A Scalable And Efficient Intrusion Detection System Based On A Shapley Value Driven Feature Selection

Giovanni Rocca, Mattia G. Spina, Floriano De Rango
Department of Informatics, Modelling, Electronics and Systems (DIMES)
Università della Calabria
Rende, Italia
{giovanni.rocca, mattiagiovanni.spina, f.derango}@dimes.unical.it

***Abstract*—The proliferation of IoT devices, amplified by 6G, has heightened security risks. To counter these, experts are improving Intrusion Detection Systems (IDSs) using Machine Learning (ML) and Deep Learning (DL) algorithms, which rely on Feature Selection (FS) to identify key features and optimize performance in detecting attacks. In this context, this paper proposes a novel informed FS technique that exploits XAI (eXplainable Artificial Intelligence). The proposed algorithm extends the use of Shapley Value in XAI by enhancing it with an *accuracy-driven* binary search with the aim of finding the most representative feature and therefore reducing the dimensionality of a Network Intrusion Dataset, improving its detection ability. Through a comprehensive experimental campaign, the proposal has been validated and its benefits, which are not only limited to reduced training and testing time, but also drastically streamlining the AI model complexity. Finally, a comparison with other standard FS techniques is also provided to further highlight the advantages of the proposed algorithm. The proposal has been validated through a comprehensive experimental campaign in which results show the benefits. The experimental section shows the performance of the model when trained on the set of features returned by DeepSHAP and on the set returned by our framework, showing a drastic reduction in training and testing time.**

***Index Terms*—Intrusion Detection System, Explainable Artificial Intelligence, Network Security**

I. Introduction

The rise of connected devices, driven by the IoT paradigm [1], has made networks easier targets for hackers. Even novice attackers can now disrupt networks for hours, exploiting IoT devices' weak security [2]. A skilled attacker controlling numerous compromised devices, or "zombies," could launch a large-scale DDoS attack. In this context, IDSs play a vital role. IDSs are generally divided into Signature-Based and Anomaly-Based [3]. Signature-Based IDSs aim to identify key patterns of attacks to enable accurate detection. When the IDS detects a pattern similar to a previously observed attack, it triggers an alert. However, the rise of new threats showed how this approach has become less effective, prompting researchers to focus more on Anomaly-Based IDSs. This type of IDS analyzes normal network traffic and flags any significant deviation as an anomaly. Often, these systems rely on ML algorithms to make decisions [4], using data quality as a foundation for accuracy. These types of IDSs are showing impressive results on the most recent datasets [5]. Anyway, this domain suffers from a limitation. The widespread availability of data raises issues like the curse of dimensionality, which refers to data sparsity in high-dimensional spaces. In addition to that, models that deal with datasets having a large number of features could become very complex and not behave well, as they tend to overfit the unknown data. To reduce the size of a dataset, one of the most used techniques is the FS method. The effectiveness of the IDS becomes paramount since it could be employed in critical and disaster scenarios [6]. This work presents a new FS technique, driven by XAI. A new framework is proposed that extends our previous research [7], based on the concatenation of Statistical Methods to reduce the size of a dataset. Thanks to SHAP (SHapley Addictive exPlanations), the dataset's most important features are extracted. By applying a binary search-inspired algorithm, the smallest subset of features that enables the ML model to achieve at least the desired *accuracy* is identified, with *accuracy* serving as the target within this feature ranking. The analysed dataset is the CSE-CIC-IDS2018. The attacking infrastructure used by the Canadian Institute for Cybersecurity includes 50 machines, and the victim organization includes 420 machines and 30 servers. The heterogeneity of the attacking infrastructure, along with the 80 features extracted from the captured traffic, makes this dataset the most suitable choice for the purposes of this work. The rest of the paper is organised as follows: Section II shows the related works. In Section III, a brief theoretical background about the Shapley Values is provided. Section IV describes the framework proposed. Section V presents the tests made on a real Intrusion Detection dataset, CSE-CIC-IDS2018. Finally, section VI concludes the work.

