IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 19, 2021, accepted August 3, 2021, date of publication August 12, 2021, date of current version August 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104472

Transfer Learning Strategies for Credit Card Fraud Detection

BERTRAND LEBICHOT[1], THÉO VERHELST[1], YANN-AËL LE BORGNE[1], LIYUN HE-GUELTON[2], FRÉDÉRIC OBLÉ[2], AND GIANLUCA BONTEMPI[1], (Senior Member, IEEE)
[1]Machine Learning Group, Computer Science Department, Faculty of Sciences, Université Libre de Bruxelles (ULB), 1050 Brussels, Belgium
[2]Performance and Intelligence Research and Development Department, Worldline, 69100 Lyon, France

Corresponding author: Bertrand Lebichot (bertrand.lebichot@ulb.be)

This work was supported by Innoviris through the TeamUp DefeatFraud Project under Grant 2017-R-49a. The work of Bertrand Lebichot was supported by the LouRIM, Université catholique de Louvain, Belgium.

ABSTRACT Credit card fraud jeopardizes the trust of customers in e-commerce transactions. This led in recent years to major advances in the design of automatic Fraud Detection Systems (FDS) able to detect fraudulent transactions with short reaction time and high precision. Nevertheless, the heterogeneous nature of the fraud behavior makes it difficult to tailor existing systems to different contexts (e.g. new payment systems, different countries and/or population segments). Given the high cost (research, prototype development, and implementation in production) of designing data-driven FDSs, it is crucial for transactional companies to define procedures able to adapt existing pipelines to new challenges. From an AI/machine learning perspective, this is known as the problem of *transfer learning*. This paper discusses the design and implementation of transfer learning approaches for e-commerce credit card fraud detection and their assessment in a real setting. The case study, based on a six-month dataset (more than 200 million e-commerce transactions) provided by the industrial partner, relates to the transfer of detection models developed for a European country to another country. In particular, we present and discuss 15 transfer learning techniques (ranging from naive baselines to state-of-the-art and new approaches), making a critical and quantitative comparison in terms of precision for different transfer scenarios. Our contributions are twofold: (i) we show that the accuracy of many transfer methods is strongly dependent on the number of labeled samples in the target domain and (ii) we propose an ensemble solution to this problem based on self-supervised and semi-supervised domain adaptation classifiers. The thorough experimental assessment shows that this solution is both highly accurate and hardly sensitive to the number of labeled samples.

INDEX TERMS Fraud detection, domain adaptation, transfer learning.

I. INTRODUCTION

Global card fraud losses amounted to 24.3 billion US dollars in 2017 and are foreseen to continue to grow to more than 34 billion by 2022 [1]. In recent years, AI/machine learning techniques played a significant role in automatic detection solutions to deal with massive amounts of transactions [2]. State-of-the-art work showed that an effective detection strategy needs to take into account the peculiarities of the fraud phenomenon [3], [4]: unbalancedness (frauds are less than 1% of all transactions), concept drift (typically due to seasonal aspects and evolving fraudster strategies), high overlap between fraudulent and non-fraudulent transactions [5], and the big data and streaming nature [6] of the problem. Disregarding those aspects might lead to a high false alert rate, low detection accuracy, or slow detection (see [7] for more details).

As a result, the design of an accurate Fraud Detection System (FDS) goes beyond the integration of some standard off-the-shelf learning libraries and requires a deep understanding of the fraud context. It follows that the reuse of existing FDS in new settings, like a new market or a new payment system, is neither immediate nor straightforward. For this reason, strategies allowing to adapt battle-proofed FDS to new markets or systems can be an asset for credit card issuing companies.

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

In literature there are mainly three approaches to address this problem:

- Re-use previous models: this is the simplest solution but reusing existing models may exhibit poor performance if the target domain is far from the original one (e.g. concept drift). Also, quantifying the distance between domains is not trivial.
- Train new models from scratch: due to the lack of data from the target domain, this solution can be impractical or very costly in terms of research, development, and implementation.
- Adapt existing models: this approach is known as transfer learning. Depending on which assumptions can be made about the problem, different sub-fields of transfer learning can be used. These are presented in Section II.

This paper discusses the role of transfer learning strategies [8] in the adaptation of existing detection models to new domains. In particular, we focus on the heterogeneous nature of e-commerce credit-card transactions, related to the different behavior in different countries (see Section V-B for more details). In a previous work [9], the authors studied how to transfer a detection model trained on e-commerce data to the face-to-face (F2F) setting.[1] Here the challenge is to address the shift due to the different behaviors of both fraudsters and genuine users in two neighboring countries.

It is crucial for the card issuer company to define which portion of the accurate detection system, carefully tuned for a specific market, can be reused and transferred to another one. Note that the case study is based on a real business need of our industrial partner, leading issuer in Belgium, which recently opened new market lines in other countries.

The paper reviews the topic of transfer learning, presents state-of-the-art and original approaches, and assesses them in the fraud detection case. The main original contributions of the paper are:

- I. a comprehensive literature review of domain adaptation (Section III);
- II. a comparison of the three main domain adaptation approaches in the fraud detection context: self-supervised, semi-supervised, and supervised (Section IV);
- III. the proposal of a novel approach based on the combination of two settings: a self-supervised generative naive Bayesian classifier and a non-linearly normalized (to adapt to the target distribution) classifier (Section V). This approach is both highly accurate and hardly sensitive to variations in the number of labeled samples;
- IV. an extensive comparison and assessment of 15 transfer learning approaches in a real case study based on a six-month dataset (more than 200 million e-commerce transactions) provided by the industrial partner (Section V).

[1] Face-to-face transactions occur when the buyer and merchant physically meet to complete a purchase. In e-commerce, transactions can take place when the cardholder is not physically with the merchant (e.g. exchange of goods or services through a computer network, like the internet).

An important aspect of this paper is the novel application domain of transfer learning, typically restricted so far to natural language processing and image recognition tasks [8]. To the best of our knowledge, only one other work [10], and our preliminary study [9], addressed transfer learning techniques in credit card fraud detection.

To summarize, our work is one of the first attempts to apply transfer learning to fraud detection. We present a wider comparison (Contribution II and IV) and propose an accurate and robust model in this context (Contribution III).

The rest of this paper is structured as follows: Section II introduces background and notation. Section III reviews related work. Section IV details the methodological contributions of the paper. Experimental comparisons are presented and analyzed in Section V while Section VI discusses these results. Finally, Section VII concludes the paper with some future perspectives.

II. BACKGROUND AND NOTATION

AI and *Machine learning* often rely on human intelligent behavior as a source of inspiration for their strategies (e.g. neural networks, semi-supervised learning, ...). The idea of *Transfer learning* (TL) originates from the consideration that humans take advantage of previously learned skills (e.g. recognize apples or playing the piano) to speed up the learning of somewhat related tasks (recognize pears or playing the organ) [8]. The rationale of transfer learning is to fill the gap between two supervised learning tasks by reusing what was learned from the former (called *source*) to better address the latter (referred to as *target*).

Let us now introduce in more formal terms the notions of domain and task [8], [11]:

- A *domain* D is a tuple $(X, P(X))$, where X denotes the multivariate input of size m and $P(X)$ its marginal probability distribution.
- Given a domain, a *task* T is defined as a tuple $(y, P(y|X))$ where y is the labeled output and $f(x) = P(y|X = x)$ the conditional distribution which formalizes the dependency between inputs and output.

Given a source domain D_s, a learning task T_s, a target domain D_t and a learning task T_t, transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ by using knowledge about D_s and T_s where $D_s \neq D_t$ or $T_s \neq T_t$.

Depending on the assumptions that can be made on D_s, T_s, D_t, and T_t, different sub-fields of TL can be considered. Here, we will limit ourselves to consider the one that matches our case study. For a complete overview of TL, we refer the reader to [11].

