Investigating the Contribution of Privileged Information in Knowledge Transfer LUPI by Explainable Machine Learning

Niharika Gauraha NIHARIKA@KTH.SE
Henrik Boström BOSTROMH@KTH.SE
KTH Royal Institute of Technnology
Stockholm, Sweden

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

Learning Under Privileged Information (LUPI) is a framework that exploits information that is available during training only, i.e., the privileged information (PI), to improve the classification of objects for which this information is not available. Knowledge transfer LUPI (KT-LUPI) extends the framework by inferring PI for the test objects through separate predictive models. Although the effectiveness of the framework has been thoroughly demonstrated, current investigations have provided limited insights only regarding what parts of the transferred PI contribute to the improved performance. A better understanding of this could not only lead to computational savings but potentially also to novel strategies for exploiting PI. We approach the problem by exploring the use of explainable machine learning through the state-of-the-art technique SHAP, to analyze the contribution of the transferred privileged information. We present results from experiments with five classification and three regression datasets, in which we compare the Shapley values of the PI computed in two different settings; one where the PI is assumed to be available during both training and testing, hence representing an ideal scenario, and a second setting, in which the PI is available during training only but is transferred to test objects, through KT-LUPI. The results indicate that explainable machine learning indeed has the potential as a tool to gain insights regarding the effectiveness of KT-LUPI.

Keywords: Learning Under Privileged Information, Knowledge Transfer, Shapley Value

1. Introduction

The Learning Under Privileged Information (LUPI) (Vapnik and Vashist, 2009) framework has been successfully applied in various fields. SVM+ (Vapnik and Vashist, 2009) was the first realization of the LUPI paradigm, which accelerated the learning process, and outperformed classical machine learning in a variety of applications. However, SVM+ suffered from limited scalability. To mitigate the scalability problem, various knowledge transfer methods within the LUPI framework have been proposed, see for example, (Vapnik and Izmailov, 2015, 2016, 2017), where the knowledge from the space of privileged information is transferred to the space where decision rule is constructed. In knowledge transfer LUPI (KT-LUPI), the privileged features are approximated by regressing on standard data features.

Shapley values have been used for explaining the predictions of (black-box) models by assessing feature importance when making predictions with such models. The Shapley value (Shapley, 1997) is defined as an average of all the marginal contributions to all possible coalitions of features. However, the computational cost of computing the value increases exponentially with the number of features. To address the computational complexity, various approximation methods have been proposed, with the Shapley Additive exPlanations (SHAP) method (Rodríguez-Pérez and Bajorath, 2019) perhaps being the most prominent.

The main objective of this paper is to investigate the contribution of privileged features by studying their importance using SHAP. To accomplish this objective, four main experiments have been performed, see section 4 for the details. In all our experiments, we compare the performances of the KT-LUPI models with the classical machine learning models, i.e. SVM, trained with and without PI. The SHAP values for the privileged features are compared between the KT-LUPI models and the classical machine learning models trained with all the features (including the privileged). We also explore the effect of the number of privileged features and the sample sizes on predictive performance and model explainability.

The paper is organized as follows. In section 2, we outline the background concepts and notations used throughout the paper. In Section 3, we describe the proposed approach to assess the contribution of PI by using Shapley values. In section 4, we conduct an empirical investigation using a set of real-world data sets. In Section 5, we conclude and discuss limitations and future work.

2. Background

In this section, we provide a brief background on the LUPI framework, Shapley values, and required machine learning concepts, and we fix notations and assumptions used throughout the paper.

In this paper, we will focus on binary classification and regression problems. The object space is denoted by $\mathcal{X} \in \mathbb{R}^p$, where p is the number of features; the label space is denoted by $\mathcal{Y} \in \{-1, 1\}$ and $\mathcal{Y} \in \mathbb{R}$ for binary classification and regression problems, respectively. The privileged feature space is denoted by $\mathcal{X}^* \in \mathbb{R}^m$, where m is the number of privileged features. We assume that each training example consists of corresponding objects in feature space ($x_i \in \mathcal{X}$) and privileged feature space ($x_i \in \mathcal{X}^*$) and its label ($y_i \in \mathcal{Y}$), specifically $\{(x_i, x_i^*, y_i)\}_{i=1}^l$, where l is the number of training examples. However, a test object consists of only an object in the feature space, $x \in \mathcal{X}$.

2.1. Knowledge Transfer LUPI

Let us assume that ℓ IID triplets $\{(x_i, x_i^*, y_i)\}_{i=1}^l$ are given. For the privileged features, $\phi_j(x), j = 1, \ldots, m$, m regression functions are learned by using p-dimensional vectors $\mathbf{X} = x_1, x_2, ..., x_\ell$ as explanatory variables and corresponding scalar values $\mathbf{x}^*(i)$, for $i = 1 \ldots m$, as response variables. Different learning algorithms can be used for approximating the privileged features. In all our experiments, we use linear regression for this purpose.