II. Related Works

A. Feature Selection

In recent years, lots of researchers have proposed their techniques in order to try to reduce the size of a large-dimensional dataset, improving IDS effectiveness against network attacks [8]. For instance, Saq et al. [9] present a new

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

method called Fuzzy Mutual Information-based Feature Selection to improve IDS for IoT networks. This approach integrates fuzzy logic with Gaussian membership functions to handle the uncertainty and imprecision typical of IoT data, overcoming the limitations of traditional mutual information-based feature selection methods. Even though in the article is presented an IDS for IoT networks, the dataset used is very specific and cannot generalize well the heterogeneity of the IoT devices. Another article that proposes an FS method to improve the detection of botnet attacks in IoT devices is [10]. The authors combine three feature selection techniques (correlation, GDO, and LASSO [11]) to identify the most influential attributes of a network intrusion dataset. A method that combines Pearson Correlation and Lasso is proposed in [12]. The authors apply their method to the UNSW-NB15 dataset, boosting SVM accuracy and reducing the false positive rate. It is worth noting that using a synthetic dataset, even if it is one of the most used benchmark datasets, could not be helpful to the IDS when it has to generalize several attack patterns. An interesting work published by Ren et al in [13] filters the features in order to find the optimum subset using a Recursive Feature Elimination (RFE) approach. Also their method is tested on the CSE-CIC-IDS2018 dataset. In [14], the authors proposed an FS method to improve the performance of a K-NN classifier trained on IoT traffic. To select the best features, the authors use techniques like principal component analysis (PCA), univariate statistical test, and genetic algorithm (GA). The most important result obtained by the researchers is the reduction of the prediction time, a critical factor, especially for real-time IDSs.

B. XAI Feature Selection

Recently, a new paradigm that explains ML and DL models is gaining ground: the XAI. These techniques could be used to build FS methods. The study published in [15] exploits explanations given by XAI to highlight the impact of their procedure in biomedical applications. Wilson et al. [16] also explore the use of SHAP as an FS mechanism in ML pipelines. The authors highlight the limited focus on pre-processing steps, such as FS, in explainability research. Experimental results demonstrate that SHAP not only explains model decisions but also outperforms three common FS algorithms. In [17], the authors argue that the pre-processing stage has not received the right attention in establishing models' explainability. [18] compares FS techniques for credit card fraud detection using SHAP values and model-built feature importance rankings.

Main Contribution

Nowadays, network heterogeneity is increasing the complexity of datasets. For this reason, standard FS techniques could not be sufficient to retrieve the most important features of a dataset. As shown in this section, lots of papers must concatenate several techniques to obtain a consistent subset from the initial set. Improving on our previous work [7], this paper presents an innovative FS technique leveraged by XAI to determine the most relevant characteristics of the analysed dataset. In contrast to the current state of the art about XAI FS techniques, this work proposes an innovative stop criterion. The proposed approach uses the target *accuracy*, chosen *a priori*, to employ a binary search-inspired method on the feature ranking generated by SHAP. This process identifies the minimal subset of features required to enable the AI-based IDS to achieve predictions that meet or exceed the specified *accuracy* threshold.

III. Theoretical Background

A. Shapley Value

The Shapley Value is a concept of game theory published in a work by Shapley in 1953 [19]. The main idea is to evaluate the contribution of a player in a collaborative game. AI-based IDSs process large volumes of network traffic, using this data to analyse new traffic flows and determine if they are benign. This classification (or prediction) occurs behind the scenes through AI algorithms, often viewed as "black boxes" since experts may not always investigate beyond the algorithm's output. Shapley values, however, allow experts to pinpoint which features most influence the classification of specific classes within the dataset, providing insight into why a given instance is labelled benign or not. To clarify a prediction, imagine each feature value as a "player" in a game where the prediction represents the ultimate reward. The Shapley Value for each player (or feature, in this context) is defined by equation 1:

$$\phi_j(v) = \sum_{S \subseteq \{1:p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(v\left(S \cup \{j\}\right) - v(S)\right) \tag{1}$$

where:

- S refers to a subset of the features used in the model.
- p represents the vector of feature values for a particular instance within the dataset.
- v is the prediction based on the feature values within subset S, with all other features outside of S marginalised.

Equation 1 defines the Shapley Value, which quantifies a feature's "fair" contribution to the model's prediction. This calculation is based on a weighted average of the feature's impact across all potential combinations of the other features, much like determining a player's fair share in game theory. Adapting this principle to ML, the Shapley Value output reflects each feature's contribution to the prediction for a specific instance, considering all possible combinations of features as analogous to player coalitions in a game. Consequently, the result derived from Equation 1 provides a measure of each feature's influence on the classification of a particular instance. This technique is precious for identifying features that define a specific class within a dataset. Additionally, the Shapley Value approach is versatile and can be applied to any ML/DL model, offering robustness regardless of the dataset or class under analysis.

