Journal of Data Science and Modeling, Vol. 2, No. 2, 189-215, June 2024

Research Manuscript

Credit-Card Fraud Detection: Cost-Sensitive Meta-Learning Bayesian Network Classifiers

Mohaddeseh Safakish, Vahid Rezaeitabar*

Department of Statistics, Faculty of Statistics, Mathematics and Computer Sciences,

Allameh Tabataba'i University, Tehran, Iran.

Recieved: 22/02/2025 Accepted: 23/05/2025

Abstract: In the modern era, detecting credit card fraud has become a crucial concern from both financial and security standpoints. Given the rarity of fraudulent activities, the issue is reframed as a binary classification challenge, tackling the complexities of imbalanced datasets. To address this, the authors advocate using Bayesian networks due to their theoretical robustness and capacity to model intricate scenarios while maintaining interpretability in the context of class-skewed distributions. A pivotal component of this meta-learning framework is the cost matrix, leading the authors to explore various techniques for its calculation. By employing our meta-learning framework with data from Iran's banking system, the authors demonstrate a method for determining the cost matrix. Subsequently, they develop the corresponding Cost Augmented Bayesian Network Classifiers, called CABNCs. The outcomes highlight the potential of CATAN to diminish financial loss and the effectiveness of CAGHC-K2 in predicting labels for forthcoming transactions in the context of class imbalance.

Keywords: Bayesian network, Classification, Cost-sensitive, Economic efficiency, Fraud detection, Meta-learning

Mathematics Subject Classification (2010): 62H30, 62P05.

*Corresponding Author: vhrezaei@gmail.com

1. Introduction

In recent years, particularly following the Covid-19 pandemic, the usage of credit cards as a fast mechanism for money transferring has greatly expanded. Along with this, increasing criminal usage of the card has become a new concern for financial and economic businesses.

The Iranian National Tax Administration suspects income-related transactions with no transparent source of acquisition and unpaid tax status. Banking and credit systems consider transactions originating from illegal activities and suspected of money laundering to be fraudulent and criminal. Visual and manual inspections to identify criminal and illegal activities besides their corresponding transactions are inaccurate, costly, and time-consuming. With the advent of artificial intelligence, it has become possible to utilize machine learning-based algorithms to analyze credit card transaction data to detect illegal financial behaviors.

Depending on the amount of money being transferred, an "account turnover tax" is deducted from the source account during the transaction. If an illicit transaction is carried out without detection, the banking institution may incur financial losses. Additionally, the number of suspicious transactions is quite low. As a result, these financial transactions provide a highly skewed and imbalanced data set. Accepting fraudulent transactions as legitimate is more costly than inaccurately detecting legal transactions as fraudulent, as the former results in a higher economic loss for the financial institution and requires cost reimbursement. Finding a solution for the rapid identification of transactions resulting from criminal activities, while also taking into account the imbalanced nature of the data along with the sensitivities of the evaluation criteria to errors caused by false transaction detection, is regarded as an important issue.

Many authors have investigated the issue of detecting unauthorized or illegal transactions. The most prevalent type of unlawful financial and banking behavior, identified using machine learning approaches, is the unauthorized and criminal usage of bank cards. This topic has been addressed using a range of strategies, including supervised, unsupervised, deep learning, and ensemble methods.

Among the supervised techniques, Naive Bayesian approaches (Singh and Ranjan and Tiwari , 2021), decision trees and random forests (Seera and *et al.* , 2024), and Bayesian belief networks (Kumar and Mubarak and Dhanush , 2020) have gotten the greatest attention. Two of the most recent unsupervised approaches used to identify fraud are the hidden Markov model (Lucas and *et al.* , 2020) and the local outlier factor (LOF) (Prusti and Das and Rath , 2021).

Scholars have developed a variety of methods in recent years to provide effective solutions for credit card fraud detection, including ensemble learning methods such as XGBoost (XG), CatBoost (CB), and gradient boosting algorithms (Gamini and *et al.* , 2021), as well as deep learning techniques such as artificial neural networks (ANN) (Asha and SureshKumar , 2021).

According to the literature, strategies for addressing the issue of fraud detection could be divided into two broad categories. One approach, given that such activities are uncommon, addresses the issue using anomaly detection techniques (Halvaiee and Akbari , 2014). In contrast, the other group views fraud detection as the problem of categorizing transactions as legal or illegal, and they seek an adequate algorithm for binary classification (Hens and Tiwari , 2012). Following the second group, this paper investigates credit card fraud detection using a binary classification task.

Caldeira and *et al.* (2012) employed artificial neural networks and random forests to detect fraudulent online transactions. De Sá and *et al.* (2018) explored imbalance by assigning different misclassification costs to distinct classes. Fu and *et al.* (2016) used a convolutional neural network (CNN) to identify hidden patterns in each transaction. They used a cost-based sampling method to address the issue of imbalanced data, creating synthetic fraudulent samples from real ones. Sahin and *et al.* (2013), as pioneers in the use of cost-sensitive approaches, developed a cost-sensitive decision tree algorithm that preserved the imbalanced distribution of classes through stratified sampling during the model learning phase. To overcome the imbalance, De Sá and *et al.* (2018) employed an undersampling technique in combination with two classification algorithms. They developed FRAUD-BNC, a customized Bayesian Network Classifier (BNC), using a hyper-heuristic evolutionary algorithm (HHEA).

Several works have addressed the challenge of fraud detection ((Van Vlasselaer and *et al.* , 2015); (Dal Pozzolo and *et al.* , 2014)). Unfortunately, most real-world financial fraud datasets suffer from a severe class imbalance issue, where the fraud data's proportion is significantly lower than that of non-fraud. In binary classification, class imbalance often leads to biased predictions favoring the majority class Johnson and Khoshgoftaar (2019). Consequently, the classifier's performance on the minority class is compromised, especially when encountering dissimilar frauds. Overcoming this problem poses a significant challenge, as classifiers are expected to achieve high precision and recall in the fraudulent class.

Fernández and *et al.* (2018) present a great review of the methods to overcome imbalanced data. Most literature has focused on improving statistical metrics to evaluate predictive performance. Lots of imbalanced data dealing approaches have been proposed at either the data or algorithm levels. Data-level is usually based on re-sampling methods, which mainly include increasing the number of minority examples by generating synthetic examples (over-sampling) (Cateni and *et al.* , 2014), decreasing the number of majority examples by removing some of them (under-sampling) (Brown and Mues , 2012), and synthetic minority oversampling technique (SMOTE) (Zhang and *et al.* , 2017). SMOTE interpolates between the existing minority data to synthesize minority samples (Chawla and *et al.* , 2002). By 2018, more than 85 SMOTE variations were proposed (Cheah and *et al.* , 2023). On the other hand, the algorithm-level solution primarily focuses on exploring some suitable and robust classification algorithms, including ensemble learning approaches (Yu and Ni , 2014) and reweight-learning methods (Kotsiantis , 2011). Nevertheless, duplication and uncertainty introduced by re-sampling techniques, as well as the high computational resources and time consumed by algorithm-level methods, are some issues arising from such approaches.

However, some other works have adopted a cost-sensitive approach incorporating class-dependent misclassification costs of fraud (Sahin and *et al.* , 2013). The cost-sensitive learning framework is a methodology between data-level and algorithm-level approaches. In these techniques, both data-level transformations (adding costs to samples) and algorithm-level modifications (by amending the learning process to accept costs) are considered simultaneously. The resulting classifier in these techniques is biased towards the minority class by assuming higher costs of misclassification for the whole minority class. Following the minimization of the expected total cost for both classes, it provides a framework of cost-sensitive approaches to address the imbalanced class problem. During the past years, many cost-sensitive learning methods have been developed. Among these studies, Domingos (1999) proposed the MetaCost algorithm, which is a principled method for making an arbitrary classifier cost-sensitive by wrapping a cost-minimizing procedure around it. Fan and *et al.* (1999) studied the problem of reducing misclassification cost using boosting methods and proposed the AdaCost algorithm. In AdaCost, the weight updating rule increases the weights of costly wrong classifications more aggressively, but decreases the weights of costly correct classifications more conservatively. Under this updating rule, the weights for expensive samples are higher and the weights for inexpensive samples are comparatively lower. Elkan (2001) and Zadrozny and Elkan (2001) reviewed the general structure of cost-sensitive learning and described the role of misclassification costs on different cost-sensitive learning algorithms in detail. Sheng and Ling (2006) presented the Thresholding method to make any cost-insensitive classifier cost-sensitive. Thresholding selects a proper threshold from training data according to the misclassification cost. Zhao (2008) compared the effects of weighting and the threshold adjusting approach on several classification methods.

