False Positive Identification in Intrusion Detection Using XAI

R. S. Lopes, J. C. Duarte, and R. R. Goldschmidt

Abstract—**With the increase in the use of the Internet to access sensitive data, intrusion detection has become an essential security measure. The evolution that took place in Artificial Intelligence in the last decades, notably in Machine Learning techniques, combined with the availability of network traffic datasets, opened a vast field for research and development in Intrusion Detection Systems based on anomalies. Published studies on this subject, nonetheless, are unanimous in stating that this type of detection is more prone to the occurrence of false positives. In order to mitigate this problem, we propose a more effective method of identifying them, compared to using only the algorithm's confidence. For this, we hypothesize that the relevance given by the algorithm to certain attributes may be related to whether the detection is true or false. The method consists, therefore, in obtaining these features relevance through eXplainable Artificial Intelligence (XAI) and, together with a confidence measure, identifying detections that are more likely to be false. By using the LYCOS-IDS2017 dataset, it is possible to eliminate more than 65% of the total false positives, with a loss of only 0.38% of true positives. Conversely, by using only a confidence measure, the elimination of false positives is approximately just 50%, with a loss of 0.42% of true positives.**

Index Terms—**Intrusion detection, machine learning, explainability, XAI, false positive rate.**

I. Introduction

Intrusion detection is an important activity that aims to improve the security level in computer systems. It complements other devices and techniques (e.g., firewalls and cryptography), being considered the last line of defense [1], [2]. As attackers learn to circumvent firewalls, crack passwords, steal cryptography keys, etc., Intrusion Detection Systems (IDS) become a mandatory device where sensible data is traveling.

An IDS operates as a network (NIDS) or as a host (HIDS) device and monitors events (e.g., IP packet traffic, calls to the operational system, logs, and file systems) in order to find signs of security policy violations. It is categorized in signature (sometimes called misuse) or anomaly-based detection [3]–[5]. The first one compares characteristics of the monitored data against signatures or rules related to known attacks. The second one creates a model to represent normal (or benign) data and monitors deviations from it, which has the advantage of detecting unknown attacks, albeit at the price of more false positives.

Advances in Machine Learning (ML) applied to anomaly IDS resulted, at least theoretically, in a sharp reduction in mistaken detections. It is hard to say, nonetheless, if it occurs in the real world, as we don't have real data with their ground truth to test the detection effectiveness [5]. Furthermore, it is not possible to assure the reliability of evaluations on synthetic datasets, where the highly complex open-world network traffic characteristics are hard to simulate [6].

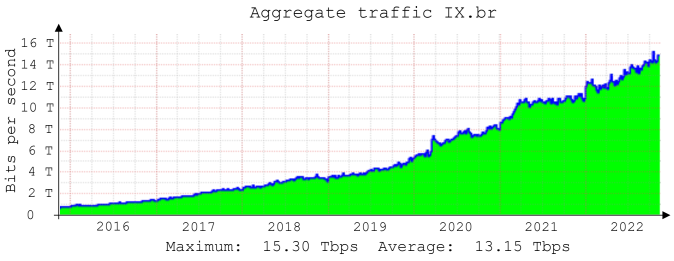

Fig. 1. Growth of Brazilian internet traffic. Source: [7].

The main consequence of false positives is the loss of detection credibility, inducing the security analyst to neglect a proper alarm analysis. In addition, active IDS, also known as IDPS ("P" stands for Prevention), causes annoyance to legitimate users when it actively blocks benign traffic wrongly deemed malicious.

The reduction of false positives can be achieved with a proper IDS sensitivity adjustment, however, this is a challenging procedure caused by a trade-off between false positives and false negatives. While it is desirable to have both reduced, these measures are generally inversely proportional. Therefore, false positives reduction results in an increase of false negatives, which can be even more harmful, as false negatives represent real unnoticed attacks, and Information Technology (IT) companies rather try to decrease them, even at the cost of more false positives [4].

With the significant growth of internet traffic, especially the benign one, even IDS with a low false positive rate can generate, in absolute numbers, a substantial amount of false alarms. This hypothesis can be supported by Fig. 1, which shows such an increase in band usage, mainly due to more services being available on the Internet, together with other ways of access (e.g., smartphones). While this is related to Brazil, it is reasonable that the same growth is happening all over the world.

