Transfer Learning in Credit Risk*

Hendra Suryanto[1][0000−0001−9318−8572] (✉), Charles Guan[1][0000−0002−6406−2599], Andrew Voumard[1][0000−0002−4414−6904], and Ghassan Beydoun[2][0000−0001−8087−5445]

[1] Rich Data Corporation, Singapore
info@richdataco.com
http://www.richdataco.com
{hendra.suryanto,charles.guan,andrew.voumard}@richdataco.com
[2] University of Technology Sydney, Sydney, Australia
http://feit.uts.edu.au
ghassan.beydoun@uts.edu.au

Abstract. In the credit risk domain, lenders frequently face situations where there is no, or limited historical lending outcome data. This generally results in limited or unaffordable credit for some individuals and small businesses. Transfer learning can potentially reduce this limitation, by leveraging knowledge from related domains, with sufficient outcome data. We investigated the potential for applying transfer learning across various credit domains, for example, from the credit card lending and debt consolidation domain into the small business lending domain.

Keywords: Credit Risk · Transfer Learning · Data Science.

1 Introduction

We studied a new domain where no or limited historical lending outcomes are available, for example: offering credit to un-banked or under-banked populations or micro to small businesses, where limited historical data is available. Currently, lenders rely mainly on expert rules for credit scoring. Due to high uncertainty in the performance of such scoring models, lenders charge a high fee or simply don't offer credit. Transfer learning from related domains is a potential solution to augment this lack of information and improve financial inclusion. For instance, transferring knowledge from credit card/debt consolidation loans to more risky small business loans or from utility bill payments to loan repayments could potentially deliver a more accurate scoring model.

We investigated the application of transfer learning during the initial stage of a credit risk model implementation, where there was limited historical labelled data available. In the credit risk domain, business priorities are stability and accuracy of model performance, in order to predict the probability of default. We present our approach, that enabled us to combine the outcome of the transferred model from related credit risk domains, with new models based on newly acquired

* Supported by Rich Data Corporation.

labelled data from new domains. Using this approach, we were able to achieve a higher accuracy and maintained stability of the overall model. Experiments on real-world commercial data showed that combining the transferred models and the new models can achieve these goals by using an incrementally transitioned approach. To allow us to publish the results and comply with the privacy requirements of our client's data, we reproduced our experiment using lendingclub.com data, https://www.lendingclub.com/info/download-data.action, which is publicly available.

When a lender expands into new market segments, a new credit risk model is required to assess the credit risk of loan applications. The current approach is based on expert rules, where the credit risk expert builds business rules based on data and available derived data, combined with the expert's experience and knowledge. Lenders initially use an expert model to gather sufficient labelled data, to build a supervised learning model. The expert model is compared against the supervised learning model. If one model performs substantially better than the other, the better model is used. Alternatively, if both models complement each other, they can be combined into an ensemble model. In commercial lending systems, lenders normally charge a higher price or limit credit offerings as there is no (or limited) labelled data to validate the expert models. The result is that many individuals and businesses are excluded from these "formal" lending systems. Organizing data access for a suitable expert to perform analysis on such typically sensitive data may be difficult, for example, when data can only be accessed on site by authorized persons.

We studied two scenarios using a large dataset of existing loan products to enhance the credit risk model for new loan products, which have a much smaller dataset. The first scenario uses Lending Club data to mimic a lender that has existing credit card and debt consolidation data and starts to offer loans to small businesses. The second scenario also uses Lending Club data, to mimic a lender that has an existing credit card loan product and starts to offer car loans.

When we pre-process the Lending Club data, we select 16 variables as inputs and the output to be predicted is the loan status. To simplify the model, we convert the loan status into a binary outcome. 1: for defaulted loans, charged-off loans or late loan payments, 0: for paid-off loans. Any current loans that are not yet due are excluded from this exercise. The pre-processing details are described in Section 9.

Credit card and debt consolidation loans are typically unsecured consumer loans. Their scoring model depends mainly on an individual's credit rating, income, expenses and other attributes like stability of their employment and residence. A hive of recent fintech activities in this space (particularly in UK, US and China) accumulated a legacy of historical data and concomitant stable and accurate scoring models, in what is now a crowded and competitive market. Small business lending is a relatively new market for fintechs; it is riskier, more diverse, more challenging to predict the outcomes, and suffers from a scarcity of data. As we can see in the Lending Club data, the quantity of the historical lending outcome for small business loans is far lower, and insufficient to develop a stable and accurate model using traditional supervised learning. With less competition and higher margin for small business lending (compared to consumer lending) it is more valuable for lenders to find ways to predict loan outcomes and serve this market. Furthermore, Micro, Small and Medium Enterprises (MSMEs) are one of the strongest drivers of economic development, innovation and employment. Access to finance is frequently identified as a critical barrier to growth for MSMEs. Creating opportunities for MSMEs in emerging markets is a key way to advance economic development and reduce poverty. 65 million (or 40% of formal MSMEs) in developing countries have unmet financing needs. The MSME finance gap in developing countries is estimated at $5.2 trillion - 1.4 times the current level of MSME lending [5].

