ENHANCED RECURRENT NEURAL NETWORK FOR COMBINING STATIC AND DYNAMIC FEATURES FOR CREDIT CARD DEFAULT PREDICTION

Te-Cheng Hsu[*]*, Shing-Tzuo Liou*[*]*, Yun-Ping Wang, Yung-Shun Huang, Che-Lin*[†]

Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

{andy810436, nick860617, tommt8541, vito31733}@gmail.com, [†]che.lin@gmail.com

ABSTRACT

Deep learning models have been shown to be capable of extracting high-level representations from the increasing amount of customer-level data generated via fast-growing financial activities. In financial data, dynamic features that evolve with time are commonly observed. However, such time dependencies are often ignored in classical classification models. In this study, we propose to learn a Recurrent Neural Network (RNN) feature extractor with GRU on credit card payment history to leverage the time dependencies embedded in these dynamic features. Input sequences are first preprocessed by this feature extractor. The extracted dynamic features along with the static features are then utilized to train an enhanced RNN model (RNN-RF) to predict credit card client defaults. Numerical experiments confirmed that the enhanced RNN predictor indeed provides the best performance in both lift index (0.659) and AUC (0.782) compared to the other benchmark models. The proposed model allows us to effectively combine static and dynamic features to provide superior predictive performance for financial data.

Index Terms— Deep learning, recurrent neural network, lift index, credit card client default, risk management

1. INTRODUCTION

Risk management has been an important issue in modern financial systems [1–5]. The main focus of risk management lies in reducing damage and uncertainty to banks via assessing the borrower's ability to repay [6]. Accumulated credit card debt and emerging delinquency are huge challenges to both banks and card holders nowadays.

Fast growing customer-level financial data makes it possible to incorporate big data analytics into building intelligent decision-making systems for banks. However, time dependencies embedded in behavioral data are often ignored in statistical models for traditional risk prediction. Deep learning models are popular owing to their strong ability to extract high-level features from a huge amount of raw data. Among those models, Recurrent Neural Networks (RNN) are specifically designed to use recursive architecture to extract patterns from input sequences [7]. They have been proven useful in applications that heavily rely on sequence (time-variant) features such as ChatBot [8] and sentiment analysis [9, 10]. As a result, it is natural to consider RNN models as feature extractors for customer behavior that often appear as sequences in financial data.

Many researches were conducted over risk management [2–5]. Rosenberg and Gleit performed credit evaluation with many static and dynamic models [2]. Hand and Henley further classified applicants into "good" and "bad" risk classes [3]. Graphical models were also introduced to provide inference over a pre-defined grpah generated from financial theories. Poalo utilized Markov Chain Monte Carlo (MCMC) over conditional independence graphs under a Bayesian framework [4]. Attigeri et al assessed credit risks with supervised machine learning algorithms and evaluated them with Chi-squared tests [5]. Yeh and Lien proposed Sorting Smoothing Method (SSM) to estimate default probabilities of credit card clients with six data mining methods [1]. However, time dependencies in data were still ignored, and few works utilized deep models to conduct automatic feature selection/extraction.

Receiver Operating Characteristics (ROC) curves and Area Under the ROC Curve (AUC) are commonly adopted as performance evaluation criteria in classification tasks. However, in risk management applications, making perfect predictions for every customer provides few advantages. In practice, preventing most of the potential cost resulted from default clients can be achieved through ranking and identifying the top 10% - 20% clients that will probably default afterwards. For this, lift index [11] has been introduced as an appropriate model performance evaluation metric in such ranking problems which provides an intuitive and practical perspective towards risk management.

In this paper, we proposed a novel model that combine the strong dynamic feature extraction capability from RNN with Random Forest (RF). We obtained superior performance to benchmark models evaluated on an open dataset from the UCI Machine Learning Repository [1] for credit card clients defaults. We demonstrated that our proposed model, RNN-

[*]These authors contributed equally to this work.

[†]Corresponding authors.

RF, performed the best in terms of lift index and AUC among models widely-used in current risk management systems implemented in banks or financial holding companies. We offered a solution that can help guide risk management strategies, boost consumer finance confidence, and lower damage and uncertainty significantly by identifying high risk customers who are likely to default in their credit card payment.

