EMPOWERING CREDIT SCORING THROUGH REJECT-AWARE MULTI-TASK NETWORKS

Tirugulla neelima[1],Surkanti sravan kumar reddy[2],Nandigama sravan kumar[3]

[1]Associate Professor,M.Tech.,BRILLIANT GRAMMAR SCHOOL EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS-INTEGRATED CAMPUS

Abdullapurmet (v), hayath nagar (m), r.r dt. Hyderabad

[2]Assistant Professor,M.Tech.,BRILLIANT GRAMMAR SCHOOL EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS-INTEGRATED CAMPUS

Abdullapurmet (V), Hayath Nagar (M), R.R Dt. Hyderabad

Department of CSE,

[3]UG Students, BRILLIANT GRAMMAR SCHOOL EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS-INTEGRATED CAMPUS

Abdullapurmet (V), Hayath Nagar (M), R.R Dt. Hyderabad

ABSTRACT

In financial credit scoring, loan applications may be approved or rejected. We can only observe default/non-default labels for approved samples but have no observations for rejected samples, which leads to missing-not-at-random selection bias. Machine learning models trained on such biased data are inevitably unreliable. In this work, we find that the default/non-default classification task and the rejection/approval classification task are highly correlated, according to both real-world data study and theoretical analysis. Consequently, the learning of default/non-default can benefit from rejection/approval. Accordingly, we for the first time propose to model the biased credit scoring data with Multi-Task Learning (MTL). Specifically, we propose a novel Reject-aware Multi-Task Network (RMT-Net), which learns the task weights that control the information sharing from the rejection/approval task to the default/non-default task by a gating network based on rejection probabilities. RMT-Net leverages the relation between the two tasks that the larger the rejection probability, the more the default/non-default task needs to learn from the rejection/approval task. Furthermore, we extend RMT-Net to RMT-Net++ for modeling scenarios with multiple rejection/approval strategies. Extensive experiments are conducted on several datasets, and strongly verifies the effectiveness of RMT-Net on both approved and rejected samples. In addition, RMT-Net++ further improves RMT-Net's performances.

I. INTRODUCTION

CREDIT scoring aims to use machine learning methods to measure customers' default probabilities of credit loans [1] [2] [3] [4] [5] . Based on the evaluated credits, financial institutions such as banks and online lending companies can decide whether to approve or reject credit loan applications. When a customer applies for credit loan, his or her application may be approved or rejected. If the application is approved, it will become an approved sample, and the customer will get the loan. After a period, if the customer repays the credit loan timely, it will be a non-default sample; if the customer fails to timely repay, it will be a default sample. In contrast, if the application is not approved, it will become a rejected sample, and the customer will not get credit loan. Since a rejected sample gets no loans, we have no way to observe whether it will be default or non-default. Above process is illustrated in Fig. 1. Credit scoring models are usually constructed based on approved samples, as we have no ground-truth default/non-default labels for rejected samples [6] [7] [8] [9]. The rejection/approval strategies are usually machine learning models or expert rules based on the features of customers, thus approved and rejected samples share different feature distributions. This makes us face the missing-not-at-random selection bias in data [9] [10] [11]. However, when serving online, credit scoring models need to infer credits of loan applications in feature distributions of both approved and rejected samples. Training models with such biased data has severe consequences that the model parameters are biased [12], i.e., the predicted relation between input features and default probability is incorrect. Using such models on samples across various data distributions leads to significant economic losses [7] [13] [14]. Therefore, for reliable credit scoring, besides the modeling of approved samples, we also need to take rejected ones into consideration and infer their true credits [15].