Morais and *et al.* (2016) used meta-learning technology to improve the existing methods of resolving the imbalanced data problem. Such approaches modify the data or model's output based on the cost information rather than adapting the algorithm. This provides an innovative path for studying the problems of class-imbalanced data. Sampling or instance-weighting (Zadrozny and *et al.* , 2003) is a kind of cost-sensitive meta-learning technique. In a class-skewed distribution scenario, the inequality between the number of instances in each of the classes is severe, and therefore the class distribution is highly skewed. Chawla and *et al.* (2002) used sampling algorithms to balance the class distribution of the training data and make the minority-class instances well-represented, and as a consequence, classifiers are allowed to place more importance on the minority class. Jiang and *et al.* (2014) incorporated an instance weighting method into various Bayesian network classifiers. They modified the probability estimation of Bayesian network classifiers by the instance weighting method, which makes Bayesian network classifiers cost-sensitive. Most cost-sensitive algorithms utilize class-dependent costs ((Höppner and *et al.* , 2020); (Krawczyk and *et al.* , 2014); (Domingos , 1999); (Chawla and *et al.* , 2008)).

In this paper, from a class-dependent cost-sensitive point of view, we propose the CABNC, a meta-learning technique for detecting fraud in imbalanced credit card data, that emphasizes cost matrix exploration through focusing on output modification rather than instance weighting. BNC models include Naive Bayes (NB) with the assumption of no intra-feature independence given the class, and two additional models without this assumption: TAN and GHC-K2. Firstly, we calculate the optimal cost matrix specific to each BNC based on economic efficiency and other statistically based evaluation criteria. In the second step, to convert BNCs to their corresponding CABNCs, a cost-sensitive class label assignment rule is developed. Depending on the BNC and evaluation metric of interest, one optimal cost matrix is calculated. Subsequently, by utilizing the matrix obtained in the previous step, CABNC is generated. The performance of the models is assessed in terms of Kappa, F1, Recall, Specificity, Accuracy, and Economic Efficiency. All the computations are carried out with the pyAgrum package in the Python programming language.

The paper is organized as follows: Section 2 introduces the Cost Augmented learning approach and associated decision rule. Section 3 discusses the application of proposed models to real data, and Section 4 concludes the paper.

2. Methodology

Imbalanced datasets are prevalent in numerous real-world scenarios where the distribution of classes in the data is significantly uneven. For the sake of simplicity, we will consider the minority or rare class as the positive class and the majority class as the negative class. Consider the dataset $S = \{(\mathbf{T}_1, Y_1), (\mathbf{T}_2, Y_2), \ldots, (\mathbf{T}_N, Y_N)\}$, where $\mathbf{T}_m = (T_1, \dots, T_l) \in T \subset R^l, m = 1, \dots, N$ represents the l-dimensional feature vector of N samples and the class variable is denoted by $Y_m \in \{0, +1\}$. The minority and majority classes are referred to as S^+ and S^-, defined as follows:

$$S^+ = \{(\mathbf{T}, Y) \in S : Y = +1\}, N_P = |S^+|$$
$$S^- = \{(\mathbf{T}, Y) \in S : Y = 0\}, N_N = |S^-|$$

where the sample sizes of the positive and negative classes are indicated as N_N and N_P, respectively.

In binary classification, the imbalance ratio (IR) signifies the ratio of samples associated with the positive class (fraudulent transactions) to those in the negative class (legitimate transactions). This is a commonly used criterion for measuring the degree of imbalance in a dataset. Typically, the size of the minority class is minimal ($N_N \gg N_P$), sometimes as low as 1% of the entire dataset. If we utilize most traditional classifiers that do not account for costs, they are likely to predict all instances as belonging to the negative class (the majority). This issue is often seen as a challenge when working with a highly skewed class distribution with a large amount of IR.

However, as noted by Provost (2000), two key assumptions are frequently made in conventional cost-insensitive classifiers. The first assumption is that the objective of the classifiers is to enhance accuracy (or reduce the error rate); the second assumption is that the class distributions in both the training and test datasets are identical. Given these two assumptions, in the case of a significantly imbalanced dataset, predicting every instance as negative is often the appropriate approach.

Consequently, the problem of class imbalance is pertinent only when one or both of the two previously mentioned assumptions are violated; particularly, if the costs associated with various error types (false positives and false negatives in binary classification) are unequal, or if the class distribution in the test data differs from that in the training data. The first scenario can be addressed effectively through techniques found in cost-sensitive meta-learning.

When the cost of misclassification is unequal, it is usually more expensive to misclassify a minority (fraudulent) transaction into a majority (legitimate) class than a majority transaction into the minority class (otherwise, it is more plausible to predict everything as legitimate). We do not mention the cases where class distributions of training and test datasets are different.

Definition 2.1 (Cost Matrix)**.** *In the context of cost-sensitive methodologies for addressing a classification issue, a cost matrix ($\boldsymbol{C}$) is established for each confusion matrix. The elements along the diagonal of this matrix indicate the costs associated with correctly classifying transactions* $\left[\boldsymbol{C}(\mathcal{C}_n, \mathcal{C}_n), \boldsymbol{C}(\mathcal{C}_p, \mathcal{C}_p)\right]$*, while the* *off-diagonal elements reflect the costs incurred from misclassifying transactions* $\big[\boldsymbol{C}(\mathcal{C}_n, \mathcal{C}_p), \boldsymbol{C}(\mathcal{C}_p, \mathcal{C}_n)\big]$.

In this paper, we consider the case where the diagonal elements remain constant. Thus, throughout the following sections, when we refer to the cost matrix, we are specifically discussing the elements that lie outside of the main diagonal, and our objective is to determine their optimal values. Consequently, we represent the matrix as $\big(\mathbf{C}_{np}, \mathbf{C}_{pn}\big)$.

There are two primary methods for estimating this matrix: direct and meta-learning. The key concept behind developing a direct cost-sensitive learning algorithm is to directly incorporate and utilize misclassification costs within the learning algorithms. Meanwhile, cost-sensitive meta-learning transforms existing cost-insensitive classifiers into cost-sensitive versions without changing their structure. Therefore, it can be considered a middleware component that either preprocesses the training data or post-processes the outputs produced by cost-insensitive learning algorithms.

2.1 Cost Matrix Calculation and Evaluation Measurements

This section outlines the different evaluation metrics for assessing models and leveraging them to achieve the optimal cost matrix required for cost-sensitive learning. The criteria used to evaluate the performance of the learned Bayesian network classifier are divided into metrics associated with economic and statistical factors.

2.1.1 Statistically Related Metrics

Based on the elements of the confusion matrix, metrics for classifier performance evaluation are defined and estimated. The accuracy of predicting positive (fraudulent) and negative (legitimate) transactions is denoted by A_p and A_n, respectively. Furthermore, A_t evaluates the prediction's accuracy, whether transactions are fraudulent or legitimate.

$$A_n(\text{Specificity}) = \frac{N_{TN}}{N_{TN} + N_{FP}} = \frac{N_{TN}}{N_N},$$
$$A_p(\text{Recall or Sensitivity}) = \frac{N_{TP}}{N_{TP} + N_{FN}} = \frac{N_{TP}}{N_P},$$

$$A_t(\text{Acc}) = \frac{N_{TP} + N_{TN}}{N_{TN} + N_{FP} + N_{TP} + N_{FN}} = \frac{N_{TP} + N_{TN}}{N_N + N_P}.$$

The balance between A_p and A_n is achieved by optimal configuration of cost matrix elements.