2. RISK MEASUREMENT

To appropriately depict performance evaluation in the risk management for credit card default, we introduce lift (or lift ratio) [11, 12]. Lift has been widely adopted in applications such as modeling customer churn, subscription renewal, and promotion targeting [13, 14]. If a prediction model can rank the customers bases on their credit card default risk, we can write lift (ratio) as a function of n, the number of customers in the top-scoring group, as

$$\text{lift}(n) = \frac{\text{\#true buyers in the top-}n\text{ scoring customers}}{n}. \quad (1)$$

Corresponding lift chart can then be plotted from (1) with the vertical and horizontal axes being (normalized) true buyers count and (normalized) number of customers visited, respectively. We can summarize lift charts into a single number called lift index [11] which reflects how well models can "skim the cream" and, more importantly, provides us a convenient way to compare performance among candidate models. Lift index is defined as

$$\text{lift index} = \sum_{i=1}^{10} w_i \times S_i. \quad (2)$$

where S_i's are the decile lift in decile i and w_i's are prespecified weights. Lift index can be viewed as a weighted average of area under the lift curve, where weights are usually larger in the top deciles since we care more about customers with the highest probabilities of default and apply risk management on them, where each decile consists of 10% of total customers.

3. PROPOSED ENHANCED RNN PREDICTOR

Classical models do not utilize time dependencies embedded in features to learn compact representations. These underlying time dependencies enable us to train deep architectures even with less samples available [7]. To better model time sequences and leverage time dependencies commonly appear in credit card payment history, we proposed to use a RNN feature extractor with GRU [7] to extract such dynamic features (see Fig. 1a). Another variation of basic RNN cells, Long-Short Term Memory (LSTM) [7], was not adopted here due to its complexity (see Sec. 4.3.1). The proposed model, RNN-RF, is illustrated in Fig. 1b.

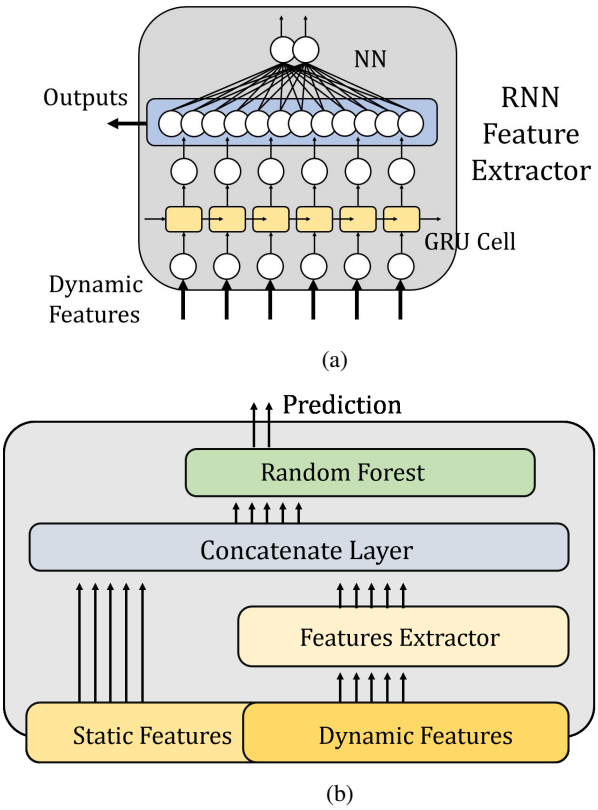

Fig. 1: Model architecture for RNN-RF; (a) RNN with GRU units as a feature extractor for dynamic features, (b) The proposed RNN-RF model that combines dynamic and static features.

