Cost-sensitive Parallel Learning Framework for Insurance Intelligence Operation

Xinxin Jiang, Shirui Pan, *Member, IEEE*, Guodong Long, Fei Xiong, Jing Jiang, and Chengqi Zhang

***Abstract*—Recent advancements in artificial intelligence (AI) are providing the insurance industry with new opportunities to create tailored solutions and services based on newfound knowledge of consumers, and the execution of enhanced operations and business functions. However, insurance data is heterogeneous, and imbalanced class distribution with low frequency and high dimensions presents four major challenges to machine learning in real-world business. Traditional machine learning algorithms can typically only be applied to standard data sets, which are normally homogeneous and balanced. In this paper, we focus on an efficient cost-sensitive parallel learning framework (CPLF) to enhance insurance operations with a deep learning approach that does not require pre-processing. Our approach comprises a novel, unified, end-to-end cost-sensitive parallel neural network that learns real-world heterogeneous data. A specifically-designed cost-sensitive matrix then automatically generates a robust model for learning minority classifications, and the parameters of both the cost-sensitive matrix and the hybrid neural network are alternately but jointly optimized during training. We also study the CPLF-based architecture for a real-world insurance intelligence operation system, and demonstrate fraud detection experiments on this system. The results of comparative experiments on real-world insurance data sets reflecting actual business cases demonstrate the effectiveness of our design.**

***Index Terms*—deep learning, heterogeneous data, imbalanced data, insurance operation, neural network**

I. Introduction

INSURANCE, companies have historically mainly achieved the significant performance differentiation by combining scale of exposures and underwriting expertise. The most important functions of insurance operations consist of: marketing, underwriting, reinsurance, legal and regulatory issues, claims adjustment, policy management, customer service, actuarial analysis and investments [1] [2].

Manuscript received May 16, 2018; revised August 22, 2018; accepted September 9, 2018. This research was funded by the Australian Government through the Australian Research Council (ARC) under grants 1) LP160100630 partnership with Australia Government Department of Health and 2) LP150100671 partnership with Australia Research Alliance for Children and Youth (ARACY) and Global Business College Australia (GBCA). (*Corresponding authors*: *Shirui Pan* and *Fei Xiong*.)

X. Jiang, S. Pan, G. Long, J. Jiang, C. Zhang are with Centre for Artificial Intelligence, FEIT, University of Technology Sydney, NSW 2007, Australia (E-mail: xinxin.jiang@student.uts.edu.au; shirui.pan@uts.edu.au; guodong.long@uts.edu.au; jing.jiang@uts.edu.au; chengqi.zhang@uts.edu.au).

Fei Xiong is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China. (E-mail: xiongf@bjtu.edu.cn).

As we enter to the big data era, developing an organization capability of utilizing huge accumulated data is critical for continued existence and profitability of insurers [3] [4]. With the increasing computational power of modern technologies, machine learning algorithms, and deep learning algorithms in particular, began to consistently win image, text, speech recognition contests, and excel in business data analysis. Driven by this trend, insurance operation can thus benefit greatly from the recent advances in artificial intelligence and machine learning. Some insurers use machine learning methods to analyze a variety of data to lower costs and improve profitability in their business. For example, they may apply the analyzed results to the underwriting process, assisting agents to sort through vast data sets collected by insurance companies to identify high risk cases, thus potentially reducing the number of claims.

Ongoing changes in technology, demography, and consumer needs and expectations continue to challenge the insurance industry. The opportunities open to insurers relate to using technology to enhance operations and execute business functions. For example, some insurers have used AI technology to enhance internal operations, which has improved efficiency and automated existing customer-facing, underwriting and claims processes. Examples can be found in usage-based and personalized insurance which leverages technology and data to develop new risk models based on behavioral factors. This also has the potential to drive radical change. However, data-driven insurance operation using artificial intelligence is still quite difficult to achieve for the most insurance companies in practice. The following four key challenges place much greater emphasis on applying artificial intelligence and machine learning approaches in insurance operation.

A. Low Frequency Transactions

Compared to other financial business transactions such as retail banking in which a customer's bank account usually transacts with high frequency, generating a long sequence by which customer behaviors can be traced, the frequency of transactions and customer contact in the insurance industry is much lower. In insurance industry, contacting the customer only once a year to notify a renewal premium is commonplace. Other transactions in a policy life cycle, such as customer services and claims, also occur with low frequency. This ignores the fact that tracing customer contacts or transactions might be significant for insurance operations. A customer taking the initiative to contact the insurance company represents a significant behavior, such as payment, surrender, complaint or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2018.2873526, IEEE Transactions on Industrial Electronics

claim, therefore deep analysis of customer transaction data will provide deeper insights into the company.

B. High Dimension Properties

Insurance companies collect an abundance of customer information, product information, and policy information to control the insured risks. In insurance data sets, the common dimensions of insurance customer and policy related objects can exceed ten thousand properties. Moreover, the values of some optional dimensions, such as a customer's insurance history, might be missing.

In insurance data sets, data density and data quality of different records can be quite different. It is very challenging to choose the most effective, correlated properties of a business target for data pre-processing on. For instance, the ratio of height and weight may be an important factor in analyzing underwriting risk, whereas customer income (i.e. payment capacity) may be a more important feature in the analysis of renewal probability. Effectively utilizing the most correlated features among high dimension properties is an important factor in insurance data analysis.

C. Heterogeneous Data Structure

Heterogeneity is key characteristics inside enterprise data to meet different enterprise requirements throughout an organization. Typical heterogeneous insurance objects can be specified by two elements, as shown in Fig.1. The description defines the properties of object that do not change over time, and the sequence records the transactions that occur over the timeline of the object's lifecycle.

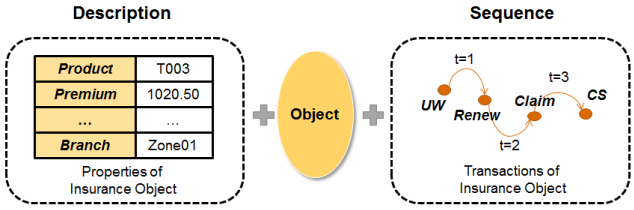

Fig. 1. An example of heterogeneous data sets in Insurance

Heterogeneous insurance data is divided into two main levels: the data-level and the structure-level [5] [6] [7]. The data-level heterogeneity is composed of data objects which contain various types of data (e.g., a simple data structure that contains different data types such as integer, float and character), while the structure-level heterogeneity combines various types of data formats and data sources (e.g., a complex data structure that combines the static properties for object description, and the dynamic transactions for time series activities). Heterogeneous data usually occupy different positions in data space, thus using a single learning method or one projection to extract patterns from heterogeneous data would not provide a useful comparison [8] [9] [10] [11]. How to use this heterogeneous information effectively in the insurance industry is a very worthwhile question.