Domain adaptation (DA) (also called transductive transfer learning [11]) is a sub-field of transfer learning. In this case, there is a change in the domain $D_s \neq D_t$ but the task is supposed to remain the same $T_s = T_t$. The change between D_s and D_t may occur due to $X_s \neq X_t$ or $P(X_s) \neq P(X_t)$. If the source and target feature sets differ (i.e. $X_s \neq X_t$) the domain adaptation is called *heterogeneous*, otherwise *homogenous*.

TABLE 1. The amount of labeled data in the target domain leads to three different settings [11]. Notice that the problem is always fully supervised in the source domain. [11] originally refer to self-supervised DA as unsupervised DA. We choose to use this name instead, as unsupervised can be misleading: the model will still receive some form of feedback, at least from source labels.

Domain Adaptation	Source labels	Target labels	Setting
Fully supervised DA	Yes	Yes	Transductive
Semi-supervised DA	Yes	Partially	Transductive
Self-supervised DA	Yes	None	Transductive

In this paper, we consider a homogeneous domain adaptation where the feature sets for source and target are the same, the task is the same (fraud detection), but there is a change in the domain distribution $P(X_s) \neq P(X_t)$.

In the following, by *source* we denote the original domain of credit card transactions from the source country (see Section I), while *target* refers to the new domain of transactions from the target country. The actual name of the source and target domain cannot be revealed for confidentiality reasons.

Source data are assumed to be completely labeled since they are collected in a fully-known production line with sufficient historical feedback. The target data, however, is only partially labeled: this means that we have only limited information about the new production line (e.g. in a start-up phase). This case is known as semi-supervised domain adaptation and it is one of the three most common configurations (see Table 1 inspired from [11]). In our previous work [9], we discussed the fully supervised case only. Here we extend this work by analyzing the three sub-cases and assessing the impact of the labeling target rate on the precision (Section V).

Homogeneous transfer learning has typically recourse to one of the following strategies [12]: (i) adapt both the marginal and conditional distributions in the source domain, (ii) focus on the marginal distribution only (e.g. by normalization), or (iii) focus on the conditional distribution only. Note that the first approach is recommended in settings where the mechanism behind the conditional and marginal distributions is not independent. This is typically the case of fraud detection where the link between inputs (e.g. transaction features) and output (fraud/genuine) is *anticausal* [13], i.e. we aim to predict the causes from the effects. As shown in recent works in literature [14], in an anticausal scenario the changes in the marginal and conditional distribution are related, i.e. the change of $P(X)$ tells us something about the change of $P(y|X)$. Those results encouraged the adoption of semi-supervised approaches in our specific application setting.

III. RELATED WORK

There are five main DA approaches in the literature according to the most cited reviews in literature [8], [11], [12]. We detail them below by taking into consideration that our detection problem is homogeneous and that a subset of labels in the target domain is available.

- Instance-based: its rationale is that the gap between source and target tasks may be reduced by an appropriate adaptation (e.g. by weighting samples or adding labels) of the training set. Three versions are available in the literature: instance weighting, cluster-based and, self-labeling approaches.
  Instance weighting focuses on the re-weighting of the source domain instances to correct for marginal distributions discrepancy. The re-weighted instances are then directly used in the target domain for training. This approach, inspired by importance sampling [11], works best when the conditional distribution is the same (or at least very similar) in both domains [12]. Examples in literature are kernel-mean matching (KKM) [15], nearest neighbor-based importance weighting [16] and, KL importance estimation procedure (KLIEP) [17].
  Cluster-based approaches construct a graph or clusters where the labeled and unlabeled samples are nodes and the edge weights are based on their similarity. Labels are then propagated according to the graphs (e.g. using graph-based classification) [8]. The main assumption is that samples connected by high-density paths are likely to have the same labels [18]. Examples of this approach are locally weighted ensemble (LWE) [18] and topic-bridged PLSA [19]. Those methods may be highly computationally intensive, especially when working with large graphs.
  Self-labeling methods include unlabeled target domain samples in the training process, initialize their labels, and then iteratively refine them. This is often done using Expectation-Maximisation (EM) algorithms (for example TrAdaBoost [20]). Hard versions add samples with specific labels while others [21] assign label confidences when fitting the model. A self-supervised approach based on Fourier transform and Wavelet transform is presented in [22].
- Feature representation: these methods aim to find a new feature representation and belong to two main categories: distribution similarity and latent approaches.
  Distribution similarity approaches aim to make the source and target domain sample distributions similar, either by penalizing/removing features whose statistics vary between domains or by learning a feature space projection in which a distribution divergence statistic is minimized [18], [23]. This strategy has been applied to the task of fraud detection in [10].
  Latent feature approaches construct new features using source and target domain data or, more in general, define a new feature space [18], [24].
  Weiss *et al.* [12] also distinguish between asymmetric and symmetric feature transformation. The asymmetric case transforms the source features by re-weighting them to match the target domain (e.g. [25]). The symmetric case discovers underlying meaningful structures

TABLE 2. This table summarises the considered methods of this paper. More details about the strategy and parameters can be found in Section IV. Run time is an indicative execution time to transfer the data, train the classifier, and label the 82 days of data (see Section VI for details). The DA class description can be found in Section III. DA configuration are resumed in Table 1 and in Section II. Parameters tuned during validation are indicated in the last column, and the value most frequently selected is underlined. Notice that the considered test set is always composed of the target domain only.

Acronym	DA class	Strategy	Train	DA configuration	Run time	Parameters
G-NB	Feat. repr.	Generative naive Bayes	s	Self-supervised	9.21 s	-
N-DNN	Baseline	Use source labels only	s	Self-supervised	128.0 s	-
B-DNN	Baseline	Use target labels only	t	Supervised	91.2 s	-
C-DNN	Baseline	Simple concatenation	s+t	Supervised	194.3 s	-
FEDA-DNN	Feat. repr.	Imputation	s+t	Supervised	258.8 s	-
Aug-DNN	Feat. repr.	Add source-related features	s+t	Semi-supervised	533.3 s	$n_{PC} = [1, 2, \underline{5}]$
Adv-DNN	Feat. repr.	Adversarial	s+t	Semi-supervised	219.2 s	$\lambda = [0.01, \underline{0.1}, 1, 10, 100]$
G-NB&DNN	Combined	Unsup. and sup. DA ensemble	s+t	Semi-supervised	211.8 s	$\alpha = [\underline{0.1}, 0.3, 0.5, 0.7, 0.9]$
N-EE	Baseline	Use source labels only	s	Self-supervised	78.3 s	-
B-EE	Baseline	Use target labels only	t	Supervised	56.2 s	-
C-EE	Baseline	Simple concatenation	s+t	Supervised	82.0 s	-
FEDA-EE	Feat. repr	Imputation	s+t	Supervised	105.9 s	-
Aug-EE	Feat. repr	Add source-related features	s+t	Semi-supervised	179.6 s	$n_{PC} = [1, \underline{2}, 5]$
TrAB-EE	Inst. based	Adaptative boosting	s+t	Semi-supervised	586.7 s	-
G-NB&EE	Combined	Unsup. and sup. DA ensemble	s+t	Semi-supervised	95.8 s	$\alpha = [0.1, 0.3, 0.5, 0.7, \underline{0.9}]$

between domains to find a common latent feature space that has better predictive qualities and reduces the marginal distribution gap between domains. Examples of this approach are frustratingly easy domain adaptation (FEDA) [24], transfer component analysis (TCA) [25], and Domain-adversarial training [23].