False positive reduction usually can be done in two ways: by improving the model accuracy or by post-processing positive detections. Both ways are complementary, therefore, they can be applied concurrently. One interesting strategy is to obtain the best model possible, from the point of view of accuracy. After that, if the number of positive detections is higher

R. S. Lopes, J. C. Duarte and R. R. Goldschmidt are with Military Institute of Engineering, Brazil, e-mails: ricardo.lopes@ime.eb.br, duarte@ime.eb.br and ronaldo.rgold@ime.eb.br.

than the security team analysis capability, the post-processing technique is applied, thus reducing the detections that need analysis to an acceptable amount. There are two common post-processing methods: filtering out detections more likely to be false positives and grouping duplicate detections or those related to the same attack. Again, both are complementary and can also be applied concurrently.

In this article, a post-processing method is proposed that aims to filter out false positives. What distinguishes this method from others is the use of recent explainable techniques in an ML-powered anomaly-based IDS.

The remainder of this paper is organized as follows: Section II provides a brief background of explainability in ML and the reason it is needed. Section III presents some works dealing with false positives reduction via post-processing and highlights the differences related to the proposed approach, which is described in Section IV. An analysis of the results is presented in Section V, where the method is applied to identify false positives generated by an anomaly-based IDS on the LYCOS-IDS2017 dataset. Finally, Section VI concludes the paper and proposes future improvements.

II. Machine Learning and XAI

XAI stands for eXplainable Artificial Intelligence and, as the name suggests, it emerged as a way to explain the outputs of ML algorithms, especially those very complex and that are considered black boxes. Such algorithms are the result of increased computational power combined with the search for better performance. A classic example is Deep Neural Networks (DNN), which sometimes have hundreds of neural layers. This results in an overwhelming amount of parameters being adjusted, which is accomplished in a reasonable amount of time only with graphical unit processors (GPU). DNN succeeded notably in computer vision and natural language processing, but its great performance comes at a cost: the lack of a profound understanding of exactly what the algorithm has learned from data to perform a prediction. Other non-transparent algorithms are those based on ensemble methods (e.g., Random Forest), and even rule-based algorithms (e.g., Decision Trees), which are self-explainable, can harm its intelligibility when their outputs are the result of a large number of mixed rules.

The need for XAI exists in several areas, being mandatory in some countries, particularly on subjects related to law and rights. One example is the use of AI systems in loan decisions, providing the so-called right to explanation. It means that if a loan company refuses to give a loan, it is obliged to explain the reason for refusal outputted by the algorithm. Other demanding areas of XAI are those with low fault toleration, as long as it is possible to know the reasons behind the failure. Thus, XAI can help fix, or at least predict, situations where failure is more prone.

Deep Learning has obtained high accuracy in intrusion detection, with a reduced false positive rate [3]. Nonetheless, these rates can represent unrealistic values, being not supported by practical applications. When this occurs, it is quite likely that the model has learned specific relationships that exist only in the evaluation dataset but not in the production environment. When the model is not very transparent, it is quite difficult to predict such behavior, which has raised the importance of developing methods that reasonably explain the reasons behind the decisions taken by these models.

Despite the large number of XAI techniques, they share the same goal: spotting which attributes are most important in a particular decision (or a set of decisions) and how they are related. Here, two methods are highlighted, named Adversarial Approach and SHAP.

A. Adversarial Approach

This technique was presented by Marino et al. [8], and starts with the search for the smallest possible change in the attributes of a misclassified sample, so that the model starts to classify it correctly. The main idea is, therefore, to use these changes as an explanation for the model error. Originally, such strategy is used by the adversary to modify the intrusion in order to go unnoticed, hence its name.

If the classification function learned by the model is smooth with respect to the input samples attributes, e.g., Linear Regression, Neural Networks, Support Vector Machines (SVM), then this minimal change in the attributes can be obtained through a gradient, and this method is also known as a sensitive analysis.

B. SHAP

SHAP (SHapley Additive exPlanations) [9] is based on the cooperative game theory Shapley values [10]. These values average each player's contribution overall possible coalitions. In ML, players are replaced by attributes from the sample, whose contributions are added to the output of the algorithm, that is, its prediction. Therefore, such technique belongs to the class of additive feature attribution methods, as well as Local Interpretable Model-Agnostic Explanations (LIME) [11], DeepLift [12] and Layer-Wise Relevance Propagation [13]. What distinguishes the Shapley values method is that it always preserves three desirable properties: local accuracy, missingness, and consistency [9].

When the amount of attributes increases, calculating the exact Shapley values becomes challenging, due to an exponential growth in the total number of possible coalitions. To overcome this issue, SHAP approximates Shapley values through LIME's regression formulation. While in the latter the parameters choices are made heuristically, SHAP finds the unique solution to this regression that maintains these three properties, hence recovering the Shapley values. As one of the LIME's parameters is called weighting kernel, this method is also called Kernel SHAP. As LIME, Kernel SHAP is model agnostic. However, there are two other faster SHAP variations that take advantage of the model's particularities: Tree SHAP and Deep SHAP, specific for Decision Trees and Neural Networks, respectively.

