On Incorporating new Variables during Evaluation

Harsimran Bhasin and Soumyadeep Ghosh
AI Garage, Mastercard
Gurgaon, India
{harsimran.bhasin@mastercard.com, soumyadeep.ghosh@mastercard.com}

Abstract

Any classification or regression model needs access to the same features and input that were utilized to train the model. However in real world scenarios, several models are in operation for years and new variables/features may be available during the inferencing stage. If such features are to be utilized, their values have to be captured in the dataset that was utilized for training the model. We propose a model agnostic approach where we trained a model without the access to those features during the training stage, which could benefit from the additional features available during testing. We show that by using the proposed approach and without any access to the extra features during the training phase, we are able to improve the performance of the model on four real world tabular datasets. We provide extensive analysis on how and which variables result in the improvement over the model which was trained without the extra feature(s).

1 Introduction

Machine learning based models have seen tremendous success in recent times on a wide variety of tasks, particularly on image, language, speech, and graph data. Deep learning models have recently gained popularity in several industrial applications such as web search, e-commerce [20], recommendation [10] engines, server fault prediction [15], and fraud detection [1] in payments. Such models need to be trained on real world data from a particular time period and then they are evaluated during several real time applications such as online or in-store payments, recommendation engines on e-commerce websites, cryptocurrency transfers, and computing server fault prediction. In most cases, the data which is used in training and testing/evaluation come from different time periods and sometimes from different regions of the world. The set of features which were used to train the model must be available for the time period of testing/evaluation in order to utilize the trained model. There might be situations where an additional set of features might be available during evaluation. If that feature was not utilized during training, then it is not possible to utilize that feature during model inferencing. In this paper we propose an approach which would allow us to utilize features not seen during training for testing.

In order to illustrate the proposed approach we take the example of a real world application, such as payment fraud detection which runs in real time. The data used to train such a model is generally tabular in nature. The model would be trained on fraud and non-fraud transaction data for a specific time period and the model is expected to score transactions on the likeliness of them being fraud in real time. It may happen that the organization that built this model may start collecting some extra attributes (eg. card type) for transactions during evaluation. Since this attribute was not collected/observed during the training period (may be an older time period), we cannot retrain the model with that additional attribute because it does not exist for that time period. However, we propose a technique to allow us to incorporate the extra attribute in the trained model. As shown in Figure 1 we treat the extra feature (say F as our label and train a surrogate model (say M) on the testing data to predict the extra feature. This model M can now be applied on the training data to

Table Representation Learning Workshop at NeurIPS 2023. (NeurIPS 2023).

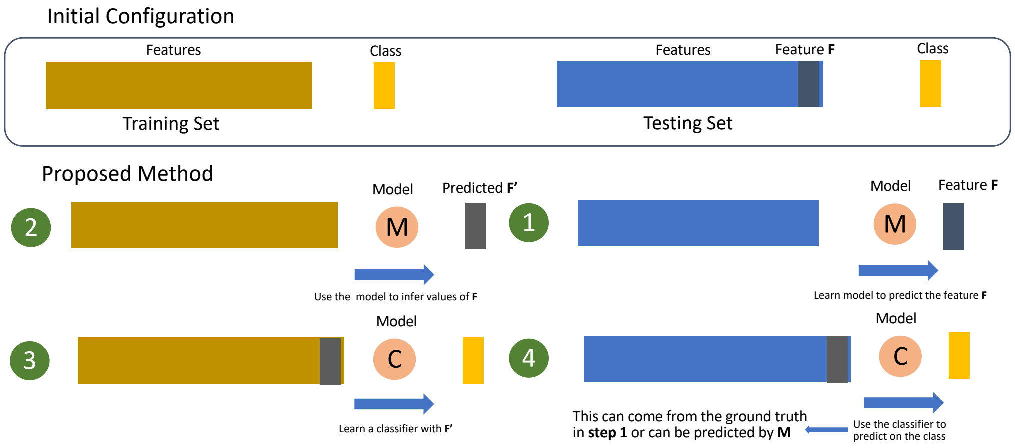
Initial Configuration
Features
Class
Features
Feature F
Class
Training Set
Testing Set
Proposed Method
Model
Predicted F’
Model
Feature F
2
M
1
M
Use the model to infer values of F
Learn model to predict the feature F
Model
Model
3
C
4
C
Learn a classifier with F’
This can come from the ground truth in step 1 or can be predicted by M
Use the classifier to predict on the class

Figure 1: Illustration for the proposed approach

generate the new feature *F’*. This newly generated feature column *F’* can now be augmented with the previous training data and another version of the classifier can be trained with the same training labels.