D. Minority Classification on Business Targets

Insurance operation focuses on the minority classes rather than the balanced classification associated with traditional techniques. Leading insurers are retooling the role of their risk function from one of incident response and compliance to that of becoming an essential partner in advancing the business strategy. The risks are usually small probability events, which means that analyzed business targets usually focus on minority classification in the insurance industry. For instance, the incidence claim of term life insurance is usually only 0.3-0.5 per thousand, and the incidence of major illness claims is usually only 1-3 per thousand. These small probability rates lead to the insurance samples being extremely imbalanced for operation optimization topics. Compared to traditional balanced classification, the imbalanced class distribution of outcome is usually extremely skewed [12] [13]. Taking fraud detection as an example, it is evident that fraudulent transactions are significantly less than normal transactions, such that one class is severely unrepresented compared to the others.

To take advantage of AI opportunities, the insurance companies require an effective machine learning framework to overcome these challenges and make meaningful connections in insurance operations. In contrast to traditional machine learning, deep learning approaches have the advantage of being able to extract features and nonlinear correlations without relying on econometric assumptions and human expertise [14] [15] [16] [17] [18]. Despite the recent advancements in deep learning with real-world heterogeneous and imbalanced data sets, applying deep neural networks to insurance applications still poses many challenges. Hybrid neural networks (HNNs), which combine the strengths of a variety of neural networks, have been the subject of interest in the fields of computer vision and natural language processing [14] [15] [19], but most current HNNs attempt to solve classification problems via a two-step serial framework [14] [19]. The serial hybrid architecture is better adapted to specific learning areas, feeding the processed data in turn to utilize the advantages of each neural network without considering what kind of data it is more suitable for processing. However, as yet, such hybrid architectures have not been well-studied with real-world heterogeneous and imbalanced data. In practical data analysis, the serial framework usually requires more data pre-processing and lacks the efficiency in data learning required to meet real-world business demands.

This study therefore focuses on learning and trend forecasting with imbalanced, heterogeneous data via deep networks. Our approach involves learning descriptions and sequences in heterogeneous data and adjusting the cost-sensitive matrix of imbalanced classifications, through an end-to-end cost-sensitive parallel learning framework (CPLF). To avoid the heavy pre-processing and inefficiency problems associated with training heterogeneous data in traditional serial hybrid neural network architectures, CPLF consists of multiple neural networks, such as a multi-layer perceptron (MLP) [20] and long short-term memory (LSTM) [21], working within a new parallel architecture that captures descriptions and sequences within the same epoch during training.

To address the problem of imbalanced classifications in deep neural networks, previous studies have tended to disturb the data distribution in the training set to capture better classifiers [22] [23] [24]. In sampling methods for imbalanced learning, the synthetic minority over-sampling technique (SMOTE) and several variants algorithms were proposed in [25] [26], where data distributions are balanced by either over-sampling the minority classes or under-sampling the majority classes. These approaches change the original data distribution but lead to increased computational costs for data pre-processing and model learning [27] [28] [29] [30]. Instead of these sampling strategies, cost-sensitive learning methods consider the costs associated with misclassifying examples [31] [32] [33]. However, directly modified class-specific loss functions usually lead to high computation costs and less flexibility, which limits their application to real-world problems. In contrast, we have directly modified the learning procedure to incorporate class-dependent costs during training. To this end, we introduce a CPLF for heterogeneous and imbalanced data sets. The key contributions of this paper are:

1) A parallel hybrid neural network to handle heterogeneous business objects that consist of both description and sequence data. The network operates within a unified parallel architecture that aggregates different types of neural networks, e.g., MLP and LSTM, into the same epoch, which greatly improves learning efficiency in real-world complex data analysis. The proposed parallel architecture is easily extended to more diverse networks and data types, which effectively optimizes the performance issues of most currently HNN serial architectures.
2) A joint optimization algorithm for the HNN parameters and the cost-sensitive matrix to solve data imbalance problems in deep neural networks within one training procedure. The effect of modified loss functions is analyzed by deriving relations for propagated gradients.
3) An intelligence architecture of insurance operating system based on CPLF, and demonstrated real-world operating optimization functions, such as risk management and policy renewal classification.
4) An empirical study using six real-world insurance data sets to validate the effectiveness of our proposed approach.

The remainder of this paper is organized as follows. Section II defines the problem formulation and the statement of heterogeneous and imbalanced data. The proposed cost-sensitive parallel learning framework is presented in Section III. Section IV demonstrates the effectiveness of our approach on real-world data sets, and Section V concludes the paper.

II. Problem Statement

This section addresses the problems of analyzing insurance operation data sets from two aspects: the heterogeneous data inputs and the imbalanced classification as the output.

Let $X = \{D, S\}$ represent the descriptions and the transactions of the heterogeneous data inputs, where $D = \{A_1, \cdots, A_n\}$ is a set of n-dimensional attributes that specify the object's characteristics, and $S = \{T_1, \cdots, T_m\}$ is the set of m transactions that record the object's activities over its lifecycle within the time series sequence S. Each transaction T_m consists of $(t_1, \cdots, t_j)$, where t_j is the jth feature that describes the context-aware information of the mth occurring transaction. Our goal is to take a heterogeneous data set $X = \{D, S\}$, with a suitable data structure, feed the data into a novel unified learning framework through weight and bias optimization, and demonstrate that this approach produces superior performance on insurance operation data sets.

TABLE I
INFORMATION TABLE OF HETEROGENEOUS AND IMBALANCED DATA

Real-world dataset	**Heterogeneous inputs**						**Imbalanced output**
	Descriptions			**Sequences**			
XID	A_1	**...**	A_n	T_1	**...**	T_m	**Y**
X_1	2	...	Z01	(a,6)	...	(e,1)	No
X_2	6	...	Z02	(b,7))	...	(f,2)	No
X_3	9	...	Z03	(c,8)	...	(a,5)	Yes
X_4	3	...	Z04	(a,2)	...	(d,8)	No

In our method, an information table is constructed that maps each description and sequence into its corresponding attribute or transaction column. The table, as illustrated by the example in Table I, consists of four samples$X = \{X_1, X_2, X_3, X_4\}$ and includes: n description columns and the corresponding attributes $\{A_1 - A_n\}$; m sequence columns and the corresponding transactions $\{T_1 - T_m\}$; and a labeled classification output, column Y. The output column $Y = \{No, No, Yes, No\}$ shows imbalanced classifications in the input samples $X = \{X_1, X_2, X_3, X_4\}$. Given a sample X_1, the values$\{T_1, \cdots, T_m\}$ are $\{(a, 6), \cdots, (e, 1)\}$, representing a sequence of m transactions with the two features of name and utilization. For example, $(a, 6)$ means that transaction T1 is named a and the amount used is 6.

Based on the constructed information table, we propose a novel parallel learning framework to learn a combination of description and sequence representations in a unified training structure, and a cost-sensitive layer to solve the imbalanced problem. The framework is explained in Section III.

III. Cost-Sensitive Parallel Learning Framework

This section presents a novel cost-sensitive parallel learning framework (CPLF) for analyzing heterogeneous and imbalanced data sets in the insurance industry. An overview of CPLF is given in Fig. 2. It consists of a parallel neural network that analyzes heterogeneous inputs, and a cost-sensitive loss function that solves imbalanced classifications. The various components are described in the following sections.

