applied sciences

MDPI

Article

Predicting Success of Outbound Telemarketing in Insurance Policy Loans Using an Explainable Multiple-Filter Convolutional Neural Network

Jinmo Gu [1,†], Jinhyuk Na [2,†], Jeongeun Park [1] and Hayoung Kim [1,*]

[1] Graduate School of Information, Yonsei University, Yonsei-ro 50, Seodaemun-gu, Seoul 03722, Korea; koos808@yonsei.ac.kr (J.G.); park.je@yonsei.ac.kr (J.P.)
[2] Personal Loan Digital Marketing Support Center, KYOBO Life Insurance, Jong-ro 1, Jongno-gu, Seoul 03154, Korea; najh79@kyobo.com
* Correspondence: hayoung.kim@yonsei.ac.kr
† These authors contributed equally.

Abstract: Outbound telemarketing is an efficient direct marketing method wherein telemarketers solicit potential customers by phone to purchase or subscribe to products or services. However, those who are not interested in the information or offers provided by outbound telemarketing generally experience such interactions negatively because they perceive telemarketing as spam. In this study, therefore, we investigate the use of deep learning models to predict the success of outbound telemarketing for insurance policy loans. We propose an explainable multiple-filter convolutional neural network model called XmCNN that can alleviate overfitting and extract various high-level features using hundreds of input variables. To enable the practical application of the proposed method, we also examine ensemble models to further improve its performance. We experimentally demonstrate that the proposed XmCNN significantly outperformed conventional deep neural network models and machine learning models. Furthermore, a deep learning ensemble model constructed using the XmCNN architecture achieved the lowest false positive rate (4.92%) and the highest F1-score (87.47%). We identified important variables influencing insurance policy loan prediction through the proposed model, suggesting that these factors should be considered in practice. The proposed method may increase the efficiency of outbound telemarketing and reduce the spam problems caused by calling non-potential customers.

Keywords: outbound telemarketing; deep learning; machine learning; convolutional neural network; insurance policy loan; explainability

Citation: Gu, J.; Na, J.; Park, J.; Kim, H. Predicting Success of Outbound Telemarketing in Insurance Policy Loans Using an Explainable Multiple-Filter Convolutional Neural Network. *Appl. Sci.* **2021**, *11*, 7147. https://doi.org/10.3390/app11157147

Academic Editor: Federico Divina

Received: 1 July 2021
Accepted: 30 July 2021
Published: 2 August 2021

1. Introduction

The recent advancements in digital technology and the accelerating development of global markets are completely changing consumers' patterns of living and spending. Consumers' preference for contactless, remote interaction channels has increased, and they have become accustomed to using mobile technology to obtain their desired services and information almost anytime and anywhere. To respond to this situation and gain a competitive economic advantage while avoiding potential negative business outcomes, companies are attempting to provide services tailored to the digital age while increasing the convenience of contactless channels and the proportion of direct marketing.

Hence, the importance of telemarketing is highlighted as a means of implementing direct marketing strategies, and the focus of telemarketing is shifting from passive inbound calls to outbound calls, which are cost-effective and active marketing methods. In the inbound method, customers are encouraged to subscribe to products or services when they call a call center. In contrast, in the outbound method, a telemarketer calls customers and invites them to subscribe to a product or service. Therefore, the development of technology to accurately select potential customers who are likely to purchase a product is important.

As shown in Table 1, several studies have proposed various machine learning and deep learning (DL) prediction models to predict telemarketing success. However, because most of these studies analyzed the success of telemarketing methods carried out by banks, the extension of their results to various financial companies such as insurance and security companies involves significant limitations. In this study, we aim to predict the success of outbound telemarketing methods in the relatively sparsely researched field of insurance, with particular emphasis on insurance policy loan prediction. An insurance policy loan is a service allowing customers to withdraw and spend a portion of an insurance coverage amount in advance while maintaining coverage. Withdrawals and repayments are possible at any time, and loans can be made without any loan review procedures, such as credit evaluation and proof of income. Customers can apply through common channels such as personal computers (PCs), mobile phones, call centers, and automated teller machines (ATMs), without visiting a branch of the insurance company. These insurance policy loans are important for insurers because they have a positive effect of suppressing the increase in insurance debt as an advance payment. Therefore, insurance companies are working to increase the size of insurance policy loans, and for this purpose, increasing the efficiency of outbound telemarketing operations is crucial.

Table 1. Comparison of present work with previous studies on predicting telemarketing success.