B. SHapley Addictive exPlanations (SHAP) Framework

SHAP is a comprehensive framework introduced by Lundberg and Lee [20] for understanding predictions. It clarifies the prediction of a specific instance x by assessing the impact of each feature on that prediction. A notable innovation of the SHAP method is its representation of Shapley value explanations as a linear model, which links the approaches of LIME (Local Interpretable Model-agnostic Explanations) and Shapley Values. Each SHAP value indicates the degree to which each feature contributes to the model's prediction, whether positively or negatively. SHAP values provide two key advantages: they can be computed for any type of model, not just straightforward linear ones, and each record has its unique set of SHAP values. A clear explanation of the math behind SHAP is provided by Molnar in his book [21]. In this work, he explains how SHAP defines the interpretation for an instance x in the following equation:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{2}$$

where:

- g is the explanation model.
- $z' \in \{0,1\}^M$ is the coalition vector.
- M is the maximum coalition size.
- $\phi_j \in \mathbb{R}$ is the feature attribution for a feature j.

In the coalition vector, a value of 1 indicates that the corresponding feature is *present*, while a value of 0 signifies that it is *absent*. Shapley values are calculated by simulating scenarios in which certain feature values are active, *present*, while others remain inactive *absent*. As it can be seen, SHAP values could be very important in future developments of IDS. There is a little drawback, though. Calculating the Shapley values for the features of a dataset involves a high computational cost due to the high number of possible feature combinations. For a single feature j in a set of N features, there are 2^{N-1} possible subsets that don't include j. Each subset requires the computation of $v(S\ U\ \{j\}) - v(S))$, which consists of evaluating the difference between the value of the evaluation function v when a particular feature j is included in a subset S of features and the value of the v when i is not included in S. v, here, represents the prediction of the model based only on the features in S. This process must be repeated for each feature of the dataset. For a single instance of the dataset, the total number of operations required to calculate the Shapley values is $O(2^{N-1} * N)$, where:

- 2^{N-1} is the number of subsets for every feature.
- N is the total number of features.

For this reason, SHAP does not evaluate the exact Shapley values for the features. As the number of features grow, calculating the exact Shapley values for each feature is computationally intensive. Hence, approximations like DeepSHAP were developed to make Shapley value computations feasible.

C. DeepSHAP

DeepSHAP is an approximation method designed to explain the predictions of DL models by combining the principles of SHAP values with a mechanism tailored for deep Neural Networks (NN). DeepSHAP leverages the DeepLIFT [22] (Deep Learning Important FeaTures) method to approximate SHAP values for NN. DeepLIFT computes the contribution of each input feature relative to a baseline by tracking the flow of activation differences through the network. DeepSHAP adheres to SHAP's axiomatic properties, such as local accuracy (the sum of feature contributions equals the model output) and feature attribution consistency (ensuring consistent rankings of feature importance).

IV. Proposed Framework

In this section it is offered a detailed description of the FS technique proposed.

Algorithm 1 Input: $F, data, target$

```
F' ← SHAP ranking(F)
if accuracy(data[F']) < target then
    F' = [ ]
    return F'
end if
start ← 0
end ← size of F'
while start ≤ end do
    mid ← ⌈(start + end)/2⌉
    if accuracy(data[F'_[0:mid]]) ≥ target then
        end ← mid
    else
        start ← mid
    end if
end while
F' = F'_[0:end]
return F'
```

The procedure is summarised by Algorithm 1, and it is explained below. With respect to the state of the art, this procedure offers a novel FS technique guided by an important metric for IDS, the *accuracy*. This framework takes as input three parameters: the dataset to analyse, the full set of features, and the target *accuracy*. Therefore, the last parameter selected by the user drives the FS technique. More specifically, the framework evaluates F', the ranking of the most important features of the dataset using the methods described in the previous section. Then, the procedure computes the *accuracy* obtained by the underlying model of the IDS, trained on F', on predicting new instances. This operation works as a preliminary check. As a matter of fact, if the *accuracy* is below the selected target, the algorithm stops and returns an empty subset of features. This means that there is not a subset of characteristics that can guarantee the *accuracy* selected by the user on the prediction of new instances. The user should launch again the procedure, this time reducing the *accuracy* selected before. Otherwise, the algorithm starts the FS, by doing a