III. Related Work

In this section, articles concerning post-processing techniques are presented. Although it is possible to find several articles related to this subject, very few deal with anomaly-based IDS, while most of them are limited in scope to signature-based IDS, even though their post-processing techniques use ML. This is evidence that anomaly-based IDS is still not widespread in real-world environments, remaining a research topic [6], meaning that signature-based IDS is mainly preferred.

Regarding signature-based IDS, most works deal with Snort IDS applied on the DARPA 1999 dataset [14]. A common approach is to sort alarms as a time sequence to perform time window analysis. Spathoulas and Katsikas [15] exemplify this procedure well while noticing that the signature distribution of alarms, in a given time window, is different depending on whether or not there is a true attack event at that time. It means that, in attack-free time windows, there is a specific and constant signature distribution, which distinguishes false alarm time windows. Such pattern distribution depends on the network configuration, and when an attack occurs, this distribution changes due to an increase in the frequency of alarms whose signatures are related to that attack. It was also noticed that the occurrence of many alarms with similarities in the source or destination IP addresses and, in close moments, is more related to true alarms. Thus, for each alarm, an analysis of its neighboring was carried out, within a time window, in order to verify the frequency of those with the same signature and source or destination IP addresses of the alarm in question. If this frequency was above a given threshold, the alarm was considered true. A reduction of approximately 75% in the number of false alarms has been reported [15].

Pitre et al. [16] published one of the few works that deal with false positive post-processing in anomaly-based IDS. Furthermore, the post-processing technique also uses ML, which makes it very similar to our method. The main difference is that they do not employ XAI in this task, but rather the same original attributes used by the IDS in the detection phase. The experiments were carried out in two small portions of the CSE-CIC-IDS2018 dataset, called stages 1 and 2. In stage 1, the IDS was trained and tested using Logistic Regression, yielding an accuracy of 0.932. The positive examples obtained in the test part of this stage were used for training a false positive filter, also by means of logistic regression. Finally, these two devices (IDS + filter) were used together in stage 2, resulting in an accuracy of 0.951. According to the authors, this improvement in accuracy, compared to stage 1, resulted in a drastic reduction in the false positives proportion, at the cost of a small increase in the false negatives proportion. Although confusion matrices for both stages have been made available, it is not possible to confirm such statements.

Finally, a metric called FOS (Feature Outlier Score), which uses SHAP values to obtain a better confidence level regarding anomaly IDS decisions, was defined by Kim et al. [17]. This evaluation represents how anomalous the SHAP values of a given sample are in relation to the SHAP values of similar examples in the training set. The more the sample SHAP values differ from the mean and standard deviation found in the training set, the lower the confidence in the decision. The difference in relation to our method is that we improve the confidence level through a second ML algorithm, and not by mean and standard deviation comparisons. To validate the FOS metric, the NSL-KDD dataset and an intrusion detector with the XGBoost algorithm were used. FOS allowed the identification of a 114% greater number of decision errors than without its use. However, the authors do not clarify how this comparison basis was evaluated.

IV. False Positive Detection using XAI

Our objective is to identify false positives more effectively, using XAI techniques. For this, we hypothesize that the aspects obtained through explainability are, in general, distinct between false and true positives. Furthermore, some explainable techniques are model-restricted, while others are model-agnostic, rendering the proposed method to behave accordingly.

An important question before detailing the method is to clarify its usefulness. An application example occurs when the security analyst needs to find false positives that are making an IDPS block normal traffic, causing annoyance to legitimate users. For security reasons, this traffic should be inspected before being released. In this case, it is more efficient to only inspect those related to alarms with a high probability of being false. If most of the alarms are true, analysts may fail to release all improperly blocked traffic, as they will waste too much time inspecting true alarms. The same reasoning can be done in the opposite way, when an analyst, for some reason, is inspecting lost true alarms among a large number of false ones.

As the proposed method is concerned only with post-processing positive outputs, i.e., intrusion alarms, it is assumed to have an already adjusted model (the anomaly-based IDS), along with a dataset used to train and evaluate such model. The method, then, identifies samples likely to be false positive more effectively compared to using, solely, the IDS confidence. Fig. 2 presents an overview of the proposed method, consisting of the following three steps.