In KT-LUPI, both the features of space $\mathcal{X}$ and the features defined by the regression functions are considered for constructing the decision rules. In our experiments, we choose a small number of important features in the privileged space. We then transfer them (using linear regression) into the decision space.

A classifier (or regressor) is trained, e.g., using SVM (or SVR), on the modified dataset $(\mathbf{Y}, \mathbf{X}_{mod})$, and a decision function, f, is learned, where X_{mod} is defined as in Equation 1. Given a new object in the feature space $x \in \mathcal{X}$, we compute m regression estimates $\hat{x}^*$, using the m regression functions learned previously, after which the inference is made in the $m+p$ dimension space using the decision function f.

$$\mathbf{X}_{mod} = \begin{bmatrix} x_1 & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_m(x_1) \\ x_2 & \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_m(x_2) \\ \dots & \dots & \dots & \dots & \dots \\ x_\ell & \phi_1(x_\ell) & \phi_2(x_\ell) & \dots & \phi_m(x_\ell) \end{bmatrix} \tag{1}$$

2.2. Model Explainability using SHAP Values

SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) is based on a game-theoretic approach, which offers a way to fairly distribute the payoffs to the individual players in a coalition (Shapley, 1997). SHAP employs the approach to measure the contribution of input features to the outcome of a machine learning model at the instance level (Rodríguez-Pérez and Bajorath, 2019). Given a specific data object, SHAP outputs a value for each feature that represents the contribution of that feature to the final prediction.

Let us denote the decision function for a machine learning method with $f(x)$, where $x \in \mathcal{X}$ and the object space is denoted by $\mathcal{X} = \mathbb{R}^p$, where p is the number of features. Let the player set be a single data object $x = \{x_i | 1 \leq i \leq p\}$. Then the payoff of a coalition $s \subseteq x$ is the scalar value prediction $f(s)$ calculated from the subset of feature values. Since the decision function takes the input in feature space $x \in \mathcal{X}$, for computing $f(s)$, the missing input feature values are imputed with reference values, e.g., the mean computed from multiple instances (Lundberg and Lee, 2017). In all our experiments, we use SHAP to compute feature scores.

3. Model Explainability of KT-LUPI using SHAP Values

In this section, we describe SHAP value computation for KT-LUPI. As discussed in the previous section, given a test object to compute the SHAP values for a subset of features it is required that the decision function takes the input for those features, and the values for the remaining features are imputed. In general, for LUPI methods, the input space for the decision rule is limited to the standard feature space, and it is not possible to compute SHAP values for the Privileged Features (PFs). However, for KT-LUPI, the input for the decision function is both standard features and transformed PFs. Hence, it is possible to compute the importance of PFs (though in the transformed space).

4. Experiments

We perform experiments with five binary classification and three regression datasets from the UCI machine learning repository (Lichman et al., 2013); see Table 1 for details about the datasets. Each dataset is randomly partitioned into a test set (30%) and a training set (70%). We use SHAP values to choose privileged features; for example, for choosing five PFs in a dataset, we first train a linear model using the training set with all the features and then compute the SHAP values using the test set, and finally, the top five features having highest SHAP values are identified as PFs. Once the PFs are chosen, we remove them from the set of features. For example, for a hypothetical dataset having 10 features, let us assume that all the features are named $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and also assume that the features $\{1, 3, 5, 7, 9\}$ are PFs. Then X consists of samples from the features $\{0, 2, 4, 6, 8\}$ and X^* consists of samples from the privileged features $\{1, 3, 5, 7, 9\}$. The set of all features are now denoted by $\{XX^*\} = \{0, 2, 4, 6, 8, 1, 3, 5, 7, 9\}$ and their indices are reset from 0 to 9 again.

Table 1: Description of the datasets from UCI repository that are used in the evaluation. *Training* refers to the number of examples in the training set, *Test* refers to the number of objects in the test set, *Features* refer to the number of features in the dataset. *Privileged Features* and *Standard Features* refer to the number of PFs and standard features, respectively. The first five datasets correspond to classification problems, while the remaining three datasets correspond to regression problems

Dataset	Training	Test	Features	Privileged Features	Standard Features
Spambase	3220	1381	57	5	52
Breast Cancer	398	171	30	5	25
Phishing Websites	7738	3317	30	5	25
Australian	483	207	14	5	9
Parkinsons	136	59	22	5	17
Wine	3428	1470	10	3	7
Energy Efficiency	537	231	8	2	6
Concrete	721	309	8	2	6

All the experiments are conducted using Python. The package "shap" is used to compute SHAP values. Support Vector Machine (SVM) and Support Vector Regression (SVR) with linear kernel are trained using Python implementations of LibSVM (Chang and Lin, 2011) in the Python package "scikit-learn" (Pedregosa et al., 2011). In all the experiments, linear regression is used to approximate the privileged features. The top m features that have the highest SHAP values are chosen as PFs, these m features are referred to as primary features and the rest of the features are referred to as standard features. We may interchangeably use primary features or privileged features. In all the experiments, the following three types of classification (or regression) models are trained:

1. SVM (or SVR) on standard features: the SVM (or SVR) algorithm is used to create a decision rule using standard features.