2 Related Work

There are three different research areas associated with unknown categories in dataset. First area deals with data drift detection which studies if the relationship between dependent and independent variables gets changed during production [2]. Second area deals with open set learning [29, 27] where there are test samples from classes that have not been observed during training. The third area deals with incremental learning [5, 11, 6] that involves processing incoming data from a data stream.

Within open set learning there are broadly two approaches towards unknown categories in dataset. First approach augments the training data to accommodate unknown categories [29] while the second approach applies post training augmentation [27]. Current problem is related to the first approach where unknown categories are discovered in the train dataset itself.

Incremental learning can be divided into two areas, one where the current model is retrained incrementally using newer data patterns to obtain a better model [11, 6]. Second, where the current model is retrained by combining new and old data to form a better representative dataset [5].

Approaches to identify unknown categories in train dataset can be classified into three broad categories. The first set of approaches identify unknown classes and augment the feature space to make such classes visible. Within these approaches some works have explored feature information in training dataset for discovery [29]. One method [17] augments training data by adding generated examples close to training data. Huang et al. [8] use unknown label detection to classify knowns and unknowns. Others use clustering based regularization to discover unobserved labels [14]. Zhou et al. [31] use structure network to differentiate class centres of known and unknown classes. Second set of approaches use semi-supervised and unsupervised training to label unlabelled data which is clustered into seen and unseen classes [22]. Zheng et al. [30] use a two-stage framework for object detection and category discovery. [4] identify new categories on-the-fly using hash coding. Yang et al. [26] handles arbitrary unknown class distributions by utilising class priors. NEal et al. [17] labels novel classes using online clustering. Wu et al. [24] use self-supervised and inductive methods for feature extrapolation. Third set of approaches uses outlier detection algorithms to identify new classes in train dataset [7]. Pal et al. [19] use outlier calibration network and meta-training. Huang et al. [9] performs location agnostic outlier detection. Pal et al. [18] use class-conditioned adversarial samples for separating closed and open spaces. Xu et al. [25] compares feature maps of train and test datasets using local outlier factor to detect open set samples.

Within the domain of incremental learning, one approach uses discard-after-learn approach where new data is dropped after using it for model retraining [11]. Second approach makes decisions to