A. Step 1 – Threshold Setup

Easy-to-classify samples usually are classified correctly in addition to receiving high confidence. This means that false positives (FP) with such confidence level are rare, which, in itself, already ensures that these are probably true detections. Thus, it is important to estimate the max false positive confidence value which can be used as a cut-off point, above which the non-existence of false positives is assumed. This estimation should be done in the dataset training portion and confirmed in the validation subset. Only positive samples with confidence under the obtained threshold are submitted to the false positive detection method. The dataset test portion satisfying it is called the analysis set and is used for evaluation, which is explained in Subsection IV-C.

The threshold value estimation is a manual process, which requires an examination of the positive examples, i.e., detected as malign, in the training set. This value may be the greatest possible confidence in false positives, however, there may be some rare false positives in the training set with a confidence level much higher than normal (e.g., an easily classified malign

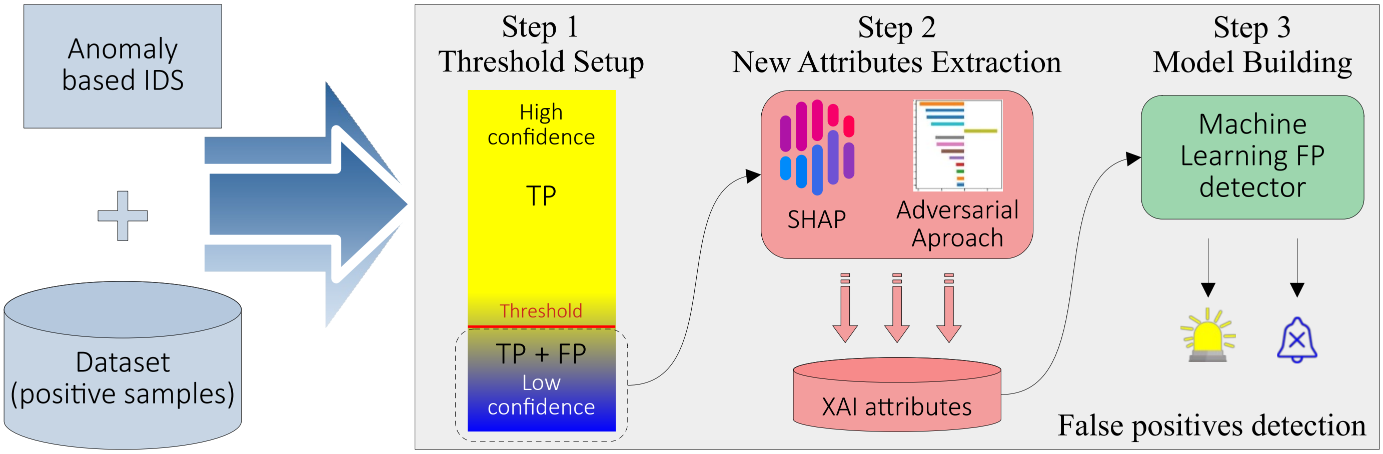

Fig. 2. Proposed method overview.

example, but mislabeled as benign). Thus, using this confidence as a threshold may result in the unnecessary inclusion of too many true positives (TP) in the analysis set, as it is quite likely that almost all false positives from the test set have confidence significantly below this value. A better estimate is a confidence value that covers a high percentage of false positives (e.g., 99%), rather than **all** of them.

The illustration of step 1 (Fig. 2) shows a quite linear increase in false positives amount, represented in a dark tone, as confidence decreases. In this case, the model's confidence is said to be calibrated, and the estimation of the false alarm probability can be done directly. However, the low confidence region may not be well behaved, requiring calibration techniques [18]. Despite this, it is expected that such confidence (calibrated or not) can be used together with XAI attributes to train a second algorithm in a true and false alarm inference.

B. Step 2 – New Attributes Extraction

This is the step where Adversarial Approach and SHAP are used. Given that the number of original attributes is n, and that XAI computes the relevance value for all of them, it follows that the number of new attributes extracted by each technique is also n. The reason to use more than one XAI technique is to extract attribute sets that capture different aspects of the same predictions. For this, it is desirable to avoid similar techniques (e.g., SHAP and LIME, which are both additive feature attribution methods). Therefore, Adversarial Approach and SHAP attributes are combined in the next step to produce better results than each technique individually.

The n XAI values, regardless of the technique employed, can be used, as such, to train the second algorithm, or they can go through some pre-processing phase first, in order to reduce the number of new attributes. This is more suitable when there are few examples for training, which may happen depending on several factors (e.g., skewed dataset, IDS performance, the threshold used in step 1, etc.). In order to accomplish that, Principal Components Analysis (PCA) may be used, although it may hinder the understandability of the XAI attributes, but, as the false alarm detection is done by a second algorithm, this is not a critical issue. Anyway, another form of reduction can also be used, taking advantage of the fact that XAI values can be considered as an n-dimensional vector. This is denoted as "XAI vector", which is, therefore, characterized by its direction and modulus.