In practice, machine learning models like Logistic Regression (LR), Support Vector Machines (SVM), Multi-Layer Perceptron (MLP) and XGBoost (XGB) are widely used for modeling credit scoring data. However, they are affected by the missing-not-at-random bias in data to produce reliable and accurate predictions. To tackle this problem, some existing approaches address the selection bias and conduct reject inference from multiple perspectives. Some approaches apply the self-training algorithm [16], which

iteratively adds rejected samples with higher default probability as default samples to retrain the model [17]. This is a semisupervised approach [18]. Besides, Semi-Supervised SVM (S3VM) [6] and Semi-Supervised Gaussian Mixture Models (SS-GMM) [7] are also deployed in credit scoring systems. In another perspective, some approaches attempt to re-weight the training approved samples to approximate unbiaseddata [14] [19] [20] [21]. These approaches are similar to counterfactual learning [10] [11] [22] [23], which attempts to re-weight observed samples to remove bias in data.

Though some of the above approaches have achieved relative improvements on some credit scoring datasets [7] [14], they cannot achieve optimal performances due to the lack of consideration of some key factors. Specifically, we find that the default/non-default classification task and the rejection/approval classification task are highly correlated in real credit scoring applications, according to both real world data study and theoretical analysis in Sec. 3. Intuitively speaking, with an effective credit approval system, rejected customers have higher default ratios, while approved customers have lower ones. Consequently, the learning of default/non-default can benefit from the learning of rejection/approval. Accordingly, it might be promising to incorporate Multi-Task Learning (MTL) [24] for modeling biased credit scoring data.

Nowadays, state-of-the-art MTL approaches mainly focus on adaptively learning weights of different tasks in a mixture-of-experts structure [25] [26] [27] [28] [29]. This makes task weights changing in different samples so that tasks can share useful but not conflict information adaptively. Such MTL approaches achieve promising performances in various scenarios. However, when we use state of- the-art MTL approaches for modeling the default/non default task and the rejection/approval task, we do not achieve satisfactory performances, and even achieve poor performances in default prediction on rejected samples. This may be because we have no observed default/non-default labels for rejected samples during model training. **The task weights, which decide how much information is shared between the two tasks, are not well optimized in the feature distribution of rejected samples**. Thus, exiting MTL approaches fail in modeling the biased credit scoring data, and we need a novel and specially-designed MTL approach.

Accordingly, we propose a **Reject-aware Multi-Task Network (RMT-Net)**. RMT-Net learns the weights that control the information sharing from the rejection/approval task to the default/non-default task by a gating network based on rejection probabilities. With larger rejection probability, less reliable information can be learned in the default/non-default network and more information is shared from the rejection/approval network. In this way, we can consider the correlation between rejected samples and default samples, as well as personalize the information sharing weights in the feature distribution of rejected samples. Furthermore, we consider cases with multiple rejection/ approval strategies, and extend RMT-Net to **RMT-Net++**, which models several rejection/approval classification tasks in the MTL framework.

In all, we verify RMT-Net and RMT-Net++ on 10 datasets under different settings, in which significant improvements are achieved for default prediction on both accepted and rejected samples. Evaluated by the commonly used Kolmogorov-Smirnov (KS) metric1 in credit scoring, comparing with conventional classifiers, i.e. LR, DNN, and XGB, RMT-Net relatively improves the performances by 47:9% on average. Comparing with the most competitive reject inference approaches, RMT-Net relatively improves the performances by 11:9% on average. In addition, we show in an extra experiment with multiple rejection/approval strategies that RMT-Net++ can further relatively improve the performances of RMT-Net by 5:8% on average.

The main contributions of this work are concluded:

_ We for the first time propose to model biased credit scoring data using an MTL approach, namely RMT Net. Instead of directly using conventional MTL approaches, we present several modifications to improve the poor performances of existing MTL approaches on credit scoring.

_ We further consider multiple rejection/approval strategies, and extend RMT-Net to RMT-Net++. In this way, our work suits different application scenarios in real applications.

_ Extensive experiments are conducted on 10 datasets under different settings. Significant improvements are achieved by our proposed RMT-Net approach on both accepted and rejected samples. In addition, we show that RMT-Net++ with multiple strategies can further improve the performances

The rest of the paper is organized as follows. In Section 2, we review some related work on reject inference, counterfactual learning and multi-task learning. Then we analyze the correlation between the default/non-default task and the rejection/approval task according to both real-world data study and theoretical analysis in Section 3. Sections 4 and 5 detail our proposed RMT-Net and RMT-Net++ under single strategy and multiple strategies respectively. In Section 6, we conduct empirical experiments to verify the effectiveness of RMT-Net and RMT-NET++. Section 7 concludes our work.