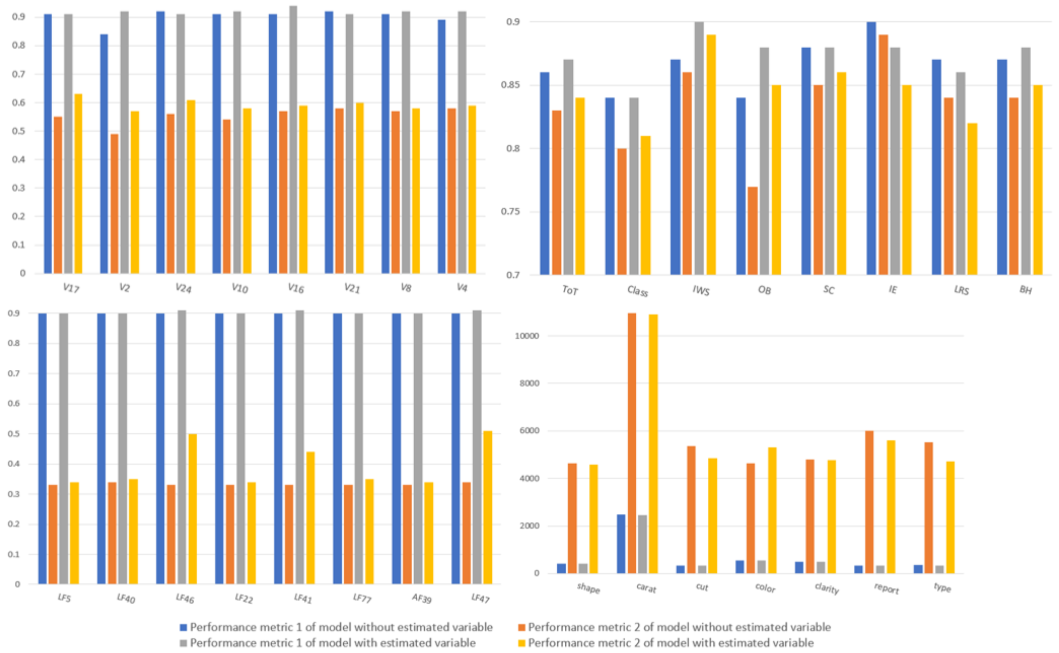

Figure 2: Experimental results using surrogate models for **left top:** Credit card fraud dataset **right top:** Airline Passenger satisfaction, **bottom left:** Elliptic Bitcoin dataset and **bottom right:** Brilliant Diamonds dataset. For the first 3 plots performance metric 1 and 3 are Area under the ROC curve and performance metric 2 and metric 4 Area under the precision recall curve.

accept or reject new attributes, where a secondary neural network is trained on newer attributes which is eventually merged with the original neural network [6]. Third approach identifies common attributes between different classes and aims to identify attributes of unseen object classes [12].

In open set learning, all the existing approaches attempts to identify new classes in the dataset. The problem where new categorical features appear for some variables while the target classes remain same, has not been explored to the best of our knowledge.

In the incremental learning literature, one approach tries to identify attributes of unseen classes [12]. Attribute identification has been carried out for computer vision applications. It has not been explored how this can be used with tabular data. This problem tries to identify attributes for unseen classes and does not explore unseen categories in the training dataset especially for tabular data.

3 Proposed Method

Let us assume a classical pattern classification scenario where a tabular data set X=$\{X_1^1, X_1^2, X_1^3, ...X_i^1, ...X_n^1\}$ is available for training from n data samples, and subsequently for testing the data given is P=$\{P_1^1, P_1^2, P_1^3, ...P_i^1, ...P_m^1\}$ for m testing samples. Let us assume that the set of features that were available for training and testing initially is given by F=$\{F_1^1, F_1^2, F_1^3, ...F_i^1, ...F_t^1\}$. Utilizing the training data, we train a model $C_\theta(c) : \mathbb{R}^t \longrightarrow \mathbb{R}$, where t is the dimensionality or the number of features available during training, and $\theta(j)$ are the parameters of the model.

Let us create a scenario where during testing an extra feature column F_{t+1} is made available in addition to F, such as $F \cup F_{t+1} = F'$. We train a surrogate model $M_\theta(m) : \mathbb{R}^t \longrightarrow \mathbb{R}$. This surrogate model is learned to predict the extra variable F_{t+1} from F using the data provided in the test set P. This model may then be used to inference on X to generate F_{t+1} on the train dataset. Now, since we have been able to generate one extra feature column for the training data, we can train another model $C'_\theta(m) : \mathbb{R}^{t+1} \longrightarrow \mathbb{R}$ on X and then use it for inferencing on the same testing data P.

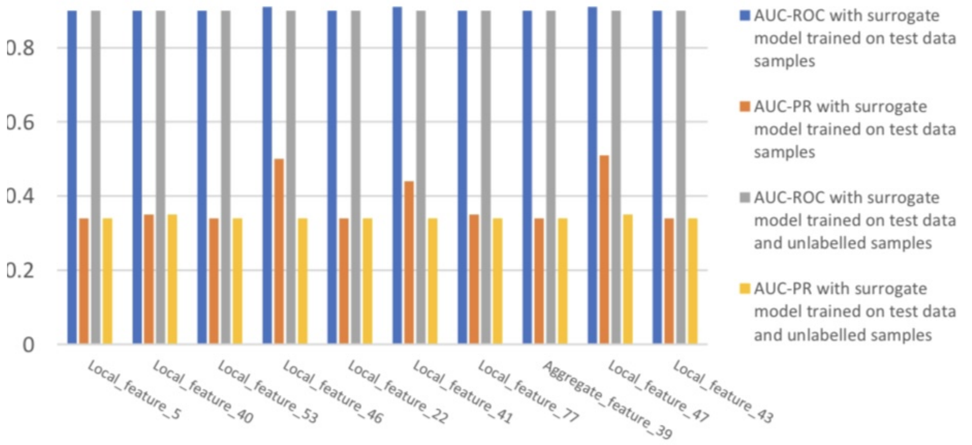

Figure 3: Illustration for surrogate model using different datasets to train on Elliptic Bitcoin dataset

4 Experiments and Results

4.1 Datasets

We demonstrate results on two different kinds of datasets in our proposed setting: financial domain and general tasks. In the financial domain, we use credit card dataset to detect fraudulent credit card transactions [3] and the elliptic dataset to detect bitcoin transaction fraud [23]. Both these are tabular datasets, however only the PCA representation of the features have been used. We further evaluate our method on two general tabular datasets like airline satisfaction dataset where the task is to predict customer satisfaction [21] and brilliant diamonds dataset [16] where the task is to predict price of a diamond.

