Towards Explainable Deep Domain Adaptation

Szymon Bobek[1][0000−0002−6350−8405], Sławomir Nowaczyk[2][0000−0002−7796−5201], Sepideh Pashami[2][0000−0003−3272−4145], Zahra Taghiyarrenani[2][0000−0002−1759−8593], and Grzegorz J. Nalepa[1][0000−0002−8182−4225]

[1] Jagiellonian Human-Centered Artificial Intelligence Laboratory (JAHCAI), Mark Kac Center for Complex Systems Research, and Institute of Applied Computer Science, Jagiellonian University, 31-007 Kraków, Poland
{szymon.bobek,grzegorz.j.nalepa}@uj.edu.pl

[2] Center for Applied Intelligent Systems Research, Halmstad University, Sweden
{slawomir.nowaczyk,sepideh.pashami,zahra.taghiyarrenani}@hh.se

Abstract. In many practical applications data used for training a machine learning model and the deployment data does not always preserve the same distribution. Transfer learning and, in particular, domain adaptation allows to overcome this issue, by adapting the source model to a new target data distribution and therefore generalizing the knowledge from source to target domain. In this work, we present a method that makes the adaptation process more transparent by providing two complementary explanation mechanisms. The first mechanism explains how the source and target distributions are aligned in the latent space of the domain adaptation model. The second mechanism provides descriptive explanations on how the decision boundary changes in the adapted model with respect to the source model. Along with a description of a method, we also provide initial results obtained on publicly available, real-life dataset.

Keywords: Explainable AI (XAI) · Domain adaptation · artificial intelligence.

1 Introduction

Domain adaptation (DA) aligns different but related domains to leverage all the available knowledge together. Typically, a source domain with an abundance of training data is used to enable models to generalize effectively in another domain, called a target domain [5]. This capability makes domain adaptation a suitable approach for overcoming the challenges of limited labeled data in many practical applications, and it has demonstrated significant success in addressing real-world problems [13]. The main challenge of DA is how to map both input data distributions, given the data shift between the source and target domains, into a common latent space. Deep domain adaptation [14], which covers a lot of recent work, aims at learning this transferable representation using deep learning. Similar to any deep learning model [1], deep domain adaptation techniques are considered black-box models, and understanding the adaptation process between source and target domains is challenging. In particular, explaining the adaptation process is an important step in many practical settings for ensuring trust and acceptance from the end user.

The success of domain adaptation depends on the difficulty of transferring knowledge from the source domain to a "different but related" target domain [12]. Neither of these two terms ("different" and "related") is generally well-defined; those concepts highly depend on the task at hand and are often impossible for domain experts, not well-versed in data science, to grasp fully. Surprisingly, these concepts have received limited attention in the existing literature [15]. In particular, there is a lack of discussions on these aspects from an explainability perspective – how to convey to humans key knowledge about the adaptation performed by a model. We claim that explainability can help in describing, in a meaningful way, the domains' variations, discrepancies, and similarities.

When performing DA, one of the most common techniques is to learn a shared feature representation that aligns both domains with each other. The final prediction model operates within this shared feature space. Understanding how this shared feature space is constructed is crucial to comprehend the differences between the domains and how the DA model addresses these differences. It is particularly important to focus only on regions of feature space that affect decision boundary in the adapted model, i.e., regions from the target domain that are incorrectly classified by the source model; discrepancies that are irrelevant to the decision-making should be hidden not to distract the expert. The second important aspect of domain adaptation is how the decision boundary differs between the original model (trained only using the source domain) and the adapted model (trained using both domains). Given that the additional data is likely to affect the decision-making, possibly by identifying new discriminative patterns, a full-picture explanation needs also to highlight those changes.

This paper proposes a model-agnostic explanation, which allows us to analyze the adapted model from two complementary perspectives explained above. First, it provides an explanation of the feature extraction process by generating an approximation of the transform that the DA performs to align two domains. Second, it gives insight into the changes in a decision boundary in the adapted model, compared to the source model. This work is the first attempt at explaining the meta-level of the domain adaptation mechanism. The expert can directly use this knowledge to gain more understanding of the technical aspect of adaptation (model debugging) but also to obtain information about semantic relations between domains that the adaptable mechanism encoded and the explainable method revealed (knowledge discovery). For example, if rich knowledge about the source domain exists, but little is known about the target domain, such a descriptive summary linking source and target domains through the lenses of an adapted model clearly brings new insights and opportunities for data analysis.

