Achieving Explainability of Intrusion Detection System by Hybrid Oracle-Explainer Approach

Mateusz Szczepański
ITTI Sp. z o.o.
Poznań, Poland
UTP University of Science and Technology
Bydgoszcz, Poland

Michał Choraś
ITTI Sp. z o.o.
Poznań, Poland
UTP University of Science and Technology
Bydgoszcz, Poland

Marek Pawlicki
ITTI Sp. z o.o.
Poznań, Poland
UTP University of Science and Technology
Bydgoszcz, Poland

Rafał Kozik
ITTI Sp. z o.o.
Poznań, Poland
UTP University of Science and Technology
Bydgoszcz, Poland

***Abstract*—With the progressing development and ubiquitousness of Artificial intelligence (AI) observed in last decade, the need for creating methods explainable and/or interpretable for humans has become a pressing matter. The ability to understand how a system makes a decision is necessary to help develop trust, settle issues of fairness and perform debugging of a model. Although there are many different techniques allowing to get insights into models' inner workings, they often come with a trade off in the form of decreased accuracy. In the context of cybersecurity, where a single false negative can lead to a breach and compromise of the whole system, such a price is unacceptable. Therefore, there is a need for a solution which allows for maximum possible model performance, and in the same time delivers human understandable interpretations. Hybrid approaches to Explainable Artificial Intelligence (XAI) have the potential to achieve this goal. In this work we present the fundamental concepts and a prototype of a system using such an architecture.**

***Index Terms*—Explainability, Artificial Intelligence, Cybersecurity, Intrusion Detection, Neural Networks, Decision Trees**

I. Introduction and Rationale

A. Principles of Explainable Artificial Intelligence

The need for understanding the decision-making process of an Artificial intelligence system is not a truly new concept. In fact, it has been an active research topic since the emergence of the field [7]. Lately, with the quickly expanding market of AI solutions [2] both legislators and developers started to invest a lot in research of explainable, fair and trustworthy AI systems [5]. Thus, the term of Explainable Artificial Intelligence becomes natural part of the vocabulary of everyone interested in AI as the whole discipline sees resurgence [2].

But could one ask what exactly XAI is trying to achieve and how? The answer to this question is quite complex and there is already rich literature on this matter [1] to [7].

As the name suggests, XAI is concerned with developing methods and metrics that allow to generate an explanation of a 'black-box' AI system [2]. It must be noted though, that there is a lot of ambiguity and confusion surrounding the issue of what explanation in context of an AI system really is [2]. Also, some authors use the terms "*explainability*" and "*interpretability*" interchangeably [4] [2], while others keep them separated [7]. On top of that, there is even less certainty as to what constitutes a good explanation [1].

For the purposes of this work and for the sake of simplicity, the terms "*explainability*" and "*interpretability*" will be used interchangeably and are defined in accordance with [4], i.e as the ability of an agent to explain or to present its decision to a human user, in understandable terms.

As a fundamental guideline for an explanation quality, authors of this work have decided to use "*XAI Desiderata*" from [7]:

1) **Fidelity**: the explanation must be a reasonable representation of what the system actually does.
2) **Understandability**: Involves multiple usability factors including terminology, user competencies, levels of abstraction and interactivity.
3) **Sufficiency**: Should be able to explain function and terminology and be detailed enough to justify decision.
4) **Low Construction Overhead**: The explanation should not dominate the cost of designing AI.
5) **Efficiency**: The explanation system should not slow down the AI significantly.

B. Explainable Artificial Intelligence in the Context of Intrusion Detection Systems

There are a few additional concerns about XAI that must be stressed in the context of Intrusion Detection Systems (IDS) and Cybersecurity in general (which are our domains/application of interest in this work). During the design of an AI (or Machine Learning based detection) system for cybersecurity there are a lot of aspects that must be taken into consideration. A developer should know the answers to the "*Six Ws*" (Who? What? Where? When? Why? How?) [3] in order to deliver reliable, secure and useful solutions (e.g. explanation for alarms, detected anomalies and so called

IoC (Indicator of Compromise)) for all the stakeholders (e.g. security operators in SOCs (Security Operations Centres)).

As for XAI in cybersecurity context, we agree that **the use of interpretability should not, under any circumstances, lead to any decrease in model performance, i.e introduce vulnerability**. As stated in [6], there are possible dangers to transparency delivered by an incorrectly designed model.

For example, there is a difference between target audience and system beneficiaries [6], as it is possible that by gaining insights into model learning functions, we can gain the means to manipulate it. While in context of recommendation system it does not really matter, it can compromise the whole IDS system.