2. Knowledge transfer LUPI (KT-LUPI): knowledge transfer from privileged features to the space of standard features is realized using linear regression. After augmenting standard features with the regressed values of privileged features, the SVM (or SVR) algorithm is used to create a decision rule on the augmented decision space.

3. SVM (or SVR) on all features: the SVM (or SVR) algorithm is used to create a decision rule using both primary and standard features.

For the assessment of model performance, we measure Area Under Curve (AUC) for classification (higher is better) and Root Mean Squared Error (RMSE) for regression (lower is better). To measure the similarity of the contribution of the privileged information in KT-LUPI and in SVM (or SVR) on all features, we compute the sum of absolute differences (SAD) of the SHAP values for privileged features. A lower SAD score indicates a higher similarity between KT-LUPI and SVM on all features in terms of the contribution of the PI.

We present results from four experiments to investigate the performance and explainability of the KT-LUPI models.

4.1. Experiment 1: Model performance and explainability for classification datasets

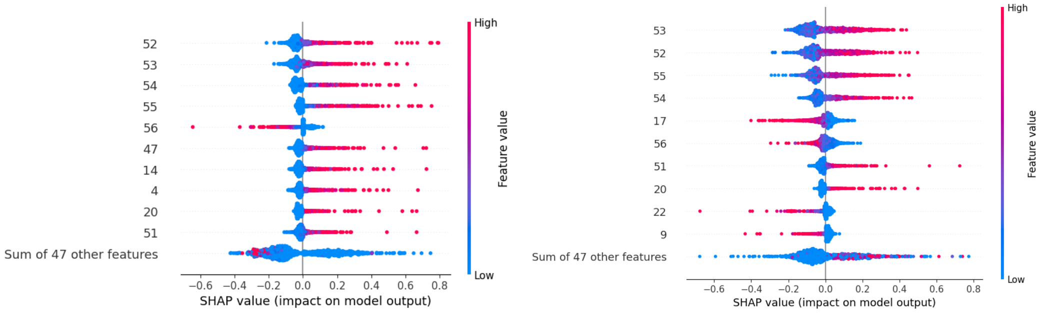
The main objective of this experiment is to analyze the contribution of the transferred privileged features using SHAP values for classification problems. We compare the performances of KT-LUPI models with the following models: SVM on standard features and SVM on all features. The SHAP values for the Privileged Features (PFs) are compared for the KT-LUPI models and the models trained using SVM on all features.

In this experiment, we train three different models for each of the classification datasets that are described in Table 1. The first model is trained with the SVM algorithm using only the standard features X. In the second model, the privileged features (PF) are approximated using linear regression, and the KT-LUPI model is trained using SVM on the combined features, X and the approximated PFs. The third model is trained using SVM on all the features. The results are reported in Table 2. The AUC scores for SVM on standard features, KT-LUPI and SVM on all features are reported in columns 1-3. The fourth column shows the sum of absolute difference between the SHAP values for the PFs as computed from the models obtained by KT-LUPI and SVM on all features. A plot of the SHAP values for the Spambase dataset is shown in Figure 1, where features are ordered in descending order (from top to bottom) according to their scores. Similar plots for the other data sets are available in Appendix A.

Table 2 shows that KT-LUPI yields higher AUC scores than SVM on standard features for most of the datasets. It also shows that the AUC scores for KT-LUPI lie in between SVM on standard features and SVM on all features. We also notice that the higher the difference in AUC scores between the two models (SVM on all features and KT-LUPI) is, the higher the SAD value. As explained previously, by construction the PFs are the five features having the highest SHAP values for the model SVM on all features. The five privileged features for KT-LUPI are also among the top 10 features with the highest SHAP values, as shown in Figures 1, 5, 6, 7 and 8.

Table 2: Experiment 1: The first column presents the name of the datasets. The AUC scores for SVM on standard features, KT-LUPI and SVM on all features are reported in the three following columns. The last column shows the sum of absolute difference between the SHAP values for the privileged features, when computed using KT-LUPI and SVM on all features, repectively.

Dataset	SVM on standard features	KT-LUPI	SVM on all features	SAD
Spambase	0.869	0.871	0.899	0.137
Breast Cancer	0.975	0.975	0.979	0.027
Phishing Websites	0.727	0.727	0.926	0.374
Australian	0.754	0.782	0.884	0.374
Parkinsons	0.757	0.769	0.808	0.099



(a) (b)

Figure 1: Results from experiment 1 for the Spambase dataset; (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI

4.2. Experiment 2: Model performance and explainability for regression datasets

The aim of this experiment is to analyze the contribution of the (transferred) privileged features using SHAP values for regression problems.