GRU was first pre-trained with the ith client's credit card payment history for the jth month, denoted as $\mathbf{c}_i^{(j)}, i = 1, 2, \ldots, K$, $j = 1, 2, \ldots, N$. Note that K denotes the total number of customers, and $N = 6$ so a total of 6 months of payment history were included in the training of our model. Let $\mathbf{z}_i^{(j)}$ denote the hidden state activation values at the jth month for the ith client. Labels were transformed into one-hot vectors, and we use $\mathbf{y}_i$ to denote the ith client's true label. A neural network reads in all activation values, $Z_i = [\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(6)}]$, to predict credit card default label, $\hat{\mathbf{y}}_{pre,i}$, for pre-training:

$$\hat{\mathbf{y}}_{pre,i} = \sigma\left(WZ_i + b\right), \quad (3)$$

where $\sigma(\cdot)$ is a sigmoid function , and W and b are the weights and biases of the neural network, respectively. The loss function for pre-training our GRU feature extractor can then be formulated as follows:

$$\mathcal{L}_{pre} = -\sum_i \log p_\theta\left(\mathbf{y}_i | \{\mathbf{c}_i^{(1)}, \ldots, \mathbf{c}_i^{(6)}\}\right), \quad (4)$$

where θ denotes all the parameters in the network, and $p_\theta\left(\mathbf{y}_i | \{\mathbf{c}_i^{(1)}, \ldots, \mathbf{c}_i^{(6)}\}\right)$ can be readily given by reading

the entry from the output vector $\hat{\mathbf{y}}_i$. Activation values of hidden states of the GRU cells, as well as demographic (static) features, were taken as a new feature set that was further fed into a RF predictor to make the final prediction (see Fig. 1b). Denote the static features of the ith client as $\mathbf{x}_i$, then the final prediction can be obtained as:

$$\hat{\mathbf{y}_i} = \mathcal{RF}\left(\mathbf{x}_i, \mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(6)}\right). \quad (5)$$

Our model combines the strength of both models and yields the best performance in terms of both lift index and AUC.

4. EXPERIMENT RESULTS AND DISCUSSION

4.1. Dataset

We adopted an open dataset from UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`) [1] that records customers credit card payment history. Customers' past 6-month credit card behavior (e.g., bill amount, payment amount, delayed payments, etc.) is provided along with their demographic information (e.g., gender, marital status, education level, etc.). In summary, this dataset consists of 30,000 samples with 23 features (seperated into 5 static features and 18 dynamic features) for each customer and a binary label that indicates wether the customers default in the next month.

We partitioned the data into testing and training sets with an 30-70 split. Our proposed RNN-RF was compared with all other benchmark models via lift/ROC curve, lift index, and AUC. Several classical models [15] were chosen as benchmarks, including logistic regression (LR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and classical RF. We simply concatenated dynamic and static features and fed them into these benchmark models where time dependencies were not exploited.

4.2. Data preprocessing and model settings

To overcome class imbalance, we adopted Synthetic Minority Oversampling TEchnique (SMOTE) [16] on the training set. All columns were first standardized with the training set distribution before training, except for random forest that is rather insensitive to normalization [12]. We further added L2-regularization to LR to prevent overfitting. Radial Basis Function (RBF) was selected as the kernel for SVM. For RNN, we use a single layer of GRU. To calculate lift index, we used $1.0, 0.9, \ldots, 0.1$ for the first, second, $\ldots$, tenth decile, respectively.

4.3. Model comparison

4.3.1. RNN cells and various combinations

In simulations, GRU performed better in terms of lift index than LSTM when trained on only dynamic features (GRU units having 2% performance gain compared to LSTM). Both GRU and LSTM achieved worse lift if static features were also included. It is interesting to see that static and dynamic features need to be carefully combined for better performance. We adopted KNN, LR, SVM, DNN, and RF to leverage static features which were latter combined with GRU to form a hybrid model. Fig. 2 confirmed that it is best to combine static and dynamic features with our proposed RNN-RF model. Specifically, lift index of RNN-RF is higher than RNN-KNN, RNN-LR, RNN-SVM, RNN-DNN by 0.103, 0.099, 0.042, and 0.024, respectively.