Besides, there is an issue of accuracy and/or efficiency ahead, that XAI methods can have [6]. Of course, with an IDS it is crucial to have as accurate a model as possible in order to deliver protection and threat mitigation. Therefore, XAI in the context of cybersecurity should be treated more as a means of reaching the end [6], **which is to foster trust and reduce risk of unwanted, unknown behaviour, rather than a goal on its own**. This idea is the foundation and motivation for the solution proposed in this work.

Therefore, **in the context of IDS and cybersecurity, there is a need for a system that fulfils the following conditions:**

- Delivers reliable predictions about potential threats,
- Delivers easy to understand explanations about its decisions,
- Keeps flexibility necessary to adapt program towards new challenges,
- Meets all of the above without detrimental effect on the performance.

C. *Our Contribution*

This paper offers a method that fulfils all the conditions laid out in the previous subsection.

At the same time, it also has the potential to realise most of the points of Desiderata described in I-A.

The proposed solution is called **Hybrid Oracle-Explainer Intrusion Detection System**. It uses two separate modules to deliver human interpretable answers about system decisions, at the same time allowing for highest possible accuracy.

This paper shows its fundamental assumptions, scheme and detailed description. To support all of that, an early prototype has been delivered and tested. We report very promising results proving the efficiency of the proposed solution.

After the in-depth introduction, context and rationale, the remainder of the paper is structured as follows: in Section II the related work is overviewed. Our contribution and the proposed solution is presented in details in Section III. Experimental setup, results and presentation of the implemented solution/prototype are given in Section IV. Conclusions are given thereafter.

II. Related Work

There are, as stated in [5], *"Different Facets of an Explanation"*. This means, that there are many ways to achieve interpretability on different levels, depending on such things as target recipients, information content or designed roles [5].

Therefore, in this section previous related works closely tied to the proposed solution i.e. either **surrogate type models** [11] or methods providing **local explanations**, as e.g. in [13] are presented.

The first term denotes the common approach of using a simple and intuitive decision algorithm to derive explanation for the decisions of a black-box model [10]. The second term means that the generated explanation concerns individual samples and shows what features were most important [5].

In [8] authors have proposed a model that became the direct inspiration for this work. Their **"Hybrid Data-Expert Explainable Style Classifier"** combines an opaque machine learning system (composed of a Random Forest or a Neural Network) with an interpretable module made of three fuzzy rule based classifiers and one decision tree. Then, it performs local explanation of the data point by taking the simplest interpretable classifier with matching prediction.

After that, it is either supported by one of the interpretable classifiers and the procedure goes as explained above or there is still no matching output and the simplest classifier with the most frequent output is being picked.

Their solution also provides a user with a textual explanation thanks to a Natural Language Generation (NLG) module. It is based upon the Linguistic Descriptions of Complex Phenomena (LDCP) architecture, having a granular linguistic model of phenomena (GLMP) in its core [8].

Work presented in [8] is closely tied to the content of position [12]. It presents an interesting approach to XAI based upon granular computing and fuzzy modelling, which allows for the creation of knowledge based models capable of modelling non-linear relations and at the same time allowing for interpretability by the usage of simplified natural language [12].

Another proposition of a surrogate-model-based system is described in [11]. The authors claim that their solution solves two important problems characteristic for this approach to XAI. Firstly, the surrogate models generally only *approximate* the decision making process of the opaque model [11]. This directly leads to the second problem, which is the inconsistency of the derived interpretation [11]. Both those issues can be solved by using a method called the "Interpretable Partial Substitute" by the authors. It relies on the simple thought, that if the interpretable model is capable of delivering a competent prediction, it should be used instead of the black-box model. In that case, the delivered explanations are fully representing the decision process. Under this framework (called the **"Hybrid Predictive Model"**) the authors have defined transparency as a percentage of how many samples are processed by the explainer [11].

Since shallow decision trees are inherently explainable [5], to encompass a complex dataset they usually, under normal circumstances, need to get deeper. This introduces higher complexity and therefore, makes them less comprehensible. In [10] a solution for this specific issue has been presented.

Authors propose to use **microagregation** to train many limited size explainers and therefore to achieve, as they say, a *"trade-off between comprehensibility and representativeness of the surrogate model on the one side and privacy of the subjects used for training the black-box model on the other side"* [10].

Then basing on the distance between a sample and the centroid of each cluster, appropriate tree is chosen as the local explanation. An example of the effect of using this method is presented in Fig. 1.

It should be noted that the library used to create this particular visualisation (and visualisations made by the prototype) is called **dtreeviz**. More about that project can be read under [15].