2 Related Work

We have seen increasing interest in transferred supervised models - from one domain to another. Most published works in this area cover image processing, for example: Yang proposed transferring parameters on SVM [14], Pan proposed domain adaptation using transfer component analysis [7]. Pan and Yang grouped transfer learning into four approaches: instance-transfer, feature-representation-transfer, parameter-transfer, and relational-knowledge-transfer [8].

Our experimentation combines the reuse of features and derivation of new features from the source (existing) domain. Source domain labels are available; limited target (new) domain labels are available. We also focus on classification. Our experimentation is similar in those ways to Transductive Transfer Learning [7] - one key addition, is to the target classification task optimization. In Transductive Transfer Learning, the source and target tasks must be the same (classification in this case). In our experimentation though, we took a new step in optimizing the target model accuracy, by introducing and experimenting with an extra optimization variable: the level of relative source/target feature data contribution proportions into the target model.

Many papers focus largely on making optimal choices of parameters, features, and source(s), to transfer learning to the target model, as summarized in [12] - which examines homogeneous and even heterogeneous data domains, symmetric and asymmetric feature transformation, for instance-based, feature-based, parameter-based, and relational-based related transfer learning. [10], [13], [3], [6] make specific efforts to minimize 'negative transfer' (a transfer that has a negative impact on the target model). While these approaches help to improve target model results - and can (in some cases) reduce target model build times, our focus was centered on optimizing the target model configuration / composition and design, for the transferred features after they were selected to be inputs to the target model.

3 Credit Risk

Lenders seek to optimize the risk return ratio across their lending portfolios. Accurately and consistently measuring credit risk is the foundation of this optimization. Lenders commonly use the concept of Expected Loss (EL) to measure credit risk. In an unsecured lending scenario, EL is mainly determined by the Probability of Default (PD). Credit scoring models are used to calculate PD. Inputs of a credit scoring model are normally attributes of the loan applicant and their application. In this paper we use a few attributes from lendingclub.com data to illustrate our approach. In credit risk, the most common metrics to assess the quality of credit scoring model are Gini, Kolmogorov-Smirnov statistics (KS), Lift, the Mahalanobis distance and information statistics [9]. In this paper we use Gini for this purpose.

The scoring model output is a score from 0 to 1. It is an estimated probability of default. Usually some part of the data is pre-allocated for calibration of the score. Lenders use a set of decision processes and rules to make an optimal decision with the derived PD and loan application data as inputs. A decision process generally starts with an eligibility test. PD is calculated for the eligible applicants, and then used to group applicants into different decision groups. For instance, the interest rate could vary for different decision groups, as could the loan amount as a percentage of net income.

In this investigation, our focus is credit scoring for unsecured lending. We measure the performance of our credit scoring model using Area Under Receiver Operating Curve (AUC) or $GiniROC$ which is $2AUC - 1$ [4]. GiniROC shares the same concept as Gini, for splitting criteria in CART [2]. Gini and GiniROC, however, have different usages. The metric GiniROC is used to allow the assessment of model quality, based on PD, without needing to convert PD into binary classifications, since the threshold to do those classifications is defined in the credit decisioning.

3.1 Credit Scoring

Credit scoring produces a PD, which is used to predict binary outcomes, loan-paid or loan-defaulted. In real-world scenarios, there are additional outcomes, such as late payment or partial payment. In credit scoring, we need a metric to assess the quality of the model without defining a threshold to convert the PD into a classification. When we have classifications, we can use a metric such as $Fscore$. In credit risk, this decision is deferred to the credit decisioning step, where expert rules are utilized to decide whether the loan is approved or not.

3.2 Credit Decisioning

Credit decisioning consumes PD and produces a decision to approve or decline a loan application. The conversion from PD to a decision is usually driven by a mapping table to map ranges of PD to decisions. The decision is not only to approve or to decline, it may also update the loan amount, interest and term.

This model is usually based on expert rules, since the data is usually too sparse and/or the search space is too large for building supervised learning models.

4 Model Development

We developed six example network configurations to empirically assess the effectiveness of our transfer learning algorithm; their detail is explained in Sections 4.1 and 5. Table 1 shows those configurations in order of increasing Progressive Shift Contribution (PSC) from the source domain to the target domain. The PSC is our novel contribution and is explained in detail in Section 5.

Model No 1 is developed by training using source domain data only. The domain contribution in Models No 2 to 5 is progressively shifted from source to target domain. The last model, Model No 6 is trained using target domain data only. The difference in contribution between the source and target domains shows up in the ratio between the number of trained layers using the target domain and the number of trained layers using the source domain. The algorithm can be generalized for any network configuration size. Further details of the algorithm will be discussed in Section 5. Source code and data for all experiments is provided, see Section 9.