Usually, in classification problems with two imbalanced categories, the F1 criterion is used along with other criteria to evaluate a cost-sensitive model. This criterion is defined as the harmonic mean of recall, A_p, and the accuracy; and is calculated by (2.1)

$$F1 = 2 \times \left(\frac{A_p \times \text{Precision}}{\text{Precision} + A_p} \right); \tag{2.1}$$
$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} = \frac{N_{TP}}{N_{\acute{P}}}.$$

Since the harmonic mean of two numbers is close to the smaller one, a large value of the F1 indicates that both recall and accuracy are simultaneously large.

In the context of applying Bayesian network classifiers to finance, the cost matrix is often viewed as a univariate function representing the cost of mistakenly classifying fraudulent transactions as legitimate, denoted as $\mathbf{C}_{np}$ (Wang and *et al.* , 2023; Bei and *et al.* , 2021). This paper treats the issue as a bivariate problem, incorporating the costs of misclassifying transactions from both classes ($\mathbf{C}_{np}$ and $\mathbf{C}_{pn}$). To determine the optimal cost matrix value, a lattice search method is employed.

After completing the calculations, the cost that produces the maximum value of *economic efficiency* $(EE(k)T, S)$ or any of the A_p(Recall), A_n(Specificity), A_t(Acc), $F1$, and Kappa criteria is deemed the optimal cost. The CABNC technique is subsequently applied to classify the transactions based on the optimal cost matrix obtained.

2.1.2 Threshold Analysis

In cases where the cost matrix is unknown, the modification of the classification output to detect the class label of the new transaction is proposed as an alternative solution in addition to heuristically calculating the cost matrix.

To estimate the class label using statistical and economic evaluation criteria specific to the problem under study from the validation dataset, this method ignores the threshold standard value of 0.5. Definition (2.2) introduces a criterion, *economic efficiency*, to determine the optimal threshold for class label assignment.

In this manner, a transaction is classified as fraudulent and belonging to the positive class if the posterior probability of the positive class for that transaction is higher than the threshold value.

The financial costs that result from the failure to detect a fraudulent transaction are of great importance when confronting actual industry issues, irrespective of the results obtained by ranking algorithms. Therefore, in the industry, the models are evaluated not only based on statistical criteria but also according to what is known as economic efficiency (EE). In the realm of financial literature, transactions accurately classified as fraudulent (N_{TP}) avert capital losses for the financial institution, whereas transactions accurately classified as legitimate (N_{TN}) yield a profit of k percent of the transaction amount for the institution. Similarly, the financial loss resulting from an incorrectly identified fraudulent transaction (N_{FN}) will be equal to $100 \times (1-k)$ percent of the transaction amount because the financial institution makes k percent from such a transaction, while it is required to reimburse the cardholder 100 percent of the unauthorized transaction cost.

Definition 2.2 (*economic efficiency*)**.** *In financial and credit institutions, the monetary return from accurately classified legal transactions after subtracting cumulative losses arising from the misclassification of fraudulent transactions is defined as economic efficiency by (2.2).*

$$
\begin{aligned}
EE(k)T, S &= \sum_{i=1}^{N_{\acute{N}}} Return(\boldsymbol{t}_i);\\
Return(\boldsymbol{t}_i) &= \big(k \times v_i\big)\delta_i - \big((1-k) \times v_i\big)(1-\delta_i),\\
\delta_i &= \begin{cases} 1 & \boldsymbol{t}_i \textit{is accurately identified as legal} \\ 0 & \boldsymbol{t}_i \textit{is misclassified as legal} \end{cases}
\end{aligned} \tag{2.2}
$$

The monetary value of the ith transaction ($\mathbf{t}_i$) is denoted by v_i, whereas k represents the fraction of the transaction's monetary value that the financial institution retains as interest. De Sá and *et al.* (2018) defines the value of k to be 0.03 and calculates EE(0.03). The term $N_{\acute{N}}$ denotes the number of legally identified transactions, which is $N_{TN} + N_{FN}$. The function δ_i indicates whether $\mathbf{t}_i$ has been accurately identified as legal or incorrectly misclassified.

2.2 Bayesian Network Classifier (BNC)

The classification problem can be described as a procedure that, given a training set $\mathcal{D} = \{(\mathbf{t}_j, y)\}_{j=1}^{N_{train}}$ and an unclassified observation $\mathbf{t} = (t_1, \ldots, t_l)$, assigns a class label y.

A Bayesian network classifier addresses this task by first modeling the joint distribution $P(y, \mathbf{t})$ with a certain Bayesian network $\mathcal{B}$, and then calculating the posterior distribution $P(y|\mathbf{t})$ by Bayes' rule. A Bayesian network is characterized by a pair $\mathcal{B} =< \mathcal{G}, \Theta >$. The first component, $\mathcal{G}$, is a directed acyclic graph. The nodes in $\mathcal{G}$ represent random variables, including features $T_1, \ldots, T_l$ and the class variable Y. The second component of the pair, namely Θ, represents the set of parameters that quantifies the network. It contains a parameter $P_{\mathcal{B}}\big(t_k|y, pa(t_k)\big)$, the conditional probabilities induced by $\mathcal{G}$. The arcs in $\mathcal{G}$ represent directed dependencies between the nodes. If T_k points directly to T_l via a directed edge (an arc), we say T_k is the parent of T_l, which belongs to the parent set $pa(T_k)$. Different Bayesian network classifiers extended from Naive Bayes assume various dependencies among the attributes, but all suppose that Y is the parent of all attributes and has no parents.

A Naive Bayesian network (NB) defines a unique joint probability distribution given by

$$P_{\mathcal{NB}}(t_1, \ldots, t_l, y) = P_{\mathcal{NB}}(y) \prod_{k=1}^{l} P_{\mathcal{NB}}\big(t_k|pa(t_k)\big).$$

where $pa(t_k)$ presents the parent set of t_k. Unlike the NB's strong assumption about feature independence given the class, in the Tree Augmented Naive Bayesian (TAN) network, each feature has as parents at most one other attribute in addition to the class variable. The TAN model defines a unique joint probability distribution given by

$$P_{\mathcal{TAN}}(\mathbf{t}, y) = P_{\mathcal{TAN}}(y) \prod_{k=1}^{l} P_{\mathcal{TAN}}\big(t_k|y, pa(t_k)\big).$$

However, in more general structures, with no assumption about independence or the class variable as the root, the joint probability distribution is given by

$$P_{\mathcal{B}}(\mathbf{t}, y) = P_{\mathcal{B}}(y) P_{\mathcal{B}}(t_{root}|y) \prod_{k=1}^{l} P_{\mathcal{B}}\big(t_k|y, pa(t_k)\big).$$

By Bayes' rule, the posterior distribution of an unclassified instance $\mathbf{t}$ can be calculated as follows:

$$P_{\mathcal{B}}(y|\mathbf{t}) = \frac{P_{\mathcal{B}}(\mathbf{t}, y)}{\sum_y P_{\mathcal{B}}(\mathbf{t}, y)}. \tag{2.3}$$

where $P_{\mathcal{B}}(\mathbf{t}, y)$ denotes the joint probability distribution defined earlier. So we can easily classify instance $\mathbf{t}$ into class $\mathbf{arg\ max}_y\big(P_{\mathcal{B}}(y|\mathbf{t})\big)$.

2.3 Cost Augmented Bayesian Network Classifier (CABNC)

In this section, for the sake of robustness and interpretability, the Bayesian network classifier $\mathcal{B}$ is transformed into cost-augmented versions by incorporating misclassification costs into the loss function for class membership via modifying the outputs in the label-assigning mechanism. As a result, determining the optimal cost matrix is crucial to our approach. In a class-dependent cost-sensitive learning algorithm, the classifier is developed by minimizing an expected cost rather than concentrating on an error rate function. This paper uses the term "cost" interchangeably with "loss."