binary search on F'. The binary search consists in finding the minimum subset of features that gives the IDS the *accuracy* selected by the user in predicting new instances. Based on the binary search principle, the model is trained only on half of the features that make up the ranking. If the model's *accuracy* during the inference phase exceeds the target value, the size of F' is further halved for the next iteration. The algorithm stops when the values of the two indexes, $start$ and end, coincide. So, the subset of F' with indexes from 0 to end, represents the subset that matches user's needs. The steps of the procedure are illustrated by Fig. 1.

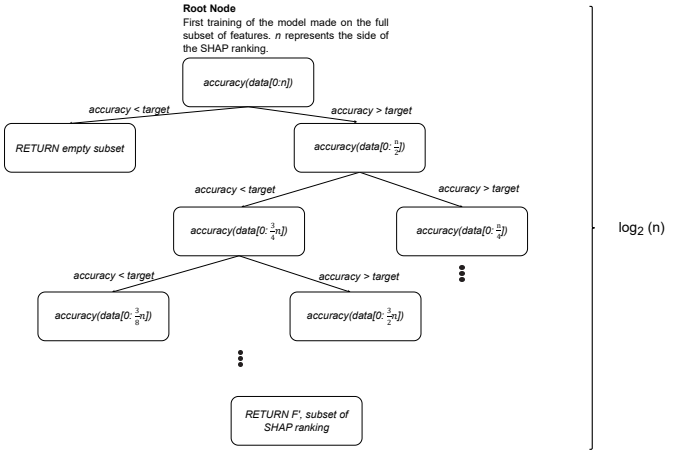

Fig. 1. Binary Search Feature Selection

The algorithm, since it works on an ordered array, the SHAP ranking, will find the minimal subset of features in a logarithmic time with respect to the size of the ranking, $log_2(n)$, where n is the total number of features. This *accuracy* driven algorithm is dual purpose. First, reducing features in a large dimensional dataset, like a Network Intrusion one, helps model reduce overfitting [23] and decrease inference time. Second, as it can be seen in the next section, this algorithm tunes the underlying model of the IDS, thus reducing its complexity.

V. Test on real dataset

This section shows a use case of our framework tested on the CSE-CIC-IDS2018 dataset [24], provided by the Canadian Institute for Cybersecurity. After a description of the dataset, the experimental setup is discussed, together with the analysis of the AI model used. Finally, the obtained results are compared with standard FS techniques.

A. CSE-CIC-IDS2018

The CSE-CIC-IDS2018 dataset is widely used for IDS development, featuring attacks DDoS and brute-force. It contains 80 features capturing network flow characteristics. The dataset simulates real network behaviour with legitimate and attacker profiles using various protocols, such as HTTP, with different attacks performed daily.

TABLE I
Number of instances of each class

Class name	# of instances	Name of the file	Attack family
Benign	1.048.213	14-02-2018	Benign
Ares	193.360	02-03-2018	Botnet
Hulk	461.912	16-02-2018	DoS
LOIC	686.012	21-02-2018	DDoS
NMAP port scan	93.063	01-03-2018	Infiltration

B. Feature Selection

The evaluation of the technique involves the development of a multiclassifier AI-based IDS. The underlying model is the NN proposed in [25]. The hyper-parameters used are summarised in Table II. All tests are runned on Python 3.8.10

TABLE II
Hyper-parameters used in the training phase

Hyper-Parameters	**Value**
Number of Epochs	200
Learning Rate	0.001
Batch Size	1024
Validation Split	20%
Optimizer	Adam
Hidden Layer Activation	ReLU
Output Activation Function	Categorical Cross-Entropy