4.1.1 Kaggle Credit Card Fraud

This dataset contains transactions made by credit card users. It has 284,807 transactions out of which 492 are fraudulent transactions. The dataset is highly imbalanced where fraudulent transactions account for 0.172% of all transactions. There are 28 numerical features obtained as an output of PCA transformation along with "amount" and "time".

4.1.2 Elliptic Bitcoin Fraud

This dataset maps bitcoin transactions to real entities belonging to licit (exchanges, wallet providers, miners, licit services, etc.) and illicit categories (scams, malware, terrorist organizations, ransomware, Ponzi schemes,etc.). The dataset is presented as a transaction graph, each node being a bitcoin transaction, and an edge represents the flow of bitcoins between one transaction and the other. The dataset contains 203,769 nodes/data samples, out of which 4,545 (around 2%) are labelled as illicit, 42,019 (around 21%) samples are labelled as licit and the rest are unlabelled.

4.1.3 Airline Satisfaction

This dataset contains results of an airline customer satisfaction survey. The total number of samples in this dataset are 103904 out of which 43.3% of the customers are satisfied with airline service while the rest are either neutral or dissatisfied. It has 24 features with 5 categorical features and the rest being numeric.

4.1.4 Brilliant Diamonds

This dataset contains records for natural and lab-created diamonds. The total number of samples in this dataset are 119307. The task here is to predict price of a diamond based on various attributes like cut, color, clarity, etc. It has 11 features with 8 categorical features and the rest numerical features.

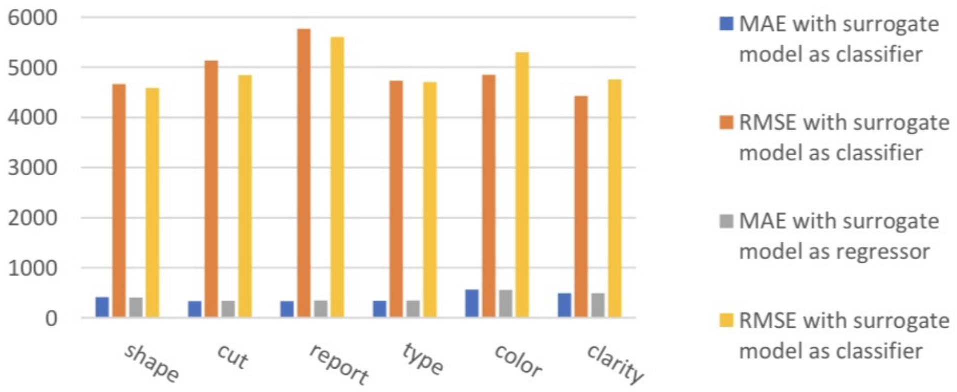

Figure 4: Illustration for the proposed approach using classifier and regressor surrogate models on Brilliant Diamonds dataset

4.2 Experiments

In this section we provide an outline of the experiments using the proposed method along with comparison on relevant baselines. We begin this section by providing a list of experiments, followed by additional details. The experiments performed are as follows:

4.2.1 Training the Surrogate Model

In this experiment, we show the effectiveness of the proposed method by training a surrogate model to estimate new variables encountered during evaluation. In this setting, the test dataset has new variables that were not seen during model training. A surrogate model is trained using the features available in the test set (omitting the target variable) to estimate this new variable. The surrogate model can then be used to estimate this new variable in the train dataset as well. This estimated variable can be used to retrain a model to predict the original target. We present experimental results for this in Fig. 2. Further, we tested the efficacy of using regression and classification tasks to build the surrogate model. In the regression task the surrogate model can estimate the new variable using continuous numerical values while the classifier surrogate model can estimate the new variable using a probability value bounded by the interval [0,1]. Experimental results on change observed in performance upon using classifier and regressor surrogate models is illustrated on Brilliant Diamonds dataset (Fig. 4). For this experiment only categorical variables were considered.