TABLE I
SAMPLES USING REDUCTION THROUGH SIMILARITY COSINE.

Confidence	$\cos_s(\nabla_x, \bar{\nabla}_{TP})$	$\cos_s(\nabla_x, \bar{\nabla}_{FP})$	$\lVert\nabla_x\rVert$	Label
0,545810	0,875526	0,848827	4,93	TP
0,534495	0,899218	0,916288	14,65	TP
0,524484	0,890373	0,911622	26,34	FP
0,517797	0,876379	0,875529	19,15	TP
0,514481	0,932285	0,945562	24,79	FP
0,512833	0,900737	0,862286	7,19	TP
0,505035	0,868779	0,886791	11,24	FP
0,502664	0,828933	0,833023	0,11	FP

Although the modulus is represented by a scalar number, the same is not true for the direction in high dimensions. Thus, to represent that using fewer attributes, two reference directions are established in the higher dimension. The XAI vector of a given sample is then compared to these two reference directions using the similarity cosine, yielding two new attributes. A natural choice for these two references is the average of XAI vectors of false positives and true positives in the training set. The TABLE I presents some samples using this type of reduction, obtained from Adversarial Approach attributes. The attributes $\cos_s(\nabla_x, \bar{\nabla}_{TP})$ and $\cos_s(\nabla_x, \bar{\nabla}_{FP})$ represent the similarity cosine between the XAI vector of the sample x and the reference directions of TP and FP, respectively, being used the gradient operator due to the Adversary Approach technique.

Therefore, consider a sample predicted to be malign about which we want to infer if it is a true or false positive based on the direction of its XAI vector. Intuitively, a high similarity with the false positive reference direction and, at the same time, a low similarity with the true positive reference direction lead to the conclusion that this is a sample with a high probability of being a false positive. Fig. 3 illustrate this procedure with two dimensions (note that $\cos\beta > \cos\alpha$).

To summarize, the extracted attributes are:

- all the original XAI attributes, when there are enough

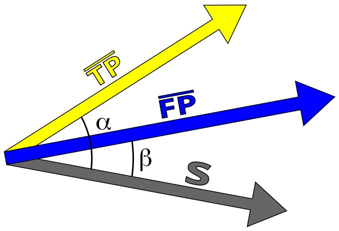

Fig. 3. Two-dimensional XAI vectors – $\overline{\text{TP}}$, $\overline{\text{FP}}$ (averaged in the training set) and a given positive sample s.

false and true positives examples below the threshold for training; or

- a reduced version of the XAI attributes, which can be done through PCA or similarity cosine comparison, whichever achieves more accuracy in the validation dataset. This way, the new attributes are the modulus and two cosine values: one between the sample XAI vector and the TP reference, and the other using FP as reference.

C. Step 3 – Model Building

One can ask about the real need for a second ML algorithm, as according to Fig. 3, a direct comparison is apparently enough to judge whether a given sample is true or false positive. However, this is an oversimplified approach, since it does not take into account the XAI vectors' direction variance. In other words, it is possible to have a significant amount of true positives, whose XAI vectors are more similar to the false positives reference direction, and vice versa. So, it is assumed that there is a non-trivial relationship between XAI attributes and the detection credibility, i.e., how likely it is false or true. Such relationship can be learned by a second algorithm, and, as the IDS confidence is also related to the detection credibility, this is a valuable attribute to be used together with the XAI ones.

Despite which kind of XAI attributes are used, special care must be taken to avoid data leakage. It means that the XAI attributes used to train, validate and test the false alarm detector must come exactly from samples that belong to the train, validation, and test portions of the original dataset, respectively. Furthermore, for each XAI technique, one model is built. Therefore, as we are using Adversarial Approach and SHAP, two models are used, each one trained on its respective XAI attributes.

Different ML models can be used (e.g., Naive Bayes, Decision Tree, Neural Network, etc.), and it is also important that such models provide the confidence that the false positive sample is classified. When this confidence is high, traffic related to it must be a priority for security analyst inspection. Thus, it is desirable that such confidence reflects the probability of the sample's tendency to be false. To evaluate this aspect, the analysis set (i.e., the positive samples from the original dataset test portion under the threshold IDS confidence) is used. Fig. 4 illustrates a desirable result, where a given analysis set, composed of samples labeled as FP and TP, are sorted by model confidence in two different cases, using or not using XAI attributes. The "Without XAI attributes" case represents two possible situations: either the analysis set was sorted according to the IDS confidence or according to a second algorithm that uses only IDS confidence as attributes. For the sake of comparison, whichever is the best is used.