- Parameter-based: they assume that the source and the target domains share some parameters or prior distributions of the hyper-parameters of the models [11]. Knowledge is transferred through shared parameters (or priors) of the source and target learners [12]. For example, a learner on the target domain can be regularised according to a cost function measuring the difference with the source parameters [26]. It is also common to use an ensemble version of this approach: create multiple source learner models and combine the re-weighted learners to form an improved target learner. Examples of this approach are form-free Gaussian process [27], task-coupling SVM [28], and Neural Network Adaptation [26].
- Relational-based: the basic assumption is that some relationships among the data in the source and target domains are similar [11]. Thus, the knowledge to be transferred is the relationship between the data. This approach, while promising on datasets with thousands of samples, is not suitable for our volume of data (millions of transactions). Instances of this approach are deep transfer via Markov logic [29] and SR2LR [30].
- Deep neural network methods (DNN): DNNs have been widely used for TL and DA since their multi-layer nature can capture the intricate non-linear representations of data, and provide useful level features for transfer learning [8]. Multitask learning [31] can be as well implemented by DNN, by training two or more related tasks with a network sharing inputs and hidden layers but having separate output layers. As far as domain adaptation is concerned, hidden layers trained by the source task can be reused on a different target task. For the target task model, only the last classification layer needs to be retrained, though any layer of the new model could be fine-tuned if needed [8]. In other configurations, the hidden parameters related to the source task can be used to initialize the target model [32]. Autoencoders can also be used to gradually change the training distribution. In [33], a sequence of deep autoencoders are trained layer-by-layer, while gradually replacing source-domain samples with target-domain samples. In [34], the authors simply train a single deep autoencoder for both domains. Finally, [23] used DNN in an adversarial way to tackle domain adaptation. We will discuss more extensively this approach in Section IV.

IV. TRANSFER LEARNING STRATEGIES FOR FRAUD DETECTION

This section introduces the set of transfer learning strategies that we designed and implemented for our fraud detection case study (Table 2). Before detailing them, however, we mention two important components of all discussed methods, i.e. the classifier used for the supervised approaches and the related normalization strategies.

A. SUPERVISED CLASSIFIERS

To better assess the impact of the transfer strategy on the final result, we consider two different base classifiers: Random Forests and DNN. The reason for our choice is that Random Forests (RF) showed good performance in several works on FDS [2], [35] while DNNs have been widely used for TL and DA, as discussed in Section III. The domain adaptation configuration corresponding to each strategy is reported in Table 2.

To mitigate the effect of an unbalanced ratio between genuine and fraudulent transactions we consider the Easy Ensemble (EE) approach [36] based on Random Forests.

Following a similar idea, we use random under-sampling to re-balance the two classes for the DNN methods.

Note that to avoid bias related to the classifier structure, all DNN strategies share the same topology composed of two fully connected hidden layers. Based on preliminary results (not reported here), we set the number of neurons in the hidden layers to 1.5 times the number of input features. For EE methods, the number of RFs in the ensemble and the number of trees in each RF are the same for all approaches.

B. FEATURE NORMALIZATION

We refer to *normalization* as a nonlinear monotonous transformation of the values of a continuous random variable X, such that the cumulative distribution function (CDF) of X after transformation matches a given CDF F. We consider here only the univariate case where each feature is normalized independently of the others. First, we compute the value of the empirical CDF of X (noted $\hat{F}$) at each observed value $x_i, i = 1, \ldots, n$. If all values x_i are sorted in ascending order (ties are allowed) the empirical estimation is $\hat{F}(x_i) = (i-1)/(n-1)$. The transformed value x'_i is then chosen such that $F(x'_i) = \hat{F}(x_i)$. In the context of TL, the normalization process is performed separately on the source and target domain data. We denote source examples by $x_i^{(s)}, i = 1, \ldots, n^{(s)}$ and target examples by $x_j^{(t)}, j = 1, \ldots, n^{(t)}$, with $n^{(s)}$ and $n^{(t)}$ respectively the number of source and target examples. We also note the value of the empirical CDF as $p_i^{(s)} = \hat{F}^{(s)}(x_i^{(s)})$ and $p_j^{(t)} = \hat{F}^{(t)}(x_j^{(t)})$. We consider two different feature normalizations (note that methods with no normalization are denoted by a subscript n).

- *normalization to a standard uniform distribution:* In this case, the values of X can be transformed to a standard uniform distribution in $[0, 1]$ simply by choosing $x'_i = p_i$, since $F(x) = x, \ \forall x \in [0, 1]$, for this particular CDF. This normalization is denoted by a subscript u in the method acronym.
- *normalization to the target domain:* the source examples are transformed to a CDF that matches the empirical CDF $\hat{F}^{(t)}$ of the target examples. The target examples are left unmodified. For each source example $x_i^{(s)}$ and the corresponding empirical CDF value $p_i^{(s)}$, we find the two consecutive empirical CDF values $p_{j_1}^{(t)}$ and $p_{j_2}^{(t)}$ framing $p_i^{(s)}$ in the target domain:

$$p_{j_1}^{(t)} \leq p_i^{(s)} < p_{j_2}^{(t)}$$

with $j_1 + 1 = j_2$. The value of $x_i^{(s)\prime}$ is then computed as the linear interpolation between the values $x_{j_1}^{(t)}$ and $x_{j_2}^{(t)}$:

$$x_i^{(s)\prime} = (1-\lambda)x_{j_1}^{(t)} + \lambda x_{j_2}^{(t)} \quad \text{with } \lambda = \frac{p_i^{(s)} - p_{j_1}^{(t)}}{p_{j_2}^{(t)} - p_{j_1}^{(t)}}.$$

This normalization is denoted by a subscript t in the method acronym.

Note that the normalization can be considered as a very simple example of feature representation strategy (Section III), symmetric in the standard case and asymmetric in the target case, respectively.

C. TRANSFER LEARNING STRATEGIES

Overall we consider 15 methods which are detailed below. When the "DNN/EE" string appears in the acronym describing the strategy, this means that the strategy has been implemented with both the Easy Ensemble Random Forest (EE) and DNN classifiers. Note that though the first three are very simple baselines, they are important to assess the added value of more complex strategies.

- *B-(DNN/EE):* this is the baseline "no-transfer" classifier (see Section IV) where the training dataset is composed of the labeled target samples only.
- *N-(DNN/EE):* this is the naive strategy which consists in training the classifier (DNN or EE) on the source dataset and test it on the target test set. This approach is also often considered in the literature as a baseline [24] to assess the added value of a transfer learning strategy.
- *C-(DNN/EE):* this approach uses both source and target data in the training phase by adding a binary feature which plays the role of flag indicating the domain of the data sample. This approach is probably the simplest conceivable supervised DA strategy.

The list of non-baseline methods is:

- *FEDA-(DNN/EE):* FEDA (Frustratingly Easy Domain Adaptation) is a basic feature representation method (Section III) which combines three versions of the original feature set: a general version, a source-specific version, and a target-specific version [24]. Each source column-feature X_s is replaced by $\phi^s(X_s) = \langle X_s, X_s, \mathbf{0} \rangle$ and each target column-feature X_t is replaced by $\phi^t(X_t) = \langle X_t, \mathbf{0}, X_t \rangle$, where $\mathbf{0}$ is a zero vector and $\phi^s(X_s)$ ($\phi^s(X_t)$) is the source (target) mapping.
  This strategy boils down to represent both domains in an extended feature space, where missing values are imputed with a null value. The augmented source data therefore contains only general and source-specific versions while the augmented target data contains both general and target-specific versions. Finally, ϕ^t is used to obtain the test set from the original target data.
- *Aug-(DNN/EE):* this is an original technique, first presented in [9], which uses information from the source domain (e.g. conditional distribution, marginal input distribution) to add potentially informative features. This strategy allows the classifier to learn from data how the *relatedness* [37] between source and target samples is associated with the classification output. Since the relatedness is not explicitly available but can only be estimated, the strategy estimates both the conditional (e.g. returned by a classifier) and marginal distribution from the source dataset (see Figure 1(a) for an illustration). Those quantities are then computed for both the source and target samples and integrated as additional variables to the original dataset. Here are more details about the two steps in our specific case:

-- We train a classifier (e.g. DNN or EE) on the source dataset. The classifier is then used to return an estimate of the conditional probability for each source and target sample. These values are used to create an additional feature *Pred1* to augment the original dataset.
-- We build a principal component analysis (e.g. PCA [38]) on the source training set. The projections of all (source and target) transactions on the first PCs return several additional variables (denoted *Pca*) to augment the original dataset.