A. Parallel Neural Network on Insurance Data

The heterogeneous insurance input $X = \{D, S\}$ contains both elements of the data, i.e., the description and the sequence. This input describes both the property-based attributes and the transaction-based sequences. Traditional monotonous methods produce low accuracy on such heterogeneous data sets in most real-world cases.

To solve this problem, we have developed a parallel hybrid concept that incorporates both the descriptive information and the sequence information in one training procedure.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2018.2873526, IEEE Transactions on Industrial Electronics

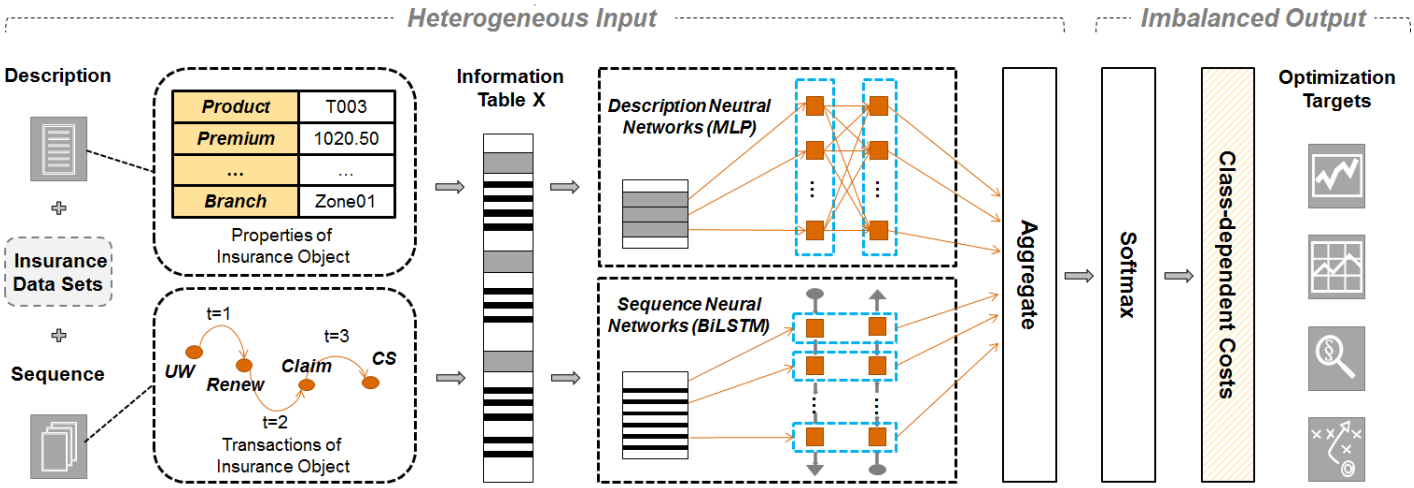

Fig. 2. An overview of the end-to-end cost-sensitive parallel learning framework in insurance operation.

The proposed parallel neural network contains the following components:

1) Heterogeneous Data Embedding: Given heterogeneous data sources with n attributes of description D and m transactions in sequence S, the data embedding task requires the heterogeneous data inputs X to be pre-processed into a uniform information table (see Information Table I as an example), in which: $X \rightarrow \{\{A_1, \cdots, A_n\}, \{T_1, \cdots, T_m\}\}$, where $\{A_1, \cdots, A_n\}$ correspond to n attributes of description D and $\{T_1, \cdots, T_m\}$ correspond to m transactions of sequence S. The categorical variables of the information table are then converted with one-hot encoding, and the numerical variables are normalized to achieve better performance. CPLF then processes the heterogeneous data inputs X through two parallel description and sequence networks with a training procedure that can be applied to whole epochs.

2) Description Neural Network: The description neural network (DsNN) trains the description elements of the data inputs $D = \{A_1, \cdots, A_n\}$ of the constructed information table. The DsNN is essentially an MLP with ≥ 3 layers $\{Input \rightarrow Hidden \rightarrow Output\}$ and several nonlinear activation functions, either $tanh$ or logistic $sigmoid$. Given a 1-hidden-layer MLP, the description parameter $\alpha = \{W_1, W_2, b_1, b_2\}$. The inference function $F(x)$ follows:

$$F(x) = \sigma(b_2 + W_2\sigma(b_1 + W_1 x)), \tag{1}$$

where the MLP inference function is formulated by the bias vectors b_1, b_2 and the weight matrices W_1, W_2. σ represents the sigmoid activation function.

3) Sequence Neural Network: The sequence neural network (SqNN) processes the sequences $S = \{T_1, \cdots, T_m\}$ in the constructed information table, using a bi-directional LSTM (BiLSTM) for learning. The BiLSTM orders the sequential inputs in two ways, one from past to future and one from future to past. Compared to traditional unidirectional LSTMs, BiLSTM networks combine both directional hidden states to preserve the information for any point in time from both the past and the future. In real-world sequential cases, BiLSTM networks usually show good results as they are better at interpreting context. Through the BiLSTM, the SqNN efficiently processes the past, via forward states, and the future, via backward states, for a specific time frame as:

$$\begin{aligned} \overrightarrow{H_t} &= \overrightarrow{LSTMU}(T_t), t \in [1, m] \\ \overleftarrow{H_t} &= \overleftarrow{LSTMU}(T_t), t \in [1, m] \end{aligned} \tag{2}$$

where LSTMU represents a standard unit of long short-term memory. Given a sequence with m transactions, the hidden outputs of a given transaction input T_t, $t \in [1, M]$ are produced from the following subfunctions.

- forget gate: $f_t = \sigma(W_f \cdot [H_{t-1}, T_t] + b_f)$
- input gate layer: $i_t = \sigma(W_i \cdot [H_{t-1}, T_t] + b_i)$
- new contribution: $\tilde{C}_t = \tanh(W_C \cdot [H_{t-1}, T_t] + b_C)$
- update cell state (memory): $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
- output gate layer: $o_t = \sigma(W_o \cdot [H_{t-1}, T_t] + b_o)$
- output to next layer: $H_t = o_t * \tanh(C_t)$

where σ represents the sigmoid activation function, $[H_{t-1}, T_t]$ represents the concatenatation of H_{t-1} and T_t. The SqNN parameters $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$ are a concatenation of the forward hidden state $\overrightarrow{H_t}$ and the backward hidden state $\overleftarrow{H_t}$. $H_t = [\overrightarrow{H_t}; \overleftarrow{H_t}]$, summarizes the information of the entire sequence of transactions centered around T_t.

4) Neural Network Aggregation: To aggregate the DsNN and the SqNN into an HNN (see Fig. 2), the outputs of DsNN and SqNN can be concatenated, multiplied or averaged. In our implementation, the outputs of both the description and the sequence networks are concatenated, followed by a softmax layer for classification:

$$Comb = \text{softmax}(W_c v + b_c), \tag{3}$$

where v is a high level vector of the combined hidden outputs. $v = [H_{des}, H_{seq}]$ represents the concatenation of

the hidden outputs H_{des} from the description network and H_{seq} from the sequence network. A softmax function on the heterogeneous data set $X = \{D, S\}$ is then used for data classification.