This experiment is similar to the first except that we here used regression instead of classification datasets and the SVR instead of the SVM algorithm for training. In this experiment, we also train three different models for each dataset. The first model is trained with Support Vector Regression (SVR) using only the standard features. In the second model, the privileged features (PF) are approximated using linear regression, and the KT-LUPI model is trained using SVR on $(X, \hat{X}*)$. The third model is trained with SVR using all the features $(X, X*)$. The results are reported in Table 3. The Root Mean Squared Error (RMSE) score for SVR on standard features, KT-LUPI and SVR on all features are reported in columns 2-4. The fifth column shows the sum of absolute difference between the privileged feature SHAP values. SHAP values for the Wine dataset are plotted in Figure 2, where the features are again ordered in descending order (from top to bottom) according to their SHAP values. Similar plots for the other data sets are available in Appendix B.

Table 3 shows that KT-LUPI yields lower (better) RMSE scores than SVR on standard features for most of the datasets. It also shows that the RMSE scores for KT-LUPI lie in between SVR on standard features and SVR on all features. As before, the PFs are the top features having the highest SHAP values for the model SVR on all features, and we can observe that some of the privileged features for KT-LUPI are also among top five features with the highest SHAP values, as reported in Figure 2.

Table 3: Experiment 2: The first column presents the name of the datasets. The RMSE scores for SVR on standard features, KT-LUPI and SVR on all features are reported in the three following columns. The last column shows the sum of absolute difference between the privileged feature SHAP values computed by KT-LUPI and SVR on all features.

Dataset	SVR on standard features	KT-LUPI	SVR on all features	SAD
Wine	1.011	1.010	0.478	1.576
Energy Efficiency	4.651	4.236	2.937	1.296
Concrete	14.323	11.918	11.594	2.252

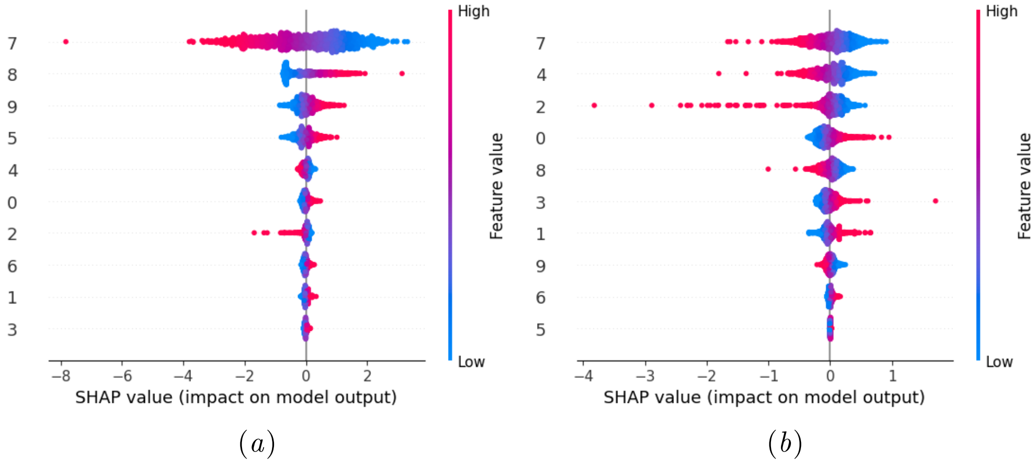


Figure 2: Results from experiment 2 for the Wine dataset; (a) SVR on all features $(X + X^*)$ (b) KT-LUPI using SVR with X as standard features and X^* as PI

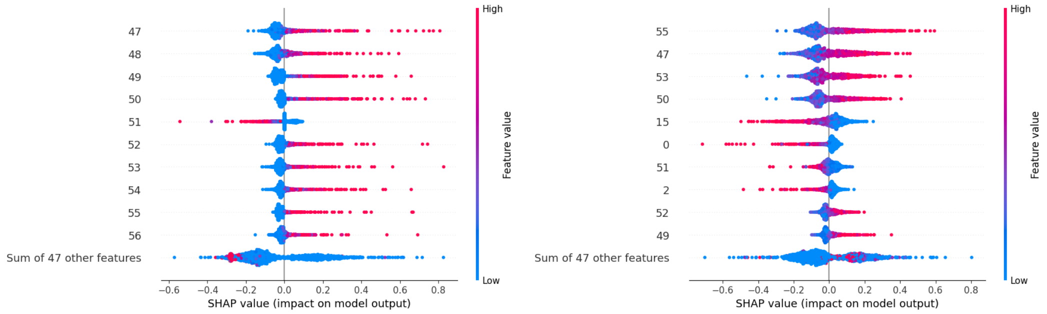
4.3. Experiment 3: Varying numbers of privileged features

In this experiment, we analyze the effect of the number of PFs on predictive performance and explainability of KT-LUPI. The results are reported in Table 4 and Figure 3. This experiment is similar to the first except that we use only the Spambase dataset, and it is repeated for the following number of PFs: $\{5, 10, 15, 20\}$.