and the NN is implemented using the Keras library. For this evaluation 5 different traffic classes were chosen. The Benign traffic class is accompanied by 4 attacks, namely Ares, a Botnet written in Python, Hulk, a volumetric DoS attack, LOIC, which is the acronym of Low Orbit Ion Cannon, a DDoS volumetric attack, and the last one is an attempt of Information gathering using NMAP. The number of instances per class is illustrated by table I. The data passed to the NN was appropriately preprocessed. So, for instance, all NaN values were removed, and the data was scaled between 0 and 1. Then, before to the application of DeepSHAP, the approximation method for the evaluation of the Shapley Values of the features that feed the NN, an important factor must be taken into account. The Shapley Values fairly distribute the contribution made by every feature used by the model. In case of strongly correlated features, i.e. where there are features that represent the same information more than once, the Shapley Value assigned to each feature may be distorted. For this reason, we use our previous work [7] to remove strongly correlated features from the dataset, in order to obtain a ranking that is as correct as possible. The first training of the framework was carried out on the entire dataset, deprived only of highly correlated features. This first stage returns us the ranking of the most important features, shown in Fig. 2. The plot shown in this picture is called Summary Plot and highlights the contribution that each feature has on the prediction of the model. Once the ranking has been obtained, the research of the subset of features guided by the desired *accuracy* starts. For our purposes, we chose to search the minimum subset of features that makes the algorithm work with an *accuracy* of at least 99%. The subset of features

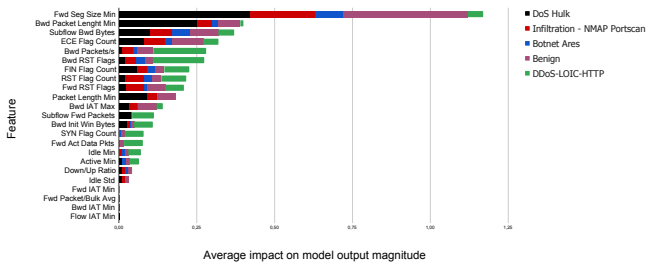

Fig. 2. Summary Plot

returned by SHAP has a size of 24. Features that have zero impact on model prediction are automatically excluded from this ranking. With just 5 iterations, basically $log_2(24)$, our algorithm finds that the minimum subset of features that makes the model work with an *accuracy* of at least 99% is 10. Thus, the NN described in the first part of this section, if trained on the first 10 features shown on picture 2, has an *accuracy* of 99% in predicting new instances of the dataset.

C. Model Tuning

As stated before, not only does this procedure reduce the size of a high-dimensional data set, but it also streamlines the structure of the model used, especially when dealing with DL models. In the NN used in this work the number of neurons of each hidden layer strongly depends on the input layer, i.e. the number of features. If n is the number of features used to train the network, the hidden layers follow this scheme: [n, $n \times 80\%$, $n \times 60\%$, $n \times 40\%$, $n \times 30\%$, $n \times 40\%$, $n \times 60\%$, $n \times 80\%$, n]. So, this FS technique acts as a model tuner, because when it finds a minimum subset guided by the *accuracy* target, it returns a smaller set of features with which the model can work.

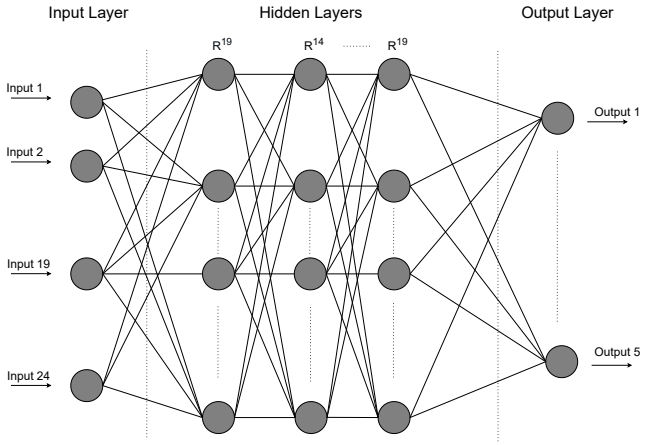

Fig. 3. Neural Network trained on all features

Fig. 3 and 4 show the difference between the two NNs. The first picture represents the NN on the initial set of features without the highly correlated ones. Fig. 4 shows the structure of the NN trained after the FS technique. The number of

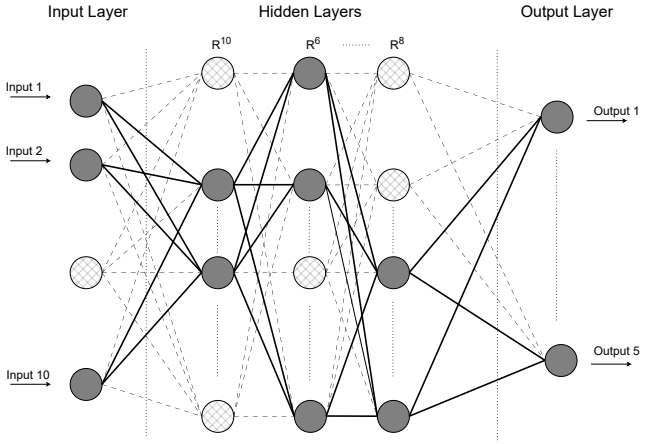

Fig. 4. Neural Network after the FS technique

hidden nodes, which depends on the number of input nodes, has been reduced by approximately 58%, as we can see in the pictures.