B. Imbalanced Cost-Sensitive Classification

Class imbalance problems are addressed during training. A cost-sensitive classification function minimizes the expected risk $\mathcal{R}(p|x)$, where x is an input sample, and p is the output classification of the classifier. The expected risk can be expressed as

$$\mathcal{R}(p|x) = \sum_{q} \delta_{p,q} P(q|x)$$

where cost matrix $\delta_{p,q}$ denotes the cost of misclassifying a sample belonging to a class p as a different class q. $P(q|x)$ is the posterior probability over all possible classes given a sample x. The cost-sensitive error function is expressed as the loss function over the training set:

$$E(\alpha, \beta, \delta) = \ell(y, \hat{y}_{(\alpha,\beta,\delta)}) \tag{4}$$

where $\hat{y}_{(\alpha,\beta,\delta)}$ is parameterized by the HNN, $\alpha = \{W_1, W_2, b_1, b_2\}$ are the weights and biases in the description network, $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$ are the weights and biases of the sequence network, and δ is the matrix of class-sensitive costs. $y \in \{0, 1\}^{1\times N}$ is the desired output, and N denotes the total number of neurons in the output layer, which equals the number of classes. For example, in Table I, $N = 2$ according to column Y. Therefore, the cost-sensitive classification optimization objective is

$$(\alpha^*, \beta^*, \delta^*) = \arg\min_{\alpha,\beta,\delta} E(\alpha, \beta, \delta) \tag{5}$$

where the optimal parameters $(\alpha^*, \beta^*, \delta^*)$ are the objectives of the learning algorithm, giving the minimum possible cost E in Eq. (5). The loss function $\ell(\cdot)$ in Eq. (4) could be any suitable loss function, such as the cross-entropy loss function used here.

Cost-sensitive Cross Entropy Loss: The cross-entropy loss function maximizes the predictions for the desired output and is given by

$$\ell(y, \hat{y}) = -\sum_{n} y \log \hat{y}_{(\alpha,\beta,\delta)}, \tag{6}$$

where y incorporates the class dependent cost δ. The output relates to the previous combination layer output via the softmax function to calculate the probability distribution of different possible outcomes.

C. Optimal Parameter Learning

The goal in optimizing the learning parameters is to jointly learn the three types of parameters using the functions in CPLF: the description hypothesis parameters $\alpha = \{W_1, W_2, b_1, b_2\}$, the sequence hypothesis parameters $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$, and the class-dependent loss function parameters δ. The three types of parameters are solved alternately by keeping two fixed and minimizing the cost with respect to the other. Stochastic gradient descent with a backpropagation error is used to optimize α and β. A gradient descent algorithm is used to optimize the cost-sensitive matrix δ by calculating the direction of the step to update the parameters.

Algorithm 1 Leaning Optimization for Parameters (α, β, δ)

Input: Training set (X_T, Y_T), Validation set (X_V, Y_V), Max epochs (Max_{ep}), Learning rate $(\gamma_\alpha, \gamma_\beta, \gamma_\delta)$
Output: Learned parameters (α^*, β^*, δ^*)

Net ← construct-Parallel-Neural-Net()
{*Random initialization*}
α, β ← initialize-Net(Net), $\delta \leftarrow 1$, val-err ← 1
{*Looping in number of epochs*}
for $e \in [1, Max_{ep}]$ **do**
  $grad_\delta \leftarrow 1$, compute-Grad $(X_T, Y_T, F(\delta))$
  $\delta^* \leftarrow$ update-CostParams $(\delta, \gamma_\delta, grad_\delta)$
  $\delta \leftarrow \delta^*$
  for $b \in [1, batchSize]$ **do**
    $out^b \leftarrow$ forwardPass$(X_T^b, Y_T^b, Net, \alpha, \beta)$
    {***Training Description of neural network***}
    $grad_\alpha^b \leftarrow$ backwardD$(out^b, X_T^b, Y_T^b, Net, \alpha, \beta, \delta)$
    $\alpha^* \leftarrow$ update-DNet-Params$(Net, \alpha, \beta, \gamma_\alpha, grad_\alpha^b)$
    {***Training Sequence of neural network***}
    $grad_\beta^b \leftarrow$ backwardS$(out^b, X_T^b, Y_T^b, Net, \alpha^*, \beta, \delta)$
    $\beta^* \leftarrow$ update-SNet-Params$(Net, \alpha^*, \beta, \gamma_\beta, grad_\beta^b)$
    α, $\beta \leftarrow \alpha^*$, β^*
  end for
  val-err* ← forwardPass$(X_V^b, Y_V^b, Net, \alpha, \beta)$
  if val-err* >val-err **then**
    $\gamma_\delta \leftarrow \gamma_\delta * 0.01$
    val-err ← val-err*
  end if
end for
return (α^*, β^*, δ^*)

The following cost function is used for the gradient computation to update δ, which can be understood as a squared L_2 norm of the difference between the vectors $\hbar$ and δ:

$$f(\delta) = \frac{1}{2} \sum_{c} (\hbar_c - \delta_c)^2, c \in [1, N] \tag{7}$$

where N is the total number of distinct classes in the training set, and $\hbar$ denotes the histogram vector that encodes the distribution of classes in the training set. The minimization objective to find the optimal δ^* is expressed as:

$$\delta^* = \arg\min_{\delta} f(\delta), \tag{8}$$

The gradient descent algorithm that calculates the direction of updated steps and optimizes the cost function is

$$\begin{aligned} \nabla f(\delta) =& \nabla((\hbar - \delta)(\hbar - \delta)^T) \\ =& (\hbar - \delta) J_\delta^T = -(\hbar - \delta)\mathbf{1}^T \end{aligned} \tag{9}$$

where J is the Jacobin matrix. To compute the dependence of $f(\delta)$ on the validation error, we take the update step only if it results in a decrease in the validation error.

In next section, we discuss the impact of the modified loss functions on the gradient computation in the backpropagation algorithm.

D. Gradient Computation on Backpropagation

1) Description NN Backpropagation: In the DsNN, the minimization objective to find the optimal α^* is expressed as

$$\alpha^* = \arg\min_{\alpha} E(\alpha), \tag{10}$$

The loss function is represented as $\ell(y, \hat{y}) = \frac{1}{2}\sum_k (y_k - \hat{y_k})^2$ with the output as the kth neuron in the training set. Using gradient descent, the mathematical expression of gradient at each neuron is given by

$$\frac{\partial \ell(y, \hat{y})}{\partial v_k} = -(y_k - \hat{y}_k)\frac{\partial \hat{y}_k}{\partial v_k}, \tag{11}$$

where v_k is the weighted sum of the input connections. $\hat{y}_k$ in sigmoid activation function is defined as

$$\hat{y}_k = (1 + \exp(-v_k))^{-1}, \tag{12}$$

Therefore, the partial derivation of $\hat{y}_k$ can be given as

$$\frac{\partial \hat{y}_k}{\partial v_k} = \frac{\exp(-v_k)}{(1 + \exp(-v_k))^2} = \hat{y}_k(1 - \hat{y}_k) \tag{13}$$

IV. Experimental Verification

The approach outlined above was evaluated on real-world heterogeneous and imbalanced data sets in the insurance industry. Three data sets were extracted from the Insurance-FD data set and three were extracted from the Insurance-RN data set. Insurance-FD focuses on fraud detection, while Insurance-RN focuses on policy renewable classification. The details of each data set and the experimental settings used follow.