The rest of the paper is organized as follows. In Section 2 we describe current trends at the intersection of explainable AI and DA. In Section 3 we introduce the theoretical background of our method and demonstrate it on a simple 2D use-case. The more advanced case-study that involves explanation of a domain adaptation in the area of network intrusion detection is presented in Section 4. Finally, we summarize our work in Section 5 and provide future possible extensions and application of our method.

2 Related Work

Explainable Domain Adaptation has been approached from different perspectives in the literature. In this section, we focus on the use of explainable methods as a tool for performing DA. Typically, some explanation is provided for every domain, explaining the model or data, and then the explanations are used to adapt the domains. In this regard, the authors of [11] propose an explanation-guided training strategy, specifically focusing on the Cross-domain Few-shot learning mechanism. To achieve this, they utilize LRP (Layerwise relevance propagation) to construct a weight vector that indicates the relative importance of features in the prediction process and feeds it into the classifier. By downscaling the weights of features with lower LRP values, they ensure that the classifier pays more attention to the features deemed more important. The authors of [18] also employ the concept of explainability, specifically saliency maps, as a tool for conducting domain adaptation. The proposed method in [18] utilizes saliency maps to create a strong influence on classifier prediction, forcing it to prioritize attention to specific regions. As a result of being forced to focus primarily on these salient regions, the model will focus more on features that are domain-invariant while neglecting features that are domain-specific (such as background information). The emphasis on domain-invariant features facilitates the mapping of the source and target domains so that source domain information can be used to make accurate predictions in the target domain. Such an approach focuses primarily on explaining the domains rather than explaining how they are adapted.

Designing DA methods that are inherently explainable is another direction, although very few papers have been published in this area. The proposed method in [7] explains the prediction of the output of the test samples based on the prototypes of source and target domains. The method focuses on aligning the prototypes between the domains, ensuring that prototypes belonging to the same class are closer to each other and farther from prototypes of other classes. Furthermore, a prototype projection layer is introduced to map the prototype vectors into visually interpretable images, enhancing human understanding of the model's inner workings. Such methods provide explanations for final predictions. However, they adapt source and target domains according to their predefined rules. Thus, the explanations provided by such approaches are aligned with what is injected into the DA model for adaptation.

In the paper [16], the authors present a method to explain DA by highlighting the impact of source samples on predicting a target sample. To achieve this, they introduce an interpretable deep classifier and integrate it into the framework of Domain-Adversarial Neural Networks (DANN). The classifier works by measuring the distance between source and target samples resulting in interpretability. In summary, the proposed method provides insights into the role of source samples in the DA process.

All of the aforementioned works focus either on explaining the final adapted model or using explanations in the process of adaptation to enhance it. In our work, we focus on explaining the adaptation itself; hence we provide explanations in the form of transform vectors that approximate the adaptation process and describe shifts in decision boundaries between source and adapted models. In the next section, we describe how we achieve that in more detail.

3 Method

In our work, we focus on descriptive explanation mechanisms that capture two aspects of domain adaptation: domain alignment and decision boundary update. In Figure 1, we present how our explanation modules fit into the architecture of the most common domain adaptation model. The first module is responsible for explaining the latent space adaptation mechanism by providing a transform (or a set of transforms) that the feature extractor performs on the original data to align source and target domains in the latent space. The second module is responsible for explaining how the decision boundary changes in the adapted model in comparison to the source model.

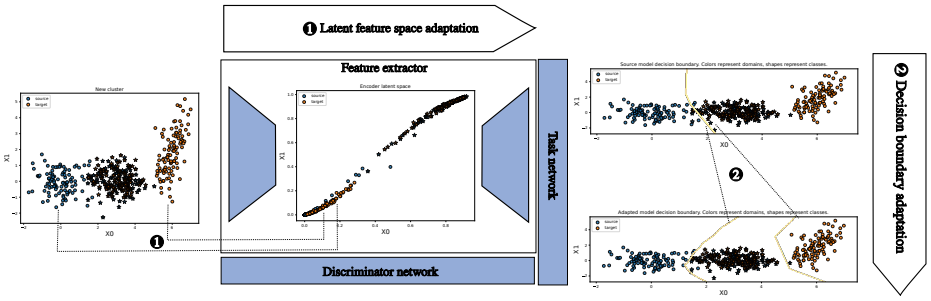

Fig. 1. Explainability modules in the architecture of the domain-adapted model. The first module describes the transform performed by the feature extractor in order to align domains in latent space. The section module describes the changes in the decision boundary.