Table 1. Six Network Configurations with PSC

No	Model Name	Source Domain Contribution	Target Domain Contribution	Layers Trained by Source	Layers Trained by Target	Network Configuration
1	$M(v)_e$	100%	0%	4	0	Fig 2
2	$M(w)_{transfer}$	75%	25%	3	1	Fig 4
3	$M(wx)_{transfer}$	71%	29%	5	2	Fig 5
4	$M(wxy)_{transfer}$	60%	40%	6	4	Fig 6
5	$M(wxyz)_{transfer}$	46%	54%	6	7	Fig 3
6	$M(u)_n$	0%	100%	0	4	Fig 1

We discover the optimum network configuration by shifting the PSC from the source to target domain and measure the Gini performance using the target domain test data. The model performance is conceptually influenced by a) the modelling techniques (e.g. deep learning, gradient boosting machine, generalized linear model), hyper parameters[3], b) the signal strength in the data and c) feature engineering; Informally, the relationship between $Gini$ and these factors can be written as follows:

$$Gini = g(test(M_e, s_e)) \tag{1}$$

where s_e is test data from the source domain, M_e is the model trained using training data from the source domain, $test()$ is an activity to test a model on the

[3] the hyper parameters optimization has been done before this step

test data producing the test results and $g()$ is a function to calculate the Gini of the results. M_e is defined as follows:

$$M_e = train(M_0, P_e, t_e, F_e) \tag{2}$$

where M_0 is a deep neural network configuration with initial random weights, P_e is a set of hyper parameters to train M_e, t_e is the training data from the source domain, F_e is a set of features derived from t_e, $train()$ is an activity to train a model based on these four factors. The result of $train()$ is a trained model.

To explain how we perform the PSC, we define a function $split()$ to conceptually split M_e into two segments: $Mfix_e$ and $Mfree_e$. $Mfix_e$ is the segment where the layers were trained using t_e and these layers are not retrainable. $Mfree_e$ is the segment where the layers were also trained using t_e, but these layers are trainable using the training data from the target domain t_n.

$$(Mfix_e, Mfree_e) = split(M_e) \tag{3}$$

The inverse function of $split()$ is $c()$, for combining $Mfix_e$ and $Mfree_e$ back into M_e

$$M_e = c(Mfix_e, Mfree_e) \tag{4}$$

To create a mixed model based on both the source and target domain data, we developed a model for the target domain $M_{transfer}$, by transferring the structure and weights of $Mfix_e$ layers and retraining the structure and weights of $Mfree_e$.

$$Mfree_n = train(Mfree_e, P_n, t_n, F_n) \tag{5}$$

Finally, we combined the target model $Mfree_n$ with $Mfix_e$. The result is the transferred model $M_{transfer}$

$$M_{transfer} = c(Mfix_e, Mfree_n) \tag{6}$$

The overall goal is to maximize $Gini_{transfer}$, where s_n is the test data from the target domain:

$$Gini_{transfer} = g(test(M_{transfer}, s_n)) \tag{7}$$

by monitoring $Gini_{transfer}$ as we shift the PSC from the source to target domain data. Finally, we discover the maximum $Gini_{transfer}$ by testing the performance of all six network configurations outlined in Table 1.

4.1 The base model

The base models were configured based on network structures. The first is shown in Figure 1. It has 16 input nodes on the input layer, 3 hidden layers, each layer has 32 nodes and 1 output node on the output layer. This network configuration is selected by using a hyper parameter search to find a near optimum configuration.

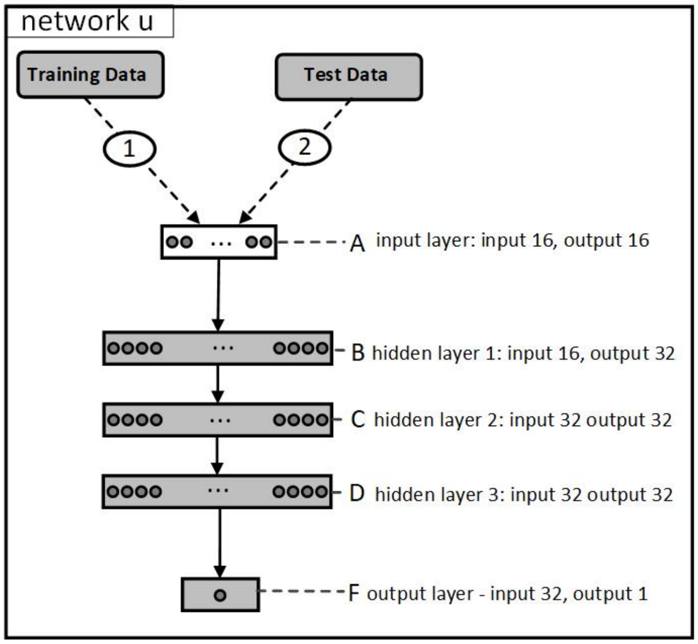

Fig. 1: Network u: the base model

4.2 The Comparison model (Model u)

The comparison model is based on network configuration u as shown in Figure 1. This model is trained using the target domain data only (no contribution from source domain data). Similar to Equation 2, the model built using target domain data, can be defined as follows:

$$M(u)_n = train(M(u)_0, P_n, t_n, F_n) \tag{8}$$

where $M(u)_n$ is a model developed using data from the target domain based on network configuration u, $M(u)_0$ is the initial model based on network configuration u with all weights randomly initialized, P_n, t_n, F_n are parameters, training data and features respectively, used to develop the model $M(u)_n$.

$$Gini = g(test(M(u)_n, s_n)) \tag{9}$$

where s_n is test data from the target domain.