The main point of [13] is that the algorithm is to sample data points around the instance which is being explained, get their predictions using the classifier and finally weight them by proximity to the instance. Then by optimising a particular equation the explanation is found. The obtained explanation is faithful locally and model agnostic, which means that this explanation could be used with any black-box model because it makes no assumptions about classifiers function [13].

Finally, [9] presents a different approach to explainability. It is named Layer-wise Relevance Propagation (**LRP**) and is not based on a surrogate model. Instead, it *"leverages graph structure of deep neural network"* [9] to redistribute, neuron by neuron, its received input to the previous layer. The distribution is controlled by specified rules (equations). This whole method allows to understand the impact of each feature upon the chosen prediction and therefore allowing to perform a better feature selection.

III. The Proposed Model

A. Three Principles

The model proposed in further part of this section is based upon three important assumptions:

1) **In the context of IDS, the accuracy and reliability of a system are the top priority.**
2) **One phenomenon can have more than one explanation a.k.a the Rashomon effect [2].**
3) **The delivered explanation should be simple and help to develop trust [13].**

Because of those principles it was decided that a surrogate type system with local explanations may be the best solution. It has low overhead and no impact on accuracy, therefore it realises the principle number one. The Rashomon effects makes an approach valid. Though the derived explanation is not a faithful representation of the opaque classifier function in general, it is a potentially possible approximation of it. Therefore, it still provides useful insights into the data and helps to develop trust. Finally, because of its model agnostic and modular approach it allows to freely use a wide range of explanatory methods and as a consequence to tailor the explanation to any potential user.

In other words, this proposed method sacrifices, to some degree, the first point of *"XAI Desiderata"* presented in I-A to better realise the rest of them and to fully solve the problem described in I-B.

B. Model Overview

Fig. 2 reveals the general scheme of Hybrid Oracle-Explainer IDS solution.

The chosen sample is first being transformed to the form used by the opaque classifier during training. In this case the role of the black-box machine learning algorithm is fulfilled by a Feed Forward Artificial Neural Network (**ANN**).

Then, after obtaining a prediction, the sample in its original form, along with the Oracle output, is being passed to Explainer module. There it is compared with the saved centroid of each cluster made during the training process in order to find n closest (most similar) in terms of l^2 (Euclidean) norm.

Following that, starting with closest centroid, Decision Tree trained on the according cluster is being retrieved. If its prediction matches that of the Oracle, the search stops and the local explainer is returned. Otherwise the algorithm continues until it finds a supporting Tree or runs out of centroids. In that case the Tree linked to the closest centroid is returned.

This introduces a divergence in some cases and development of a strategy to minimise and properly handle this is a part of the future work. Next, the scheme of the decision tree is being drawn, resembling the one in Fig. 1 but with a highlighted path to prediction made by the chosen explainer.

The created visualisation is then presented to the security analyst, who uses it to understand why the chosen sample could be classified in such a way and/or to obtain a better understanding of the potential threat's characteristics.

C. Data Preparation

Because the training data for both main modules must be the same, some standard parts of machine learning pipeline must be carried beforehand.

It includes data cleaning, formatting, balancing samples and feature selection. Afterwards the dataset is split to a training set and a testing set, which are saved as files accessed by both modules.

D. Oracle Module

This part of the solution is relatively straightforward, being a standard machine learning pipeline oriented toward maximised precision. It means that most feature engineering methods and transformations can be used, along any classifier.

In the prototype shown in section IV an ANN with Principal Component Analysis (**PCA**) is being adopted as an example (since we have a running IDS/cybersecurity system based on ANN).

E. Explainer Module

It should be reminded that because of both the modular and the agnostic nature of the whole system, the presented implementation is not the only valid one. It can be, like Oracle, changed to another or even expanded upon with additional algorithms. Of course, as long as they are model agnostic and