In the context of fraud detection, when implementing cost-sensitive binary classification based on the given BNC $\mathcal{B}$, through a meta-learning perspective, the confusion matrix is derived by heuristically modifying the elements of the cost matrix using equations (2.4) and (2.5).

$$L(\mathcal{C}_i, \mathbf{t}) = \sum_{j \in \{p,n\}} \mathbf{C}_{ij} \times P_{\mathcal{B}}(\mathcal{C}_j | t_1, \ldots, t_l), \quad i \in \{p, n\} \tag{2.4}$$

$$\mathbf{Class} = \mathbf{argmin}_{i \in \{p,n\}} L(\mathcal{C}_i, \mathbf{t}). \tag{2.5}$$

where expression $P_{\mathcal{B}}(\mathcal{C}_j | t_1, \ldots, t_l); j \in \{p, n\}$ represents the posterior probability for each class calculated by (2.3). $\mathbf{C}_{ij}$ is the cost of assigning label $\mathcal{C}_i$ instead of $\mathcal{C}_j$.

The calculation of the cost matrix and its application for modifying the classifier outputs involves a recursive process. Algorithm 1 outlines the overall framework for generating CABNCs from cost-insensitive BNCs. As stated previously, in the context of cost-augmented meta-learning, the original BNCs remain unchanged, with both the structure and associated parameters staying intact. The only aspect that varies is the classifier's outputs, which are modified according to (2.5).

3. Applications on Real Data

In this section, we implement the cost-sensitive approach proposed in this paper on a real dataset obtained from the banking system in Iran. The data were collected between May 4 and August 18, 2020, and contain information regarding commercial credit card transactions processed through a specific type of money transfer machine. To ensure confidentiality, the identification numbers of cardholders and other sensitive identifiers have been anonymized.

The issue at hand is a binary classification problem involving two imbalanced classes: fraud and legitimate. The analysis phase of the study is outlined as follows: preprocessing, BNC estimation (including structure and parameter learning), calculation of the cost matrix, and the generation of CABNC. Detailed comparisons of the developed models are also included.

3.1 Preprocessing

In the data preprocessing step, feature engineering, discretization, and dependency analysis were utilized to extract informative features that indicate the hidden social behavior of transactions. In this part of the data manipulation, we integrate multiple transactions between the same source and target cards into one, leading the underlying relationships as a one-sided network. The lack of reciprocal transactions in the data establishes a hierarchical framework within the network,

Algorithm 1 Cost Augmented BNC Generation Framework

Inputs:

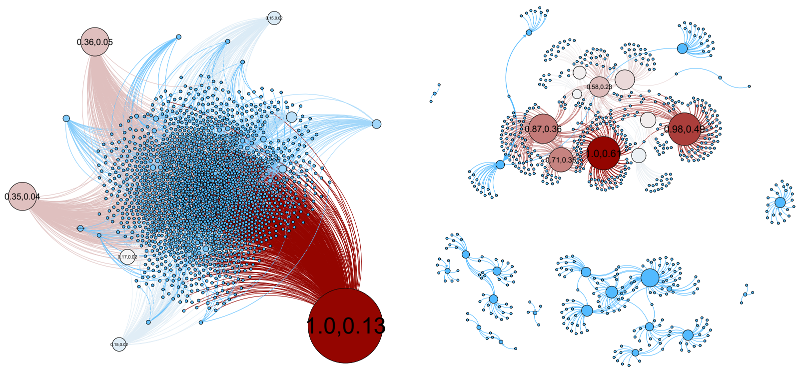
- Any cost-insensitive Bayesian network classifier including Naive Bayes (NB), Tree-Augmented Naive Bayes (TAN), and GHC-K2,
- Initialize the sets of cost matrix possible values as $\mathcal{C}_{np}$ and $\mathcal{C}_{pn}$.

1. Select the evaluation measure of interest, **eval**, from the set of relevant metrics: **eval** $\in \big\{EE(0.03)_{T,S}, A_p, A_n, A_t, Precision, F1, Kappa\big\}$,
2. Calculate the optimal cost matrix associated with the selected **eval** metric from step 1 and consider it as $\big(\mathbf{C}_{np}, \mathbf{C}_{pn}\big)$:
   (a) Initialize set $\mathbf{l_{eval}} = \emptyset$,
   (b) For $c_{np} \in \mathcal{C}_{np}$ and $c_{pn} \in \mathcal{C}_{pn}$ do:
      i. Calculate $L\big(\mathcal{C}_p, \mathbf{t}\big)$ and $L\big(\mathcal{C}_n, \mathbf{t}\big)$ by (2.4),
      ii. Do the class label assignment for $\mathbf{t}_i, i = 1, ..., N_{train}$ by (2.5),
      iii. Compute the evaluation measure $\mathbf{eval}_{(c_{np},c_{pn})}$ based on the results of 2(b)ii and add it into $\mathbf{l_{eval}} \longleftarrow \mathbf{l_{eval}} \cup \{\mathbf{eval}_{(c_{np},c_{pn})}\}$.
   (c) Calculate the maximum value for $\mathbf{l_{eval}}$ and denote $\mathbf{max\_eval}_{(\mathbf{c_{np}},\mathbf{c_{pn}})} \longleftarrow \mathbf{argmax}_{\mathbf{eval}_{(c_{np},c_{pn})}}\big(\mathbf{l_{eval}}\big)$ which corresponds to the optimal cost matrix $(\mathbf{C}_{np}, \mathbf{C}_{pn})$.
3. Considering the cost matrix $(\mathbf{C}_{np}, \mathbf{C}_{pn})$ obtained from step 2, for any new transaction $\mathbf{t}_{new}$, calculate the loss functions $L\big(\mathcal{C}_p, \mathbf{t}_{new}\big)$ and $L\big(\mathcal{C}_n, \mathbf{t}_{new}\big)$ by (2.4).
4. Update label assignment decision rule (2.5) associated with the step 3 findings.

Output: $\text{CABNC}_{\big(\mathbf{C}_{np},\mathbf{C}_{pn}\big)}$ is achieved by a combination of 3 and 4 results.

wherein source nodes, representing the cards that transmit funds, serve as hubs, while target nodes, or the cards that receive money, are regarded as authorities. In this configuration, one-sided money transfers uncover criminal patterns, with deposits made into target cards in a nonreciprocal manner. To leverage the structural characteristics of the transaction network, centrality measures pertinent to the cards are calculated and employed as variables for the development of Bayesian network models.

The interaction graph of fraudulent and legitimate transactions is illustrated in Figure 1, highlighting their distinct characteristics. The differing behaviors of these two classes are clearly observable.



(a) Legitimate transactions

(b) Fraudulent transactions

Figure 1: Subnet comparison of legitimate and fraudulent transactions. The size of the nodes corresponds to the eigenvector centrality value of the card, and the edges match the color of the recipient's card. (a) The intensity of the node color reflects the node's PageRank centrality criterion, indicating its importance. (b) The color of the node signifies the extent of the node's authority measure, which denotes the card's trustworthiness.

In the legitimate transactions subnet, larger nodes represent cards with elevated eigenvector values, serving as key recipients for other influential cards in the network. Among fraudulent transactions, the intensity of the red color indicates a higher authority centrality value, which in turn implies greater credibility for the corresponding node. Simply put, red cards represent those trusted by other cards and primarily function as key recipients. These nodes in the fraudulent transactions subnet reveal the existence of communities involved in illegal transactions.

On the other hand, in the legal transactions subnet, the intensity of the color indicates both greater PageRank centrality and the increased importance of the nodes (Figure 1(a)). In this subnet, nodes that are both large (influential) and prominent in color (important and key) represent crucial cards that significantly influence the subnet behavior due to their effect on the interactions of other transactions. Upon further investigation of the illegal transaction subnet in Figure 1(b), it becomes evident that two recipient cards play a substantial role, one having an authority close to 1 and the other at 0.98, along with eigenvector scores of 0.61 and 0.49. These cards serve both as reliable receivers of funds from illicit activities and indicate clusters of interconnected accounts due to their associations with other influential cards.

Additionally, the IR ratio of the transaction dataset is approximately three, indicating about three legitimate transactions for each fraudulent one. This results in 25.21 percent of all transactions being fraudulent. Consequently, the data are considered highly imbalanced. Figure 2 displays the explanatory dependence analysis done with the help of the mutual information matrix. The weighted degree of the target card, indicating the volume and frequency of the card's involvement in transactions as a recipient, probably plays an important role in determining the label of the transactions.

