acceptable\_FN\_rate$. The definition of the new threshold is based on the sets of benign neuron and attack neuron output values obtained during the DNNET prediction with the training data. This new threshold for each neuron is assigned according to the value corresponding to the $(i * 10)$ percentile of the set of values generated by each output neuron, where i corresponds to the current iteration. The threshold definition process continues to the next iteration, where it is repeated with the new thresholds. Thus, the limits are incremented by ten percentiles at each iteration until reaching a limit that reaches the acceptable rate of FP and FN. If the percentages of FP and FN are smaller and acceptable at any time, the ideal limits are found, and the training is finished.

3.2. Second level - Ensemble Multiclass identification

The proposed approach second step is the Identification phase, which is designed to operate in the cloud computing layer. As for second step, the method used consists of a multiclass ensemble approach proposed in [Souza et al. 2022], composed of 3 different machine learning techniques, an ET, RF, and a DNN. Ensemble methods are created by combining multiple models. Promoting the combination of the classifications of several conceptually different base machine learning classifiers to improve generalization and robustness over single classifiers.

Due to the major resource capacity of the cloud, there is no need to reduce the structure of the method to conserve resources. Thus, unlike [Souza et al. 2022], we propose using the 100 trees in the RF and ET structures ($a = 100$), but limiting the growth in depth. Each tree can grow to a maximum of 100 levels ($p = 100$). The greater resource capacity allows for working with a DNN with a more complex architecture, which was defined with two hidden layers of 150 neurons, each with a ReLU activation function.

3.3. DNNET-Ensemble Training process

The training process is responsible for generating the models of the proposed approach and making it able to analyze network traffic to identify intrusions. Initially, the training dataset goes through a pre-processing step to obtain the normalized set of information for analysis. This set is then submitted to the class balancing process.

Class balancing equals the number of training instances in each class. The challenge of working with unbalanced datasets is that most ML techniques will underperform in the minority class. One of the strategies for working with unbalanced datasets is to create new data for the minority classes to equal the quantity of the majority class. The SMOTE method can be used for this purpose. It selects nearby examples in the feature space, draws a line between the examples in the feature space, and creates new instances

at a point along that line. Nevertheless, in scenarios where there is a large imbalance between classes and a large number of classes, creating a very large number of synthetic records can damage the machine learning model's performance and make the training process more expensive. As solution, we propose the Soft-SMOTE, a less aggressive balancing strategy that seeks to define an adequate percentage of records for each class based on the total number of records. The objective is to create new records for the minority classes until they reach a minimum ($minimum$) that can contribute to the detection method learning process. The strategy does not aim to equal the number of records of all classes with the number of records of the majority class. In extremely unbalanced scenarios, this would result in a huge amount of new synthetic records. Initially, the number of existing classes in the dataset is identified ($n\_classes$), and the appropriate percentage of records for each class ($percentage$) is calculated. This percentage calculates each class's minimum number of records ($minimum$). For each of the classes, a check is performed. If the currently existing quantity (qty_per_class) is less than the minimum ($minimum$), then it is defined as the new quantity of records for the class ($final\_qty\_per\_class$), the minimum quantity defined earlier. These values are then used to indicate the intended number of records for each class to SMOTE. After balancing the original training data, a new dataset with balanced classes is generated and submitted in two different flows —one for training the binary DNNET method and another for training the multiclass method. The balanced dataset is submitted to the attribute selection step in one of the flows.

To perform the selection of attributes in this step, the same strategy presented in Section 3.1.3 is used. However, in this case, considering a multiclass scenario. The dataset with only the selected attributes is used to train the Ensemble classifier. After training, an ensemble method capable of performing multiclass detection on new data is obtained.

The other flow corresponds to the training of the DNNET method. Initially, the balanced dataset goes through a binarization step to convert all intrusive flows to label 1 and the benign ones to label 0. The binary dataset is used to train the DNNET approach. After the complete training process of the DNNET-ET approach, the architectures, information, and weights of the neural model are sent for implantation in the Fog Node.