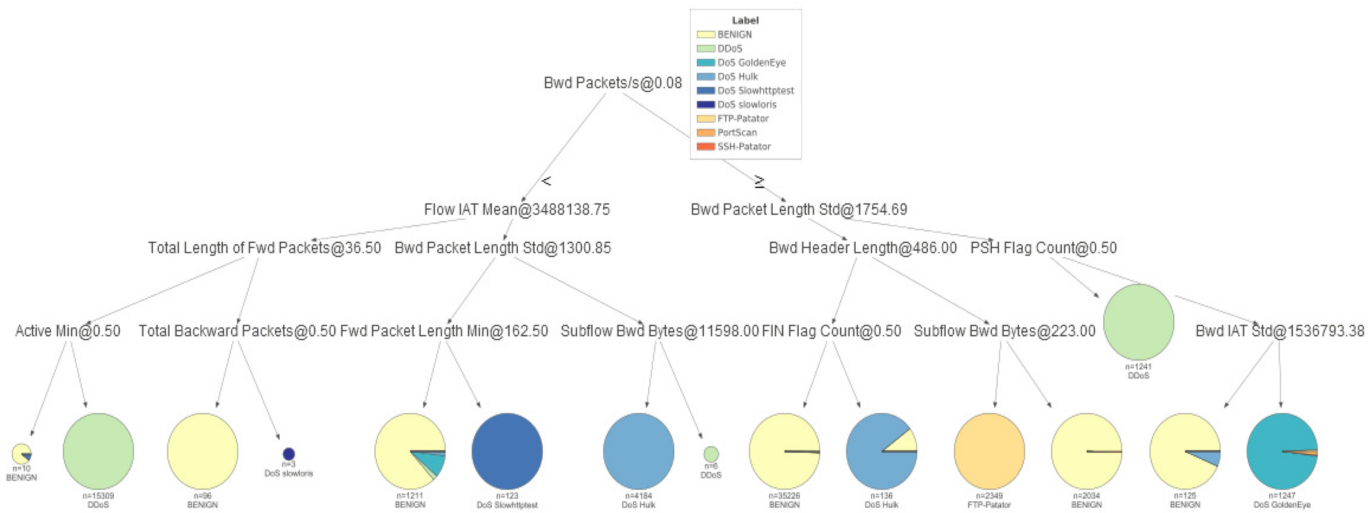

Fig. 1. An example of Decision Tree trained on CICIDS2017 dataset using microaggregation method.

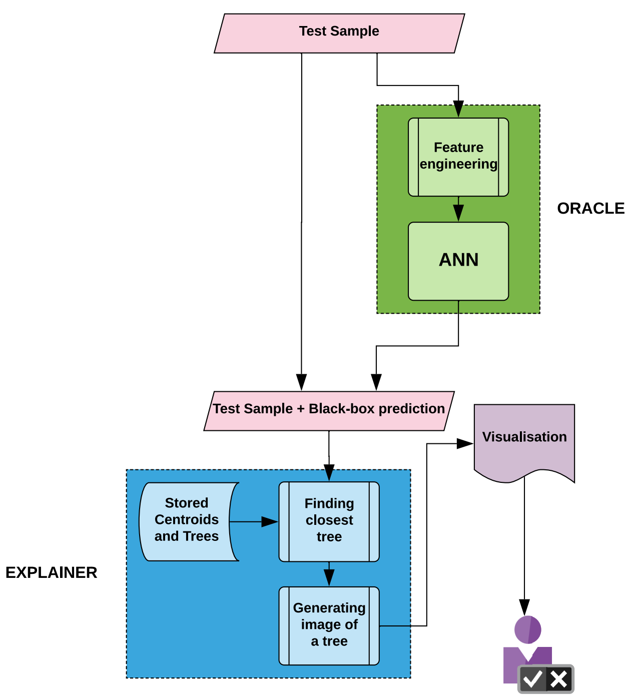

Fig. 2. Proposed system overview.

with a local scope. Experimentation with different explainers and their potential compositions is part of the future work.

The training procedure strictly follows the structure presented in [10].

For the readers' convenience it is presented here as an Algorithm 1. The number and the size of clusters is controlled by the parameter k, which indicates the level of representativity. The higher its value, the bigger the clusters, and therefore, there are fewer of them there.

To compute the clusters the method uses a microaggregation heuristic named the Mean Distance to Average Vector (**MDAV**).

Detailed description is available in [17], while the algorithm can be found in [10].

Algorithm 1 Generation of cluster-based explanations

procedure CLUSTER(Training set $\mathbf{X}$)
    Compute a clustering $C(X)$ for X based on all attributes except the class attribute
    for *each cluster* $C_i \in C(X)$ **do**
        Compute a representative, e.g. the centroid of average record $\tilde{c}_i$
    end for
    for *each cluster* $C_i \in C(X)$ **do**
        Train an interpretable model, such as a decision tree DT_i
    end for
end procedure

Prepared train set and test set (as described in III-C) are imported. No additional transformations are performed, so clusters are generated directly on training set. Having centroids, clusters and trees saved, procedure of finding explanation for chosen sample follows Algorithm 2 [10].

Next, as mentioned before, the samples with the retrieved tree are handled to the function of the library **dtreeviz** [15], which is responsible for generating the visualisation.