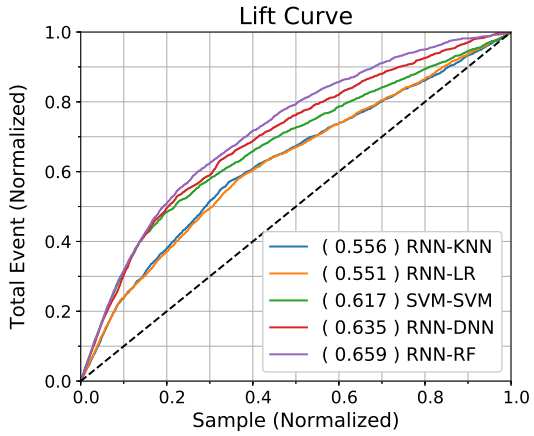

Fig. 2: Lift curves for various combinations of hybrid models.

4.3.2. Performance of enhanced RNN-RF model

Fig. 3 presents the comparison of our enhanced RNN-RF model with other benchmark models. It can be noted that our RNN-RF model achieved the maximum lift index that is higher than KNN, LR, SVM, RF by 0.062, 0.057, 0.024, and 0.011, respectively. We also plotted decile lifts for the top-3 (RNN-RF, SVM, and RF) models in Fig. 4 for a more detailed evaluation. RNN-RF was significantly better than RF and SVM in terms of decile lifts in the top 10% clients, the most relevant ones for risk management. This again verifies leveraging time dependencies can further boosts model ranking ability.

ROC curves and AUCs of RNN-RF and other benchmark models were shown in Fig. 5. RNN-RF clearly had superior performance.

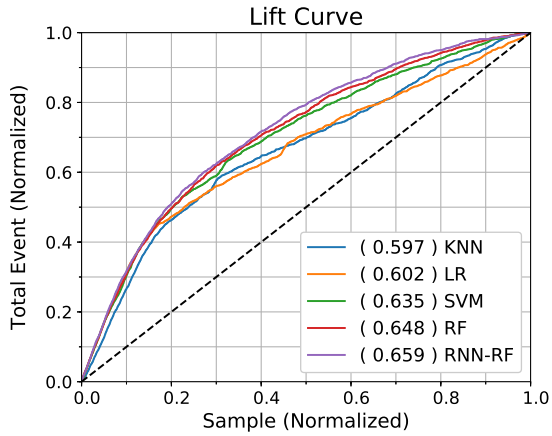

Fig. 3: Lift curves and lift indices for RNN-RF and benchmark models.

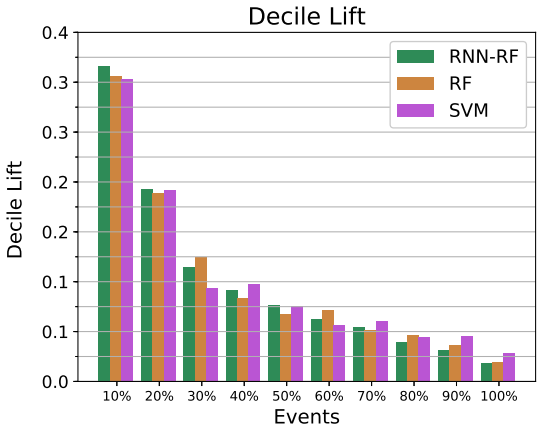

Fig. 4: Decile Lifts for RNN-RF, RF and SVM.

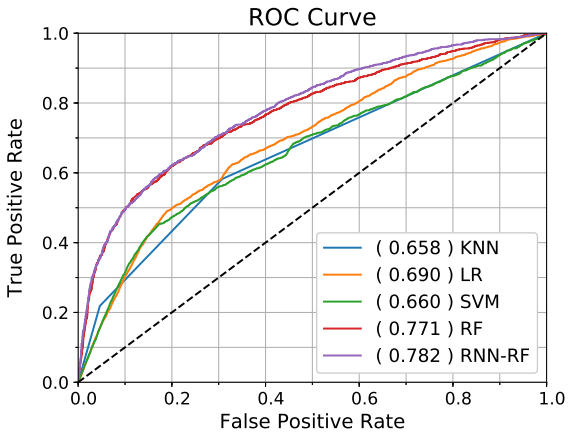

Fig. 5: ROC curves and AUCs for RNN-RF and benchmark models.