A. Data Sets and Experimental Settings

TABLE II
NETWORK SETTING FOR THE INSURANCE-FD DATA SET

Compared Method	*Learning Rate*	*Hidden Layer Neutron Setting*
DNN: MLP	γ=0.01	$\{input \rightarrow 512 \rightarrow 256 \rightarrow output\}$
RNN: BiLSTM	γ=0.01	$\{input \rightarrow 256 \rightarrow 256 \rightarrow output\}$
PNN: Parallel NN	γ_α=0.01 γ_β=0.01	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 256 \rightarrow 256 \rightarrow h\}$
SNN: SMOTE NN	γ=0.01	$\{input \rightarrow 512 \rightarrow 256 \rightarrow output\}$
CPLF: Cost-sensitive Parallel LF	γ_α=0.01 γ_β=0.01 γ_δ=0.0001	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 256 \rightarrow 256 \rightarrow h\}$

1) Fraud Detection Classification: Insurance-FD is a real-world data set generated from the information systems of a large Chinese life insurance company. It has been randomly extracted from core insurance systems based on real business optimization requirements, and verified by the insurances policies and transactions from other business systems, such as billing or commission. It contains over 138,200 samples and 411 dimensions narrowed from an original 2457 dimensions. The data objects are insurance policies, and the samples describe each policy's properties in terms of 341 attributes (descriptions), such as product list, agent, sales channel, insured, and beneficiaries. The policies are also classified into seven transaction types, which form the sequence information, such as cooling-off period, insurance additive or deductive, loan, claim, surrender, account change, people change and other information change. For the purposes of this study, these features in a policy's lifecycle are considered to belong to positive (fraudulent) and negative (non-fraudulent) classes.

Experimental Setting: To evaluate the cost-sensitive parallel learning framework (CPLF) on data sets of various scales, three data sets of different sizes were extracted from Insurance-FD and converted into information tables, as described in Section III. To represent different degrees of imbalance in the data distribution, we reduced the representations of one of the two classes in each extracted data set to 20%, 10%, and 5%. For example, an imbalance value of 5% means that the proportion of minority (fraudulent) class to majority (non-fraudulent) class is 5%. Therefore an imbalanced level 5% is more imbalanced than an imbalanced level 10% and 20%. The neural networks settings for Insurance-FD for each compared method are shown in Table II.

TABLE III
NETWORK SETTING FOR THE INSURANCE-RN DATA SET

Compared Method	*Learning Rate*	*Hidden Layer Neutron Setting*
DNN: MLP	γ=0.01	$\{input \rightarrow 512 \rightarrow 256 \rightarrow output\}$
RNN: BiLSTM	γ=0.01	$\{input \rightarrow 128 \rightarrow 128 \rightarrow output\}$
PNN: Parallel NN	γ_α=0.01 γ_β=0.01	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 128 \rightarrow 128 \rightarrow h\}$
SNN: SMOTE NN	γ=0.01	$\{input \rightarrow 512 \rightarrow 256 \rightarrow output\}$
CPLF: Cost-sensitive Parallel LF	γ_α=0.01 γ_β=0.01 γ_δ=0.0001	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 128 \rightarrow 128 \rightarrow h\}$

2) Policy Renewable Classification: Life insurance is a business with long-term products and services whose profitability cannot be measured without a long-term lens. Policy renewable classification is quite useful and important for insurers to monitor and manage their ongoing performance. Insurance-RN is a real-world data set from a large life insurance company in China. It contains the long-term insurance policies with both original underwriting information and customer behavioral data during the policy life cycle, which randomly extracted from life insurance systems based on the policy renewal requirements. The descriptions of underwriting information and transactions of customer life-cycle services

in Insurance-RN have 304 dimensions, narrowed from an original 3751 dimensions. For the purposes of this study, these features belong to positive classification that means unsuccessful renewal and negative classification to represent successful renewal.

Experimental Setting: To represent the different levels of imbalance in the data distribution, we reduced the representative samples of one of the two classes in each data set to 20%, 10%, and 5%. Again, an imbalance value of 5% means that the proportion of minority (unsuccessful renewal) class to majority (successful renewal) class is 5%. Therefore an imbalanced level 5% is more imbalanced than an imbalanced level 10% and 20%. The three data sets were then converted into information tables as outlined in Section III. The neural networks settings for Insurance-RN for each compared method are shown in Table III.

3) Comparison baselines: The following deep neural network (DNN), RNN, PNN, SNN algorithms were selected as appropriate comparisons to evaluate CPLF's performance.

- DNN: a MLP network trained on the description data
- RNN: a bi-directional LSTM (BiLSTM) network trained on the sequence data
- PNN: our proposed parallel neural network (PNN) without the cost-sensitive matrix, trained on both the description and sequence data
- SNN: a traditional neural network trained on the data with synthetic minority oversampling technique(SMOTE)

All neural networks were trained on one data set during the training procedure. SNN comparison trained on oversampling data distribution, that 80% of the samples in each data set were used as the training set after oversampling, with the remaining 20% used as the testing set. While other comparisons and CPLF using original data distribution, that 80% of the samples in each data set were used as the training set, with the remaining 20% used as the testing set. In the training set, 10% samples are excluded and used as validation set. As a result, evaluated networks with different weights and biases were generated and used to make predictions in the subsequent testing procedure.

4) Evaluation Metrics: The following classification metrics in both technical and business perspectives were used to evaluate prediction performance.

Technical Perspective:

- Accuracy [34]: $Accuracy = \frac{TN+TP}{TP+FP+FN+TN}$ where TP, TN, FP and FN represent true positive, true negative, false positive and false negative, respectively.
- Precision [34]: $Precision = \frac{TP}{TP+FP}$
- Recall [34]: $Recall = \frac{TP}{TP+FN}$
- F-measure [35]: $F_1 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$
- Receiver operating characteristic curve(ROC) [35].
- Area under curve(AUC) [35]: Area under the ROC curve, e.g. an area of 1 represents a perfect test and an area of 0.5 represents a worthless test.

Business Perspective: We analyze business performance by insurers who use the predictive outcomes to optimize their insurance operation.

- Expense Ratio: $ExpenseRatio = \frac{UnderwritingExpenses}{NetPremiumsWritten}$. The lower the expense ratio the better because it means more profits to the insurance company.
- Return on Revenues: This figure determines the profitability of an insurance company. $ReturnonRevenues = \frac{NetOperatingIncome}{TotalRevenues}$. It is the profits after all expenses and taxes are paid by the insurance company.