5 Progressive Shift Contribution Models

In Section 4, we introduced six models where the Progressive Shift Contribution (PSC) shifts between the source and target domain data. To perform the PSC, we extended the split function defined in Equation 3 with an additional parameter to define the proportion of PSC. The value of this parameter one of: v, w, wx, wxy, or $wxyz$. Each value results in a different network configuration as shown in Table 1, and a different PSC from the source to the target domain. Using these five values, we developed five PSC models. We also include the baseline Comparison Model discussed in Subsection 4.2. The following subsections explain models No 2 to 6 in detail.

5.1 Model v

Model v is only created from source domain data (no target domain contribution whatsoever). To create Model v, we started by training model $M(v)_e$, based on Equation 10, using configuration shown in Figure 2

$$M(v)_e = train(M(v)_0, P_e, t_e, F_e) \tag{10}$$

Then the model was tested on target domain data, and a Gini value was calculated from the test results.

$$Gini = g(test(M(v)_e, s_n)) \tag{11}$$

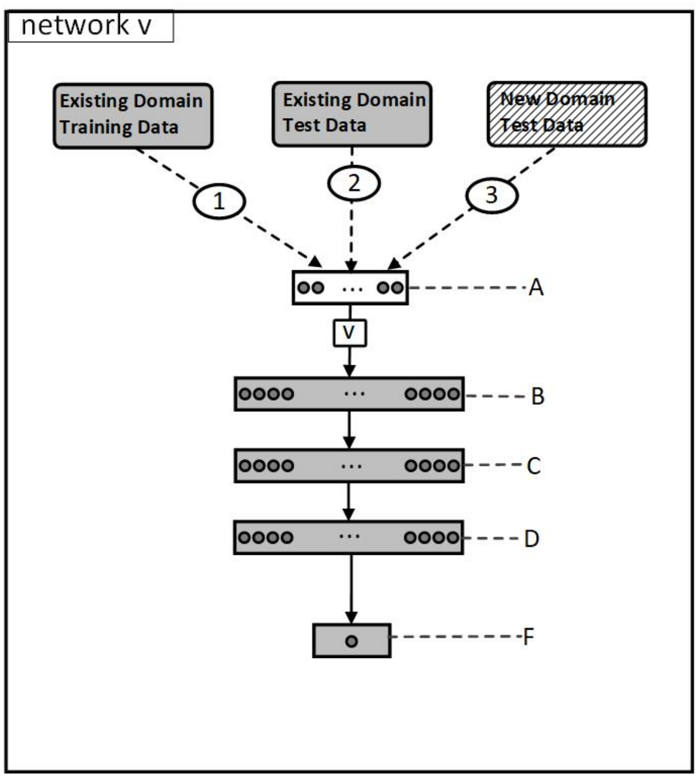

Fig. 2: Network v

5.2 Model wxyz

This model was created using four parallel networks - each with three hidden layers, connected to the input and output layers. To create this model, we initially copied hidden layers of network v (both the structure and the weights) into networks w, x, y and z. Conceptually, we illustrate the transformation using Equation 12.

$$M(wxyz)_e = transform(M(v)_e) \tag{12}$$

Networks w, x, y and z were setup as shown in Figure 3.

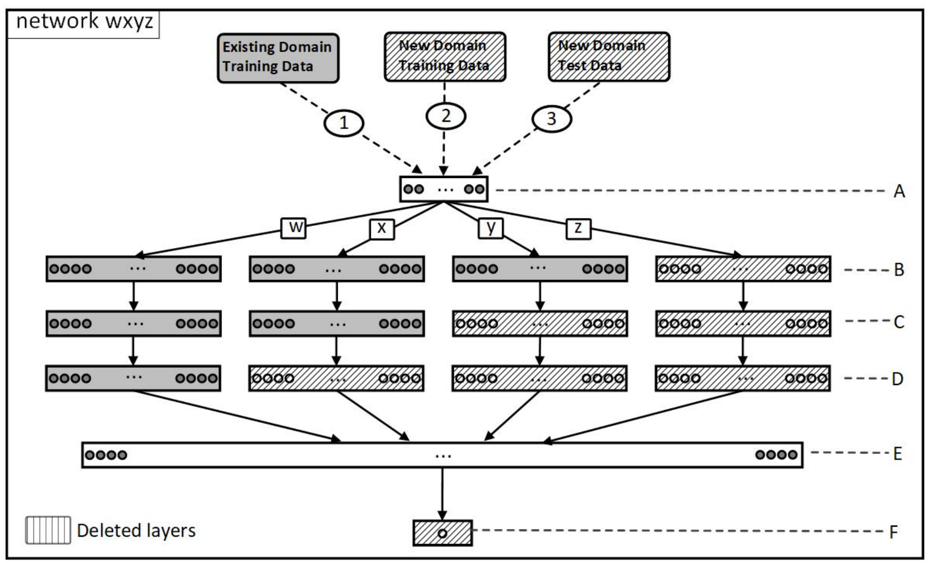

Fig. 3: Network wxyz

After the structure and weights were set (as shown in Figure 3), we then set the following as trainable, using the target domain data: The 3rd hidden layer of Network x, the 2nd and 3rd hidden layers of Network y, and all hidden layers of Network z. The next three steps are indicated in numbers 1, 2, 3 within ellipses in Figure 3:

1. Weights for networks w, x, y, z were derived from training using t_e. Some layers in networks x, y, z and w, the output layers are set as trainable, using t_n.
2. Train these layers using t_n.
3. Test the performance of the whole parallel network (w, x, y, z) on s_n, then calculate the Gini value from the test result.

The development of Model wxyz can be summarized by three equations: Equation 13, Equation 14, Equation 15.

$$(Mfix(wxyz)_e, Mfree(wxyz)_e) = split(M(wxyz)_e) \tag{13}$$

$$Mfree(wxyz)_n = train(Mfree(wxyz)_e, P_n, t_n, F_n) \tag{14}$$

$$M(wxyz)_{transfer} = c(Mfix(wxyz)_e, Mfree(wxyz)_n) \tag{15}$$

In Model wxyz, six hidden layers were trained using the source domain data and seven layers were retrained using the target domain data, i.e. six hidden layers and the output layer were retrained.

5.3 Model w

Model w is developed based on Model wxyz, where Networks x, y and z are deleted. This network configuration is shown in Figure 4. In Model w, three hidden layers were trained using the source domain data and only the output layer was retrained using the target domain data. The development of model w is shown in Equation 16, Equation 17, Equation 18.

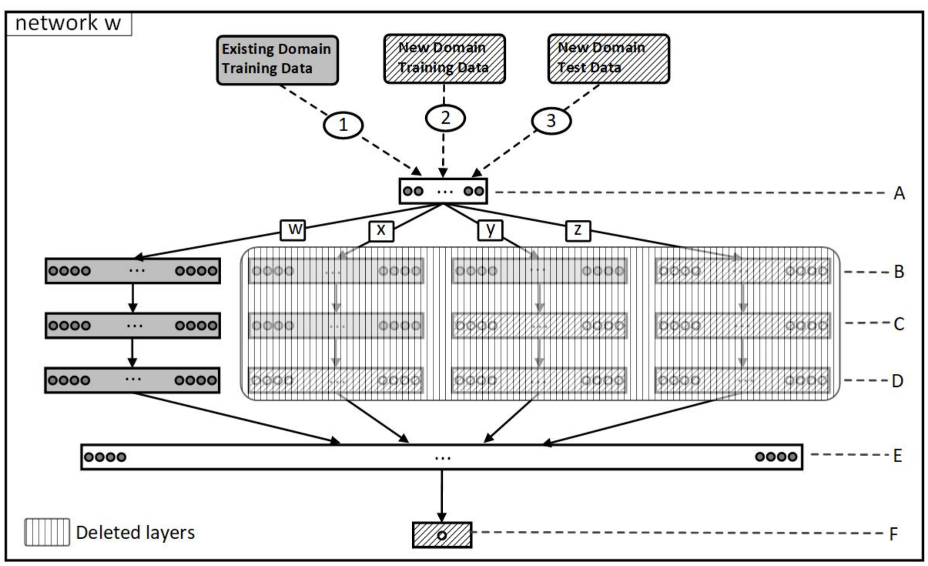

Fig. 4: Network w

$$(Mfix(w)_e, Mfree(w)_e) = split(M(w)_e) \tag{16}$$

$$Mfree(w)_n = train(Mfree(w)_e, P_n, t_n, F_n) \tag{17}$$

$$M(w)_{transfer} = c(Mfix(w)_e, Mfree(w)_n) \tag{18}$$

5.4 Model wx

Model wx is developed based on Model wxyz, where Networks y and z are deleted. This network configuration shown in Figure 5. In Model wx, five hidden layers were trained using the source domain data. One hidden layer and the output layer were retrained using the target domain data. The development of model wx is shown in Equation 19, Equation 20, Equation 21.