II. LITERATURE REVIEW:

RMT-Net: Reject-Aware Multi-Task Network for Modeling Missing-Not-At-Random Data in Financial Credit Scoring, Qiang Liu; Yingtao Luo; Shu Wu; Zhen Zhang; Xiangnan Yue; Hong Jin; Liang Wang, In financial credit scoring, loan applications may be approved or rejected. We can only observe default/non-default labels for approved samples but have no observations for rejected samples, which leads to missing-not-at-random selection bias. Machine learning models trained on such biased data are inevitably unreliable. In this work, we find that the default/non-default classification task and the rejection/approval classification task are highly correlated, according to both real-world data study and theoretical analysis. Consequently, the learning of default/non-default can benefit from rejection/approval. Accordingly, we for the first time propose to model the biased credit scoring data with Multi-Task Learning (MTL). Specifically, we propose a novel Reject-aware Multi-Task Network (RMT-Net), which learns the task weights that control the information sharing from the rejection/approval task to the default/non-default task by a gating network based on rejection probabilities. RMT-Net leverages the relation between the two tasks that the larger the rejection probability, the more the default/non-default task needs to learn from the rejection/approval task. Furthermore, we extend RMT-Net to RMT-Net++ for modeling scenarios with multiple rejection/approval strategies. Extensive experiments are conducted on several datasets, and strongly verifies the effectiveness of RMT-Net on both approved and rejected samples. In addition, RMT-Net++ further improves RMT-Net's performances.

III. EXISTING SYSTEM

Augmentation is a re-weighting approach [19] [20] [21], in which accepted samples are re-weighted to represent the entire distribution. A common way to achieve this is reweighting according to the rejection/approval probability. Moreover, the augmentation approach has been extended in a fuzzy way [14]. Parcelling is also a re-weighting approach, where the re-weighting is determined by the default probability by score-band that is adjusted by the credit modeler [8] [21]. To be noted, these re-weighting methods are similar to the researches on counterfactual learning [10] [11] [22] [23]. Counterfactual learning aims to remove data bias, in which the re-weighting of training samples is widely adopted.

Meanwhile, semi-supervised approaches are also applied to deal with the reject inference task. In [17], the authors use a self-training algorithm to improve the performance of SVM on credit scoring. Self-training, also known as self-labeling or decision-directed learning, is the most simple semi-supervised learning method [16] [30] [31]. This approach trains a model on approved samples, and labels rejected samples with largest default probabilities as default samples according to model predictions. Then, the newly labeled samples are added to retrain the model, and this process continues iteratively. Though the self-training algorithm is only used to promote SVM in [17], it can also promote other classifiers, such as LR, MLP and XGB. Besides, another semi-supervised version of SVM called S3VM [6] is also applied in reject inference. S3VM uses approved and rejected samples to fit an optimal hyperplane with maximum margin, but have problem in fitting large-scale data [7]. Meanwhile, earlier works have used some statistical machine learning methods, such as Expectation-Maximization (EM) algorithm [32], Gaussian Mixture Models (GMM) [33] and survival analysis [34], for reject inference. Based on GMM and inspired by semi-supervised generative models [35] [36], SS-GMM [7] is proposed for modeling biased credit scoring data. The counterfactual re-weighting and semi-supervised learning are the main methods for reject inference, but neither approach considers the correlation between the learning of rejection/approval and the learning of default/non-default.

MTL learns multiple tasks simultaneously in one model, and has been proven to improve performances

through information sharing between tasks [24] [26]. It has succeed in scenarios such as computer vision [29] [59] [60], recommender systems [25] [26] [27] [28] [61] [62], healthcare [63], and other prediction problems [64] [65]. The simplest MTL approach is hard parameter sharing, which shares hidden representations across different tasks, and only the last prediction layers are special for different tasks [24].