In both cases, the explanation Φ for an adapted model M_a with respect to the source model M_s is defined as a vector $\Phi^{M_s \to M_a} = (v_1, v_2, \ldots, v_n)$, where $v_i \in \mathbb{R}$ is a value associated with the feature i. In the case of the feature space adaptation, the $(v_1, v_2, \ldots, v_n)$ represents a transform vector that aligns the source and target domains in the latent space, while in case of the decision boundary adaptation, the vector represents the change in the separation hyperplanes in the source and the adapted model. In the next paragraphs, we describe how the explanations for these two modules are constructed.

3.1 Explanation of a feature space adaptation

One of the tasks of adaptable models such as DANN is the discovery of latent space representation of the input data that makes the source and target domains indistinguishable. This stage is often referred to as feature extraction because the latent space becomes the new feature space for both source and target domains. In our work, our aim is to explain what is an interpretation of such an alignment in the input space, i.e., what transform $(v_1, v_2, \ldots, v_n)$ of the target domain input space makes it indistinguishable from the source input space.

To solve this problem, we first select unaligned samples from source and target domains. We are interested in samples $X_e = \{x_i \in X_s \cup X_t : M_s(x_1) \neq M_a(x_i)\}$,

where X_s are source domain samples and X_t are target domain samples. Next, we build a classifier C that is trained to distinguish samples from X_e as either source domain samples or target domain samples. In this step, we do not use latent space representation of the samples, where they are indistinguishable, but operate on original input space.

Based on the classifier C, we define counterfactual sub-spaces for each sample from the target domain. The counterfactual subspace for a sample $x_i : C(x_i) = l$ is a set of samples $X^i_{cf} \subseteq X_e$ such that for the majority of samples from X^i_{cf} the $C(x_j) \neq C(x_i)$. The counterfactual sub-spaces are constructed with LUX explainer [2], which uses a decision tree to partition input space into homogeneous areas with respect to class label and returns counterfactual sub-spaces by traversing the decision tree in a search for partitions that contain a majority of samples from opposite class, i.e., $C(x_j) \neq C(x_i)$. It is worth noting that we do not define a counterfactual as the nearest sample from the input space with the opposite class label, as this approach is prone to noise. Instead, we are interested in finding all of the groups of counterfactual samples, which form more stable and representative counterfactuals than single nearest instances. This is also motivated by the fact that later in our method we move the analysis to the latent space, where similarities from input space do not have to be preserved. This step can also be used to obtain an explanation of the differences between the domains in the input space, as shown in Figure 2.

In the subsequent step, we transform each sample x_i and its associated counterfactual subspace X^i_{cf} into latent representations, becoming $\overline{x}_i$ and $\overline{X}^i_{cf}$ respectively. Then we select the nearest neighbor $\overline{x}_{cf} \in \overline{X}^i_{cf}$ of $\overline{x}_i$, which now forms a pair $(\overline{x}_i, \overline{x}_{cf})$ and so can be easily traced back to its original input space representation (x_i, x_{cf}). Finally, for each sample, we calculate the transform vector $(v_1, v_2, \ldots, v_n)$ as a difference between its real representation and the real representation of a nearest latent counterfactual: $x_i - x_{cf}$. Due to the fact that the transform performed in the adaptation model might not be linear, we cluster detected instance-based transforms according to cosine similarity as depicted in Figure 2.

In the case of low-dimensional space, the visualization as presented in Figure 2 is satisfactory as a way of presenting the explanation. In higher-dimensional spaces, we adapted the SHAP waterfall plots to depict the transformations. In Figure 3, the example waterfall plot for the transform of one of the clusters from Figure 2 is presented.

It is worth noting that the waterfall plot from Figure 3 transfers the information about the counterfactual explanations, as it is built based on the counterfactual subspaces X_{cf} discussed earlier in this section. In other words, the transform depicted in the plot is a generalized version of a counterfactual explanation for a whole cluster of data.

The transform clusters defined in this stage are used as input for the phase of explaining the decision boundary adaptation.