Optimizing strategy: If a positive classification is predicted, the insurer will contact the customer immediately to avoid potential performance risk, such as fraudulent leak or unsuccessful renewal. Otherwise, if there is a negative classification (non-fraudulent or a successful renewal) to be predicted, the customer only requires to be contacted as normal. With the supports of accurate testing results and optimizing strategy. The insurer will find a way to keep profitable balance between operating costs and premium incomes, by increasing premiums or decreasing risks with low cost operation optimization.

B. Experimental Results

In general, accurately classifying the minority class rather than the majority class is more important when the data is imbalanced. Without loss of generality, we mainly focused on classification performance in the minority class, which was treated as the positive class in these experiments.

The results of the experiments we conducted on the three Insurance-FD data sets and the three Insurance-RN data sets are shown in Tables IV and V, respectively.

TABLE IV
EVALUATION ON INSURANCE-FD DATA SET

Datasets	**Accuracy**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.888	0.929	0.938	0.884	0.937
Insur + Imb. level 10%	0.949	0.948	0.948	0.935	0.958
Insur + Imb. level 5 %	0.961	0.973	0.971	0.958	0.975
Datasets	**Precision**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.694	0.778	0.868	0.560	0.811
Insur + Imb. level 10%	0.671	0.745	0.669	0.572	0.787
Insur + Imb. level 5 %	0.703	0.731	0.750	0.465	0.746
Datasets	**Recall**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.638	0.692	0.674	0.755	0.718
Insur + Imb. level 10%	0.339	0.547	0.450	0.555	0.622
Insur + Imb. level 5 %	0.222	0.485	0.396	0.509	0.515
Datasets	**F-measure**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.665	0.732	0.759	0.643	0.762
Insur + Imb. level 10%	0.450	0.631	0.547	0.563	0.695
Insur + Imb. level 5 %	0.337	0.583	0.518	0.486	0.609
Datasets	**AUC**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.795	0.830	0.828	0.830	0.845
Insur + Imb. level 10%	0.652	0.758	0.770	0.760	0.804
Insur + Imb. level 5 %	0.608	0.739	0.695	0.743	0.754

*Imb. denotes imbalance

With each data set, we observed that the more imbalanced the data, the worse the classification performance, as illustrated by the general downward trend in terms of both F-measure and AUC as the degree of imbalance increased. More importantly,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2018.2873526, IEEE Transactions on Industrial Electronics

however, the CPLF and the PNN network based on parallel neural network without the cost-sensitive matrix demonstrated better classification accuracy on most of the data sets than the baseline DNN and RNN networks at the same imbalance level. Additionally, CPLF showed obvious improvements in terms of F-measure and AUC in the data sets with an extreme imbalance of 5%. This is a promising result for effectively classifying imbalanced data sets. Different from other methods using original imbalanced data sets, the SNN using oversampling balanced data sets. The SNN performed better than the DNN, the RNN and the PNN methods, however it has changed original data distribution and increased computational costs for data pre-processing and model learning. In most of the data sets, the CPLF outperformed all other four comparative baseline: the DNN, the RNN, the PNN, and the SNN.

TABLE V
EVALUATION ON INSURANCE-RN DATA SET

Datasets	**Accuracy**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.872	0.852	0.835	0.745	0.876
Insur + Imb. level 10%	0.904	0.899	0.907	0.859	0.908
Insur + Imb. level 5 %	0.947	0.942	0.953	0.946	0.956
Datasets	**Precision**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.666	0.636	0.489	0.304	0.675
Insur + Imb. level 10%	0.622	0.318	0.620	0.309	0.616
Insur + Imb. level 5 %	0.722	0.333	0.516	0.466	0.477
Datasets	**Recall**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.386	0.104	0.478	0.511	0.480
Insur + Imb. level 10%	0.167	0.051	0.210	0.424	0.438
Insur + Imb. level 5 %	0.168	0.015	0.281	0.293	0.333
Datasets	**F-measure**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.489	0.180	0.483	0.381	0.505
Insur + Imb. level 10%	0.272	0.087	0.314	0.358	0.408
Insur + Imb. level 5 %	0.270	0.028	0.364	0.360	0.393
Datasets	**AUC**				
Experimental settings	***DNN***	***RNN***	***PNN***	***SNN***	***CPLF***
Insur + Imb. level 20%	0.675	0.546	0.691	0.649	0.696
Insur + Imb. level 10%	0.580	0.519	0.598	0.664	0.672
Insur + Imb. level 5 %	0.572	0.506	0.634	0.638	0.657

*Imb. denotes imbalance

Fig. 3 plots the ROC of all comparative approaches in the Insurance-FD data sets at an imbalance level of 10%, where the horizontal axis stands for the true positive rate, and the vertical axis represents the false positive rate.

We also tested CPLF in terms of the loss value trend when training and testing the Insurance-FD data sets at an imbalance level of 5%, as shown in Fig. 4. Even with highly imbalanced heterogeneous data sets, CPLF significantly decreased the loss value trend during both the training and testing procedures as the number of epochs increased. This result empirically verifies the theoretical analysis in the previous sections, demonstrating that CPLF delivers effective classification performance on real-world heterogeneous and imbalanced data sets.

In business perspective, renewal premium is one of the important indicators that reflects operating performance and premium incomes. In practical, because of the limitation of operating expense, insurance company usually randomly chooses 20% policy customers to discuss their policy renewal. Fig.5 plots a renewal premium comparison in the Insurance-RD data sets at an imbalance level of 10%. Given by fixed operating costs of 20% customer discussion, renewal premium was calculated by standard statistical test results. The blue represents the renewal premium by neural network methods: DNN, RNN, PNN, SNN and proposed CPLF. The orange represents the renewal premium by traditional random method. It demonstrated that CPLF outperforms all other comparative methods in Fig.5.

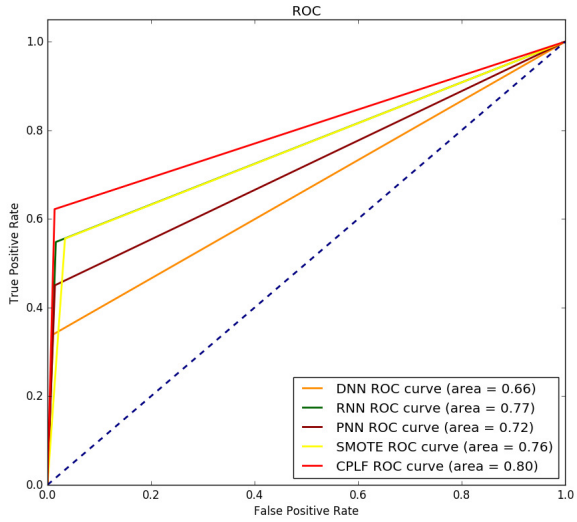

Fig. 3. ROC in the Insurance-FD data sets at an imbalance level of 10%

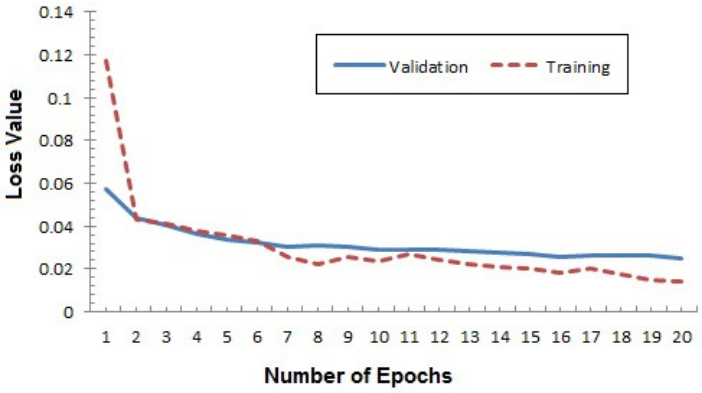

Fig. 4. CPLF effectively decreases the loss value with the increase in the number of epochs on the Insurance-FD data set with an imbalance level of 5%.