4.2.2 Utilizing unlabelled data

In this experiment, we try to observe how does surrogate model perform when additional data is provided for training. In Elliptic Bitcoin Fraud dataset, about 77% of data is unlabelled which cannot be used for direct modelling. We performed an experiment to compare our method's performance when the surrogate model is trained using different data sources. In first case, only test dataset is used for training the surrogate model. In the second case, test dataset along with unlabelled data is used to train the surrogate model. (Illustration 3) shows results for this experiment.

4.3 Experimental Protocol

For experiments, the dataset was divided into two parts: train and test. The test dataset has variables that were not seen during training. A surrogate model was trained on the test set to predict the variables not seen in train dataset. This surrogate model was then used to estimate these variables, that were not present in the train dataset. A model is trained to predict the target using two different set of features. First model, uses only those features that are originally present in the train dataset. The second model uses the estimated features along with the originally present features.

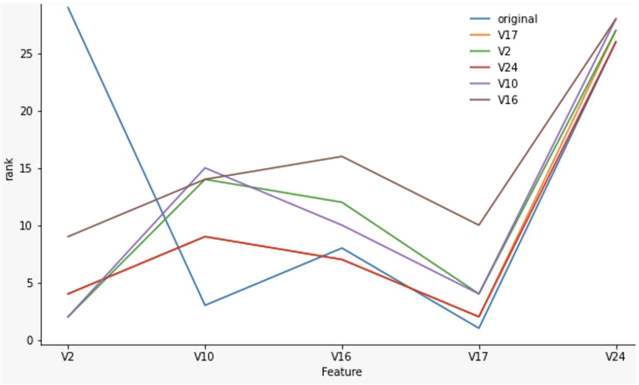

Figure 5: Illustration for relative change in feature importance rank after feature estimation for Credit Card dataset

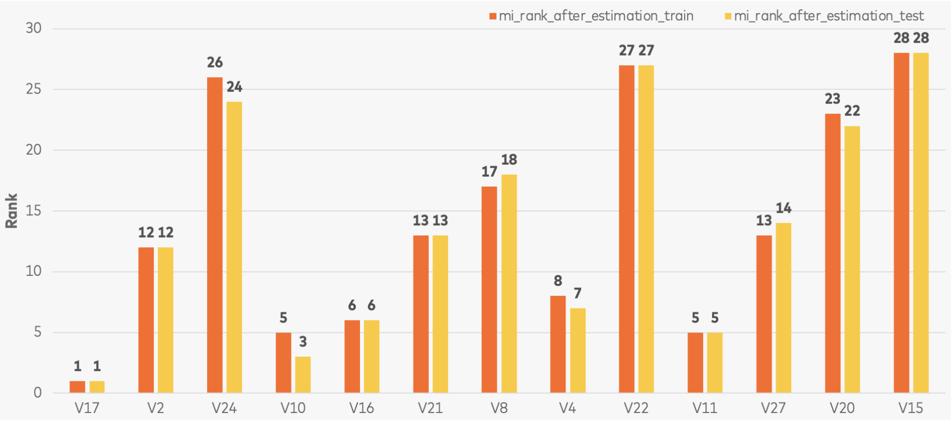

Figure 6: Illustration for mutual information rank of estimated variable with the target

4.4 Results

In Credit Card dataset as shown in Fig. 2 (top left), we observe that AUC-PR (area under the precision-recall curve) of the model using the estimated variable increases compared to the model utlising only the originally present features in the train dataset. The features "V17" and "V2" show the greatest performance lift on estimation. In Elliptic Bitcoin dataset Fig. 2 (bottom left), we observe that the features "local_feature_47", "local_feature_46" and "local_feature_41" show greatest improvement on estimation. Similarly, Airline Satisfaction dataset (Fig. 2 (top right)), shows performance improvement for many variables on estimation using the surrogate model. Variables "online boarding" and "inflight wifi service" show the greatest improvement. It is also observed that some variables show poor performance on estimation. For example, "inflight entertainment" and "leg room service" variables show inferior performance compared to the model without using these estimated variables. For Brilliant Diamonds dataset Fig. 2 (bottom right), variables "type" and "cut" show the greatest improvement in terms of RMSE (root mean squared error). The variable "color" shows poor performance on estimation. In Fig. 3, unlabelled data is used to train the surrogate model for variable estimation. It is observed that the surrogate model using only labelled data performs better than the model using both labelled and unlabelled data for variable estimation. In Fig. 4, different model types are used for variable estimation, namely: classifier and regressor. There is no clear distinction in terms of model performance while using these two types of models.