4. Evaluation

The proposed approach and machine learning methods were evaluated through experiments with the IoTID20 and NSL-KDD datasets. Each experiment had run five times, wherein 70% of the data were used for training and 30% for testing.

4.1. Evaluation binary classifiers

The NSL-KDD database is used in many current works [Diro and Chilamkurti 2018, Mohamed Omar et al. 2021, Zhao et al. 2022] to assess intrusion detection methods. Each base instance has 42 attributes (also called resources) of which 41 are behavioral characteristics of network connections extracted from packet analysis. The NSL-KDD dataset has 125,973 records.

Additionally, Table 2 compares the detection metrics obtained by the binary approaches proposed in experiments with the NSL-KDD dataset. Both proposed approaches presented higher detection rates than the other approaches. The method proposed by

[Mohamed Omar et al. 2021] had a high recall rate; however, the accuracy rate was below 99%. The accuracy rate obtained by DNNKNN was 99.43%, and that of DNNET was 99.64%. In addition, both achieved attack detection rates (DR) of over 99%.

Table 2. Binary detection results of approaches.

Abordagens	ACC	PRE	DR	Train (s)	Test (s)
[Mohamed Omar et al. 2021]	99,4	98,7	99,4	-	-
[Sahar et al. 2021]	95,4	96,2	95,4	-	-
[Gopalakrishnan and Purusothaman 2022]	95,6	98,3	92,2	-	-
DNNKNN [Souza et al. 2020]	99,43	99,36	99,43	247.47	7.97
DNNET	99,64	99,88	99,36	113.24	2.36

The DNNKNN method proposed in [Souza et al. 2020] optimized the KNN algorithm about the computational cost, with a 90% reduction in the prediction time; however, the approach remains expensive. Thus, the DNNET method was designed to optimize the prediction time, seeking to maintain the quality of the binary detection presented by DNNKNN. In the experiments conduct, the DNNET approach achieved a binary detection quality similar or superior to DNNKNN's. In addition, the objective of reducing the computational cost was also achieved. The DNNKNN approach needed approximately 247.5 seconds to complete the training process, while the DNNET needed 113.2 seconds, a reduction of 55.3%. Furthermore, the DNNET approach also reduced the prediction time from 7.96 seconds to just 2.35 seconds, corresponding to a reduction of 70.4%.

4.2. Evaluation DNNET-Ensemble with NSL-KDD

Table 3 presents the results obtained in experiments with the NSL-KDD dataset. The kNN, DNN, RF, ET, and the DNNET-Ensemble achieved excellent results in identifying benign traffic, DoS, and probing attacks. They achieved detection rates greater than 99% in all three categories. In detecting R2L attacks, DNN achieved a recall of approximately 89%. Therefore, it obtained inferior performance to the kNN, RF, ET approaches and the proposed approach, which reached approximately 95% detection.

Table 3. Results of experiments with the NSLKDD dataset.

Works	BACC	Class				
		Benign	DoS	Pro	R2L	U2R
DNN	85.84	99.69	99.87	99.15	89.26	41.25
kNN	88.30	99.71	99.92	99.34	95.03	47.50
RF	86.52	99.94	99.98	99.55	95.64	37.50
ET	87.71	99.92	99.96	99.49	94.16	45.00
[Souza et al. 2022]	88.41	99.86	99.93	99.43	92.82	50.00
Our work	92.60	99.89	99.95	99.19	95.23	68.75

About R2L attacks, the proposed approach reached the second highest rate, behind the approach of [Blanco et al. 2018], which got 99.18%, as can be seen in Figure

2. However, this work has failed to detect U2R attacks. The U2R attack category has few flows in the dataset. State-of-the-art approaches and classic machine learning methods have difficulty in detecting these attacks. DNN was able to identify 41%. The kNN detected approximately 47%, the RF 37%, and the ET 45%. The approach proposed by [Du et al. 2020] detected less than 40% of U2R attacks. The DNNET-Ensemble overcame these rates, achieving a recall of approximately 69%, surpassed only by the approach of [Almiani et al. 2020], which obtained 77.2% detection. However, this approach does not have a detection rate for normal flows. Furthermore, the works by [Almiani et al. 2020] and [Liang et al. 2022] had detection difficulties in the R2L category, achieving only 65% and 86% detection, respectively, while the proposed approach achieved 95%.