It can be observed in Table 4 that, as we increase the number of PFs, the performance of both KT-LUPI and SVM on standard features decreases, while KT-LUPI still performs better of the two. It can also be seen (in column four of Table 5) that with the increase in the number of PFs, the SAD score increases. In other words, with a larger number of PFs, it becomes more difficult to explain the contribution of the (transferred) privileged information. The corresponding plot of SHAP values for Spambase dataset with 10 PFs is reported in Figure 3. Similar plots with 5, 15 and 20 PFs are available in Appendix C.

Table 4: Experiment 3: The first column presents the number of privileged features selected. The AUC scores for SVM on standard features, KT-LUPI and SVM on all features are reported in the three following columns. The last column shows the sum of absolute difference between the PFs SHAP values computed by KT-LUPI and SVM on all features.

# PFs	SVM on standard features	KT-LUPI	SVM on all features	SAD
5	0.869	0.871	0.899	0.13868
10	0.834	0.846	0.899	0.29582
15	0.781	0.794	0.899	0.37658
20	0.762	0.784	0.899	0.52306



(a) (b)

Figure 3: Results from experiment 3 for the Spambase dataset (with ten PFs); (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI

4.4. Experiment 4: Varying the sample size

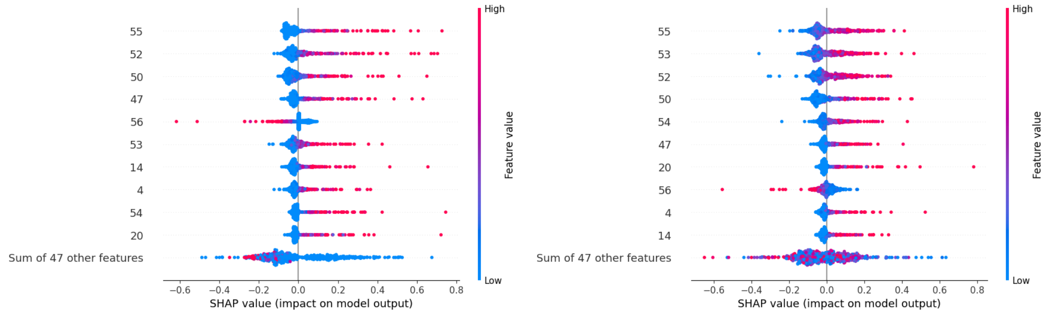
The main goal of this experiment is to investigate the effect of the sample size on the predictive performance and explainability of KT-LUPI.

This experiment is again similar to the first except that we use only the Spambase dataset, and it is repeated for the following sample sizes: $\{100, 200, 500, 1000, 20000\}$. The results are reported in Table 5 and Figure 4.

Table 5 shows that for a very small sample size (100), the performance of KT-LUPI is worse than SVM on standard features. However, with moderate sample sizes (500 and above) it is more efficient. The performance of KT-LUPI is always lower as compared to SVM on all features. However, the explainability is comparable for the two models as shown in Figure 4. As expected, with an increase in the sample size, the SAD score decreases, as shown in column five of Table 5. In other words, when increasing the sample size, the contribution of PI becomes more similar for both KT-LUPI and SVM on all features.

Table 5: Experiment 4: The first column presents the number of sample size. The AUC scores for SVM on standard features, KT-LUPI and SVM on all features are reported in the following three columns. The last column shows the sum of absolute difference between the PF SHAP values computed by KT-LUPI and SVM on all features.

Sample size	SVM on standard features	KT-LUPI	SVM on all features	SAD
100	0.833	0.766	0.866	0.198
200	0.802	0.802	0.867	0.105
500	0.780	0.784	0.855	0.086
1000	0.850	0.864	0.866	0.072
2000	0.908	0.921	0.933	0.064



(a) (b)

Figure 4: Results from experiment 4 for the Spambase dataset (with 2000 samples size); (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI

5. Conclusion

In this study, we have investigated the predictive performance and explainability of KT-LUPI. Similar to earlier studies, for example in (Gauraha et al., 2019), KT-LUPI was observed to have higher predictive performance than SVM (or SVR) using standard features only. However, in terms of explainability, KT-LUPI was comparable with SVM (or SVR) using all features. Most interestingly, we observed a positive correlation between the relative performance of KT-LUPI, when compared to having access to privileged information also during testing, and the agreement on how important the privileged features are; in case of a lower agreement, then KT-LUPI performed relatively worse compared to having access to privileged information. We also investigated the effect of the number of PFs and sample sizes on predictive performance and explainability. As the number of PFs increased, the predictive performance of both KT-LUPI and SVM using standard features were observed to decrease, but KT-LUPI still outperformed the latter. We also noticed that with an increase of the number of PFs, the SAD score increased. Similar to findings reported earlier (Nouretdinov, 2022), for very small sample sizes, the predictive performance of KT-LUPI was worse than that of SVM using standard features. However, for moderate sample sizes, KT-LUPI was more efficient. As expected, when increasing the sample size, the SAD score was observed to decrease.