As a result, we augment the original dataset with several new features: *Pred1*, and $Pca_1, Pca_2, \ldots$. The number of PCA components, n_{PC}, is tuned during validation. The expectation is that such features could encode in the training set the relatedness between the source and target distributions, both from a marginal and conditionally dependent perspective. Note that a similar idea has been discussed in [24] where a binary predicted value is used instead.

- *Adv-DNN:* this is an adaptation of the approach [23] to the fraud detection setting. The rationale is that the prediction model should use features that cannot discriminate between the source and target domains. The original approach has been proposed by Ganin for image recognition and combines a labeled source domain and an unlabeled target domain. In our case, both source and target are (at least partially) labeled. The method learns domain invariant features by jointly optimizing the feature layer from the label predictor (discriminating genuine versus fraudulent) and the domain label (discriminating source versus target) predictor. The domain classifier uses a gradient reversal layer (GRL) and a few fully connected layers [23]. The effect of the GRL is to multiply all domain-related gradients by a negative constant λ during back-propagation.
  During the training, the feature layer is optimized to minimize the label classifier loss and maximize (thanks to GRL) the domain classifier loss at the same time. The hyperparameter λ, to be tuned by validation, weights the contribution of the two terms. This approach promotes the emergence of features that are discriminative for the main learning task on the source domain and non-discriminative with respect to the domain tag [23]. A network illustration can be found in Figure 1(b). As this approach is based on a DNN architecture, the EE version was not considered in this case.
- *TrAB-EE:* this strategy implements TrAdaBoost [20], a technique which uses a small amount of target domain labeled data to leverage the source domain data and return a high-quality classification model for the target domain. This is implemented as a boosting algorithm where, at each iteration, a re-weight of the source domain is computed by using the error calculated on the target data. A classifier is then trained using the instance-weighted source data and target data. This technique has the advantage of being at the same time self-labeling and instance-based (see Section III). As in the original paper, only the EE classifier is included as a base learner. The DNN classifier was initially considered as well, but it failed to converge after only a few iterations of the boosting procedure, leading to poor performance. Note that the original approach described in [20] demanded a specific adaption to our highly imbalanced dataset. We excluded from the computation of the training error the genuine transactions that were not used by the EasyEnsemble classifier in the current iteration. Since the EasyEnsemble selects a balanced subset of the training set, this modification makes the training error more balanced as well.
- *G-NB:* this approach implements a Naive Bayes classifier trained exclusively on the source domain. We take advantage of the abundant labeled data present in the source domain to estimate the conditional probabilities in the source domain and the priors. This leads to

$$\hat{P}(y=1|X) = \frac{P_s(X|y=1)\,P_s(y=1)}{P_s(X|y=0)\,P_s(y=0) + P_s(X|y=1)\,P_s(y=1)}.$$

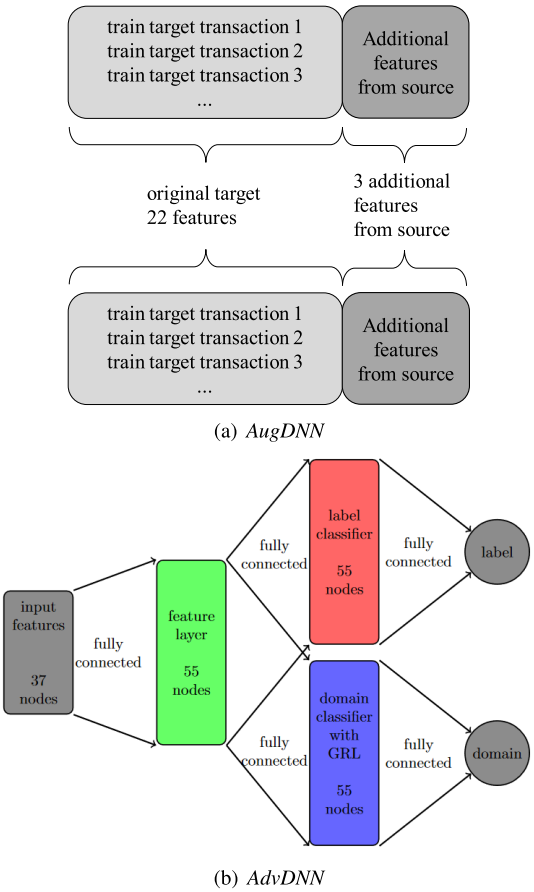


FIGURE 1. Illustration of methods *AugDNN* and *AdvDNN*. For (b), notice that removing the domain classifier reduces the network to the BDNN and NDNN baselines.

Assuming that all variables in X are conditionally independent, each univariate density function is estimated using univariate histograms. After a search on the range of the number of bins, we found that a histogram with 100 equally-spaced bins works best.

Note that this is an example of generative classifier (i.e. fitting first the dependency $Y \rightarrow X$ to estimate $P(y|X)$) unlike the discriminative ones (i.e. fitting directly the dependency $X \rightarrow Y$) implemented by EE and DNN in the previous strategies [39]. Also, it is an instance of self-supervised DA since it does not require any target samples. Given that by construction the generative classification approach is more compliant with the causal mechanism underlying the transactional data, we expect that this approach should be particularly robust and insensitive to distribution shifts.

We evaluated several alternative models: i) adding known frauds from the target domain into the density estimation of $P(X|y = 1)$, ii) modeling important or statistically dependent variables with joint histograms to avoid the assumption of conditional independence, iii) performing feature selection. In all cases, no significant performance improvement was noted.

- *G-NB*&(DNN/EE): this is an ensemble of the self-supervised approach (*G-NB*) and the supervised approach *C-(DNN/EE)*. The predicted probabilities of fraud from both classifiers are averaged using a weighted arithmetic mean where the weight α (with $\alpha \in [0, 1]$) is tuned by validation. In our experiments we considered five values of α: [0.1, 0.3, 0.5, 0.7, 0.9].

V. EXPERIMENTAL ASSESSMENT

This section describes the experimental assessment procedure and is structured as follows. Section V-A presents the dataset while Section V-B shows some visualizations to provide some visual insight into the nature of the transfer task. Section V-C details the experimental setting while Section V-D presents the experimental results.

A. DATA

The source database is made up of about 143M e-commerce transactions that occurred in the source country during 183 days (91 training, 10 validation days, and 82 test days). The source fraud ratio is 0.13% and each transaction is described by 23 features (in particular, there are no geographical data about cardholders as the country is different for both domains). The target database is composed of about 60M e-commerce transactions from the target country (the same days and features as the source database) with a fraud ratio amounting to 0.21%. All data were standardized per domain, before proceeding to the non-linear normalization step (see Section IV). Validation days are used to tune the hyperparameters introduced in the methodological section and detailed in Table 2. We do not analyze the impact of the training set size on the fraud detection accuracy, since this issue has already been extensively studied in previous works [2].

B. VISUALIZATION OF SOURCE AND TARGET DISTRIBUTIONS

In order to give a flavor of the complexity of our transfer learning task, we present here a low variate visualization of the source and target distributions by means of Principal Component Analysis.

PCA is a well-known unsupervised visualization technique [38]. To give an insight into the two domains of our problem, we show in Figure 2 the two first principal components derived from the most discriminative features. For this purpose, we restrained to consider the most relevant features (notably the ones whose importance computed by Random Forest is in the top third). The plot is made of four subplots representing four possible combinations (fraud/genuine, source/target). Figures 2(a) and 2(b) show that the distribution of fraudulent and genuine transactions is similar in the two countries. This is encouraging if we aim to use transfer learning approaches. However, Figures 2(c) and 2(d) indicate that both tasks are difficult since they are scarcely separable (i.e. the two classes overlap in both domains).