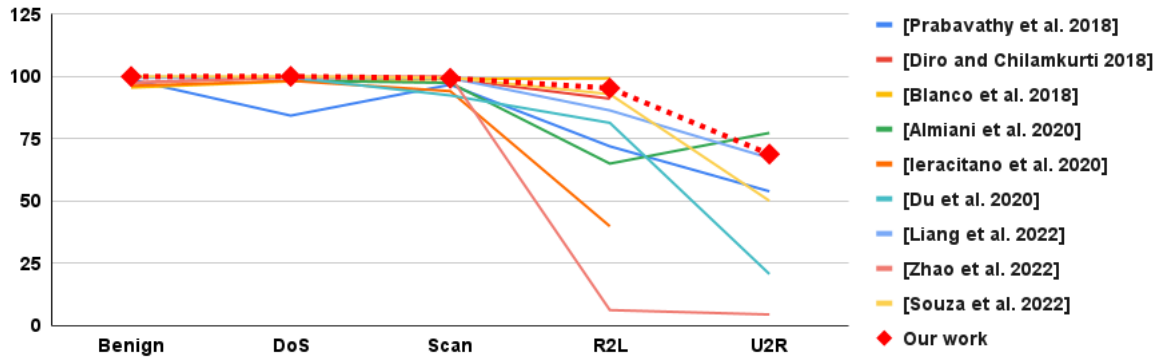

Figure 2. Results of experiments with the NSLKDD dataset.

As presented in Section 2, the approaches present difficulties related to the identification of benign traffic [Blanco et al. 2018, Ieracitano et al. 2020, Zhao et al. 2022]. The approach of [Blanco et al. 2018] obtained 95.4% identification of benign flows. This can be considered a weakness of the approaches as it indicates that a large amount of benign traffic may be mistakenly detected as an attack. False positives are a major problem with anomaly-based techniques and can damage the network. The DNNET-Ensemble obtained approximately 99.89% detection of normal flows, being the approach that obtained the highest metric. Therefore, it performs the detection of attacks generating a few false positives. This is due to the good detection capacity of the DNNET binary module deployed in the first detection level and the possibility of recovering false positives through the second detection level.

Observing the Balanced Accuracy (BACC), a metric that considers the imbalance between classes, it is noted that the proposed approach was able to overcome the others. The DNNET-Ensemble reached 92.6% of balanced average accuracy, while DNN obtained 85.8%, kNN 88.3%, RF 86,5%, ET 87.7% and [Souza et al. 2022] 88.41%.

4.3. Evaluation DNNET-Ensemble with IoTID20

The IoTID20 dataset has data referring to network traffic of IoT devices and interconnected structures typical of a Smart Home [Ullah and Mahmoud 2020]. Among the devices present in the monitored architecture are security cameras, for example.

Figure 3 shows that some works presented difficulties detecting some attacks. [Sarwar et al. 2022] had a recall of 50% in Scan attacks and only 13% for MITM. The authors [Qaddoura et al. 2021] identified only 55% of DoS and 74% of MITM attacks.

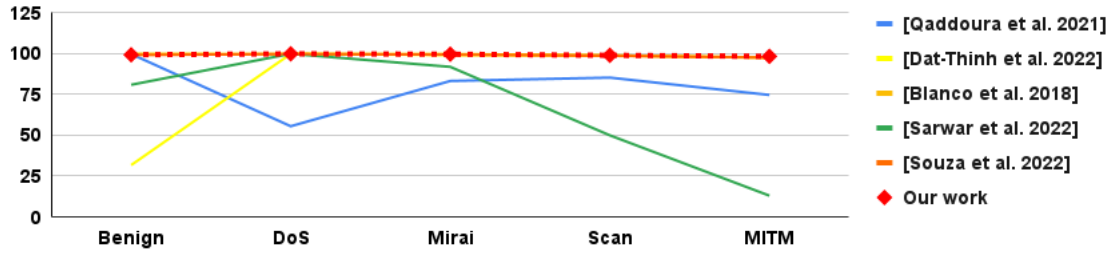

Figure 3. Results of experiments with the IoTID20 dataset.