The above results indicate that explainable machine learning indeed has the potential to be used as a tool to gain insights regarding the effectiveness of KT-LUPI. The current study is just a start and can be extended in several directions. One direction for future research is to analyze the contribution of privileged features for split KT-LUPI (Gauraha et al., 2019). Other directions for future research include investigating scenarios in which there is a cost associated with obtaining privileged information and explore strategies to exploit the outcome from analyzing the contribution of such information.

References

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Niharika Gauraha, Fabian Söderdahl, and Ola Spjuth. Split knowledge transfer in learning under privileged information framework. In *Conformal and Probabilistic Prediction and Applications*, pages 43–52. PMLR, 2019.

Moshe Lichman et al. Uci machine learning repository, 2013.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Ilia Nouretdinov. On efficiency of learning under privileged information. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 239–252. PMLR, 24–26 Aug 2022.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Raquel Rodríguez-Pérez and Jürgen Bajorath. Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values. *Journal of medicinal chemistry*, 63(16):8761–8777, 2019.

Lloyd S Shapley. A value for n-person games. *Classics in game theory*, 69, 1997.

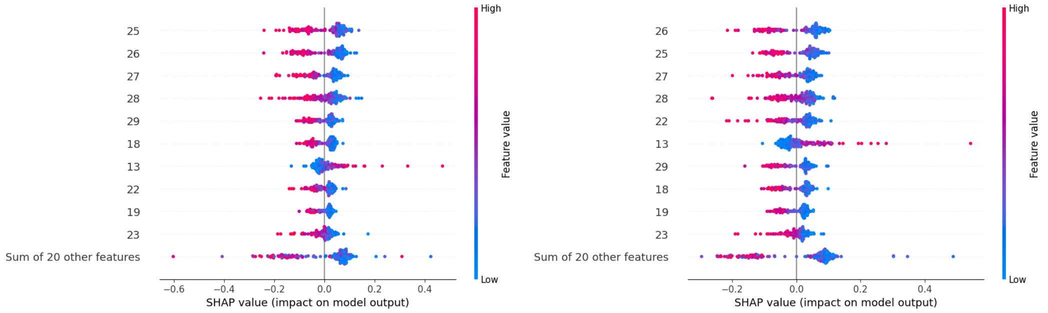
Vladimir Vapnik and Rauf Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In *International Symposium on Statistical Learning and Data Sciences*, pages 3–32. Springer, 2015.

Vladimir Vapnik and Rauf Izmailov. Learning with intelligent teacher. In Alexander Gammerman, Zhiyuan Luo, Jesús Vega, and Vladimir Vovk, editors, *Conformal and Probabilistic Prediction with Applications*, pages 3–19, Cham, 2016. Springer International Publishing.

Vladimir Vapnik and Rauf Izmailov. Knowledge transfer in svm and neural networks. *Annals of Mathematics and Artificial Intelligence*, 81(1-2):3–19, 2017.

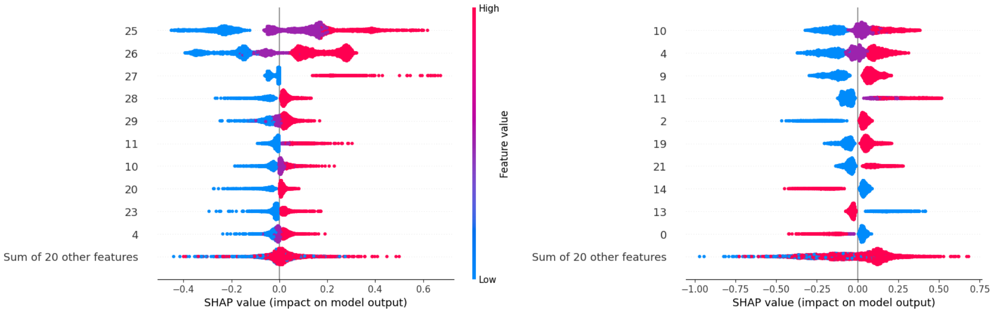
Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Appendix A. Results from Experiment 1