TABLE III
COMPARISON BETWEEN DEEPSHAP AND THE PROPOSED FRAMEWORK

	DeepSHAP	LASSO	Mutual Information	Proposed Framework
Accuracy	∼99%	∼99%	∼77%	∼99%
F1 Score	∼99%	∼99%	∼71%	∼99%
Recall	∼99%	∼99%	∼77%	∼99%
Training Time	203.167 s	177.492 s	129.220 s	129.623 s
Testing Time	2.203 s	4.477 s	2.323 s	0.358 s

Table III shows the performances of the NN trained on the full ranking returned by DeepSHAP compared with the performances of the proposed algorithm, along with some other known FS techniques. For the purposes of this work, the proposed algorithm is compared to LASSO and Mutual Information (MI). The first one selects 34 out of the 80 features of the dataset. This still guarantees good *accuracy* and F1 score performances of the IDS, but with a higher number of features, consequently increasing training and testing time. Finally, MI extracts 9 features out of 80. MI measures only the statistical dependence between each feature and the target variable. MI does not take into account correlations between features. If the dataset has highly correlated features, MI may choose only some of them, losing useful information. For this reason, NN performances with the features extracted by MI are indeed lower compared to the one of the proposed method. The FS technique proposed in this paper, together with the tuning of the model, has halved Training and Testing time of the NN, maintaining optimum performances in the inference phase.

Table IV and V show a comparison of the parameters of the two NNs. The number of neurons has been reduced by nearly 88%. This reduction of neurons brings the following computational benefits:

- Reducing the number of neurons leads to a decrease in the total number of parameters that the model must update during training.

TABLE IV
NUMBER OF CONNECTIONS AND BIASES FOR EACH LAYER OF THE NN TRAINED ON 24 FEATURES

Layer	N_{in}	N_{out}	Connections	Bias
Input → Dense(19)	24	19	$24 \times 19 = 456$	19
Dense(19) → Dense(14)	19	14	$19 \times 14 = 266$	14
Dense(14) → Dense(10)	14	10	$14 \times 10 = 140$	10
Dense(10) → Dense(7)	10	7	$10 \times 7 = 70$	7
Dense(7) → Dense(10)	7	10	$7 \times 10 = 70$	10
Dense(10) → Dense(14)	10	14	$10 \times 14 = 140$	14
Dense(14) → Dense(19)	14	19	$14 \times 19 = 266$	19
Dense(19) → Dense(5)	19	5	$19 \times 5 = 95$	5
Total	–	–	**1503**	**98**

TABLE V
NUMBER OF CONNECTIONS AND BIASES FOR EACH LAYER OF THE NN TRAINED ON 10 FEATURES

Layer	N_{in}	N_{out}	Connections	Bias
Input → Dense(8)	10	8	$10 \times 8 = 80$	8
Dense(8) → Dense(6)	8	6	$8 \times 6 = 48$	6
Dense(6) → Dense(4)	6	4	$6 \times 4 = 24$	4
Dense(4) → Dense(3)	4	3	$4 \times 3 = 12$	3
Dense(3) → Dense(4)	3	4	$3 \times 4 = 12$	4
Dense(4) → Dense(6)	4	6	$4 \times 6 = 24$	6
Dense(6) → Dense(8)	6	8	$6 \times 8 = 48$	8
Dense(8) → Dense(5)	8	5	$8 \times 5 = 40$	5
Total	–	–	**288**	**44**

- Weights take up memory space, so with fewer neurons, the network requires less memory.
- By reducing neurons, the total number of operations required during each training step decreases. For this reason, the training phase of the NN that works with 10 features is faster than the training on the other network.
- The same applies to the inference phase. A faster inference phase could be crucial in a real-time scenario.