C. Insurance Intelligence Operation System

Based on CPLF, we also study the intelligence application of insurance operation initiative. We have identified six key insurance operation opportunities by combining market and organizational priorities with artificial intelligence techniques, opening the door to both external and internal perspectives. External opportunities primarily relate to social and technological trends and pertain to the shift in customer needs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2018.2873526, IEEE Transactions on Industrial Electronics

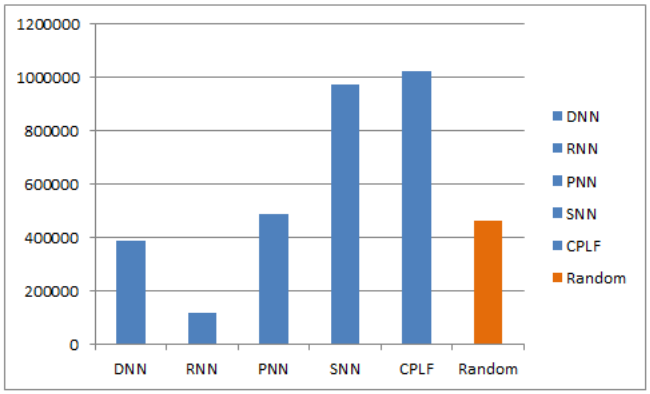

Fig. 5. Renewal premium comparison in the Insurance-RD data sets at an imbalance level of 10%.

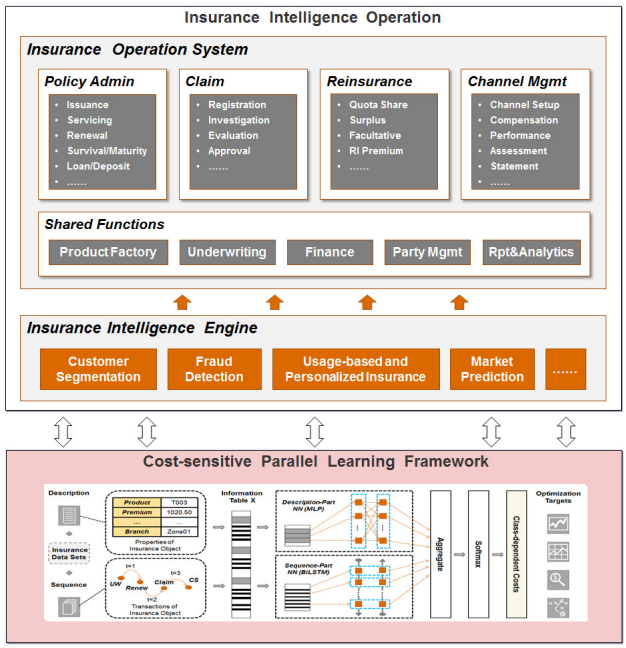

Fig. 6. The architecture of the insurance intelligence operation system.

and expectations by taking action in these areas to stay relevant in the market and maintain market position. External opportunities include: a) Are mainly driven by customer expectations and needs and enabled by technology. b) Offer front runners the opportunity to gain market relevance and position themselves. c) Also offer fast followers opportunity because value propositions can be quickly replicated. Internal opportunities relate to using technology to enhance operations and business function execution. For example, utilizing artificial intelligence technology to enhance internal operations, which has improved efficiencies and automated existing customer-facing, underwriting and claims processes. Internal opportunities are: d) Mainly driven by technological advancements. e) A source of competitive advantage but demand deeper change. f) An opportunity to set the foundation for how the company understands and manages risk.

We illustrate the following architecture of an insurance intelligence operation system in Fig.6, where the bottom of the image shows the cost-sensitive parallel learning framework (CPLF) under an insurance intelligence engine which provides meaningful data mining insights into the insurance operation system. The major components of the insurance intelligence operation in Fig.6 are as follows:

1) A cost-sensitive parallel learning framework (CPLF) to handle the insurance data analysis process that consists of hybrid neural networks and cost-sensitive layers, acting as an intelligence factory for the whole insurance intelligence operation.
2) An insurance intelligence engine that utilizes the data insights from CPLF, and combines them with business analysis targets, such as customer segmentation, fraud detection, usage-based and personalized insurance, and market prediction. An example is usage-based and personalized insurance that leverages technology and data to develop new risk models based on behavioral factors. Insurance fraud detection can help insurers to decrease losses caused by fraud and reduce false positives in insurance claims improving investigator efficiency. We have demonstrated the effectiveness of fraud detection in an insurance intelligence operation system in the previous experimental subsections.
3) An insurance operation system to monitor the core processes in insurance business operation that consists of policy administration, claim, reinsurance, channel management, and shared functions, such as product factory, underwriting, finance, party management, reporting and analytics. The intelligence operation utilizes the output models or decisions from the insurance intelligence engine and integrates them with the core business processing to enhance the qualities and effectiveness of the insurance operation.
   - Policy administration: is used to execute a number of core policy processes. Taking life insurance as an example, including policy issuance, servicing, renewal, survival or maturity, loan or deposit.
   - Claim: is a formal request to an insurance company for coverage or compensation for a covered loss or policy event.
   - Reinsurance: is insurance that is purchased by an insurance company. Reinsurance allows insurance companies to remain solvent after major claims events, such as major disasters like hurricanes and wildfires.
   - Channel management: is a core process in which a company develops various marketing techniques as well as sales strategies to reach the widest possible customer base.
   - Product factory, underwriting, finance, party management, reporting and analytics: are shared functions that also place high demand on enhancing performance and effectiveness by data-driven insights.

The CPLF-based insurance intelligence operation system can assist insurers to become more automated and efficient by focusing on higher risks and improving profitability through appropriate management. By learning the historical operation-related data, CPLF can assist in the drafting of specialized

contracts that are better tailored to the specifics of a given transaction. For example, analyzing multiple data sets from Internet of Things (IoT) sensors will provide personalized data to pricing platforms, allowing safe drivers to pay less for vehicle insurance and people with healthy lifestyles to pay less for health insurance. Fraud detection is a major concern for insurance companies, and CPLF's parallel hybrid architecture can assist in detecting suspected fraud. The experimental results on fraud detection in previous sub-sections prove the effectiveness of CPLF.