3.2 Explanation of a decision boundary adaptation

In the explanation of a decision boundary adaptation, we focus on describing how the decision boundary changed between the source model M_s and the adapted model M_a. The decision boundaries of M_s and M_a are assumed to be non-linear and possibly

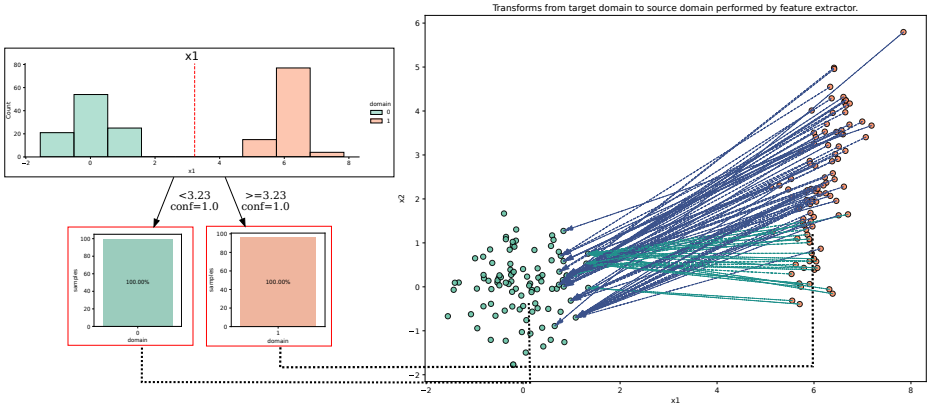

Fig. 2. Visualization of a transform-based explanation for feature space adaptation. On the left-hand side, there is a decision tree generated by the LUX algorithm that divides the space into two homogeneous subspaces and helps in defining counterfactuals. On the right-hand side, there are transforms in the input space, conditioned on their representation in latent space.

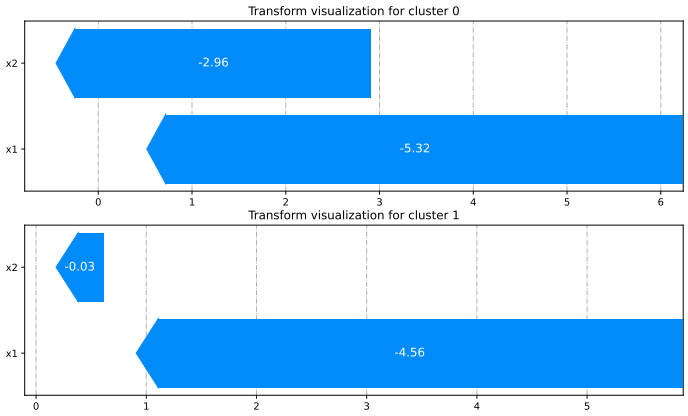

Fig. 3. Visualization of a transform-based explanation for feature space adaptation with SHAP-like waterfall plot. Three important pieces of information can be read from the plot: 1) how many transform clusters are there in the data 2) what are cluster centroids defined by the origins of bars in the plot, and 3) the transform itself depicted as colored bars which define a shift in dimension represented by a particular feature.

complex; therefore, we approximate it locally with a linear interpretable model such as LIME. To achieve that, we use transform clusters defined in previous steps as initial samples for which we calculate two approximations of decision boundaries with a local linear surrogate model: one for M_s and one for M_a. As a result, we obtain two vectors of coefficients $(\theta_1^s, \theta_2^s, \ldots, \theta_n^s)$ and $(\theta_1^a, \theta_2^a, \ldots, \theta_n^a)$ which define the hyperplanes perpendicular to them. Such a situation for the toy example used in this section is presented in Figure 4. The yellow lines represent decision boundaries for the source model (solid line) and the adapted model (dashed line). The other straight lines represent the linear approximations of the decision boundaries for particular transform cluster points (different colors for different transform clusters). The arrows are associated with each transform clusters and are vectors perpendicular to the decision boundary, which locally approximate the M_s and one for M_a boundaries. From the visualization of the vectors, one can immediately notice that for both of the transform clusters, the decision boundary has flipped by more than 180 degrees.