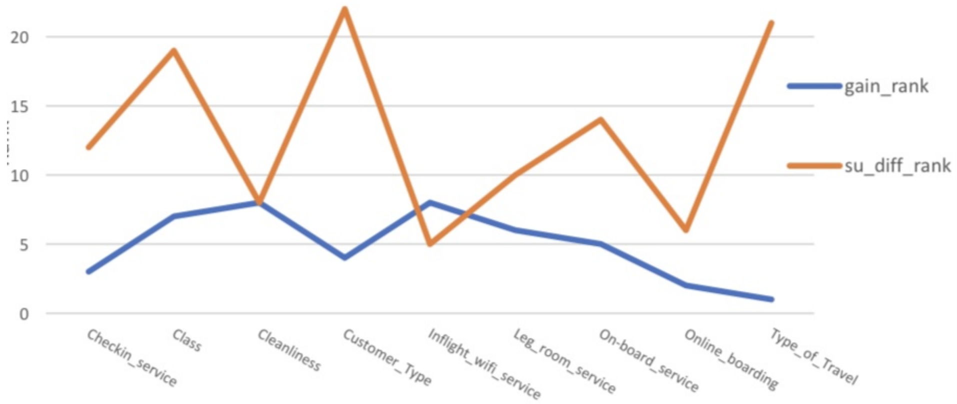

Figure 7: Comparison of gain observed on estimating feature vs symmetrical uncertainty of feature with target for Airline Satisfaction dataset

5 Analysis of Results

5.1 Relationship between the estimated and original variables

In this analysis we explore the relationship between the estimated variable and the original variable. We use mutual information [13] as a criterion to evaluate how much information from the original variable has been captured by the estimated variable.

5.2 Relationship between estimated variable and target variable

In this analysis we explore the relationship between the estimated variable and target variable. This analysis further helps to understand if the relationship exhibited by the estimated and target variable holds in the train as well as the test datasets. In Fig. 6, mutual information rank is observed for variable with the target for the train and test datasets. It is observed that most of the variables show consistent mutual information rank for both the datasets.

5.3 Feature importance of estimated and original variables

In this analysis we first check feature importance of different variables in the dataset where all variables (without estimation) are used for model training. After this, each of the variables is estimated using the surrogate model. The estimated variable is utilized instead of the original variable, and the change in feature importance rank of the estimated variable is observed. In Fig. 5, relative change in feature importance rank is observed for the Credit Card dataset. Some variables like "V16" show immense change in the feature importance rank after variable estimation using the surrogate model.

5.4 Symmetrical uncertainty

Following from the work done by [28], a feature is good if it is relevant with respect to the target and is not redundant with respect to the other relevant features.

If a feature is relevant enough with the target, then even if it is correlated with other features, it would be regarded as a good feature for the prediction task.

Furthermore, [28] states that information gain is biased towards features with more values. Values should be normalized to ensure that they are comparable and have similar effect. Hence, symmetrical uncertainty is used.

$$SU(X,Y) = 2\frac{IG(X|Y)}{H(X)+H(Y)}$$

We undertake an experiment where we check performance gain observed on estimating a feature (denoted by "gain_rank" in Fig. 7). Further, first Symmetrical Uncertainty of estimated feature with the target variable is calculated. Second, Symmetrical Uncertainty of estimated feature with other features is calculated. The absolute difference between these two is ranked.(denoted by "su_diff_rank" in 7)

Performance gain and symmetrical uncertainty difference with respect to the estimated variable is analyzed to establish the hypothesis stated in [28].

In Fig. 7, we compare the performance gain observed when a feature is estimated with the maximum difference of Symmetrical Uncertainty of the feature with other features and the Symmetrical Uncertainty of the feature with the target. We try to corroborate if features showing high performance gain on estimation have higher Symmetrical Uncertainty difference as well. On the Airline Satisfaction dataset, we observe that this relationship holds for some of the variables like "Class" and "on board service", while it does not hold for some variables like "Customer type" and "Type of travel".

6 Conclusion

Real world AI models are trained on a set of features acquired in a fixed timeline. These models are then used for testing/evaluation in the real world in a different time period that was used for training. After a certain period of time, new features/variables may be available for inferencing in addition to the existing features that were utilized for training the model. We propose a simple technique to allow us to incorporate these features in the trained model, so that we may be able to train a model which yeilds better performance on the downstream task on the test set.