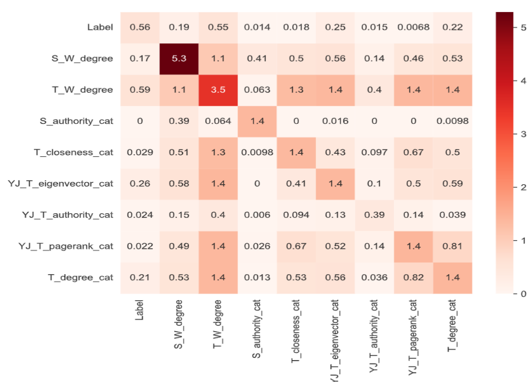


Figure 2: Pairwise mutual information matrix of extracted features reflecting the internal behavior of the network

3.2 Comparative analysis of BNCs

In this section, we divide the data into 30 percent for testing while using the remaining portion for training the model. In addition, to evaluate how the model performs on new transactions, the train-test split is conducted in such a way that the imbalance ratio (IR) remains consistent with the entire dataset, which is approximately 3 for both the training and testing subsets. The Bayesian network classifiers under review include Naive Bayes (NB), Tree-Augmented Naive Bayes

(TAN), and the results obtained from the Greedy Hill Climbing (GHC) search utilizing the K2 score function (GHC-K2). Remember that the structure and parameter learning of the Bayesian networks is not the primary focus of the paper, so we left this part of the model estimation to the methods found in the literature.

The classifiers are assessed through Economic Efficiency (EE(0.03)T,S) and various statistical metrics such as Ap (Recall), An (Specificity), Precision, F1, Kappa, and AUC scores for both ROC and PR curves.

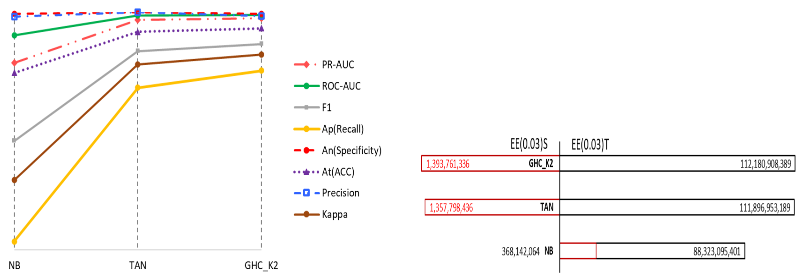
Table 1 reports the evaluation measures of the BNCs under investigation. Based on the economic efficiency determined by the amount of money transferred per transaction, GHC-K2, with a value of $112,180,908,389$, emerges as the top performer. This model excels in terms of Recall (0.951), F1 (0.973), Kappa (0.964), and At (0.987) as well. The findings of the AUC for both ROC and PR curves in Figure 3(a) and Table 1 indicate that the GHC-K2 classifier, with PR-AUC and ROC-AUC values of 0.9949 and 0.9980, respectively, outperforms the other two models.

Table 1: Comparison between BNCs based on Statistical and Economical Measures

Measure	NB	TAN	GHC-K2
Kappa	0.859	0.956	0.964
F1	0.892	0.967	0.973
An(Specificity)	0.999	1.000	0.999
At(ACC)	0.949	0.984	0.987
Ap(Recall)	0.807	0.937	0.951
Precision	0.996	1.000	0.997
EE(0.03)T	$88,323,095,401$	$111,896,953,189$	$112,180,908,389$
EE(0.03)S	$-368,142,064$	$1,357,798,436$	$1,393,761,336$
PR-AUC	0.9577	0.9939	0.9949
ROC-AUC	0.9807	0.9976	0.9980
Threshold	0.8096	0.3320	0.3054

Figure 3(b) illustrates that the GHC-K2 model demonstrated the highest economic efficiency regarding the flow of incoming transactions to the recipient cards, reaching a total of EE(0.03)T = 112,180,908,389. On the other hand, the NB model achieved the lowest economic efficiency related to incoming transactions to the target card, revealing a difference of 23,857,812,988 compared to the GHC-K2.

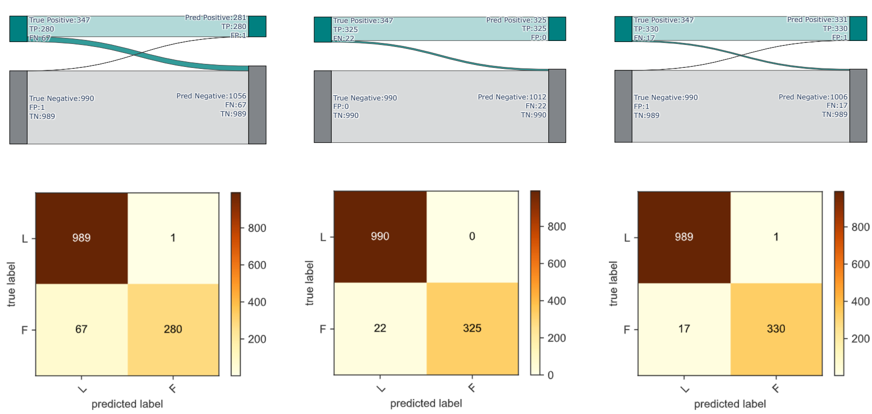
When evaluating outgoing transactions from issuer cards (EE(0.03)S), employing the NB model probably results in no profit for the financial institution, and it might also incur a maximum financial loss of 368,142,064.



(a) (b)

Figure 3: Comparison between BNCs, (a) statistical evaluation measures, (b) economic efficiency, EE(0.03)T,S.

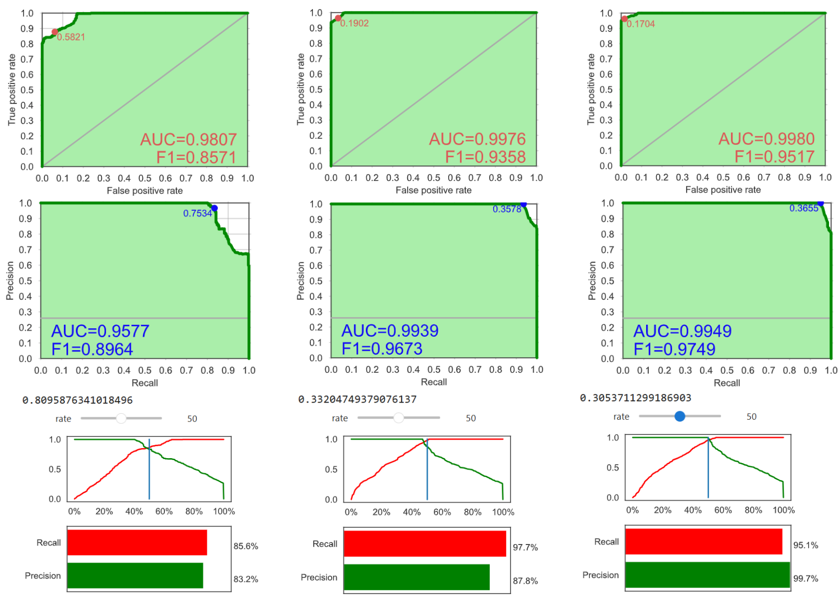
Figure 4 displays the performance of the BNCs in terms of confusion and Sankey matrices. According to this figure, evidently GHC-K2 performs well in creating the balance between precision and recall.



(a) NB (b) TAN (c) GHC-K2

Figure 4: Confusion Sankey-matrix for BNCs

This study does not adhere to the standard threshold of 0.5 for assigning labels according to the probabilities calculated for each class. Instead, it performs a threshold analysis using the PR curve to determine the most suitable value tailored to the dataset. Based on the results presented in Figure 5, the optimal threshold values are 0.33 for TAN and 0.31 for GHC-K2. For the Naive Bayes classifier, the ideal threshold is notably above 0.5, calculated as 0.8, indicating that transactions are marked as fraudulent when the estimated probability exceeds 80 percent. This significantly high threshold indicates that Naive Bayes functions as a very conservative classifier.