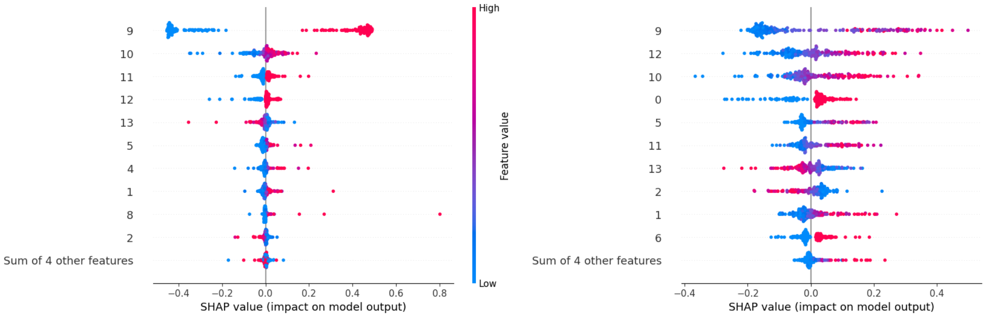
(a) (b)

Figure 5: Results from experiment 1 for the Breast Cancer dataset; (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI



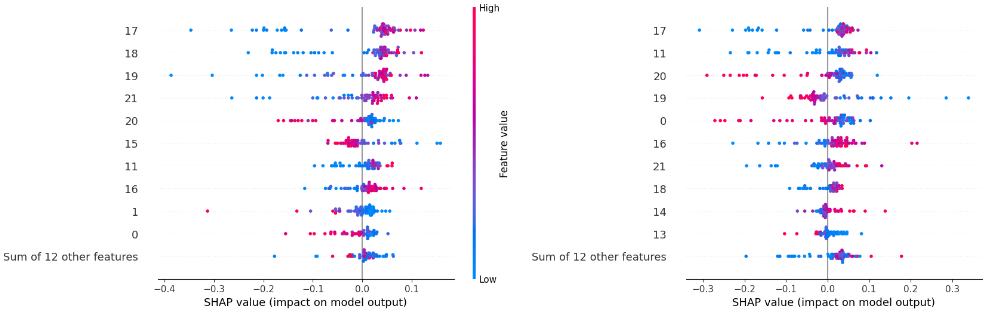
(a) (b)

Figure 6: Results from experiment 1 for the Phishing dataset; (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI



(a) (b)

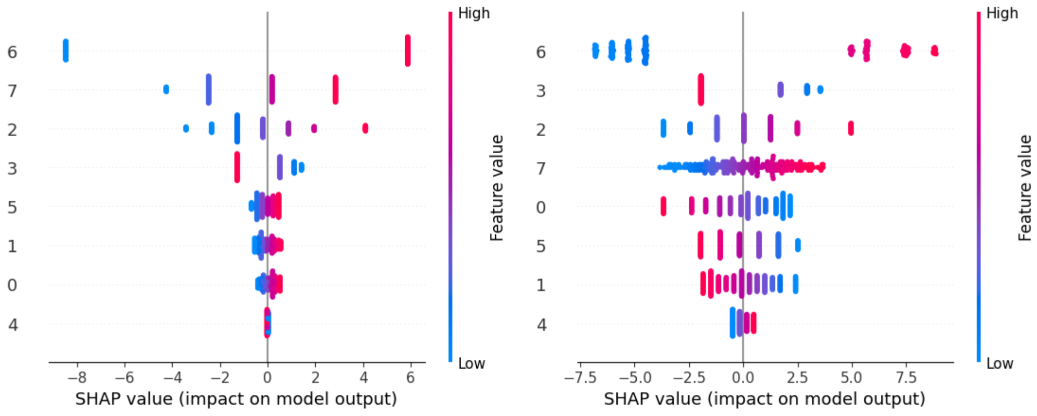
Figure 7: Results from experiment 1 for the Australian dataset; (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI



(a) (b)

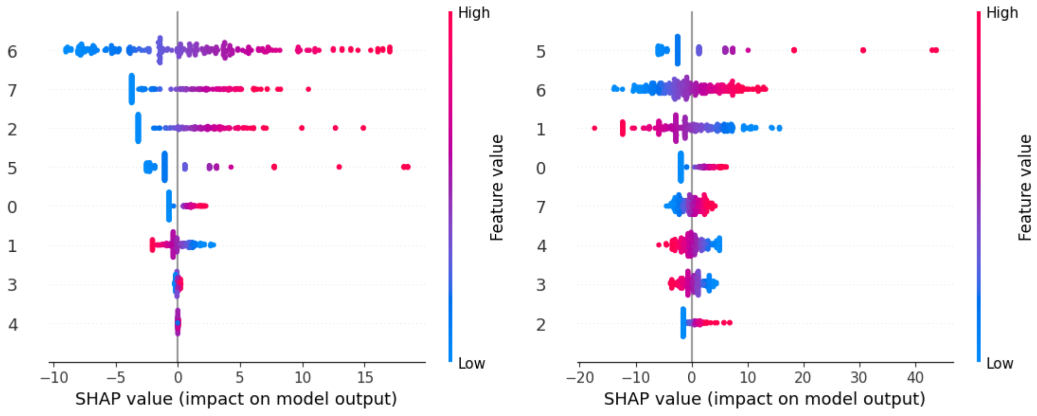
Figure 8: Results from experiment 1 for the Parkinsons dataset; (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI