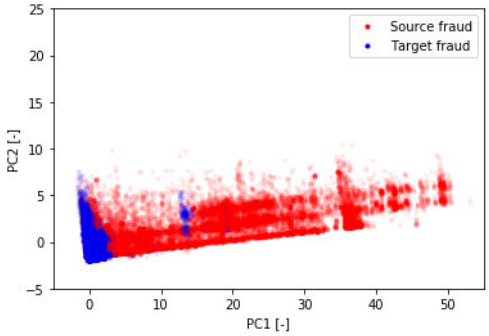
1) SOURCE/TARGET PREDICTION PER CLASS

In this second visualization, we train two random forests with random undersampling for the majority class: one is trained on half of the source data (denoted as source classifier) and the other is trained on half of the target data (denoted as target classifier). We then classify the rest (source and target) of the data and report the histograms of the a-posteriori, for both forests, in Figure 3. Figure 3(a) shows the distribution $P(\hat{Y} = 1|Y = 0)$ for the source classifier and Figure 3(b) shows the distribution $P(\hat{Y} = 1|Y = 0)$ for the target classifier. From these two plots, we can speculate that the origin of the training data (source or target) does have an impact on the classification results.

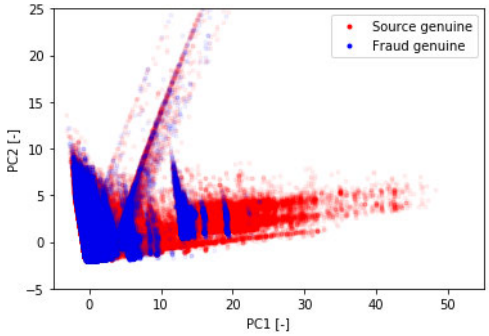
C. EXPERIMENTAL SETTING

In order to proceed to a paired assessment, we split the target dataset into a training and a test portion.

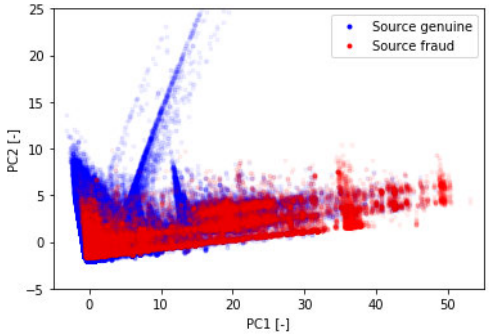
The accuracy is measured in terms of Precision@100 (Pr@100) which represents the number of true compromised cards among the first 100 alerts. The number 100 is chosen since this is compliant with the daily effort of the team of human investigators who manually check the transactions. We include also accuracy results in terms of AUPRC, a recommended alternative to the well-known ROC AUC for unbalanced classification problems [40]. For a detailed justification of the adoption of such measures, we refer the reader to [2], [3], [6], [41]. Note that in literature sometimes transaction-based precision is used instead of card-based precision. Since we obtained similar conclusions for transaction-based detection we limit to present here card-based results.



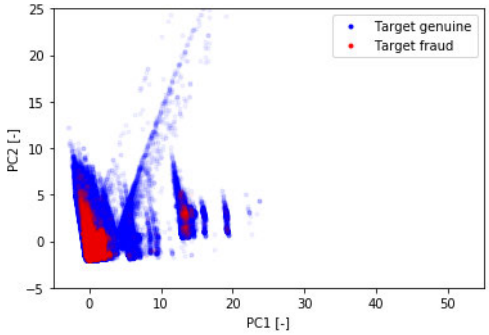
(a) $P_s(X|Y=1)$ vs $P_t(X|Y=1)$



(b) $P_s(X|Y=0)$ vs $P_t(X|Y=0)$



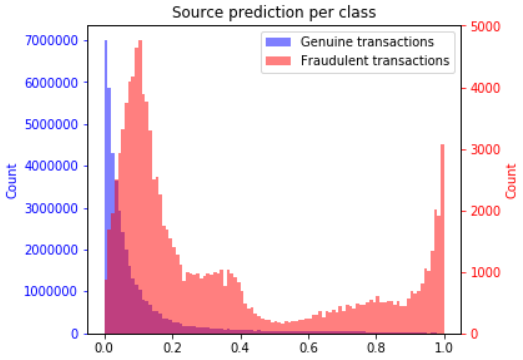
(c) $P_s(X|Y=0)$ vs $P_s(X|Y=1)$



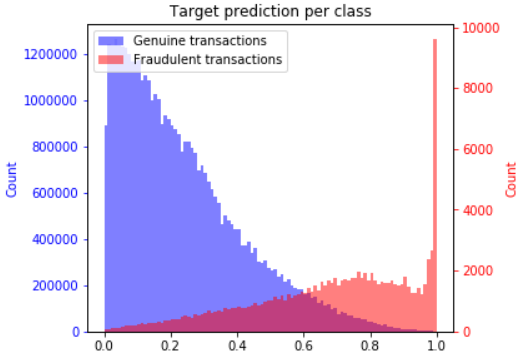
(d) $P_t(X|Y=0)$ vs $P_t(X|Y=1)$

FIGURE 2. **PC1 and PC2 stand for the main and second principal components, respectively. The four subplots show that the fraud detection task is related for both countries (source and target). Also, the task is highly non separable in the PC space since the two classes tend to overlap.**

We denote by r (or simply ratio), the ratio of labeled transactions in the target domain. For our analysis, it will vary



(a) With source classifier



(b) With target classifier

FIGURE 3. **The two subplots show that the origin of the training data (source or target) does have an impact on the classification results. Notice that $P_s(\hat{Y}=1)$ (on the top, in blue) and $P_t(\hat{Y}=1)$ (on the bottom, in blue) somewhere have similar curves: The left part of $P_s(\hat{Y}=1|Y=1)$ is similar to $P_t(\hat{Y}=1|Y=1)$ and the $P_s(\hat{Y}=1|Y=0)$ looks like $P_t(\hat{Y}=1|Y=0)$ but dilated on the x-axis.**

from 1 (fully labeled dataset) to 0.0001 (leaving less than five actual labels). The rest of the transaction is not discarded but is considered genuine instead: the probability to be a fraud is less than $1/1000$ and this is often assumed in real-life settings. This can be viewed as an advantage of working in an unbalanced world.

D. RESULTS

This section aims to provide a quantitative and paired assessment of the 15 methods in several transfer configurations, each characterized by a different ratio r of target labeled transactions. Note that the ratio $r = 0$ corresponds to a self-supervised DA setting (i.e. no labeled target sample) while $r = 1$ denotes the configuration where all the target labeled data are available for learning (and consequently transfer from the source domain is of little use).

Figures 4 and 5 (resp. 6 and 7) report the Pr@100 (resp. AUPRC) accuracy of the DNN and EE-based methods. A boxplot is used to summarize the 82 evaluations (one per test day) of the metric for each method. To avoid variability in the DNN/EE training, each result is the mean of 10 different initializations. The ratio r of labeled transactions is indicated on the x-axis. DA self-supervised strategies are insensitive

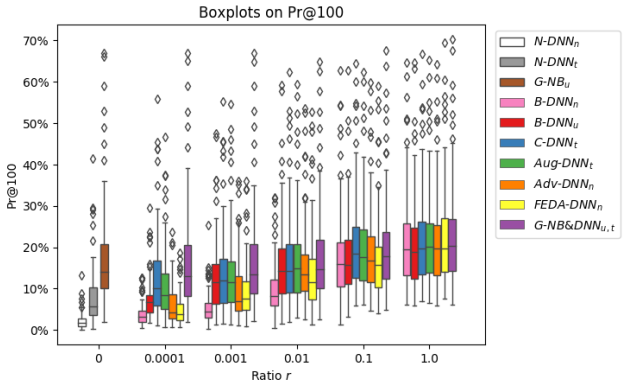


FIGURE 4. Boxplots representing the card-based precision@100 for all DNN and baseline methods (82 days with one precision score per day). Notice that the a priori fraud ratio based on cards is 0.34%.