(a) NB {0.8096}

(b) TAN {0.3320}

(c) GHC-K2 {0.3054}

Figure 5: ROC-PR Curves for BNCs: {*Threshold for Label Assignment*}

3.3 Optimal Cost Matrix Calculation

As outlined in the methodology, in this part of the analysis we convert the ordinary cost-insensitive BNCs into their corresponding cost-augmented versions estimated earlier. To achieve this, by implementing Algorithm 1, introduced in Section 2.3, we compute a cost matrix optimized for any evaluation criteria specified in Table 1. From a misclassification cost standpoint, we disregard the diagonal elements of the cost matrix related to accurate classifications, $\mathbf{C}_{nn}$ and $\mathbf{C}_{pp}$, assigning both a near-zero value of 10^{-8}. Furthermore, for the remaining off-diagonal elements of the cost matrix, we apply the values presented in equation (3.6).

$$\begin{aligned}
&\mathbf{C}_{pn} \in \left\{10^4, 10^3, 100, 10, 1, 0.1, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\right\} \\
&\mathbf{C}_{np} \in \left\{10^4, 10^3, 100, 10, 1, 0.1, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\right\}
\end{aligned} \tag{3.6}$$

Subsequently, a grid search is performed using these two sets to identify the optimal CABNCs according to equation (2.4), with labels assigned by equation (2.5). The most effective CABNCs are determined by evaluating all the metrics listed in Table 1. The combination of matrix elements corresponding to the best-estimated CABNCs yields the optimal cost matrix. These elements, along with their related CABNCs, are represented as $\left(\mathbf{C}_{np}, \mathbf{C}_{pn}\right)$ and CABNC$_{\left(\mathbf{C}_{np}, \mathbf{C}_{pn}\right)}$, respectively.

3.4 Cost Augmented BNC Generation

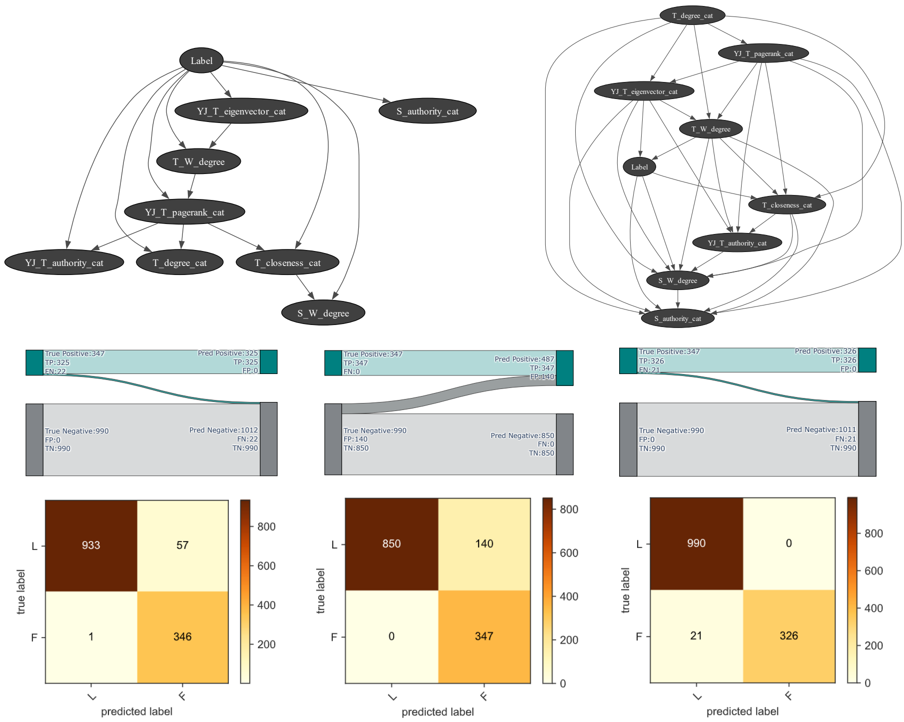
The main goal of the paper is class label prediction in a cost-sensitive meta-learning approach called CABNC models. This section analyzes such cost-sensitive classifiers.

The findings of the three top-performing cost-sensitive CABNCs, evaluated by various metrics, are presented in Table 2 and illustrated in Figure 6. Table 2 identifies the CATAN$_{\left(10^4,10^3\right)}$ obtained through cost matrix optimization following the EE(0.03)T,S measure as the best performer. This model achieves the highest economic efficiency value (EE(0.03)T=113,002,862,129). In addition, the cost matrix that is optimal for Ap (Recall) is determined as $\left(10^4, 10^2\right)$, regardless of the BNCs being analyzed. Furthermore, the corresponding CATAN$_{\left(10^4,10^2\right)}$, which has values of F1=0.832, At=0.895, Ap=1.000, Precision=0.713, An=0.859, Kappa=0.759, and EE(0.03)T=112,781,093,731, ranks first among the other two cost-augmented BNCs in this category.

Additionally, an exploration of Table 2 indicates that when considering cost matrix calculations using different statistically derived metrics beyond Recall, CAGHC-K2$_{\left(10^4,10^4\right)}$ stands out as the most effective cost-sensitive classifier compared to its two counterparts. This model exhibits strong performance in accurately distinguishing between fraudulent and legitimate transactions, achieving metrics of Ap=0.939 and An=1.000. With an EE(0.03)T value of 111,965,861,989, it demonstrates a satisfactory level of economic efficiency as well.

3.5 CABNCs versus BNCs

To improve the performance of BNCs in scenarios with class imbalance, the authors propose CABNCs. This section reports the comparative analysis of BNCs alongside their corresponding CABNCs.



(a) CATAN$_{(10^4,10^3)}$ $\{EE(0.03)T, S\}$

(b) CATAN$_{(10^4,10^2)}$ $\{Ap(Recall)\}$

(c) CAGHC-K2$_{(10^4,10^4)}$ $\{F1, Kappa, At, An, Precision\}$

Figure 6: Confusion Sankey-matrix for best CABNs: $\{Metric\ for\ Cost\ Matrix\ Optimization\}$

According to Table 3, typically, the effectiveness of cost-sensitive TANs does not improve when assessed using statistical measures. However, comparison of TAN and CATAN$_{(10^4,10^4)}$ reveals that the closest resemblance is achieved when the cost matrix is optimized based on statistical metrics including Kappa, F1, An, and Specificity. Nonetheless, applying CATANs, especially when the cost matrix is calculated by financial efficiency (EE(0.03)T,S) and Ap, boosts the model's capability to accurately detect fraudulent transactions while also enhancing financial efficiency (comparison of cost-sensitive CATAN$_{(10^4,10^3)}$ and CATAN$_{(10^4,10^2)}$ models with the ordinary TAN model). Moreover, analyzing the standard NB and GHC-K2 models alongside their cost-sensitive counterparts depicts that computing the cost matrix using statistical metrics apart from Ap tends to diminish model effectiveness, while merely providing a slight improvement in the model's capacity to accurately identify legitimate transactions, quantified by An and Precision.

Table 3 demonstrates that when a model is designed to generalize findings and make decisions about new instances, evaluating the model through statistical measures indicates that optimizing the cost matrix based on these criteria is not advisable. Compared to traditional models, the economic efficiency of cost-sensitive versions is not only diminished, but the CANB and CAGHC-K2 also reveal a slight improvement in performance regarding An and Precision (comparison of CAGHC-K2$_{\left(10^4,10^4\right)}$ and CANB$_{\left(10^3,10^4\right)}$ with GHC-K2 and NB, accordingly). However, CATANs show no differences from the standard TAN model, even when evaluated using these metrics.

3.6 Discussion

According to the findings, the GHC-K2 classifier stands out as the top performer among the BNCs analyzed in this research, showcasing an EE(0.03)T of $(112,180,908,389)$, with a Recall of (0.951), F1 score of (0.973), Kappa value of (0.964), and At of (0.987), making it the best option from both a financial and statistical viewpoint.

Furthermore, through the analysis of the AUC of the PR-curve, we determined the threshold values for assigning class labels for the GHC-K2, TAN, and Naive Bayes classifiers at 0.31, 0.33, and 0.81, correspondingly.