IV. Practical Implementation, Experiments and Results

A. Experimental Setup and Dataset

This section presents developed prototype and results of the system described in III.

System was trained on the CICIDS2017 dataset [16]. It was chosen because it is one of the most up-to-date datasets, containing a diverse range of attacks [14] [16] with 2 830 540 distinct samples [17].

TABLE I
TESTED VARIANTS OF EXPLAINERS

k	clusters	samples in each cluster	achieved accuracy	referring confusion matrix
0.2	5	253 202	95%	fig. 4
0.005	200	6 330	99%	fig. 5

Algorithm 2 Guided provision of explanation

Require: list of centroids C, list of interpretable models DT
  procedure GUIDED EXPLANATION($sample, prediction, n$)
    for *each centroid* $C_i \in C$ **do**
      calculate Euclidean distance $dist(sample, C_i)$ and add result to the dictionary $dict(C_i, dist(sample, C_i))$
    end for
    using dictionary sort C, where C_1 is the closest representative
    define iterator $i = 0$
    while $i < n$ **do**
      take interpretable model DT_i corresponding to the C_i
      if decision $(d = DT_i(sample)) == predictions$ **then**
        return d, C_i, DT_i
      else
        $i = i + 1$
      end if
    end while
    return d, C_1, DT_1
  end procedure

This includes DDos, XSS and SQL Injection attacks [16], to the total sum of 15 categories, each described with 83 features [18].

In current implementation, heavily underrepresented classes were removed, reducing their number to 9.

Finally, because samples with missing values were removed together with those belonging to disposed classes and those removed by the Random Undersampler from the training set, a total number of used distinct data points is equal to 1 971 937. The train-test is split 75% to 25% accordingly.

B. Implementation Details

The prototype was written in Python 3.7.4. For matrix/vector operations numpy 1.17.4 is being used, while for data import and basic prepossessing pandas in version 0.25.3 is applied.

The access to the popular machine learning algorithms and methods is covered by the scikit learn package 0.21.3. Simple plots are generated using pyplot (python version of matplotlib) in version 2.2.2. To create the trees dtreeviz 0.8.1 is also used [15].

Deep learning is realised on tensorflow 2.0.0 and keras 2.3.1. The code responsible for microaggregation and the explainer search is taken from a Jupyter notebook available for downloading from [10].

Finally graphical user interface (**GUI**) is developed with pyqt 5.12.1.

C. Oracle Quality and Implementation

The Oracle module used on the test dataset achieves currently 98%.

The confusion matrix is presented in Fig. 3.

The percentage of correctly classified samples is shown in the bottom-left top-right diagonal.

As for implementation, the data is first scaled to be in the value range from 0 to 1, and then is standardised to have mean 0 and the standard deviation equal to 1. It was required because PCA is applied to perform feature engineering [19].

Thanks to this step, 77 starting features are reduced to 35, which explains around 99% of variance, which increases accuracy and speeds up training. Of course, all transformations were carried separately for both the training and test sets.

The ANN is composed of 5 hidden layers, with 512, 512, 512, 512, 512 neurons accordingly. Each hidden layer has the dropout rate of 20% and uses the Rectifier Linear Function (ReLU). The architecture was empirically chosen after performing a number of separate tests.

The output layer uses the Softmax function instead. Loss is calculated with Categorical Cross-Entropy. ADAM fulfils the role of the optimiser. The Batch size is set to 10 000 and we employ early stopping to avoid overfitting. The ANN was made cost-sensitive and the weights of classes are calculated and used to counter the data imbalance problem.

D. Explainer Quality and Implementation

We have tested 2 explainers each made using different k values. They are all presented in table I.

There are few things that can be noticed. First off, accuracy alone is not the best indicator of quality in context of used dataset. Though difference in accuracy between explainer with $k = 0.2$ and $k = 0.005$ is only 4%, quality of the first one is drastically lower. Second, quality of the explainer relies heavily on the value of variable k. The more clusters there are, and therefore, amount of explainers, the better accuracy. With $k = 0.005$ it is even capable to surpass the Oracle on this specific dataset. Though it does not have to always be the case and may vary from dataset to dataset.

The implementation strictly adheres to the process presented in subsection III-E. Decision trees trained on those clusters have default configuration delivered by the scikit-learn package, with only maximal depth of a tree being limited to 4. The algorithm searches for matching explainer from the 3 closest centroids.