4.3.3. Varying the amount of training data

To further investigate the advantage of our proposed model when more training data is available, we varied the amount of training data available for RNN-RF, RF, and RNN-GRU, where RNN with GRU cells was used to handle both dynamic and static features. As we can observe in Fig. 4, lift indices of RNN-RF and RNN-GRU grew steadily as more data is available while classical RF did not improved further and saturated, which indicates that RNN-RF may continue to improve and eventually out-performs RF significantly when enough data is available.

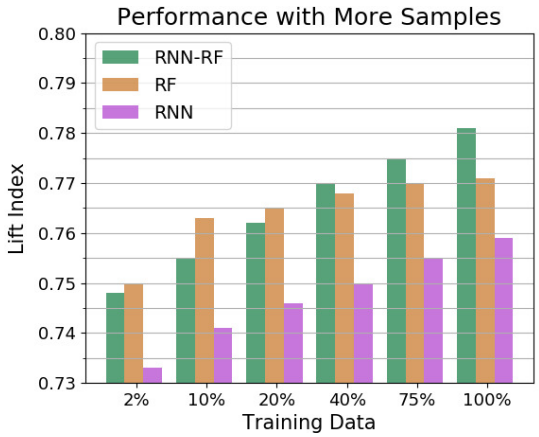

Fig. 6: Lift indices for RNN-RF, RF and RNN-GRU with growing amounts of training data.

5. CONCLUSION

With the introduction of RNN model as dynamic feature extractor, we were able to leverage time information embedded in data. RNN-RF is the best-performing one among benchmark models both in terms of lift index and AUC (Table 1). Furthermore, we found that by increasing the number of samples available for training, we obtained better results while the RF didn't. We therefore expect to achieve even better performance when huge amount of customer-level financial data is available. Lift index was introduced as a model performance evaluation metric to genuinely express model ranking ability which is closely aligned to practical risk management activities.

	KNN	LR	SVM	RF	RNN-RF
Lift Index	0.597	0.602	0.635	0.648	**0.659**
AUC Score	0.658	0.690	0.660	0.771	**0.782**
ACC Score	0.648	0.634	0.652	0.798	**0.802**

Table 1: Performance evaluation for RNN-RF and benchmark models.

Acknowledgements

We are grateful for feedback and support on lift index from Prof. Galit Shmueli in the Institute of Service Science, National Tsing Hua University.

6. REFERENCES

[1] I-Cheng Yeh and Che-hui Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[2] Eric Rosenberg and Alan Gleit, "Quantitative methods in credit management: a survey," *Operations research*, vol. 42, no. 4, pp. 589–613, 1994.

[3] David J Hand and William E Henley, "Statistical classification methods in consumer credit scoring: a review," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 160, no. 3, pp. 523–541, 1997.

[4] Paolo Giudici, "Bayesian data mining, with application to benchmarking and credit scoring," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 1, pp. 69–81, 2001.

[5] Girija V Attigeri, MM Pai, and Radhika M Pai, "Credit risk assessment using machine learning algorithms," *Advanced Science Letters*, vol. 23, no. 4, pp. 3649–3653, 2017.

[6] Ken Brown and Peter Moles, *Credit Risk Management*, Great Britain, 2nd edition edition, 2016.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[8] Shang-Yu Su, Kai-Ling Lo, Yi Ting Yeh, and Yun-Nung Chen, "Natural language generation by hierarchical decoding with linguistic patterns," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, vol. 2, pp. 61–66.

[9] Reeshad Khan and Omar Sharif, "A Literature Review on Emotion Recognition Using Various Methods," *Global Journal of Computer Science and Technology*, vol. 17, no. 1, Apr. 2017.

[10] Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young, "Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking," *arXiv:1508.01755 [cs]*, Aug. 2015.

[11] Charles X. Ling and Chenghui Li, "Data mining for direct marketing: Problems and solutions.," in *KDD*, 1998, vol. 98, pp. 73–79.

[12] Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, and Kenneth C. Lichtendahl Jr, *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*, John Wiley & Sons, Sept. 2017.

[13] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, Apr. 2009.

[14] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche, "Evaluation measures for models assessment over imbalanced datasets," *J Inf Eng Appl*, vol. 3, no. 10, 2013.

[15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics New York, NY, USA:, 2001.

[16] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.