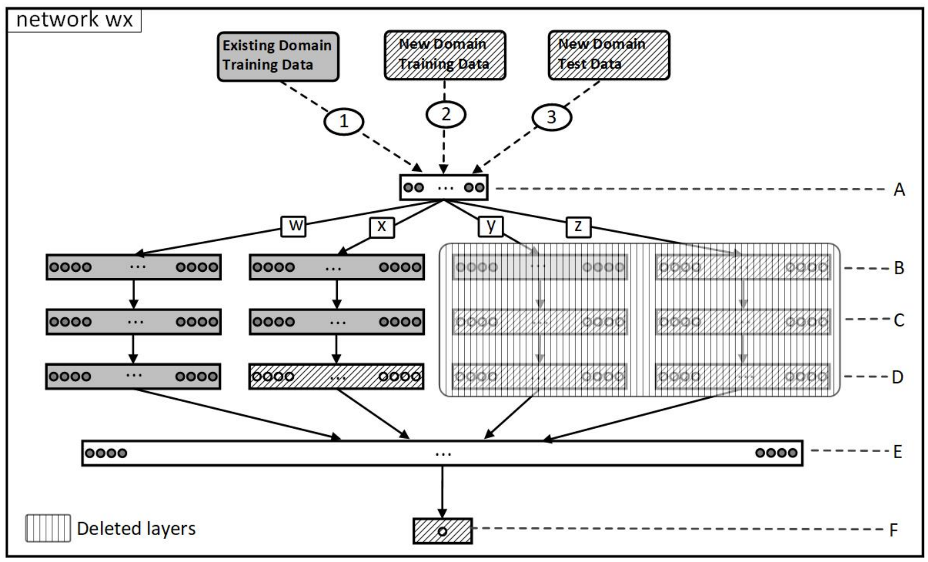

Fig. 5: Network wx

$$(Mfix(wx)_e, Mfree(wx)_e) = split(M(wx)_e) \tag{19}$$

$$Mfree(wx)_n = train(Mfree(wx)_e, P_n, t_n, F_n) \tag{20}$$

$$M(wx)_{transfer} = c(Mfix(wx)_e, Mfree(wx)_n) \tag{21}$$

5.5 Model wxy

Model wxy is developed based on Model wxyz, where only Network z is deleted. This network configuration is shown in Figure 6. In Model wxy, six hidden layers were trained using source domain data. Three hidden layers and the output layer were retrained using target domain data. The development of model wxy is shown in Equation 22, Equation 23, Equation 24.

$$(Mfix(wxy)_e, Mfree(wxy)_e) = split(M(wxy)_e) \tag{22}$$

$$Mfree(wxy)_n = train(Mfree(wxy)_e, P_n, t_n, F_n) \tag{23}$$

$$M(wxy)_{transfer} = c(Mfix(wxy)_e, Mfree(wxy)_n) \tag{24}$$

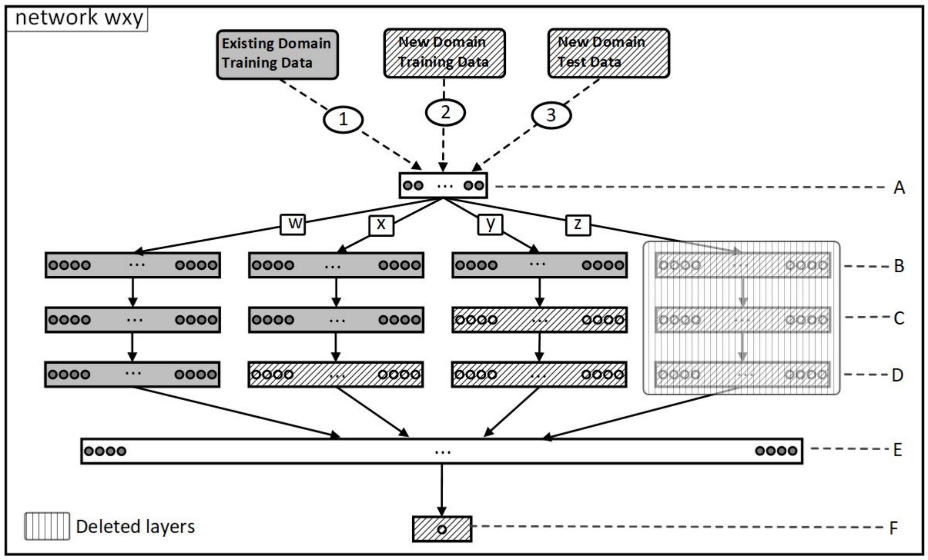

Fig. 6: Network wxy

6 Experiments

In our experiments, we used data based on lendingclub.com data, which is similar to our client's data, with a time range of 2007 to 2018, see Section 9. We first created base models - training them from scratch, without transfer learning. We applied a grid search to discover a near optimal set of hyper parameters for the Deep Learning (DL) structure. In these Transfer Learning experiments, we used Credit Card/Debt Consolidation (CD) as the source domain and Small Business (SB) as the target domain. Our goal is to transfer learning from CD to SB, as the data in SB is limited.

Table 2. Performance of Gradient Boosting Machine (GBM) and Deep Learning (DL) on Credit Card and Debt Consolidation (CD) and Small Business Loan (SB) datasets, evaluated using 10 fold cross validation

Name	Sampling	Gini GBM	Gini DL
$CreditCard/DebtConsolidation$	random	0.43 ±0.01	0.43 ±0.01
$SmallBusiness$	random	0.30 ±0.05	0.31 ±0.02

We validated the performance using CD and SB data extracts, by developing comparison models that used Gradient Boosting Machines (GBM). The comparison of performances is shown in Table 2. There is no significant difference between the performance of GBM and DL. The next experiments only focused on DL.