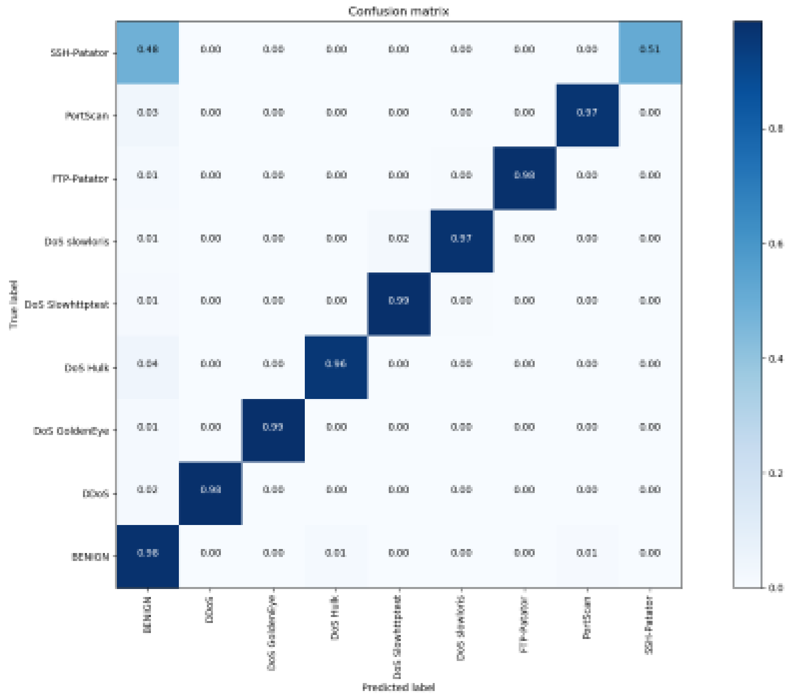

Fig. 3. Confusion Matrix of Oracle output.

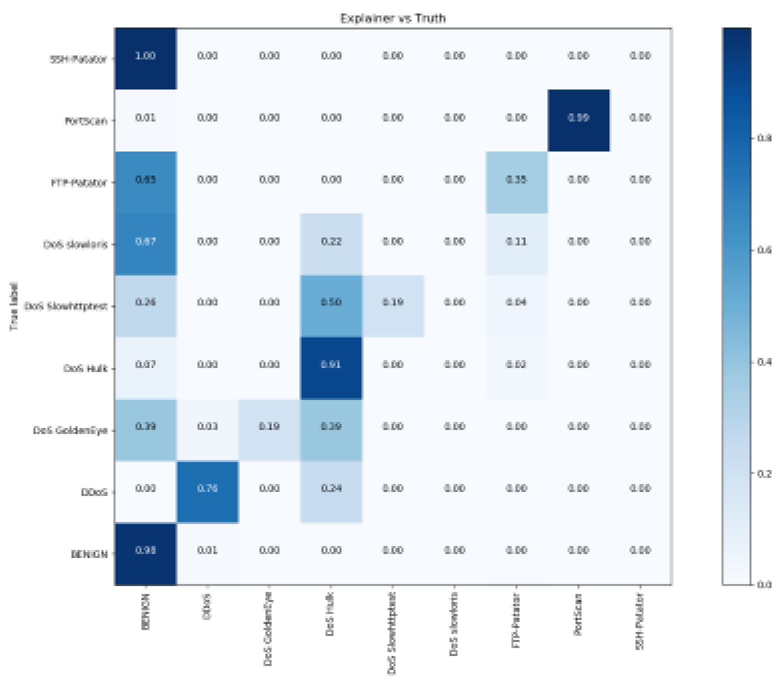

Fig. 4. Confusion Matrix of Explainer output with $k = 0.2$.