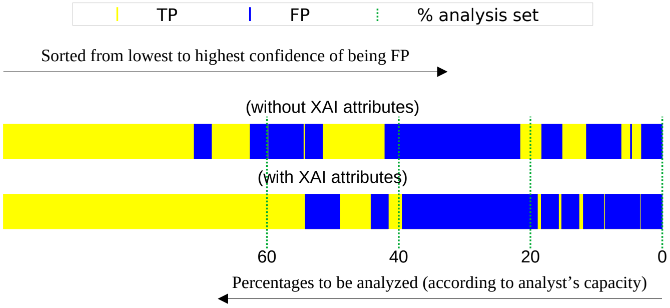

Fig. 4. Analysis set sorted by confidence.

It is possible to note that, at the bottom of Fig. 4, a greater amount of false positives fall on the right side of the confidence region when XAI attributes are used. Whenever the analyst's inspection capacity is not enough to cover the whole analysis set, it is more efficient to use this region to find and eliminate as many false positives as possible. If this capacity is for example 20% of the analysis set and as long as the analyst prioritizes samples with high confidence to be false, more false positives will be found and eliminated when XAI attributes are used.

V. Experiments and Results

An interesting dataset recommended by Ring et al. [19], that can be considered for testing the proposed method is the CIC-IDS2017, released by the Canadian Institute for Cybersecurity. This dataset is relatively new, contains a wide range of attack types, and is publicly available. However, it has several issues caused by bugs in the data flow attributes extractor, named CICFlowMeter. Two corrected versions of the dataset were independently proposed: one by Engelen et al. [20], and another by Rosay et al. [21]. The latter is used in the reported experiments, as it is the most recent. The extractor tool used to correct the dataset is open source and publicly available [22]. The "corrected" version of the CIC-IDS2017 dataset was renamed to LYCOS-IDS2017. It is worth mentioning that there is a newer dataset released by the same institute, named CSE-CIC-IDS2018. As its data flow attributes were generated by CICFlowMeter, several bugs still remain [20], [21]. Unfortunately, there is not a fully corrected version for this dataset yet. LYCOS-IDS2017's data flow is characterized by eighty-two attributes, which are continuous or discrete. The continuous attributes carry statistical flow data (e.g., the time between packets, flow duration, number of packets/bytes per second, mean length of packets, etc.). The discrete attributes represent counters that record the number of packets with some flags active (e.g., SYN, FIN, RST, ACK, PUSH, etc.) or the very information that defines the flow, which is the flow ID (a unique identifier for each sample), source and destination IP and ports. With the exception of the destination port, which is related to the application protocol (e.g., 80 for HTTP, 443

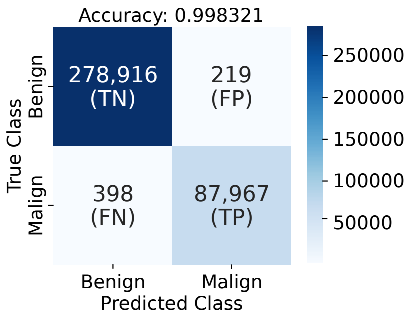

Fig. 5. Confusion matrix on the test set.

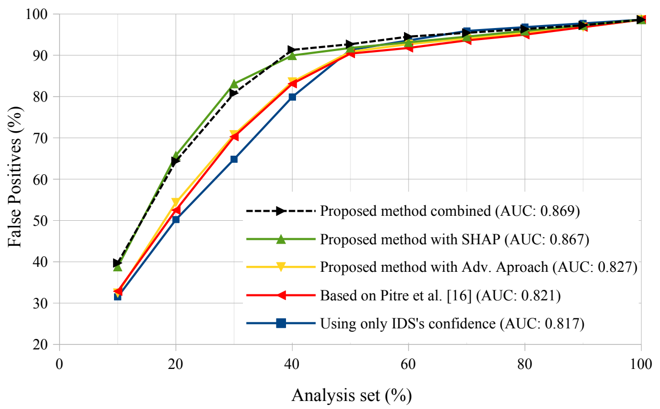

Fig. 6. False positives in percentages of the analysis set, where samples were ordered according to the method's confidence.

for HTTPS, 22 for ssh), all other flow definition attributes must not be directly used for intrusion detection, as they carry peculiarities specific from the network environment used to generate the dataset.