VI. CONCLUSIONS

This work tackles cybersecurity challenges from IoT expansion and 6G by introducing an *Accuracy-driven* FS method using XAI for IDSs. The approach leverages SHAP rankings and a binary search-inspired strategy to identify the smallest feature subset meeting a target accuracy. The experiments conducted on an NN, using the approximation method DeepSHAP, show its effectiveness in reducing dataset size, simplifying models, and accelerating both the training and testing phases. A final comparison with standard FS techniques confirms the value of the proposed work, enhancing AI-based IDSs.

ACKNOWLEDGEMENTS

This work was partially supported by project SERICS (PE00000014), under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

REFERENCES

[1] B. Bala and S. Behal, "Ai techniques for iot-based ddos attack detection: Taxonomies, comprehensive review and research challenges," *Computer Science Review*, vol. 52, p. 100631, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013724000157

[2] M. Malik *et al.*, "Defending ddos in the insecure internet of things: A survey," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, S. S. Dash, P. C. B. Naidu, R. Bayindir, and S. Das, Eds. Singapore: Springer Singapore, 2018, pp. 223–233.

[3] A. Khraisat *et al.*, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, 12 2019.

[4] F. Salatino *et al.*, "Detecting DDoS Attacks Through AI driven SDN Intrusion Detection System," in *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*. IEEE, pp. 06–09.

[5] O. Abdulganiyu *et al.*, "A systematic literature review for network intrusion detection system (ids)," *International Journal of Information Security*, vol. 22, 03 2023.

[6] M. Tropea, M. G. Spina, and F. De Rango, "Supporting Dynamic IDS Deployment with Load Balancing Strategy for SDN-enabled Drones in Emergency Scenarios," in *ACM Conferences*. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 297–300.

[7] G. Rocca *et al.*, "Integrating statistical methods and game theory for enhanced iot intrusion detection," in *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*, 2025, pp. 1–4.

[8] M. Tropea, M. G. Spina, and F. De Rango, "The Evolution of Network Intrusion Detection Systems: From Legacy to Programmable Networks," in *Mastering Intrusion Detection for Cybersecurity*. IntechOpen, Jul. 2025.

[9] A. Saq *et al.*, "Intrusion detection in iot using gaussian fuzzy mutual information-based feature selection," *Engineering, Technology Applied Science Research*, vol. 14, pp. 17 564–17 571, 12 2024.

[10] H. Y. A. Alshaeaa and Z. Ghadhban, "Developing a hybrid feature selection method to detect botnet attacks in iot devices," *Kuwait Journal of Science*, vol. 51, p. 100222, 04 2024.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[12] I. H. Putro and T. Ahmad, "Feature selection using pearson correlation with lasso regression for intrusion detection system," in *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*, 2024, pp. 1–6.

[13] K. Ren *et al.*, "Id-rdrl: a deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports*, vol. 12, 09 2022.

[14] M. Mohy-eddine *et al.*, "An efficient network intrusion detection model for iot security using k-nn classifier and feature selection," *Multimedia Tools and Applications*, vol. 82, 02 2023.

[15] H. Wang *et al.*, "Explanations as a new metric for feature selection: A systematic approach," *IEEE journal of biomedical and health informatics*, vol. PP, 05 2023.

[16] W. E. Marcílio and D. M. Eler, "From explanations to feature selection: assessing shap values as feature selection mechanism," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2020, pp. 340–347.

[17] J. Zacharias *et al.*, "Designing a feature selection method based on explainable artificial intelligence," *Electronic Markets*, vol. 32, 12 2022.

[18] H. Wang *et al.*, "Feature selection strategies: a comparative analysis of shap-value and importance-based methods," *Journal of Big Data*, vol. 11, 03 2024.

[19] L. S. Shapley, "A value for n-person games," *Contribution to the Theory of Games*, vol. 2, 1953.

[20] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017. [Online]. Available: https://arxiv.org/abs/1705.07874

[21] M. Christoph, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.

[22] S. Avanti *et al.*, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3145–3153. [Online]. Available: https://proceedings.mlr.press/v70/shrikumar17a.html

[23] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022022, feb 2019.

[24] I. Sharafaldin *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[25] M. A. Salahuddin *et al.*, "Chronos: Ddos attack detection using time-based autoencoder," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 627–641, 2022.