Additionally, during the cost-sensitive analysis phase, irrespective of the BNCs being investigated, the optimal cost matrix specific to Ap is established as $\left(10^3, 10^2\right)$. The cost matrix values optimized according to F1, Specificity, and Accuracy for the GHC-K2 and TAN models are determined to be $\left(10^4, 10^4\right)$, while for the NB model, the values are $\left(10^3, 10^4\right)$. Moreover, concerning the economic efficiency measure, the corresponding matrix for the NB and TAN models is calculated to be $\left(10^4, 10^3\right)$, and for the GHC-K2 model, it is $\left(10^{-6}, 10^{-7}\right)$.

Based on the comparative analysis done in this paper, two economically efficient and powerful models to detect fraudulent transactions are Cost Augmented models created by TAN with cost matrix optimized in terms of Ap (Recall) and EE(0.03)T,S, denoted by CATAN$_{\left(10^4,10^3\right)}$ and CATAN$_{\left(10^4,10^2\right)}$, respectively. However, a statistically well-performing model obtained in this study is the CAGHC-K2$_{\left(10^{-6},10^{-7}\right)}$ model associated with the F1-specific optimal cost matrix.

Ultimately, refining the cost matrix according to statistical evaluation measures affects the statistical efficacy and economic efficiency of CABNCs when compared to conventional BNC models. On the other hand, by determining the ideal cost matrix using the EE(0.03)T,S and Ap metrics, the economic efficiency of the cost-sensitive model is enhanced, and the model's ability to accurately identify fraudulent transactions improves as well.

4. Conclusion and Future Work

This paper investigates the issue of credit card fraud detection using a binary classification approach consisting of two steps. Firstly, we address the problem by employing three different types of Bayesian network classifiers, ranging from Naive Bayes to more sophisticated models such as TAN and GHC-K2. Traditional Bayesian network classifiers are designed to minimize misclassification errors. When applied to class imbalance learning tasks, their performance decreases generally. Hence, to improve their effectiveness, we incorporate the cost-sensitive meta-learning technique into various BNCs and develop the CABNCs in this research. So, we contribute to the literature by enhancing the performance of the traditional BNCs via CABNC generation and cost-sensitive modification of the BNC's output in a meta-learning line of research. The proposed approach allocates varying costs to different class instances, motivating classifiers to concentrate on class instances with greater misclassification costs.

Due to the financial impact of the misclassification of illicit transactions, we evaluate the classifiers based on Economic Efficiency, EE(0.03)T,S, as defined in this paper. To generate CABNCs, we perform cost matrix calculations specific to various evaluation measurements. The developed cost-sensitive CABNCs are compared regarding both economic efficiency and relevant statistical metrics, including F1, Ap (recall), An (specificity), At (accuracy), and Kappa. Experimental findings demonstrate that our recommended cost-sensitive CABNCs significantly outperform the original cost-insensitive BNCs, particularly in terms of economic efficiency and recall.

In industrial applications where Economic Efficiency and accurate prediction of fraudulent transactions are critical simultaneously, it is advisable to employ the cost-sensitive CATAN classifier. However, if the principal objective is to forecast the class label of forthcoming transactions, it is recommended to utilize the CAGHC-K2 associated with a cost matrix computed based on statistical metrics other than Ap.

There are several possible directions for future research. To keep it straightforward, we limit our investigation to three fundamental Bayesian Network structures: Naïve Bayes, TAN, and its improved version utilizing a hill-climbing method. Consequently, a key area for future research will be to broaden our existing analysis to incorporate the K2 algorithm for developing Bayesian Network structures. Investigation of cost-sensitive Bayesian Networks through an instance-dependent weighting approach would also be another attractive line of future research. Furthermore, studying the temporal aspects of transactions through utilizing dynamic Bayesian networks within a cost-sensitive context presents another promising research avenue.

Acknowledgment

The ICT Research Institute (IRAN Telecommunication Research Center (ITRC)) and Technology Development Council in the field of Connectivity and Communications, Vice-Presidency for Science, Technology, and Knowledge-Based Economy, have provided financial support for this research under the agreement number $500//12018 - 1402/7/23$. The paper's authors would like to thank them for their unwavering dedication to facilitating research-related activities. Additionally, they are deeply appreciative of the research mentor, Dr. Marjan Goodarzi, for her insightful notes and encouragement, which significantly improved the quality of the paper. Lastly, they would like to express their appreciation to the members of the editorial board and reviewers for their constructive criticism and valuable feedback, which improved the clarity and consistency of this research paper.

References

Asha, R.B., and SureshKumar KR. (2021), Credit Card Fraud Detection using Artificial Neural Network, *Global Transitions Proceedings,* **2(1)**, 35-41.

Bei, H., Wang, Y., Ren, Zh., Jiang, Sh., Li, K., and Wang, W. (2021), A Statistical Approach to Cost-Sensitive AdaBoost for Imbalanced Data Classification, *Mathematical Problems in Engineering,* **2021**, 3165589.

Brown, I. and Mues, C. (2012), An experimental comparison of classification algorithms for imbalanced credit scoring data sets, *Expert systems with applications,* **39(3)**, 3446-3453.

Caldeira, E., Brandao, G., Campos, H., and Pereira, A. (2012), Characterizing and evaluating fraud in electronic transactions, in: *Proceedings of the Latin American Web Congress,* 115–122.

Cateni, S., Colla, V. and Vannucci, M. (2014), A method for resampling imbalanced datasets in binary classification tasks for real-world problems, *Neurocomputing,* **135**, 32-41.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002), SMOTE: synthetic minority over-sampling technique, *Journal of artificial intelligence research,* **16**, 321-357.

Chawla, N.V., Cieslak, D.A., Hall, L.O. and Joshi, A. (2008), Automatically countering imbalance and its empirical relationship to cost, *Data Mining and Knowledge Discovery,* **17**, 225-252.

Cheah, P.C.Y., Yang, Y. and Lee, B.G. (2023), Enhancing financial fraud detection through addressing class imbalance using hybrid SMOTE-GAN techniques, *International Journal of Financial Studies,* **11(3)**, 110.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S. and Bontempi, G. (2014), Learned lessons in credit card fraud detection from a practitioner perspective, *Expert systems with applications,* **41(10)**, 4915-4928.

De Sá, A.G.C., Pereira, A.C.M., and Pappa, G.L. (2018), A Customized Classification Algorithm for Credit Card Fraud Detection, *Engineering Applications of Artificial Intelligence,* **72**, 21-29.

Domingos, P. (1999), Metacost: A general method for making classifiers cost-sensitive, In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining,* 155-164.

Elkan, C. (2001), The foundations of cost-sensitive learning, In: *International joint conference on artificial intelligence, Lawrence Erlbaum Associates Ltd.,* **17(1)**, 973-978.

Fan, W., Stolfo, S.J., Zhang, J. and Chan, P.K. (1999), AdaCost: misclassification cost-sensitive boosting, *Icml,* **99**, 97-105.

Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., and Herrera, F. (2018), *Learning from Imbalanced Data Sets*, Switzerland: Springer.

Fu, K., Cheng, D., Tu, Y., and Zhang, L. (2016), Credit Card Fraud Detection Using Convolutional Neural Networks, In: *Proceedings of the International Conference on Neural Information Processing, Springer*, 483–490.

Gamini, P., Yerramsetti, S.T., Darapu, G.D., Pentakoti, V.K., Raju, V.P. (2021), Detection of Credit Card Fraudulent Transactions using Boosting Algorithms, *Journal of Emerging Technologies and Innovative Research (JETIR),* **8(2)**, JETIR2102248.

Hens, A.B., Tiwari, M.K. (2014), A Novel Model for Credit Card Fraud Detection Using Artificial Immune Systems, *Applied Soft Computing,* **24**, 40–49.

Hens, A.B., Tiwari, M.K. (2012), Computational Time Reduction for Credit Scoring: an Integrated Approach Based on Support Vector Machine and Stratified Sampling Method, *Expert Systems with Applications,* **39(8)**, 6774–6781.

Höppner, S., Stripling, E., Baesens, B., vanden Broucke, S. and Verdonck, T. (2020), Profit driven decision trees for churn prediction, *European journal of operational research,* **284(3)**, 920-933.