However, hard parameter sharing suffers from conflicts among tasks, due to the simple sharing of representations. To deal with this problem, some approaches propose to learn weights of linear combinations to fuse hidden representations in different tasks, such as Cross-Stitch Network [59] and Sluice Network [60]. However, in different samples, the weights of different tasks stay the same, which limits the performances of MTL. This inspires the research on applying gating structures in MTL [25] [26] [27] [66]. Mixture-Of-Experts (MOE) first proposes to share and combine several experts through a gating network [66]. Based on MOE, to make the weights of different tasks varying across different samples and to improve the performances of MTL, Multigate MOE (MMOE) [25] proposes to use different gates for different tasks. Progressive Layered Extraction (PLE) further extends MMOE, and incorporates multi-level experts and gating networks [26]. Besides, attention networks are also utilized for assigning weights of tasks according to different feature representations [28] [29].

Disadvantages

• The complexity of data: Most of the existing machine learning models must be able to accurately interpret large and complex datasets to detect Financial Credit Scoring.

• Data availability: Most machine learning models require large amounts of data to create accurate predictions. If data is unavailable in sufficient quantities, then model accuracy may suffer.

• Incorrect labeling: The existing machine learning models are only as accurate as the data trained using the input dataset. If the data has been incorrectly labeled, the model cannot make accurate predictions.

IV.PROPOSED SYSTEM

In the proposed system, the system proposes a **Reject-aware Multi-Task Network (RMT-Net)**. RMT-Net learns the weights that control the information sharing from the rejection/approval task to the default/non-default task by a

gating network based on rejection probabilities. With larger rejection probability, less reliable information can be learned in the default/non-default network and more information is shared from the rejection/approval network. In this way we can consider the correlation between rejected samples and default samples, as well as personalize the information sharing weights in the feature distribution of rejected samples. Furthermore, we consider cases with multiple rejection/ approval strategies, and extend RMT-Net to **RMT-Net++**, which models several rejection/approval classification tasks in the MTL framework.

In all, we verify RMT-Net and RMT-Net++ on 10 datasets under different settings, in which significant improvements are achieved for default prediction on both accepted and rejected samples. Evaluated by the commonly used Kolmogorov-Smirnov (KS) metric1 in credit scoring, comparing with conventional classifiers, i.e. LR, DNN, and XGB, RMT-Net relatively improves the performances by 47:9% on average. Comparing with the most competitive reject inference approaches, RMT-Net relatively improves the performances by 11:9% on average. In addition, we show in an extra experiment with multiple rejection/approval strategies that RMT-Net++ can further relatively improve the performances of RMT-Net by 5:8% on average.

Advantages

_ We for the first time propose to model biased credit scoring data using an MTL approach, namely RMTNet. Instead of directly using conventional MTL approaches, we present several modifications to improve the poor performances of existing MTL approaches on credit scoring.

_ We further consider multiple rejection/approval strategies, and extend RMT-Net to RMT-Net++. In this way, our work suits different application scenarios in real applications.

_ Extensive experiments are conducted on 10 datasets under different settings. Significant improvements are achieved by our proposed RMT-Net approach on both accepted and rejected samples. In addition, we show that RMT-Net++ with multiple strategies can further improve the performances.