Appendix B. Results from Experiment 2



(a) (b)

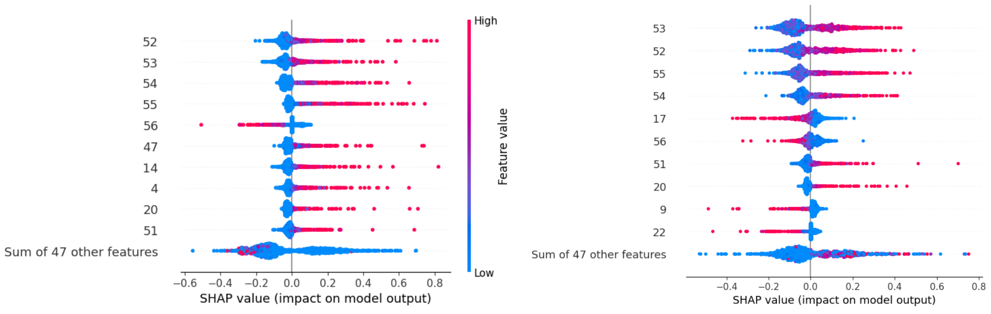
Figure 9: Results from experiment 2 for the Energy Efficiency dataset; (a) SVR on all features $(X + X^*)$ (b) KT-LUPI using SVR with X as standard features and X^* as PI



(a) (b)

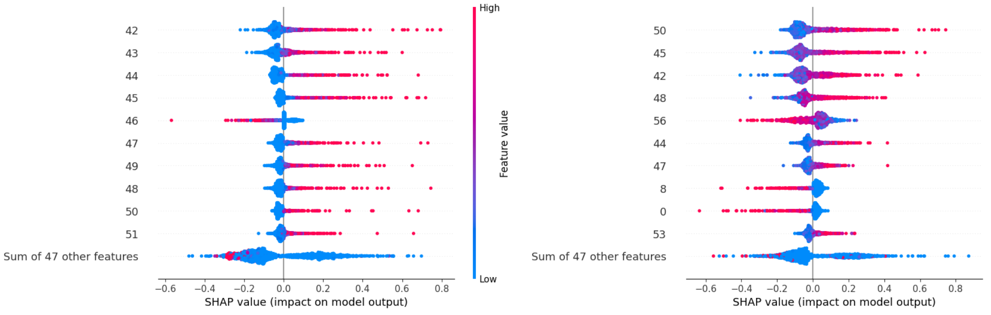
Figure 10: Results from experiment 2 for the Concrete dataset; (a) SVR on all features $(X + X^*)$ (b) KT-LUPI using SVR with X as standard features and X^* as PI

Appendix C. Results from Experiment 3



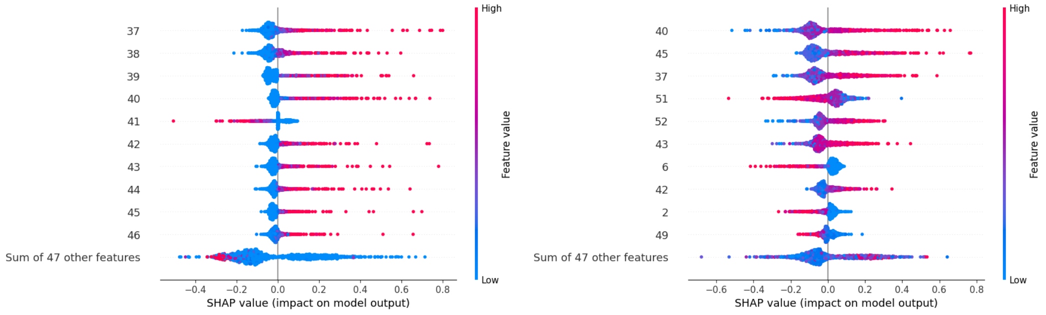
(a) (b)

Figure 11: Results from experiment 3 for the Spambase dataset (with 5 PFs); (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI



(a) (b)

Figure 12: Results from experiment 3 for the Spambase dataset (with 15 PFs); (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI



(a) (b)

Figure 13: Results from experiment 3 for the Spambase dataset (with 20 PFs) (a) SVM on all features $(X + X^*)$ (b) KT-LUPI using SVM with X as standard features and X^* as PI