Jiang, L., Li, C. and Wang, S. (2014), Cost-sensitive Bayesian network classifiers, *Pattern Recognition Letters,* **45**, 211-216.

Johnson, J.M., Khoshgoftaar, T.M. (2019), Survey on deep learning with class imbalance, *Journal of big data,* **6(1)**, 1-54.

Kotsiantis, S.B. (2011), Cascade generalization with reweighting data for handling imbalanced problems, *The Computer Journal,* **54(9)**, 1547-1559.

Krawczyk, B., Woźniak, M. and Schaefer, G. (2014), Cost-sensitive decision tree ensembles for effective imbalanced classification, *Applied Soft Computing,* **14**, 554-562.

Kumar, M.D., Mubarak, A., and Dhanush, M.S. (2020), Credit card fraud detection using Bayesian belief network, *International Journal of Research in Engineering, Science and Management,* **3(7)**, 316-319.

Lucas, Y., Portier, P.E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., and Calabretto, S. (2020), Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs, *Future Generation Computer Systems,* **102**, 393-402.

de Morais, R.F., Miranda, P.B. and Silva, R.M. (2016), A meta-learning method to select under-sampling algorithms for imbalanced data sets, In: *Proceedings of the 5th Brazilian Conference on Intelligent Systems (BRACIS),* 385-390.

Provost, F. (2000), Machine learning from imbalanced data sets 101, In: *Proceedings of the AAAI'2000 Workshop on Imbalanced Data.*

Prusti, D., Das, D., and Rath, S.K. (2021), Credit card fraud detection technique by applying graph database model, *Arabian Journal for Science and Engineering,* **46(9)**, 1-20.

Sahin, Y., Bulkan, S., and Duman, E. (2013), A Cost-Sensitive Decision Tree Approach for Fraud Detection, *Expert Systems with Applications,* **40(15)**, 5916–5923.

Seera, M., Lim, C.P., Kumar, A., Dhamotharan, L., and Tan, K.H. (2024), An intelligent payment card fraud detection system, *Annals of operations research,* **334(1)**, 445-467.

Sheng, V.S. and Ling, C.X. (2006), Thresholding for making classifiers cost-sensitive, *Aaai,* **6**, 476-481.

Singh, A., Ranjan, R. K., and Tiwari, A. (2021), Credit Card Fraud Detection under Extreme Imbalanced Data: A Comparative Study of Data-level Algorithms, *Journal of Experimental and Theoretical Artificial Intelligence,* **34(3)**, 571–598.

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M. and Baesens, B. (2015), APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions, *Decision support systems,* **75**, 38-48.

Wang, X., Bouzembrak, Y., Lansink, A. G. J. M. O., and Fels-Klerx, H. J. van der (2023), Weighted Bayesian network for the classification of unbalanced food safety data: Case study of risk-based monitoring of heavy metals, *Risk Analysis,* **43(12)**, 2549-2561.

Yu, H. and Ni, J. (2014), An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data, *IEEE/ACM transactions on computational biology and bioinformatics,* **11(4)**, 657-666.

Zadrozny, B. and Elkan, C. (2001), Learning and making decisions when costs and probabilities are both unknown, In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining,* 204-213.

Zadrozny, B., Langford, J. and Abe, N. (2003), Cost-sensitive learning by cost-proportionate example weighting, In: *Third IEEE international conference on data mining,* 435-442.

Zhang, Y.D., Zhang, Y., Phillips, P., Dong, Z. and Wang, S. (2017), Synthetic minority oversampling technique and fractal dimension for identifying multiple sclerosis, *Fractals,* **25(04)**, 1740010.

Zhao, H. (2008), Instance weighting versus threshold adjusting for cost-sensitive classification *Knowledge and Information Systems,* **15**, 321-334.

Table 2: Comparison between CABNCs by optimization metrics

Optimization Metric	Measure Name	Model Name: CANB	CATAN	CAGHC-K2
EE(0.03)T,S	Cost Matrix	$(10000, 1000)$	$(10000, 1000)$	$(10e^{-6}, 10e^{-7})$
	Kappa	0.701	0.893	0.871
	F1	0.794	0.923	0.907
	An(Specificity)	0.820	0.942	0.933
	At(ACC)	0.866	0.957	0.948
	Ap(Recall)	0.997	0.997	0.988
	Precision	0.660	0.859	0.839
	EE(0.03)T	$111, 435, 110, 441$	$113, 002, 862, 129$	$112, 784, 055, 239$
	EE(0.03)S	$1, 419, 459, 848$	$1, 454, 840, 966$	$1, 431, 696, 966$
Kappa,F1 At(Acc) Precision An(Specificity)	Cost Matrix	$(1000, 10000)$	$(10000, 10000)$	$(10000, 10000)$
	Kappa	0.843	0.954	0.958
	F1	0.879	0.966	0.969
	An(Specificity)	1.000	1.000	1.000
	At(ACC)	0.944	0.983	0.984
	Ap(Recall)	0.784	0.934	0.939
	Precision	1.000	1.000	1.000
	EE(0.03)T	$86, 409, 761, 839$	$111, 852, 546, 589$	$111, 965, 861, 989$
	EE(0.03)S	$-484, 522, 564$	$1, 294, 457, 436$	$1, 365, 267, 436$
Ap(Recall)	Cost Matrix	$(10000, 100)$	$(10000, 100)$	$(10000, 100)$
	Kappa	0.489	0.759	0.679
	F1	0.666	0.832	0.781
	An(Specificity)	0.648	0.859	0.803
	At(ACC)	0.74	0.895	0.854
	Ap(Recall)	1.000	1.000	1.000
	Precision	0.499	0.713	0.64
	EE(0.03)T	$104, 069, 427, 877$	$112, 781, 093, 731$	$111, 972, 542, 792$
	EE(0.03)S	$1, 148, 457, 905$	$1, 414, 735, 523$	$1, 400, 164, 733$

Table 3: Comparison of the classifiers: BNCs vs CABNCs

Model Name	Kappa	F1	Precision	At	Ap	An	EE(0.03)T	EE(0.03)S
NB	0.859	0.892	0.996	0.949	0.807	0.999	88, 323, 095, 401	−368, 142, 064
CANB$_{(10^4,10^3)}$ EE(0.03)T,S	0.701	0.794	0.660	0.866	0.997	0.820	111, 435, 110, 441	1, 419, 459, 848
CANB$_{(10^4,10^2)}$ Ap(Recall)	0.489	0.666	0.499	0.739	1	0.648	104, 069, 427, 877	1, 148, 457, 905
CANB$_{(10^3,10^4)}$ F1,Kappa,At,An,Precision	0.843	0.879	1	0.944	0.784	1	86, 409, 761, 839	−484, 522, 564
TAN	0.956	0.967	1	0.984	0.937	1	111, 896, 953, 189	1, 357, 798, 436
CATAN$_{(10^4,10^3)}$ EE(0.03)T,S	0.893	0.923	0.859	0.957	0.997	0.942	113, 002, 862, 129	1, 454, 840, 966
CATAN$_{(10^4,10^2)}$ Ap(Recall)	0.759	0.832	0.713	0.895	1	0.859	112, 781, 093, 731	1, 414, 735, 523
CATAN$_{(10^4,10^4)}$ F1,Kappa,At,An,Precision	0.954	0.966	1	0.983	0.934	1	111, 852, 546, 589	1, 294, 457, 436
GHC-K2	0.964	0.973	0.997	0.987	0.951	0.999	112, 180, 908, 389	1, 393, 761, 336
CAGHC-K2$_{(10^{-6},10^{-7})}$ EE(0.03)T,S	0.871	0.907	0.839	0.948	0.988	0.933	112, 784, 055, 239	1, 431, 696, 966
CAGHC-K2$_{(10^4,10^2)}$ Ap(Recall)	0.679	0.781	0.640	0.854	1	0.803	111, 972, 542, 792	1, 400, 164, 733
CAGHC-K2$_{(10^4,10^4)}$ F1,Kappa,At,An,Precision	0.958	0.969	1	0.984	0.939	1	111, 965, 861, 989	1, 365, 267, 436