6.1 Experimentation datasets

We downloaded all datasets from lendingclub.com in mid October 2018. The datasets were filtered based on *purpose* (CD and SB) and *year* (2007 to 2018). 10 datasets were created from the downloaded data. The first extract was dataset $CD4$, time range: 2007 to 2018. The size was 100,000 records, extracted randomly from 940,948 records, where the loan purpose was paying Credit Card and Debt Consolidation. The bad debt rate from this dataset was 21%. Next was dataset $SB4$, time range: 2007 to 2018, the size was 13,794 records where the loan purpose was for investing in Small Business; this type of loan is riskier; the bad debt rate is 30%. No outlier filtering was performed for these two datasets. Datasets $CD1$, $CD2$, $CD3$ are subsets of dataset $CD4$, filtered based on different time ranges. Similarly datasets $SB1$, $SB2$, $SB3$ are subsets of dataset $SB4$. Dataset CCD is also a subset of dataset $CD4$, filtered on Credit Card Loans. Similarly the Car Loan data is extracted from the Lending Club datasets, see Section 9.

Table 3. List of datasets for transfer learning experiments, the type column indicates whether the dataset is used as the source or the target of the transfer learning process.

ID	Dataset	Period	Size	Type	Gini
CD1	$CreditCard/DebtConsolidation$	2007-2011	23,813	Source	0.364 ±0.023
SB1	$SmallBusinessLoan$	2007-2011	1,831	Target	0.272 ±0.067
CD2	$CreditCard/DebtConsolidation$	2007-2014	100,000	Source	0.417 ±0.016
SB2	$SmallBusinessLoan$	2007-2014	6,686	Target	0.274 ±0.040
CD3	$CreditCard/DebtConsolidation$	2007-2016	100,000	Source	0.447 ±0.013
SB3	$SmallBusinessLoan$	2007-2016	12,114	Target	0.331 ±0.032
CD4	$CreditCard/DebtConsolidation$	2007-2018	100,000	Source	0.448 ±0.012
SB4	$SmallBusinessLoan$	2007-2018	13,794	Target	0.351 ±0.024
CCD	$CreditCard$	2007-2018	100,000	Source	0.463 ±0.014
CAR	$CarLoan$	2007-2018	12,734	Target	0.436 ±0.036

All experiments were based on the data in Table 3. They were performed using 10 fold cross validation, repeated 5 times. The base model to be transferred was developed using the dataset $CD1$, $CD2$, $CD3$, $CD4$, CCD and the network configuration u, as shown in Figure 1 and defined in Equation 2. One factor that influenced model performance was the strength of signal [4] from the data. Table 3 also shows, the larger the dataset the higher the Gini; as the data is becoming more mature, it represents the real-world better.

[4] associated with the outcome being predicted

Similarly, the Gini for $SB1$, $SB2$, $SB3$, $SB4$ and CAR shown in Table 3 and Table 4 is calculated from the test result of model $M(u)_n$ by applying the function $g()$ on the test results, as defined in Equation 9.

6.2 Experimentation results

The experiment results are summarized in Table 4 where we applied Progressive Shifted Contributions (PSC) from the source to the target domain data. We found that $M(w)_{transfer}$ had the highest Gini of 0.301 a (10.7% improvement compared to $M(u)_n$) for the experiment over source/target:$CD1/SB1$. As the target data became more mature in $CD2/SB2$, $M(w)_{transfer}$ still had the highest Gini of 0.287, however the improvement was only 4.7%. As the target data became more mature in $CD3/SB3$ the contribution shifted toward the target data; model $M(wx)_{transfer}$ had the highest Gini of 0.337. It contributed marginal improvement (1.8%) over $M(u)_n$. Finally, in $CD4/SB4$, the contribution was completely shifted into the target, resulting in $M(u)_n$ having the highest Gini of 0.351.

The experiment on datasets CCD/CAR, $M(w)_{transfer}$ had the highest Gini of 0.447, a small improvement (2.5%) over $M(u)_n$. It shows the Car Loan dataset had similar maturity to the credit card dataset.

The experiments show the contribution was shifted from source to target as the target data matured. The contribution was based on the number of trainable layers using the source and target domain data. All these experiments were performed using 10 fold cross validation, repeated 5 times. That is, each experiment was repeated 50 times; we then record the average of the Gini scores and the standard deviation.

Table 4. Experimentation Results (source:Credit Card/Debt Consolidation, target:Small Business Loan), six models with progressively shifted contribution, built based on source and target datasets described in Table 3, the best performing models are marked with the symbol *.