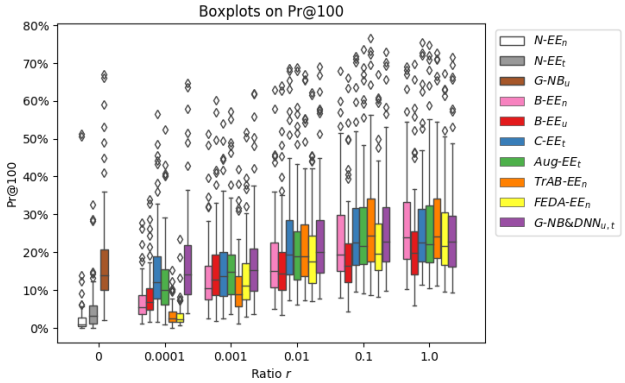


FIGURE 5. Boxplots representing the card-based precision@100 for all EE and baseline methods (82 days with one precision score per day).

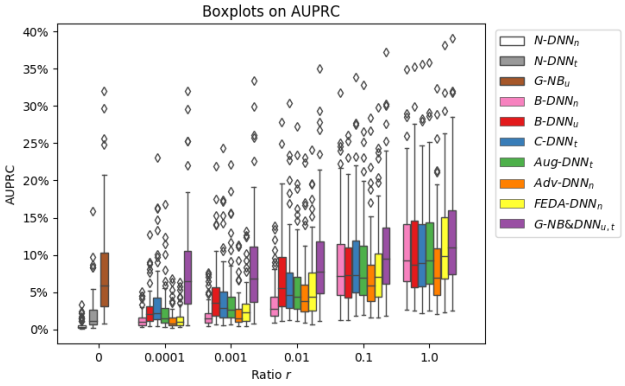


FIGURE 6. Boxplots representing the card-based area under the precision-recall curve [40] for all DNN and baseline methods (82 days with one precision score per day).

to the labeling ratio and are then presented only once on the leftmost side.

Note that for each method, we report the accuracy associated with the best normalization method (see Section IV-C). For the sake of comparison, the version with no normalization (noted by a subscript n) is also provided for the two baselines (*B-(DNN/EE)* and *N-(DNN/EE)*).

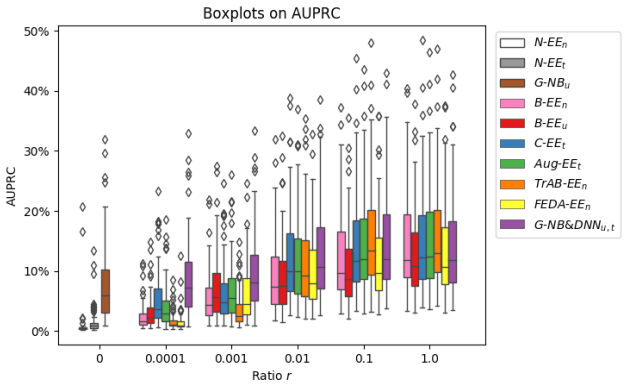


FIGURE 7. Boxplots representing the card-based area under the precision-recall curve [40] for all EE and baseline methods (82 days with one precision score per day).

Figures 8 and 9 summarizes the previous results in the form of Friedman/Nemenyi (F/N) tests [42], for DNN-based methods and EE-based methods, respectively. There is one F/N test per ratio r. A method is considered significantly better than another if the difference between their mean ranks is larger than the critical difference CD.

All experiments were carried on a server with 10 cores, 256 GB RAM, and an Asus GTX 1080 TI. The relative execution time (feature manipulation and classification only, no time for tuning), expressed as a ratio with respect to the fastest method (G-NB), appears in the last column of Table 2. Note that the B-DNN, N-DNN, and C-DNN execution times are slower when the training set is larger. More advanced methods (e.g. FEDA-DNN, Aug-DNN, Adv-DNN, TrAB-EE) typically require more time than their naive counterparts. In particular, Aug-DNN needs to train two classifiers and is therefore the slowest approach.

Model hyperparameters were tuned using a validation set on the 10 first days of data. The parameter value selected most often, per method, is $n_{PC} = 5$ for Aug-DNN, $n_{PC} = 2$ for Aug-EE, and $\lambda = 0.1$ for Adv-DNN. The optimal value of α for G-NB&DNN and G-NB&EE depends on the ratio r and is generally higher when r is higher (i.e. the supervised part of the model is favored when more samples are available). For all EE-based models, we use 25 balanced forests, each composed of 25 trees. For all DNN-based models, we use two hidden layers with $1.5d$ hidden neurons, d being the number of input features.

VI. DISCUSSION

Several considerations can be made based on the results shown in the previous section.

In this section, the different methods of Table 2 are compared on a real-life credit card transaction dataset obtained from our industrial partner. To complete the names of the methods from Table 2, we will also specify the normalization method (only the best is reported), and the actual train and test set for this method (under the form

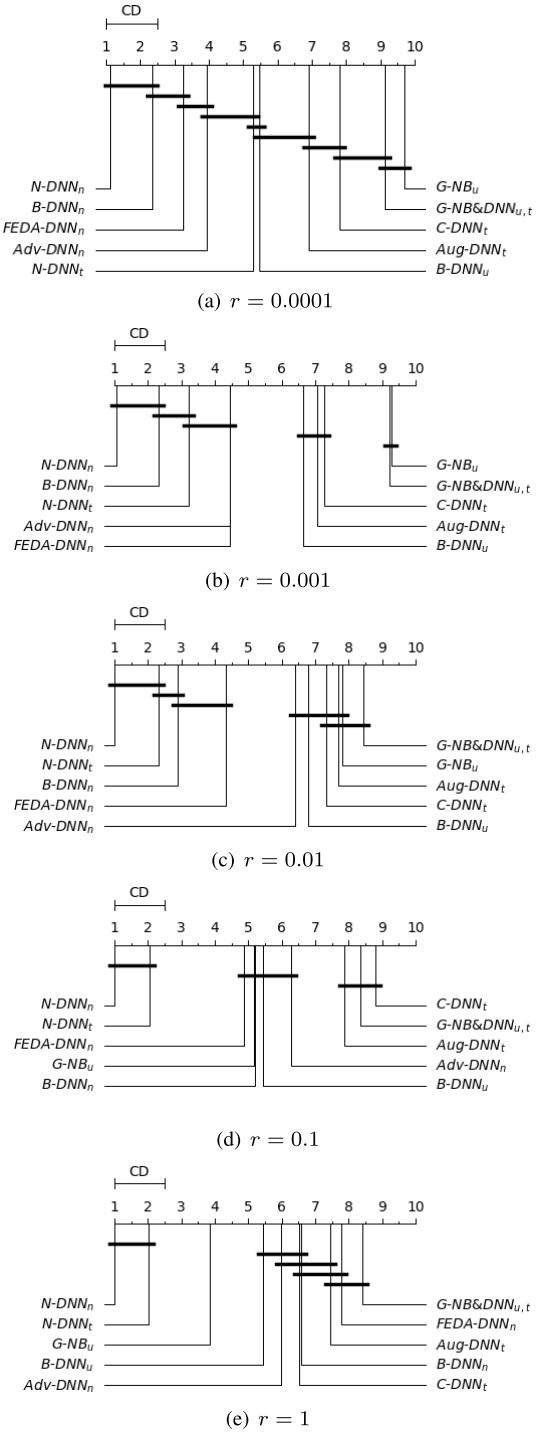


FIGURE 8. Mean rank (the higher, the better) and critical difference (*CD*) of the Friedman/Nemenyi test for each labeling rate *r*, for all the DNN-based methods. A method is considered significantly better than another if its mean rank is larger by more than the critical difference *CD* = 1.496. The bold horizontal lines indicate clusters of methods having equivalent performances (difference between the mean less than *CD*).