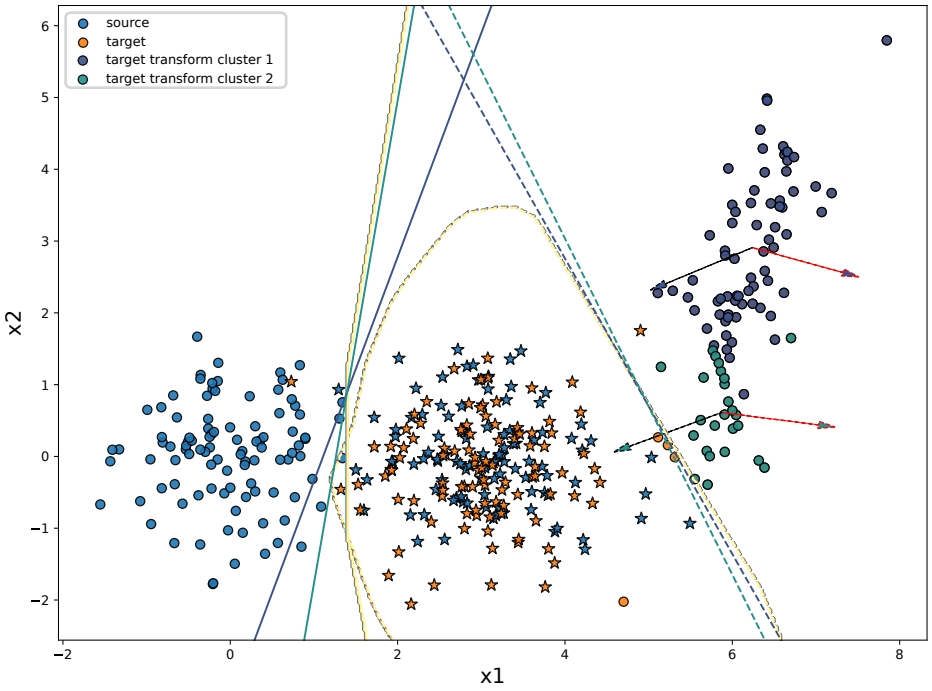

Fig. 4. Linear approximations of decision boundaries for M_s (solid line) and M_a (dashed lines) and corresponding vectors that define these hyperplanes. Arrows represent the contribution of features according to LIME for clusters of transforms. Dotted lines – adapted model, solid lines – source model.

Although such a visualization is straightforward for simple cases, it becomes infeasible for the multidimensional case. In such a situation, we adapted the summary plot

from SHAP plots that shows the variant and invariant features for the domain adaptation procedure, as shown in Figure 5. The smaller the value associated with the feature, the smaller the change of the decision boundary related to this feature in the source and adapted models. For instance, in Figure 5 one can observe that for both transform clusters the `x2` attribute is considered invariant, while `x1` variant feature. This means that the biggest change in the decision boundary in the adapted model was made along the feature `x1`, which is also visible in Figure 4.

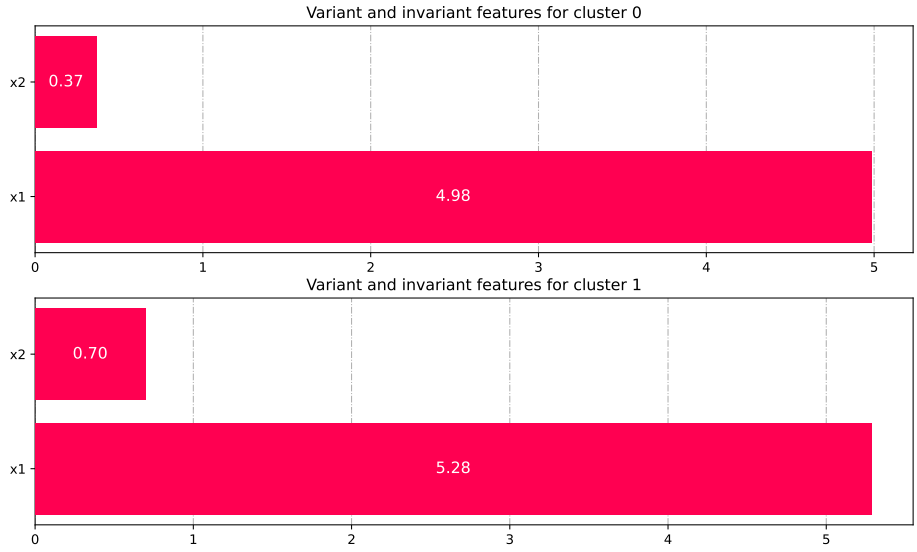

Fig. 5. Summary plot for decision boundary comparison presented in Figure 4. The low values are associated with invariant features, while high values denote the variant features.

In the following section, we demonstrate the method on a real-life, multidimensional dataset.