	Source/Target				
Model	$CD1/SB1$	$CD2/SB2$	$CD3/SB3$	$CD4/SB4$	CCD/CAR
$M(v)_e$	0.157 ±0.022	0.236 ±0.051	-0.191 ±0.260	0.196 ±0.026	0.262 ±0.355
$M(w)_{transfer}$	*0.301 ±0.097	*0.287 ±0.051	0.334 ±0.029	0.350 ±0.029	*0.447 ±0.037
$M(wx)_{transfer}$	0.292 ±0.091	0.272 ±0.054	*0.337 ±0.030	0.350 ±0.028	0.434 ±0.035
$M(wxy)_{transfer}$	0.230 ±0.087	0.217 ±0.057	0.300 ±0.032	0.310 ±0.030	0.376 ±0.040
$M(wxyz)_{transfer}$	0.174 ±0.010	0.172 ±0.051	0.254 ±0.029	0.273 ±0.030	0.310 ±0.050
$M(u)_n$	0.272 ±0.067	0.274 ±0.040	0.331 ±0.032	*0.351 ±0.024	0.436 ±0.036
% improvement	10.7%	4.7%	1.8%	0.0%	2.5%

6.3 Additional Experiments

We investigated the hypothesis that the Gini performance improvement was due to the complexity of the network structure. We conducted experiments as described in Equation 25 and Equation 26. The model with network configuration wxyz was trained and retrained on source domain data. The performance of this model was 0.39 ±0.01, which is lower than the base model Gini 0.43 ±0.01. It showed that the additional complexity of $M(wxyz)_{transfer}$ did not improve Gini performance. The improvement was instead due to the diversity of the source data, complementing the target data.

$$Mfree(wxyz)_e = train(Mfree(wxyz)_e, P_e, t_e, F_e) \tag{25}$$

$$M(wxyz)_{retrain} = c(Mfix(wxyz)_e, Mfree(wxyz)_e) \tag{26}$$

7 Conclusion

We propose an algorithm to progressively shift the contribution from the source to target domains. The PSC algorithm lets us evaluate incremental complements of target domain data with source domain data. While we undertook some activities manually, the underlying goal has been to devise a framework that can automatically search for the optimum balance between the source and target domain data, generating the highest Gini score for that combination. Six PSC models were built, ranging from Model v (using source domain data only) all the way to Model u (using target domain data only) as described in Table 1.

8 Future Work

The presented research is part of a larger effort to develop transfer learning knowledge based systems. The presented experiment and results are the first of a series of experiments which will be used to discover and formulate a stream of rules. The rules will be incrementally incorporated in a knowledge base, following the Ripple Down Rule framework, specifically geared towards incremental construction of rule-based systems [1], [11].

To realise the knowledge based system, an appropriate representation of the transfer context and the transfer recommendations will first be needed, to enable appropriate encoding of rules within the system. To formulate the representation, we will need to identify an adequate set of features for the context transfer, requiring further experiments with additional source data, such as utility payments, taxation, etc. Through these experiments, we will also seek ways to accommodate different PSC levels from each data source, and assess target model Gini impact. Further, the representation will need to account for articulating the recommendations output from the rule-based system. We will also require new features to represent the following:

1. Domain adaptation, to adjust the variables/features before performing the transfer learning process.
2. Selection of optimization approaches, by assessing their impact on the target model Gini.

9 Software and Data

The software and steps to pre-process the data are available at the following Gitlab URL: https://gitlab.com/richdataco/rdc-public/rdc-ic/research/transfer-learning/ecmlpkdd2019.

The datasets for Credit Card, Debt Consolidation, Small Business Loan, Car Loan are available from https://www.lendingclub.com/info/download-data.action.

References

1. Beydoun, G., Hoffmann, A.: Theoretical basis for hierarchical incremental knowledge acquisition. International Journal of Human-Computer Studies **54**(3), 407–452 (2001)
2. Breiman, L.: Some properties of splitting criteria. Machine Learning **24**(1), 41–47 (1996)
3. Chattopadhyay, R., Ye, J., Panchanathan, S., Fan, W., Davidson, I.: Multi-source domain adaptation and its application to early detection of fatigue. In: KDD (2011)
4. Flach, P.A.: The geometry of roc space: understanding machine learning metrics through roc isometrics. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). pp. 194–201 (2003)
5. Forum, S.F.: Msme finance gap (Jan 2019), `https://www.smefinanceforum.org/data-sites/msme-finance-gap/`
6. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD (2008)
7. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks **22**(2), 199–210 (2011)
8. Pan, S.J., Yang, Q., et al.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)
9. Řezáč, M., Řezáč, F.: How to measure the quality of credit scoring models. Finance a úvěr: Czech Journal of Economics and Finance **61**(5), 486–507 (2011)
10. Sun, S., Xu, Z., Yang, M.: Transfer learning with part-based ensembles. In: MCS (2013)
11. Suryanto, H., Compton, P.: Invented predicates to reduce knowledge acquisition. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) Engineering Knowledge in the Age of the Semantic Web. pp. 293–306. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
12. Weiss, K.R., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data **3**, 1–40 (2016)
13. Xiao, J., Wang, R., Teng, G.E., Hu, Y.: A transfer learning based classifier ensemble model for customer credit scoring. 2014 Seventh International Joint Conference on Computational Sciences and Optimization pp. 64–68 (2014)
14. Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. In: Proceedings of the 15th ACM international conference on Multimedia. pp. 188–197. ACM (2007)