Table 4. Results of experiments with the IoTID20 dataset.

Works	BACC	Class				
		Benign	DoS	Mirai	Scan	MITM
DNN	96.97	98.72	99.85	97.93	97.42	90.95
kNN	98.03	98.43	99.83	99.39	97.41	95.09
RF	98.10	98,55	99,91	99,66	97,52	94,89
ET	97.12	98.23	99.89	99.27	96.09	92.14
[Souza et al. 2022]	98.99	99.21	99.89	99.47	98.78	97.67
Our work	99.33	99.40	99.92	99.77	99.09	98.45

The approach proposed by [Dat-Thinh et al. 2022] identified only 31.7% of benign traffic. Other approaches also presented difficulties in identifying benign traffic, obtaining recall rates below 90% [Sarwar et al. 2022]. This indicates that the approach had false positive problems, where normal traffic is erroneously detected as intrusive. On the other hand, the DNNET-Ensemble detected approximately 99.4% of the existing normal traffic, thus generating a low rate of false positives. Furthermore, the proposed approach achieved recall rates of 98% for all attacks. This is mainly due to the proposed architecture that allows employing a robust ensemble method in the second detection level. It is also noteworthy that the good detection performance presented in each of the classes reflects the balanced accuracy metric, which reached 99.33%, the highest among the evaluated techniques, as can be seen in Table 4. The fact that it has a BACC of 99.3% allows us to conclude that the approach operates with a lower false positive rate than the other techniques, which presented greater difficulty in identifying benign traffic.

5. Conclusions and future works

This work proposes a two-level intrusion detection and identification approach in Fog Computing and IoT environments called DNNET-Ensemble. We propose improvements to a recent binary detection approach and generate the new DNNET binary approach. The results obtained in experiments with renowned intrusion datasets demonstrate that the approach can achieve performance superior to other classical machine learning techniques about prediction metrics. The DNNET-Ensemble achieved a balanced average accuracy of 92,6% for the NSL-KDD and 99,3% for IoTID20. Furthermore, compared with state-of-art approaches, the proposed approach capacity to generate a low rate of false positives is observed. Future work includes proposing countermeasure mechanisms and mapping

between attacks and countermeasures. Furthermore, conducting more experiments with new realistic IoT scenario datasets, such as the MQTTset [Vaccari et al. 2020] would be interesting.

References

[Albulayhi et al. 2022] Albulayhi, K., Abu Al-Haija, Q., Alsuhibany, A., A., Ashrafuzzaman, M., and Sheldon, F. T. (2022). Iot intrusion detection using machine learning with a novel high performing feature selection method. *Applied Sciences*, 12(10).

[Almiani et al. 2020] Almiani, M., AbuGhazleh, A., and Al-Rahayfeh, A. (2020). Deep recurrent neural network for iot intrusion detection system. *Simulation Modelling Practice and Theory*, 101:102031. Modeling and Simulation of Fog Computing.

[Blanco et al. 2018] Blanco, R., Malagón, P., Cilla, J. J., and Moya, J. M. (2018). Multiclass network attack classifier using cnn tuned with genetic algorithms. In *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 177–182.

[Bonomi et al. 2012] Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16, New York, NY, USA. Association for Computing Machinery.

[Dat-Thinh et al. 2022] Dat-Thinh, N., Xuan-Ninh, H., and Kim-Hung, L. (2022). Midsiot: A multistage intrusion detection system for internet of things. *Wireless Communications and Mobile Computing*, 2022.

[Diro and Chilamkurti 2018] Diro, A. A. and Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems*, 82:761 – 768.