V. Conclusion

A novel cost-sensitive parallel learning framework (CPLF) has been proposed in this paper to handle insurance operation problems with end-to-end processes. CPLF feeds a heterogeneous information table into a parallel architecture. Alternating optimization algorithms then efficiently learn a resulting imbalanced cost-sensitive matrix along with CPLF's parameters at epoch level. The results of experiments with real-world insurance data sets demonstrate the effectiveness of our design.

References

[1] E. Z. Baranoff and E. Z. Baranoff, *Risk management and insurance*. Wiley Danvers, 2004.

[2] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling extremal events: for insurance and finance*, vol. 33. Springer Science & Business Media, 2013.

[3] B. F. Bowne, N. R. Baker, D. L. Marzinzik, M. E. Riley, N. U. Christopulos, B. M. Fields, J. L. Wilson, B. T. Wilkerson, D. W. Thurber *et al.*, "Systems and methods using a mobile device to collect data for insurance premiums," Jan. 9 2018, uS Patent 9,865,018.

[4] G. Hayward, S. T. Christensen, C. E. Gay, S. C. Cielocha, and T. Binion, "Systems and methods for generating vehicle insurance policy data based on empirical vehicle related data," Jan. 9 2018, uS Patent 9,865,020.

[5] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, "Gene functional classification from heterogeneous data," in *Proceedings of the fifth annual international conference on Computational biology*, pp. 249–255. ACM, 2001.

[6] R. Hu, C. P. Yu, S.-F. Fung, S. Pan, H. Wang, and G. Long, "Universal network representation for heterogeneous information networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 388–395. IEEE, 2017.

[7] F. Emmert-Streib, R. de Matos Simoes, G. Glazko, S. McDade, B. Haibe-Kains, A. Holzinger, M. Dehmer, and F. C. Campbell, "Functional and genetic analysis of the colon cancer network," *BMC bioinformatics*, vol. 15, no. 6, p. S6, 2014.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[9] M. Rudolph, F. Ruiz, S. Mandt, and D. Blei, "Exponential family embeddings," in *Advances in Neural Information Processing Systems*, pp. 478–486, 2016.

[10] M. J. Greenacre, "Theory and applications of correspondence analysis," 1984.

[11] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Embedding heterogeneous data using statistical models," in *Proceedings of The National Conference on Artificial Intelligence*, vol. 21, no. 2, p. 1605, 2006.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[13] F. Provost, "Machine learning from imbalanced data sets 101," in *Proceedings of the AAAI 2000 workshop on imbalanced data sets*, pp. 1–3, 2000.

[14] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2285–2294. IEEE, 2016.

[15] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

[16] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder," *arXiv preprint arXiv:1802.04407*, 2018.

[17] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 889–898. ACM, 2017.

[18] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1895–1901, 2016.

[19] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 225–230, 2016.

[20] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[22] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[23] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[24] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *arXiv preprint arXiv:1305.1707*, 2013.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[26] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 107–119. Springer, 2003.

[27] R. Mollineda, R. Alejo, and J. Sotoca, "The class imbalance problem in pattern classification and learning," in *II Congreso Español de Informática (CEDI 2007). ISBN*, pp. 978–84, 2007.

[28] S. Pan, J. Wu, and X. Zhu, "Cogboost: Boosting for fast cost-sensitive graph classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 2933–2946, 2015.

[29] S. Pan, J. Wu, X. Zhu, and C. Zhang, "Graph ensemble boosting for imbalanced noisy graph stream classification," *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 954–968, 2015.

[30] S. Pan and X. Zhu, "Graph classification with imbalanced class distributions and noise," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 1586–1592, 2013.

[31] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1, pp. 973–978. Lawrence Erlbaum Associates Ltd, 2001.

[32] M. Liu, C. Xu, Y. Luo, C. Xu, Y. Wen, and D. Tao, "Cost-sensitive feature selection by optimizing f-measures," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1323–1335, 2018.

[33] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Cost-sensitive learning with noisy labels," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5666–5698, 2017.

[34] D. L. O. Dr and D. D. Dr, *Advanced Data Mining Techniques*. Springer Berlin Heidelberg, 2008.

[35] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2018.2873526, IEEE Transactions on Industrial Electronics

Xinxin Jiang is a Ph.D. Candidate in the Centre for Artificial Intelligence (CAI), Faculty of Engineering and Information Technology, University of Technology Sydney (UTS). She received her Master's degree in computer science from University of Jiangsu, China in 2001. She worked as an application architect in International Business Machines Corporation, Shanghai, China from 2007, and as a consulting manager in PricewaterhouseCoopers, Beijing, China from 2015.

Her research focuses on machine learning and data mining.

Shirui Pan (M'16) Shirui Pan received his Ph.D. degree in computer science from University of Technology Sydney (UTS), Australia, in 2015. He is a Lecturer in the Centre for Artificial Intelligence (CAI),UTS. Since 2010, he has published over 50 research papers in top-tier journals and conferences, including IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Cybernetics (TCYB), Pattern Recognition, International Joint Conference on Artificial Intelligence (IJCAI), International Conference on Data Engineering (ICDE) and IEEE International Conference on Data Mining (ICDM).

His current research interests include data mining, machine learning, and graph data analytics.

Guodong Long received his Ph.D. degree in computer science from the University of Technology Sydney, Australia, in 2014. He is currently a Senior Lecturer and core member in the Centre for Artificial Intelligence (CAI), Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

His research focuses on machine learning, data mining and cloud computing.

Fei Xiong received his B.E. degree and Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2007 and 2013 respectively. He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. From 2011 to 2012, he was a visiting scholar at Carnegie Mellon University. He has published over 60 papers in refereed journals and conference proceedings. He was the recipient of a grant from National Natural Science Foundations of China and several other research grants.

His current research interests include the areas of web mining, complex networks and complex systems.

Jing Jiang is currently a Lecturer at the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology at the University of Technology Sydney (UTS), Australia. She has a PhD degree in Information Technology.

Her research interest lies in data mining and machine learning applications with the focuses on deep reinforcement learning and sequential decision-making.

Chengqi Zhang received the PhD degree from the University of Queensland, Brisbane, Australia, in 1991 and the DSc degree (higher doctorate) from Deakin University, Geelong, Australia, in 2002. Since December 2001, he has been a professor of information technology with the University of Technology, Sydney, Australia. Since November 2005, he has been the chairman of the Australian Computer Society National Committee for Artificial Intelligence. He has published more than 200 research papers, including several in first-class international journals, such as Artificial Intelligence, and IEEE and ACM Transactions.He has published six monographs and edited 16 books, and has attracted 11 Australian Research Council grants.

His research interests mainly focus on data mining and its applications. He has served as an associate editor for three international journals, including the IEEE Transactions on Knowledge and Data Engineering (2005-2008); and as general chair, PC chair, or organizing chair for five international conferences including ICDM 2010 and WI/IAT 2008. He was/is general co-chair of KDD 2015 in Sydney, the local arrangements chair of IJCAI-2017 in Melbourne, and a fellow of the Australian Computer Society.