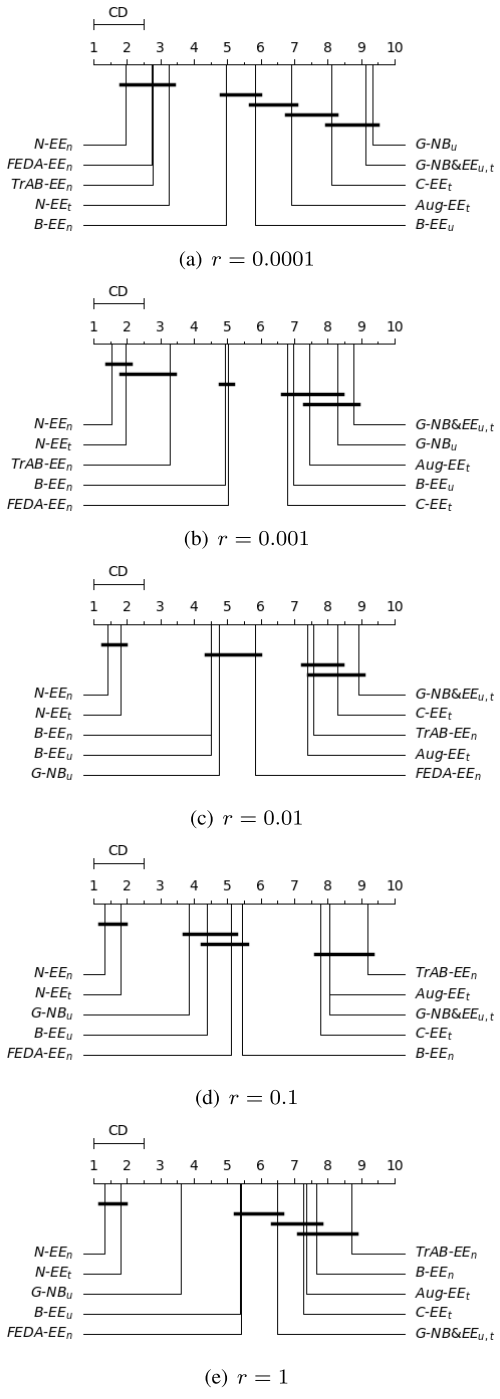


FIGURE 9. Friedman/Nemenyi test for each labeling rate *r*, for all the EE-based methods. See also the caption of Figure 8. Here, the critical difference *CD* equals 1.496.

train2test). As this leads to quite long names, they will be formatted in italic form for clarity.

Figure 4 to 7 presents the performance for the 15 methods of Table 2, plus the four versions without normalization of the baselines: *B-(DNN/EE)*$_n$ and *N-(DNN/EE)*$_n$.

- Accuracy in terms of Pr@100 vs. AUPRC metric: though the Pr@100 is considered by our partner as the most relevant metric, for the sake of completeness we reported the AUPRC results as well. It is then interesting to check if they provide similar insights about the accuracy of the assessed methods. For instance, from Figures 4 and 6 it appears that the AUPRC and the Pr@100 rankings are similar, with the exception that

Adv-DNN$_n$ is less accurate in terms of AUPRC. For EE-based methods, the AUPRC ranking (Figure 7) is nearly identical to the Pr@100 one (Figure 5). Due to the low and varying proportion of fraud in our data, there is a large variability for a given model in terms of Pr@100 and AUPRC. This variability is acceptable in our setting because the semi-supervised approach is deployed in a new environment only in order to obtain a sufficient number of labels, which are eventually used to train a fully supervised model.

- Accuracy vs. target labeling ratio r: from Figure 4, it appears that all methods perform similarly when the ratio r is high. However, as expected, when the ratio r decreases, the accuracy of all methods decreases as well. In order to compare the different strategies, it is then important to consider the trend of the accuracy for decreasing r (i.e. the slower the deterioration the better). In this perspective, the most accurate DNN methods are *C-DNN$_t$* and *Aug-DNN$_t$*, while the most sensitive to a decreasing r are *B-DNN$_n$* and *FEDA-DNN$_n$*. Among EE-based methods, *C-EE$_t$* and *Aug-EE$_t$* are the best methods while the ones with the lowest accuracy are *B-EE$_n$* and *FEDA-EE$_n$*. Interestingly, *TrAB-EE$_n$* has the best accuracy when r is large but becomes one of the worst methods when r is small.
  From Figure 8, *G-NB$_u$* is the best option when few labels are available. When r increases it is outperformed by supervised techniques: mainly by *C-DNN$_t$* and *Aug-DNN$_t$* and *FEDA-DNN$_t$*.
  For the EE-based methods, on Figure 9, *G-NB$_u$* is also the best methods, significantly equivalent to *C-EE$_t$* (for $r = 0.0001$) and *Aug-EE$_t$* (for $r = 0.001$). When more labels become available, the best option becomes TrAB-EE$_n$, with a few equivalent methods: mainly *C-EE$_t$*, *Aug-EE$_t$* and *B-EE$_n$* (when $r = 1$).
- Supervised vs. self-supervised DA approaches: self-supervised approaches (notably *G-NB$_u$*) are competitive with most supervised methods for low r though they are outperformed for large values of r.
- Baseline accuracy: the naive strategies *C-DNN$_t$* and *C-EE$_t$* (using both source and target domain samples) are among the best approaches though less accurate for small r. The other baselines (using only one domain, either source or target) have much worse accuracy. In particular, *N-DNN$_n$* is the poorest approach, confirming that just reusing the source classifier for the target domain is inadequate and that the use of some transfer learning principle is recommended.
- EE vs. DNN: overall, EE methods tend to be better than DNN methods, probably because a medium-complex baseline NN was used. This is also confirmed when AUPRC is used as the performance metric.
- Impact of normalization: normalization is a key feature for enhancing transfer performance. The normalization to the target domain is often the best strategy. However, some methods (*Adv-DNN$_n$*, *FEDA-(DNN/EE)$_n$* and *TrAB-EE$_n$*) are more accurate for raw data.
- Combined approach G-NB&(DNN/EE): The challenge was to design a method that keeps constant performances when the number of available labels in the target domain decrease. The *G-NB&(DNN/EE)* leverages on two of the two best methods of this section: *G-NB$_u$* and *C-DNN/EE$_t$*. The hyperparameter α, which allows weighting the contribution of the two approaches, is easily tuned using a few data of validation (in our case the ten first days of data, see Section V-A). From the F/N tests in Figures 8 and 9, it appears that G-NB&(DNN/EE) outperforms, or is not worse than, all corresponding DNN (or EE) methods. Therefore, we recommend this approach for transfer in FDS.

To summarize, the baseline combining source and target (C-DNN/EE) and the augmented approach Aug-(DNN/EE) are the best approaches when a sufficient number of labels is available. The self-supervised approach G-NB is the best approach when few or no labels are available. Feature normalization is key to obtain the best performance, and the combined approach G-NB&(DNN/EE) is the best approach overall, by leveraging the strengths of both supervised and self-supervised models.

VII. CONCLUSION

The paper is, to the best of our knowledge, one of the first [9], [10] to study the use of transfer learning strategies in transaction-based fraud detection systems. Though the case study is limited to 6 months of data, we consider it fully realistic from a business perspective. It is indeed a top priority for transactional companies to develop strategies to reuse detection models trained on consolidated markets to new ones.

The paper discusses, implements and assesses 15 transfer learning techniques in a number of settings characterized by different amounts of supervised labeling in the target domain. It is interesting to note for very low amounts of target labels, generative classifiers (e.g. Naive Bayes) outperform discriminative ones. This robustness might be due to the fact that generative approaches are more compatible with the causal relationship existing between the inputs and the output of a fraud classifier: indeed, the most commonly used input features are not causes of the output binary class (fraud or genuine), but descriptors of the fraudulent event (and as such effects of the output binary variable). However, if we increase the number of target labels, the adoption of adversarial or augmented feature strategies is recommended. Overall the most accurate method is an ensemble of unsupervised and semi-supervised domain adaptation classifiers, which outperforms all considered approaches. Indeed, self-supervised DA is better suited for situations where few (or no) labels in the target domain are known, whereas semi-supervised domain adaptation is more suited if enough target domain labels can be gathered. The adoption of a weighting hyperparameter allows tuning the contributions of the two approaches.