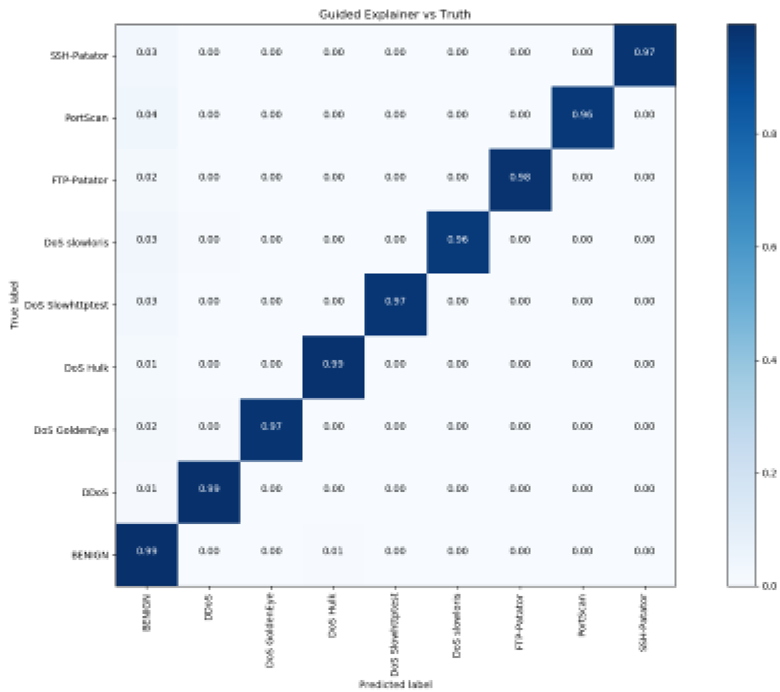

Fig. 5. Confusion Matrix of Explainer output with $k = 0.005$.

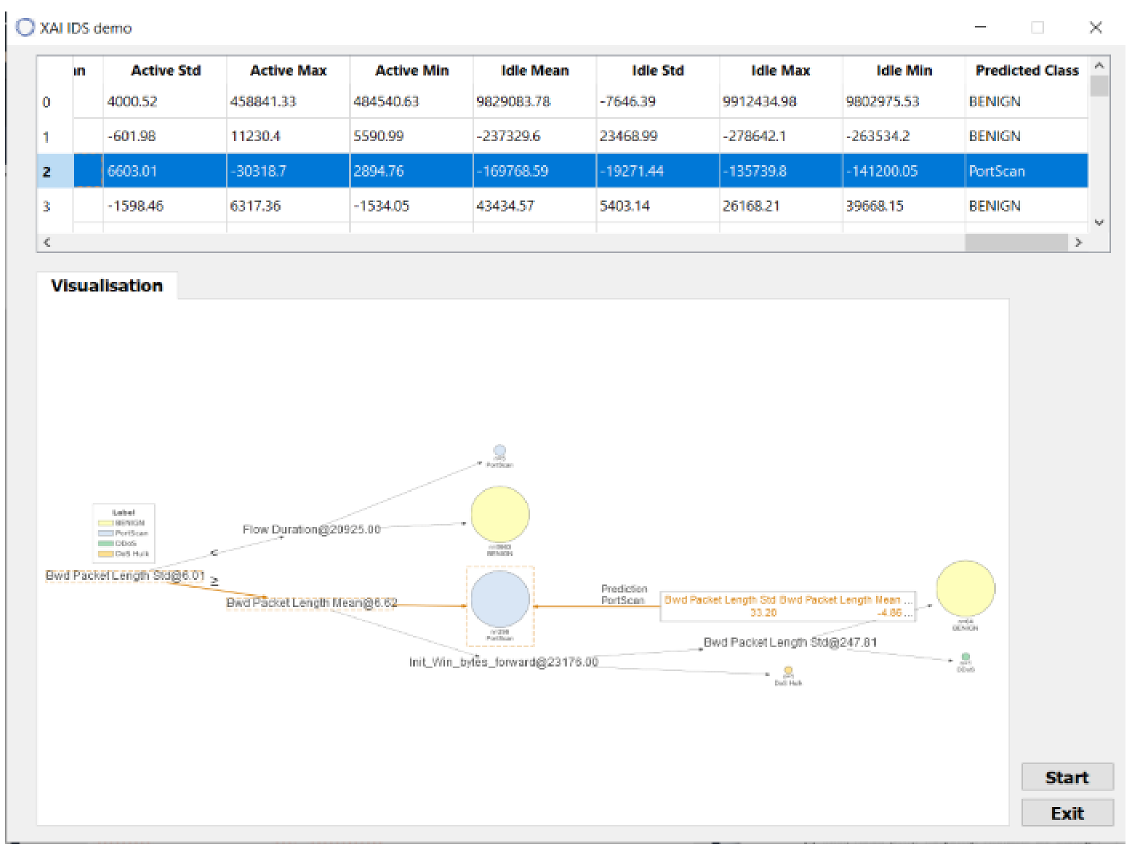

Fig. 6. Current GUI of the system.

E. Overview of the prototype application

Fig. 6 presents the current view/interface of the proposed system.

In the table at the top, the data points with the oracle predictions are displayed. After the Oracle classifies all samples, used transformations are reversed to closer correlate with the decision rules displayed by the trees.

Visualisation is provided for the sample chosen by the user.

After a double click on a row of the table, the chosen data point with prediction is being handled to the explainer module, where it searches for best tree in a way described in III-E. Library dtreeviz generates plot in Scalable Vector Graphics (**SVG**). After conversion to Portable Network Graphics (**PNG**) it is sent to be displayd at the bottom.