[Du et al. 2020] Du, R., Li, Y., Liang, X., and Tian, J. (2020). Support vector machine intrusion detection scheme based on cloud-fog collaboration. In *International Conference on Security and Privacy in New Computing Environments*, pages 321–334. Springer.

[Geurts et al. 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

[Gopalakrishnan and Purusothaman 2022] Gopalakrishnan, B. and Purusothaman, P. (2022). A new design of intrusion detection in iot sector using optimal feature selection and high ranking-based ensemble learning model. *Peer-to-Peer Networking and Applications*, pages 1–28.

[Ieracitano et al. 2020] Ieracitano, C., Adeel, A., Morabito, F. C., and Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, 387:51 – 62.

[Liang et al. 2022] Liang, H., Liu, D., Zeng, X., and Ye, C. (2022). An intrusion detection method for advanced metering infrastructure based on federated learning. *Journal of Modern Power Systems and Clean Energy*, pages 1–11.

[Mohamed Omar et al. 2021] Mohamed Omar, H. O., Goyal, S. B., and Varadarajan, V. (2021). Application of sliding window deep learning for intrusion detection in fog computing. In *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, pages 1–6.

[Nguyen et al. 2019] Nguyen, T. G., Phan, T. V., Nguyen, B. T., So-In, C., Baig, Z. A., and Sanguanpong, S. (2019). Search: A collaborative and intelligent nids architecture for sdn-based cloud iot networks. *IEEE Access*, 7:107678–107694.

[Ni et al. 2018] Ni, J., Zhang, K.and Lin, X., and Shen, X. (2018). Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Communications Surveys & Tutorials*.

[Prabavathy et al. 2018] Prabavathy, S., Sundarakantham, K., and Shalinie, S. M. (2018). Design of cognitive fog computing for intrusion detection in internet of things. *Journal of Communications and Networks*, 20(3):291–298.

[Qaddoura et al. 2021] Qaddoura, R., M. Al-Zoubi, A., Faris, H., and Almomani, I. (2021). A multi-layer classification approach for intrusion detection in iot networks based on deep learning. *Sensors*, 21(9).

[Sahar et al. 2021] Sahar, N., Mishra, R., and Kalam, S. (2021). Deep learning approach-based network intrusion detection system for fog-assisted iot. In *Proceedings of international conference on big data, machine learning and their applications*, pages 39–50. Springer.

[Sarwar et al. 2022] Sarwar, A., Hasan, S., Khan, W. U., Ahmed, S., and Marwat, S. N. K. (2022). Design of an advance intrusion detection system for iot networks. In *2022 2nd International Conference on Artificial Intelligence (ICAI)*, pages 46–51.

[Souza et al. 2022] Souza, C. A., Westphall, C. B., and Machado, R. B. (2022). Two-step ensemble approach for intrusion detection and identification in iot and fog computing environments. *Computers & Electrical Engineering*, 98:107694.

[Souza et al. 2022] Souza, C. A., Westphall, C. B., Machado, R. B., Loffi, L., Westphall, C. M., and Geronimo, G. A. (2022). Intrusion detection and prevention in fog based iot environments: A systematic literature review. *Computer Networks*, 214:109154.

[Souza et al. 2020] Souza, C. A., Westphall, C. B., Machado, R. B., Sobral, J. B. M., and Vieira, G. S. (2020). Hybrid approach to intrusion detection in fog-based iot environments. *Computer Networks*, 180:107417.

[Ullah and Mahmoud 2020] Ullah, I. and Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in iot networks. In *Canadian Conference on Artificial Intelligence*, pages 508–520. Springer.

[Vaccari et al. 2020] Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., and Cambiaso, E. (2020). Mqttset, a new dataset for machine learning techniques on mqtt. *Sensors*, 20(22):6578.

[Zhao et al. 2022] Zhao, R., Mu, Y., Zou, L., and Wen, X. (2022). A hybrid intrusion detection system based on feature selection and weighted stacking classifier. *IEEE Access*, pages 1–14.