According to Fig. 2, it is necessary to have an IDS model already trained in addition to the dataset. For the sake of argument, a neural network model is chosen, provided there are several explainable techniques specific to it. Furthermore, the Adversarial Approach depends on the cost function gradient with respect to the original attributes, which can be computed efficiently through backpropagation. As neural networks do not process categorical attributes directly, destination ports, which may have too many different values, are transformed into numerical vectors using the *ip2vec* technique [23].

Fig. 5 shows the IDS performance on the testing portion of the dataset. For security reasons, reducing the 398 false negatives should be a priority. Although it can be done by an increase in the IDS sensitivity, which in turn causes more false positives, such tuning procedure is out of scope. Thus, our goal is just to identify most of these 219 FP as efficiently as possible, since they are mixed up among the 89,967 true ones, as the ground truth labels are unknown in a production environment.

An analysis of false positive samples on the training set reveals that 99% of them are predicted as malign with confidence below 0.9987629. Using this as a threshold on the test set results in the so-called analysis set, which consists of 2,231 TP + 216 FP. Then, XAI attributes from these 2,447 samples are extracted and fed to a second ML algorithm – the FP detector. The K-Nearest Neighbors (KNN) algorithm is the one used for simplicity since it has few hyperparameters to adjust. The model is built using XAI attributes extracted from the samples belonging to the training and validation sets. For Adversarial Approach, such attributes are reduced through similarity cosine comparison, while for SHAP the reduction is done through PCA (using the nine first components). With the exception of the "number of neighbors", selected on the validation process, all other hyperparameters are left at default values, and two KNN models are generated, one for each type of XAI attribute.

Fig. 6 shows this final result. The x-axis denotes percentages of the analysis set, whose total amount in absolute values is 2,447 samples. The y-axis contains percentages of the total amount of false positives on the test set, which is 219. The idea here is to obtain high percentages of false positives with the minimum percentage of the analysis set. This is accomplished by sorting samples according to the algorithm's confidence, and then, limiting them to a percentage suitable to analyst capacity. The dashed line represents a combination of the two KNN models, through the average of their confidences. Although this has the best overall result, characterized by the area under the curve (AUC), it fails to outperform the SHAP-related KNN model in the region of interest (left side of the graph). Moreover, we implement the main idea present in the method developed by Pitre et al. [16], which is to reuse the original attributes to train an FP filter, instead of the XAI attributes. This procedure generates the curve in red with left-triangle marks.

Although Fig. 6 presents the analysis set in percentages, it is the absolute values that will tell if this is under the analyst capacity. Consider, for example, the 20% portion of the analysis set with the highest confidence that the detection is false. This portion contains, in absolute values, 489 samples, which can be over the analyst's inspection capacity. If applicable, smaller percentages can be used, however, containing a smaller amount of FP.

The inspection procedure aims to maximize security, as it prevents the loss of TP. If the analyst capacity is too low, and if FP annoyance is severe, one can consider, after careful risk management, eliminating fractions of the analysis set with high confidence to be FP. In the 20% case, it contains 65% of all FP when the proposed method is used, which stands for 148 FP in absolute values. Even if the remaining 341 samples (489 - 148) are TP, they still represent a tiny fraction of all the 89,967 TP ones (0.38%). Therefore, the elimination of this 20% portion represents a high percentage of FP and, at the same time, a low percentage of TP. The same computation can be done using only the IDS confidence, where the percentage of FP spotted decreases to 50%, with a TP loss of 0.42%.

Finally, if the analyst's capacity allows the inspection of the whole analysis set (or the great majority of it), this method may become unnecessary. As can be seen in Fig. 6, from the analysis set's percentages greater than 50%, there is a convergence where the non-use of XAI (line with square dots) comes very close, and even surpasses XAI techniques individually.

VI. Conclusion

An anomaly-based IDS has the potential to detect new unknown attacks, but it is also more prone to generate false positives. Although some approaches have been proposed, this is still an obstacle to its mainstream adoption, which makes further research necessary. Unlike misuse-based IDS, whose signature in itself explains the reason for the (false) detection, it is not trivial to understand wrong detections from the IDS powered by complex ML algorithms. In this sense, XAI arises as a new possibility to handle false positives.

The use of XAI attributes, especially SHAP ones, makes it possible to obtain percentages of analysis sets with a higher density of false positives. The method acts as a way of triage, shortening the number of samples where the analysts search for false positives, thus enhancing their efficiency.

Even though the better performance was obtained compared to not using XAI attributes, it is not always possible to obtain percentages with a majority of false positives. This points to a need for improvement, which can be achieved in future works. One suggestion is to use other XAI techniques in order to reach better results with the confidence combination. Improvements also can be done on the second ML algorithm (the FP detector) choice, preferably those more suitable to unbalanced sets. There is also a need for a study related to the impact of feature selection before applying XAI techniques. SHAP, for example, assumes statistical independence of the attributes, which may not happen in the general case. Then, the minimization of correlation through feature selection can result in SHAP values with better quality, which in turn can improve the method.