4 Case study

For the evaluation study, we selected real-life multidimensional datasets, CICIDS17 [9] and InSDN [4], from the computer network security area. These two datasets are collected using the same network monitoring tools resulting in homogeneous feature sets [17]. Despite sharing the same feature set, these two datasets differ greatly from one another due to two factors: they are collected from two different networks, and the existing attacks in each dataset are different. This characteristic of these datasets makes them suitable for performing DA. As a source domain dataset, we used the CICDS17 dataset, and as a target domain dataset, we used the InSDN dataset.

Datasets include samples for different attacks (5 attacks) that can be used for multilabel network intrusion classification. However, in order to have the same label set in different domains (while keeping the divergence between them), we changed it to binary classification. We altered the labels of all of the different types of attacks to be *abnormal* state focusing on building a classifier that distinguishes this state from *normal* state.

First, we trained a model on the source dataset (CICIDS17) and evaluated it on both source and target dataset (InSDN). We obtained F1 scores of 0.96 (macro average 0.95) and 0.22 (macro average 0.25), respectively, which indicated that in order to achieve adequate performance on the target dataset, an adaptation to a new domain is required. We used the CCSA algorithm [8] to perform the adaptation and achieved F1 scores for the source and target domain of an adapted model equal to 0.96 (macro average 0.96) and 0.99 (macro average 0.99), respectively, proving that the adaptation was performed correctly.

Next, we applied our method to explain the adaptation process. We distinguished the sets of samples X_e from the target domain that are misclassified by the source model. Then we created the interpretable classifier C (the left plot in Figure 6) that distinguishes between the domains in the input space. This classifier was later used to generate the counterfactual sub-spaces X_{cf}^i based on which we obtained explanations in the form of cluster transforms. The generated cluster of transforms are presented in the right-hand graph in Figure 6.

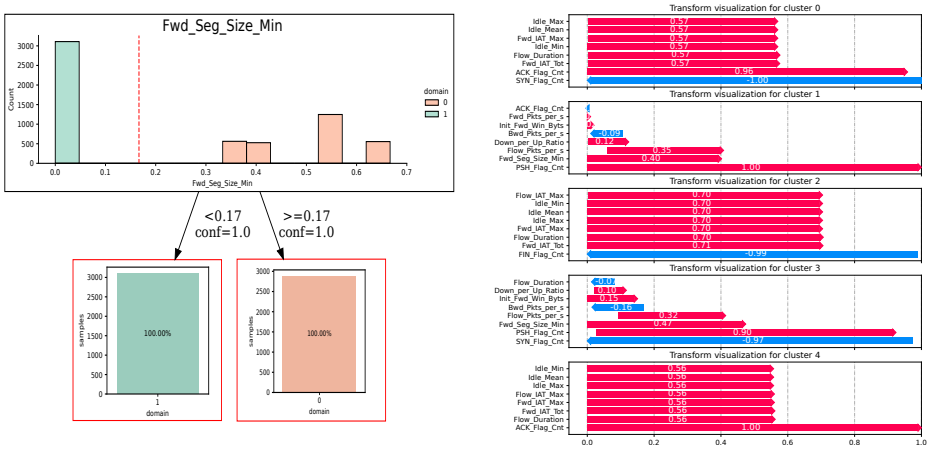

Fig. 6. Explainable decision tree (on the left) describing the most discriminative feature that allows for distinguishing between source and target domains. Transform visualization for the top five most important features (on the right) depicting how the adaptation aligns domains in a feature space.

From these results, several conclusions can be drawn. First, the most discriminative feature that makes the two domains different is the `Fwd_Seg_Size_Min` feature. In

the target domain, the value of this feature is much lower than in the source domain. Our method detected five distinct cluster transforms (the optimal number of clusters was obtained by the analysis of silhouette score and K-means clustering). These clusters depicted in the waterfall plot in Figure 6 show how samples from the target domain are transformed to the source domain in the input space.

It can be seen that `Fwd_Seg_Size_Min`, the most discriminative feature, is not present as the most important feature in the transforms-based explanations. One can conclude that the domain alignment that is performed in the latent space is a much more complex operation, and it cannot be derived only by looking at the differences between the samples in the input space. Furthermore, the cluster transforms reveal additional information on how the alignment is done with respect to the semantics of the samples. For instance, the datasets used by us in this case study were originally multi-labeled datasets, which we converted to a binary classification problem. Each of the labels in the original dataset corresponded to an attack performed on the network infrastructure, which we interpreted as anomalous behavior.