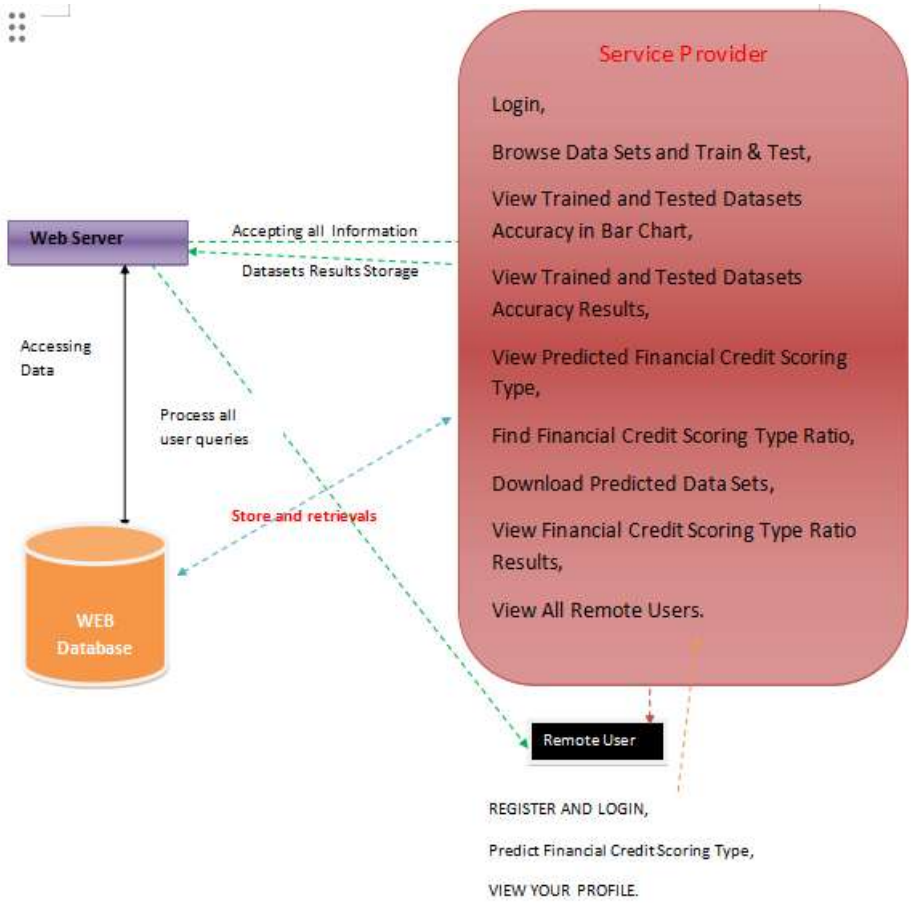

Fig1: System diagram

V. MODULES :

1.Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Browse Data Sets and Train & Test, View Trained and Tested Datasets Accuracy in Bar Chart, View Trained and Tested Datasets Accuracy Results, View Predicted Financial Credit Scoring Type, Find Financial Credit Scoring Type Ratio, Download Predicted Data Sets, View Financial Credit Scoring Type Ratio Results, View All Remote Users.

2.View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

3. Remote User

4. In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration

successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, Predict Financial Credit Scoring Type, VIEW YOUR PROFILE.

VI. ALGORITHAMS:

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

- Simple, but a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given

- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (**Weka 3.6.0**, **R 2.9.2**, **Knime 2.1.1**, **Orange 2.0b** and **RapidMiner 4.6.0)**. We try above all to understand the obtained results.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a

multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

VII. CONCLUSION

In this paper, we focus on modeling biased credit scoring data, in which we have only ground-truth labels for approved samples and no observations for rejected samples. Such bias affects the reliability of default prediction, and we aim to improve the prediction accuracy on both approved and rejected samples. We find that the default/non-default classification task and the rejection/approval classification task are highly correlated in credit scoring applications, according to both real-world data study and theoretical analysis. We for the first time propose to model biased credit scoring data using an MTL framework, and propose a novel RMT-Net approach, which learns the task weights that control the information sharing from the rejection/approval task to the default/non-default task by a gating network based on rejection probabilities. According to empirical experiments on 10 datasets under different settings, RMT Net improves the poor performances of existing MTL approaches, and significantly outperforms several state-of-the art approaches from different perspectives. Furthermore, we extend RMT-Net to RMT-Net++ for modeling scenarios with multiple rejection/approval strategies. According to an extra experiment, RMT-Net++ with multiple strategies can further improve the performances of RMT-Net in a more complex multi-policy scenario.

VIII. REFERENCES

[1] D. West, "Neural network credit scoring models," Computers &

Operations Research, vol. 27, no. 11-12, pp. 1131–1152, 2000.

[2] B. Hu, Z. Zhang, J. Zhou, J. Fang, Q. Jia, Y. Fang, Q. Yu, and Y. Qi,
"Loan default analysis with multiplex graph learning," in CIKM, 2020, pp. 2525–2532.