Future work will focus on extending the set of considered methods and in using transfer strategies to address problems related to the single market case, e.g. nonstationarity and drift. At the same time, we expect to assess the robustness of the approaches by applying them to other transfer problems (e.g. new countries). Thanks to our promising results, our industrial partner Worldline has already implemented in production a combination of self-supervised and semi-supervised approaches [43].

ACKNOWLEDGMENT

The authors would like to thank Innoviris for allowing them to conduct both fundamental and applied research. The authors and the parties cited above have no competing interests.

REFERENCES

[1] HSN Consultants, Inc. (Oct. 17, 2019). *The Nilson Report 2018*. [Online]. Available: https://nilsonreport.com

[2] A. D. Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, Aug. 2014.

[3] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection and concept-drift adaptation with delayed supervised information," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[4] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[5] F. Carcillo, Y. L. Borgne, O. Caelen, and G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization," *Int. J. Data Sci. Anal.*, vol. 5, no. 4, pp. 285–300, 2018.

[6] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with spark," *Inf. Fusion*, vol. 41, pp. 182–194, May 2018.

[7] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016.

[8] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015.

[9] B. Lebichot, Y.-A. Le Borgne, L. He-Guelton, F. Oblé, and G. Bontempi, "Deep-learning domain adaptation techniques for credit cards fraud detection," in *Recent Advances in Big Data and Deep Learning*, L. Oneto, N. Navarin, A. Sperduti, and D. Anguita, Eds. Cham, Switzerland: Springer, 2020, pp. 78–88.

[10] Y. Zhu, D. Xi, B. Song, F. Zhuang, S. Chen, X. Gu, and Q. He, "Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection," in *Proc. Web Conf.*, Apr. 2020, pp. 928–938.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[12] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016.

[13] J. Peters, D. Janzing, and B. Scholkopf, *Elements of Causal Inference*. Cambridge, MA, USA: MIT Press, 2017.

[14] D. Janzing and B. Schölkopf, "Semi-supervised interpolation in an anticausal learning scenario," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1923–1948, 2015.

[15] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2006, pp. 601–608.

[16] M. Loog, "Nearest neighbor-based importance weighting," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2012, pp. 1–6.

[17] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2007, pp. 1433–1440.

[18] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 283–291.

[19] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2008, pp. 627–634.

[20] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 193–200.

[21] S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive Bayes to domain adaptation for sentiment analysis," in *Proc. 31th Eur. Conf. IR Res. Adv. Inf. Retr.* Berlin, Germany: Springer, 2009, pp. 337–349.

[22] R. Saia and S. Carta, "Evaluating the benefits of using proactive transformed-domain-based techniques in fraud detection tasks," *Future Gener. Comput. Syst.*, vol. 93, pp. 18–32, Apr. 2019.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[24] H. Daume, III, "Frustratingly easy domain adaptation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2007, pp. 256–263.

[25] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[26] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7947–7951.

[27] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Red Hook, NY, USA: Curran Associates, 2008, pp. 153–160.

[28] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 109–117.

[29] J. Davis and P. Domingos, "Deep transfer via second-order Markov logic," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 217–224.

[30] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising Markov logic networks for transfer learning," in *Proc. 22nd Nat. Conf. Artif. Intell.*, vol. 1. Palo Alto, CA, USA: AAAI Press, 2007, pp. 608–614.

[31] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *Proc. ECCV*, vol. 3, Oct. 2008, pp. 69–82.

[32] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Transfer learning for Latin and Chinese characters with deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–6.

[33] S. Chopra, S. Balakrishnan, and R. Gopalan, "DLID: Deep learning for domain adaptation by interpolating between domains," in *Proc. ICML Workshop Challenges Represent. Learn.*, 2013, pp. 1–8.

[34] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28ht Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 1–8.

[35] I. Sohony, R. Pratap, and U. Nambiar, "Ensemble learning for credit card fraud detection," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2018, pp. 289–294.

[36] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[37] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2th ed. New York, NY, USA: Springer, 2009.

[39] I. Guyon, C. Aliferis, and A. Elisseeff, *Causal Feature Selection*. London, U.K.: Chapman & Hall, 2007, pp. 63–85.

[40] D. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve," *Mach. Learn.*, vol. 77, no. 1, pp. 103–123, 2009.

[41] B. Lebichot, F. Braun, O. Caelen, and M. Saerens, *A Graph-Based, Semi-Supervised, Credit Card Fraud Detection System*. Cham, Switzerland: Springer, 2017, pp. 721–733.

[42] J. Demsar, "Statistical comparaison of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[43] W. Siblini *et al.*, "Transfer learning for credit card fraud detection: A journey from research to production," 2021, *arXiv:2107.09323*. [Online]. Available: https://arxiv.org/abs/2107.09323

BERTRAND LEBICHOT received the M.Sc. and Ph.D. degrees in engineering from the Université catholique de Louvain (UCL), Belgium, in 2011 and 2018, respectively. He worked as a Postdoctoral Researcher at the Université Libre de Bruxelles (ULB), Belgium, and is currently a Research Associate with the University of Luxembourg. He is also a part-time Lecturer at UCL. His research interests include graph mining, deep learning, and fintech applications.

THÉO VERHELST received the bachelor's degree in computer science from the Université Libre de Bruxelles (ULB), in 2017, and the M.Sc. degree in artificial intelligence program, under the Erasmus Program, from Southampton University. He completed his second year of master's degree (Hons.) in computer science from ULB. After a six months research contract on machine learning for credit card fraud detection, he is currently pursuing the Ph.D. degree with the ULB Machine Learning Group on machine learning and causal analysis for telecom customer data, in collaboration with Orange Belgium.

YANN-AËL LE BORGNE received the M.Sc. degree in cognitive sciences from Joseph Fourier University, France, in 2003, and the Ph.D. degree in computer science, under EU Marie Curie Fellowship, from the University of Brussels, Belgium, in 2009. He is a Senior Consultant in machine learning and scientific collaborator at the Machine Learning Group, University of Brussels. His research interests include machine learning and big data technologies, with a focus on applications related to scalable time series forecasting, fraud detection, and the Internet of Things.

LIYUN HE-GUELTON received the degree in engineering from Telecom Bretagne, France, and the Ph.D. degree from the French Institution of Research for Sea Exploitation (IFREMER), in 2014. She then worked as a Research Engineer with the National Institute for Research in Computer Science and Automation (INRIA). Since 2015, she has been working with the Research and Development Department, Worldline. Her current research interests include AI, machine learning, big data, and fintech applications.

FRÉDÉRIC OBLÉ received the Ph.D. degree in computational fluid dynamics from the Université de Lille, in 1997. He has been working with Worldline, since 2000. After ten years journey within operational units making business and pushing technical innovation, he has been leading an Research and Development Department for ten years and has led big data and artificial intelligence research and development programs for Worldline and Atos Group. He is currently the Head of the scientific and technical direction at Worldline Labs and leading a research program related to AI, trust, and hyper automation.

GIANLUCA BONTEMPI (Senior Member, IEEE) is a Full Professor with the Computer Science Department, Université Libre de Bruxelles (ULB), Brussels, Belgium, the Co-Head of the ULB Machine Learning Group. He has been the Director of (IB)2, ULB/VUB Interuniversity Institute of Bioinformatics, Brussels, from 2013 to 2017. He was a Marie Curie Fellow Researcher. He is the author of more than 200 scientific publications. He is also a coauthor of several open-source software packages for bioinformatics, data mining, and prediction. His research interests include big data mining, machine learning, bioinformatics, causal inference, predictive modeling, and their application to complex tasks in engineering (time series forecasting and fraud detection) and life science (network inference and gene signature extraction). He is a member of the Scientific Advisory Board of Chist-ERA. He was awarded in two international data analysis competitions and took part in many research projects in collaboration with universities and private companies all over Europe.