References

[1] A. M. Riyad, M. Ahmed, and H. Almistarihi, “A quality framework to improve ids performance through alert post-processing,” *International Journal of Intelligent Engineering and Systems*, 2019.

[2] R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan, “Using neuro-fuzzy approach to reduce false positive alerts,” in *Fifth Annual Conference on Communication Networks and Services Research (CNSR ’07)*, pp. 345–349, 2007.

[3] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, “From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3369–3388, 2018.

[4] K. A. Scarfone and P. M. Mell, “Sp 800-94. guide to intrusion detection and prevention systems (idps),” tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, USA, 2007.

[5] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *2010 IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.

[6] E. K. Viegas, A. O. Santin, and L. S. Oliveira, “Toward a reliable anomaly-based intrusion detection in real-world environments,” *Computer Networks*, vol. 127, pp. 200–216, 2017.

[7] Internet Steering Committee project in Brazil, “Total internet data traffic in brazil,” 2022. https://ix.br/agregado/. Accessed on: Nov. 11, 2022.

[8] D. L. Marino, C. S. Wickramasinghe, and M. Manic, “An adversarial approach for explainable ai in intrusion detection systems,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3237–3243, 2018.

[9] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, pp. 4765–4774, Curran Associates, Inc., 2017.

[10] L. S. Shapley, *A Value for n-Person Games*, pp. 307–317. Princeton University Press, 1953.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should i trust you?": Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), p. 1135–1144, Association for Computing Machinery, 2016.

[12] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, “Not just a black box: Learning important features through propagating activation differences,” *ArXiv*, vol. abs/1605.01713, 2016.

[13] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLOS ONE*, vol. 10, pp. 1–46, 07 2015.

[14] MIT Lincoln Laboratory, “1999 darpa intrusion detection evaluation dataset,” 1999. https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset. Accessed on: Nov. 16, 2022.

[15] G. P. Spathoulas and S. K. Katsikas, “Reducing false positives in intrusion detection systems,” *Computers & Security*, vol. 29, no. 1, pp. 35–44, 2010.

[16] P. Pitre, A. Gandhi, V. Konde, R. Adhao, and V. Pachghare, “An intrusion detection system for zero-day attacks to reduce false positive rates,” in *2022 International Conference for Advancement in Technology (ICONAT)*, pp. 1–6, 2022.

[17] H. Kim, Y. Lee, E. Lee, and T. Lee, “Cost-effective valuable data detection based on the reliability of artificial intelligence,” *IEEE Access*, vol. 9, pp. 108959–108974, 2021.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” 2017.

[19] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” *Computers & Security*, vol. 86, pp. 147–167, 2019.

[20] G. Engelen, V. Rimmer, and W. Joosen, “Troubleshooting an intrusion detection dataset: the cicids2017 case study,” in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 7–12, 2021.

[21] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, “Network Intrusion Detection: A Comprehensive Analysis of CIC-IDS2017,” in *8th International Conference on Information Systems Security and Privacy*, pp. 25–36, SCITEPRESS - Science and Technology Publications, Feb. 2022.

[22] http://lycos-ids.univ-lemans.fr/

[23] M. Ring, A. Dallmann, D. Landes, and A. Hotho, “IP2Vec: Learning similarities between ip addresses,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 657–666, 2017.

Ricardo da Silveira Lopes Received the B.Sc. and M.Sc. degrees in 2007 and 2014, respectively, from the Military Institute of Engineering, Brazil, where he recently defended his D.Sc. thesis in Defense Engineering. His research interests include machine learning intrusion detection and explainable artificial intelligence.

Julio Cesar Duarte Graduated from the Military Institute of Engineering (1998), Master’s in Computer Science from the Pontifical Catholic University of Rio de Janeiro (2003), and Ph.D. in Computer Science from the Pontifical Catholic University of Rio de Janeiro (2009). He has multidisciplinary experience, working on the following topics: machine learning, artificial intelligence, and natural language processing.

Ronaldo Ribeiro Goldschmidt Received the B.Sc. in Mathematics from Fluminense Federal University (1990), the M.Sc. in Computer Systems from the Military Institute of Engineering (1992), and the D.Sc. in Electrical Engineering from the Pontifical Catholic University of Rio de Janeiro (2004). He currently works as an associate professor at the Military Institute of Engineering and his research interests include Data Science and Artificial Intelligence.