We traced back which classes from the original dataset were mapped with each other by the domain adaptation mechanism; it appeared that the only class from the target domain that was incorrectly classified by the source model was 'DDoS' attack. Additionally, by analyzing instances linked by the transforms obtained from our method, we discovered that the 'DDoS' attack from the target domain (which was missing in the source domain) was aligned with the 'Patator' attack in the source domain (not present in target domain). Such information can be used by the expert to judge whether the alignment performed by the domain adaptation mechanism is consistent with background knowledge. In this case, one can argue that this alignment does make sense, as the 'Patator' attack, which is a brute-force password cracking mechanism, may resemble DDoS or DoS attacks. The transform clusters give more details on how such alignment was performed. For example, when analyzing the cluster transform 1 in Figure 6, we can observe that there exist several features for which the transform was minimal, such as `Init_Fwd_Win_Byts` or `ACK_Flag_Cnt`. This means that samples from source and target domains had similar values for these parameters. According to the analysis of the source domain dataset [10], these features happen to be the most important features for identifying 'Patator' attacks. Thanks to transform clusters and available knowledge about the source domain, we can derive a conclusion that the type of attack that is associated with cluster transform 1 resembles 'Patator' attacks from the source domain, and the alignment done by the adaptation mechanism is semantically sound.

In Figure 7, linear decision boundaries approximations for source and adapted model for cluster transform 1 was presented. It can be seen that the biggest difference between decision boundaries (right plot) is observed in features related to the number of packages sent over the network per second. This is again consistent with the background knowledge about the difference between 'Patator' and 'DDoS' attacks. The former is performed from a single computer, and the latter is a distributed attack that results in larger packet intensity.

Similarly to transform cluster analysis, the feature which is the most discriminative from the perspective of data distribution (i.e. `Fwd_Seg_Size_Min`) is not present as

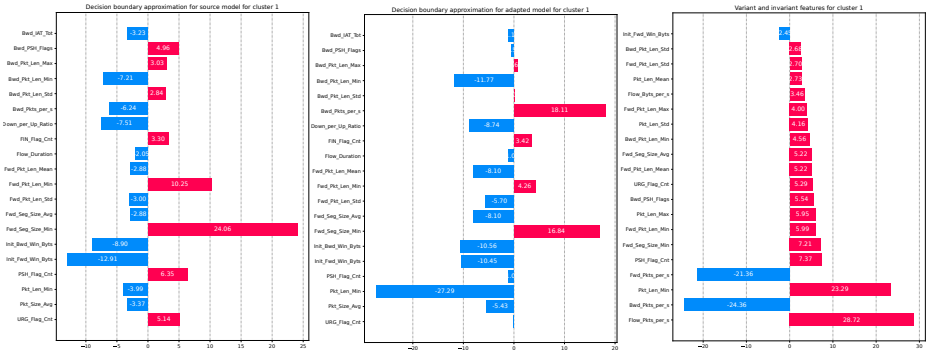

Fig. 7. Linear decision boundary approximation in source model for transform cluster 1 (left plot). Linear decision boundary approximation in the adapted model for transform cluster 1 (middle plot). Variant and invariant features between domains (right plot).

important in decision boundary explanations. Because the model fits decision boundary with a different objective than separating domains, this suggests a conclusion that domain adaptation explanation cannot be done purely based on data distribution analysis.

5 Summary

In this paper, we proposed explainability mechanisms for feature-based domain adaptation algorithms. The explanations provide two complementary perspectives on the domain adaptation process: (1) feature alignment and (2) decision boundary updates. Our initial investigation has shown that it is not enough to look at domain adaptation through the perspective of differences between data distributions of source and target domains. Instead, looking deeper into how these distributions are aligned by the adaptation mechanism and observing in which directions the decision boundaries are changing can give a new opportunity to relate the two domains on a more semantic level and open the possibilities to transfer background knowledge from the source to the target domain.

Although the work presented in this paper contributes the most towards the theoretical analysis of XAI applications in the domain adaptation area, the potential practical value is much broader. One of the possible applications of the method we presented herein is to distinguish domain shifts from anomalies in a data-stream scenario. The problem of differentiating between these two phenomena in data streams was recently reported in [6]. The usage of explainable domain adaptation can help in detecting anomalies or failures in industrial applications, separating them from domain shifts.