The produced graph shows the tree's structure, as the path leading to the prediction with important features highlighted. The circles are pie-charts showing how many samples of each class are within leaves. In this case, all leaves are pure, meaning every one of them contains samples belonging to the one category.

V. CONCLUSIONS

This paper presented the fundamental ideas behind the Hybrid Oracle-Explainer Intrusion Detection System along with the details on prototype implementation and achieved results.

It is a surrogate type approach to XAI motivated by properties such as low overhead, no detrimental effect on accuracy and high flexibility. We believe it is an interesting proposition for explainability in the context of cybersecurity applications.

Hereby we presented the practical implementation as a combination of ANN with Decision Trees trained using microaggregation.

Though it sacrifices fidelity, it fulfils the other requirements stated for an XAI system and delivers decent practical results.

Further exploration of this path, together with improvements to the current implementation, is the goal of the future work. For example, one of the things worth looking at is the solution proposed in [11].

The presented work is a part of SAFAIR (Secure And Fair AI systems for citizens) Programme of the H2020 project SPARTA, where explainability is a key research topics and therefore our solution will be further improved.

ACKNOWLEDGEMENT

This work is funded under the SPARTA project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

REFERENCES

[1] T. Miller, "Explanation in artificial intelligence: insights from the social sciences", Artificial Intelligence, vol 267 pp. 1-38, 2019.

[2] A. Richardson, A. Rosenfeld, "A survey of interpretability and explainability in human-agent systems," 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, Stockholm, Sweden, pp. 137-144, July 13-19 2018.

[3] L. Vigano, D. Magazzeni, "Explainable security," 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, Stockholm, Sweden, pp. 158-164, July 13-19 2018.

[4] D. V. Carvalho, E. M. Pereira, J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," Electronics, vol 8. 832, July 2019.

[5] W. Samek, KR. Müller, "Towards explainable artificial intelligence," in Samek W., Montavon G., Vedaldi A., Hansen L., Müller KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol 11700. Springer, Cham, pp. 5-22, September 2019.

[6] A. Weller, "Transparency: motivations and challenges," in Samek W., Montavon G., Vedaldi A., Hansen L., Müller KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol 11700. Springer, Cham, pp. 23-40, September 2019.

[7] L. K. Hansen, L. Rieger, "Interpretability in intelligent systems – a new concept?," in Samek W., Montavon G., Vedaldi A., Hansen L., Müller KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol 11700. Springer, Cham, pp. 41-50, September 2019.

[8] J. M. Alonso, A. RamosSoto, C. Castiello, C. Mencar, "Hybrid data-expert explainable beer style classifier," 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, Stockholm, Sweden, pp. 1-5, July 13-19 2018.

[9] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, KR Müller,"Layer-Wise relevance propagation: an overview," in Samek W., Montavon G., Vedaldi A., Hansen L., Müller KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol 11700. Springer, Cham, pp. 193-210 , September 2019.

[10] A. Blanco-Justicia, J. Domingo-Ferrer, "Machine learning explainability through comprehensible decision trees," in Holzinger A., Kieseberg P., Tjoa A., Weippl E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2019. Lecture Notes in Computer Science, vol 11713. Springer, Cham, August 2019.

[11] T. Wang, "Gaining free or low-cost transparency with Interpretable Partial Substitute", arXiv preprint, arXiv:1802.04346v2, May 2019.

[12] C. Mencar, J.M. Alonso, "Paving the way to explainable artificial intelligence with fuzzy modeling: tutorial," in Fullér R., Giove S., Masulli F. (eds) Fuzzy Logic and Applications. WILF 2018. Lecture Notes in Computer Science, vol 11291. Springer, Cham, February 2019.

[13] M. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?: explaining the predictions of any classifier," Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, San Diego, California, pp. 97-101, June 2016.

[14] A. Boukhamla, J. Coronel, "CICIDS2017 dataset: Performance Improvements and Validation as a Robust Intrusion Detection System Testbed," International Journal of Information and Computer Security, September 2018.

[15] T. Parr, P. Grover, "Explained.ai", https://explained.ai/decision-tree-viz/index.html. Last accessed 8 Jan 2020.

[16] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

[17] J. Domingo-Ferrer, V. Torra, "Ordinal, continuous and heterogeneous k-anonymity through microaggregation," Data Mining and Knowledge Discovery, 11(2), pp.195-212, August 2005.

[18] R. Panigrahi, S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," International Journal of Engineering Technology, [S.l.], vol 7. n. 3.24, pp. 479-482, August 2018.

[19] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," Electronics, vol 8. 322, March 2019.