References

[1] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113, 2016.

[2] Supriya Agrahari and Anil Kumar Singh. Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10):9523–9540, 2022.

[3] Reid A. Johnson Andrea Dal Pozzolo, Olivier Caelen and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. in symposium on computational intelligence and data mining (cidm). 2015.

[4] Ruoyi Du, Dongliang Chang, Kongming Liang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. On-the-fly category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11691–11700, 2023.

[5] Giulia Boato† Francesco Marra, Cristiano Saltori† and Luisa Verdoliva. Incremental learning for the detection and classification of gan-generated images, 2019.

[6] Sheng-Uei Guan and Shanchun Li. Incremental learning with respect to new incoming input attributes. *Neural Processing Letters*, 14:241–260, 12 2001.

[7] Mehadi Hassen and Philip K Chan. Learning a neural-network-based representation for open set recognition. In *Proceedings of the SIAM International Conference on Data Mining*, pages 154–162, 2020.

[8] Jun Huang, Yu Yan, Xiao Zheng, Xiwen Qu, and Xudong Hong. Discovering unknown labels for multi-label image classification. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 797–806, 2022.

[9] Yi Huang, Ying Li, Guillaume Jourjon, Suranga Seneviratne, Kanchana Thilakarathna, Adriel Cheng, Darren Webb, and Richard Yi Da Xu. Calibrated reconstruction based adversarial autoencoder model for novelty detection. *Pattern Recognition Letters*, 169:50–57, 2023.

[10] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273, 2015.

[11] Prem Junsawang, Suphakant Phimoltares, and Chidchanok Lursinsap. Streaming chunk incremental learning for class-wise data stream classification with fast learning speed and low structural complexity. *PLOS ONE*, 14:e0220624, 09 2019.

[12] Pichai Kankuekul, Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Online incremental attribute-based zero-shot learning. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3657–3664, 2012.

[13] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, Jun 2004.

[14] Sanjay Kumar, Nadira Ahmadi, and Reshma Rastogi. Multi-label learning with missing labels using sparse global structure for label-specific features. *Applied Intelligence*, pages 1–16, 2023.

[15] Aurel A Lazar, Weiguo Wang, and Robert H Deng. Models and algorithms for network fault detection and identification: A review. *[Proceedings] Singapore ICCS/ISITA92*, pages 999–1003, 1992.

[16] Miguel Corral Jr. Brilliant diamonds, 2020. `https://www.kaggle.com/datasets/miguelcorraljr/brilliant-diamonds`,.

[17] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision*, pages 613–628, 2018.

[18] Debabrata Pal, Shirsha Bose, Biplab Banerjee, and Yogananda Jeppu. Morgan: Meta-learning-based few-shot open-set recognition via generative adversarial network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6295–6304, 2023.

[19] Debabrata Pal, Valay Bundele, Renuka Sharma, Biplab Banerjee, and Yogananda Jeppu. Few-shot open-set recognition of hyperspectral images with outlier calibration network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3801–3810, 2022.

[20] Zhiting Song, Yanming Sun, Jiafu Wan, Lingli Huang, and Jianhua Zhu. Smart e-commerce systems: current status and research challenges. *Electronic Markets*, 29:221–238, 2019.

[21] TJ Klein. Airline passenger satisfaction, 2020. `https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction`,.

[22] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501, 2022.

[23] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.

[24] Qitian Wu, Chenxiao Yang, and Junchi Yan. Towards open-world feature extrapolation: An inductive graph learning approach. *Advances in Neural Information Processing Systems*, 34:19435–19447, 2021.

[25] Jiawen Xu, Matthias Kovatsch, and Sergio Lucia. Open set recognition for machinery fault diagnosis. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–7, 2021.

[26] Muli Yang, Liancheng Wang, Cheng Deng, and Hanwang Zhang. Bootstrap your own prior: Towards distribution-agnostic novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3459–3468, 2023.

[27] Tim Kraska Yeounoh Chung, Peter J. Haas and Eli Upfal. Learning unknown examples for ml model generalization, 2019.

[28] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.

[29] Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. Exploratory machine learning with unknown unknowns, 2020.

[30] Jiyang Zheng, Weihao Li, Jie Hong, Lars Petersson, and Nick Barnes. Towards open-set object detection and discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2022.

[31] Da-Wei Zhou, Yang Yang, and De-Chuan Zhan. Learning to classify with incremental new class. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2429–2443, 2022.