Furthermore, in the case of consecutively changing domains, especially in industrial settings, where each domain represents a new generation of products or processes, one can use explainable domain adaptation to build a predictive model on top of the discovered feature adaptation transforms and use it to tune future new models better.

Finally, one of the important research paths related to explainable domain adaptation is exploring more sophisticated ways of visualizing explanations. We plan to evaluate

methods that are more interactive and better suited for multidimensional dataset analysis, such as interactive parallel coordinate plots (IPCP) [3], and combine them with explanations obtained from our method.

Acknowledgment

The paper is funded from the XPM project funded by the National Science Centre, Poland under the CHIST-ERA programme (NCN UMO-2020/02/Y/ST6/00070) and the Swedish Research Council under grant CHIST-ERA-19-XAI-012.

References

1. Berenji, A., Nowaczyk, S., Taghiyarrenani, Z.: Data-centric perspective on explainability versus performance trade-off. In: Crémilleux, B., Hess, S., Nijssen, S. (eds.) Advances in Intelligent Data Analysis XXI. pp. 42–54. Springer Nature Switzerland, Cham (2023)
2. Bobek, S., Nalepa, G.J.: Introducing uncertainty into explainable ai methods. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science – ICCS 2021. pp. 444–457. Springer International Publishing, Cham (2021)
3. Bobek, S., Tadeja, S.K., Struski, L., Stachura, P., Kipouros, T., Tabor, J., Nalepa, G.J., Kristensson, P.O.: Virtual reality-based parallel coordinates plots enhanced with explainable ai and data-science analytics for decision-making processes. Applied Sciences **12**(1) (2022). https://doi.org/10.3390/app12010331, https://www.mdpi.com/2076-3417/12/1/331
4. Elsayed, M.S., Le-Khac, N.A., Jurcut, A.D.: Insdn: A novel sdn intrusion dataset. Ieee Access **8**, 165263–165284 (2020)
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(59), 1–35 (2016), http://jmlr.org/papers/v17/15-239.html
6. Jakubowski, J., Stanisz, P., Bobek, S., Nalepa, G.J.: Towards online anomaly detection in steel manufacturing process. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M. (eds.) Computational Science – ICCS 2023. pp. 469–482. Springer Nature Switzerland, Cham (2023)
7. Kamakshi, V., Krishnan, N.C.: Explainable supervised domain adaptation. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2022)
8. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. CoRR **abs/1709.10190** (2017), http://arxiv.org/abs/1709.10190
9. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
10. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: International Conference on Information Systems Security and Privacy (2018)
11. Sun, J., Lapuschkin, S., Samek, W., Zhao, Y., Cheung, N.M., Binder, A.: Explain and improve: Cross-domain few-shot-learning using explanations. arXiv preprint arXiv:2007.08790 **1**(3), 8 (2020)
12. Taghiyarrenani, Z., Fanian, A., Mahdavi, E., Mirzaei, A., Farsi, H.: Transfer learning based intrusion detection. In: 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 92–97 (2018). https://doi.org/10.1109/ICCKE.2018.8566601

13. Taghiyarrenani, Z., Nowaczyk, S., Pashami, S., Bouguelia, M.R.: Multi-domain adaptation for regression under conditional distribution shift. Expert Systems with Applications **224**, 119907 (2023). https://doi.org/https://doi.org/10.1016/j.eswa.2023.119907, https://www.sciencedirect.com/science/article/pii/S0957417423004086
14. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018). https://doi.org/https://doi.org/10.1016/j.neucom.2018.05.083, https://www.sciencedirect.com/science/article/pii/S0925231218306684
15. Zamir, A., Sax, A., Shen, W., Guibas, L., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 6241–6245. International Joint Conferences on Artificial Intelligence Organization (7 2019). https://doi.org/10.24963/ijcai.2019/871, https://doi.org/10.24963/ijcai.2019/871
16. Zhang, Y., Yao, T., Qiu, Z., Mei, T.: Explaining cross-domain recognition with interpretable deep classifier. arXiv preprint arXiv:2211.08249 (2022)
17. Zoppi, T., Ceccarelli, A., Bondavalli, A.: Towards a general model for intrusion detection: An exploratory study. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 186–201. Springer (2022)
18. Zunino, A., Bargal, S.A., Volpi, R., Sameki, M., Zhang, J., Sclaroff, S., Murino, V., Saenko, K.: Explainable deep classification models for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3233–3242 (2021)