[3] Y. Liu, X. Ao, Q. Zhong, J. Feng, J. Tang, and Q. He, "Alike and
unlike: Resolving class imbalance problem in financial credit risk assessment," in CIKM, 2020, pp. 2125–2128.

[4] Q. Liu, Z. Liu, H. Zhang, Y. Chen, and J. Zhu, "Mining cross features for financial credit risk assessment," in CIKM, 2021, pp. 1069–1078.

[5] D. Babaev, M. Savchenko, A. Tuzhilin, and D. Umerenkov, "Et-rnn:
Applying deep learning to credit loan applications," in KDD, 2019.

[6] Z. Li, Y. Tian, K. Li, F. Zhou, and W. Yang, "Reject inference in credit scoring using semi-supervised support vector machines," Expert Systems with Applications, vol. 74, pp. 105–114, 2017.

[7] R. A. Mancisidor, M. Kampffmeyer, K. Aas, and R. Jenssen, "Deep generative models for reject inference in credit scoring," Knowledge-Based Systems, 2020.

[8] A. Ehrhardt, C. Biernacki, V. Vandewalle, P. Heinrich, and S. Beben, "Reject inference methods in credit scoring," Journal of Applied Statistics, pp. 1–21, 2021.

[9] N. Goel, A. Amayuelas, A. Deshpande, and A. Sharma, "The importance of modeling data missingness in algorithmic fairness: A causal perspective," in AAAI, 2021, pp. 7564–7573.

[10] B. M. Marlin and R. S. Zemel, "Collaborative prediction and ranking with non-random missing data," in RecSys, 2009, pp. 5–12.

[11] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation," in ICML, 2016, pp. 1670–1679.

[12] M. B¨ ucker, M. van Kampen, and W. Kr¨amer, "Reject inference in consumer credit scoring with nonignorable missing data," Journal of Banking & Finance, vol. 37, no. 3, pp. 1040–1045, 2013.

[13] G. G. Chen and T. Astebro, "The economic value of reject inference in credit scoring," Department of Management Science, University of Waterloo, 2001.

[14] H.-T. Nguyen et al., "Reject inference in application scorecards: evidence from france," University of Paris Nanterre, EconomiX, Tech. Rep., 2016.

[15] D. J. Hand and W. E. Henley, "Can reject inference ever work?" IMA Journal of Management Mathematics, vol. 5, no. 1, pp. 45–55, 1993.

[16] A. Agrawala, "Learning with a probabilistic teacher," IEEE Transactions on Information Theory, vol. 16, no. 4, pp. 373–379, 1970.

[17] S. Maldonado and G. Paredes, "A semi-supervised approach for reject inference in credit scoring using svms," in ICDM, 2010, pp. 558–571.

[18] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," Synthesis lectures on artificial intelligence and machine learning, vol. 3, no. 1, pp. 1–130, 2009.

[19] D. C. Hsia, "Credit scoring and the equal credit opportunity act," Hastings LJ, vol. 30, p. 371, 1978.

[20] J. Banasik and J. Crook, "Reject inference, augmentation, and sample selection," European Journal of Operational Research, vol. 183, no. 3, pp. 1582–1594, 2007.

[21] J. Banasik, J. Crook, and L. Thomas, "Sample selection bias in credit scoring models," Journal of the Operational Research Society, vol. 54, no. 8, pp. 822–832, 2003.

[22] N. Hassanpour and R. Greiner, "Counterfactual regression with importance sampling weights." in IJCAI, 2019, pp. 5880–5887.

[23] H. Zou, P. Cui, B. Li, Z. Shen, J. Ma, H. Yang, and Y. He, "Counterfactual prediction for bundle treatment," NeurIPS, 2020.

[24] R. Caruana, "Multitask learning," Machine learning, vol. 28, no. 1, pp. 41–75, 1997.

[25] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixtureof-experts," in KDD, 2018, pp. 1930–1939.

[26] H. Tang, J. Liu, M. Zhao, and X. Gong,