Authors	Title	Models	Number of Input Features
Moro et al. [1]	A data-driven approach to predict the success of bank telemarketing	- Logistic regression - Decision tree - Artificial neural networks - Support vector machine	22
Kim et al. [2]	Predicting the success of bank telemarketing using deep convolutional neural network	- Deep convolutional neural networks	16
Asare-Frempong et al. [3]	Predicting customer response to bank direct telemarketing campaign	- Artificial neural networks - Decision tree - Logistic regression	16
Koumétio et al. [4]	Optimizing the prediction of telemarketing target calls by a classification technique	- Naïve Bayes classifiers - Decision tree - Artificial neural networks - Support vector machine	21
Ghatasheh et al. [5]	Business analytics in telemarketing: cost-sensitive analysis of bank campaigns using artificial neural networks	- Artificial neural networks - Support vector machine - Random forest	16
Turkmen [6]	Deep learning-based methods for processing data in telemarketing-success prediction	- Long short-term memory - Gated recurrent unit - Simple recurrent neural networks	20
Authors of the present study	Predicting Success of Outbound Telemarketing in Insurance Policy Loans Using an Explainable Multiple-Filter Convolutional Neural Network	- Random forest - Support vector machine - Gradient boosting machine - eXtreme gradient boosting - light gradient boosting machine - Deep neural networks - Deep convolutional neural networks	210

However, insurance policy loan outbound telemarketing data are large-scale and high-dimensional. Predicting the success of telemarketing requires a variety of data related to customer attributes, insurance characteristics, insurance transactions, insurance policy loan transactions, and marketing campaigns. With such high-dimensional data, traditional machine learning (ML) modeling cannot achieve suitable predictive performance. Also, basic deep neural networks (DNNs) composed of only fully connected layers are likely to be subject to overfitting. Therefore, we propose a deep learning model to prevent overfitting even when many variables are used.

1.1. Objectives

The main objective of the present work is to predict the success of outbound telemarketing for insurance policy loans from data comprising a large number of users and transactions. To this end, we propose a convolutional neural network (CNN)-based prediction model that can prevent overfitting even for large numbers of variables using local connections and weight sharing.

As poor predictive success can lead to customer dissatisfaction, high practical performance capabilities are a key priority in the prediction of outbound telemarketing success. If an insurer makes an incorrect call to a customer who has never used an insurance policy loan or has no intention of doing so, such customers are likely to regard the call as spam, which could lead to a decline in customer experience. Therefore, the development of models with a high prediction success rate is very important. Hence, in this study, we focus on extracting high prediction performance by using all the various data related to insurance policy loans available in practice by utilizing the advantages of deep learning. Moreover, we observe that the results of deep learning techniques are generally difficult to interpret owing to the black-box nature of such models. To address this issue in this context, we propose an explainable multiple-filter CNN (XmCNN) model with enhanced explanatory power to enable users visually to identify variables with an important effect on predicting outbound telemarketing success and to rapidly interpret the meaning of such information in practical outbound telemarketing operations.

1.2. Contributions

The main contributions of this study can be summarized as follows:

- First research in the field of insurance policy loans: We propose an explainable deep learning model based on a CNN architecture to predict the success of outbound telemarketing using insurance policy loan data. To the best of our knowledge, the present work is the first in the field.
- New dataset configuration: We utilize newly constructed data to predict the success of outbound telemarketing of insurance policy loans. This dataset comprised 153 variables extracted from 44,412 customers.
- Information loss minimization: We used high-dimensional insurance policy loan data consisting of more than 200 input dimensions without feature selection, which allowed the advantages of a deep learning model to be exploited by extracting features from input data and minimizing information loss due to feature selection.
- Performance superiority and feasibility in practice: We confirmed that the proposed XmCNN model significantly outperformed the machine learning and deep learning models used for comparison. In particular, an ensemble model built with the proposed deep learning model showed the lowest false positive rate and the highest F1-score. Therefore, the experimental results indicate that our proposed model can reduce negative outbound telemarketing outcomes, which are detrimental to customer experience.
- Improvement of model explanatory power: The proposed interpretable model exhibited the ability to identify important variables applicable in practical operations.

The remainder of this study is organized as follows: Section 2 summarizes the characteristics of outbound telemarketing related to the research topic and prior studies related to predicting the success of outbound telemarketing. In addition, the concepts of DNN and CNN are briefly explained along with some relevant prior works. Section 3 describes the architecture and components of the proposed CNN-based prediction model. Section 4 presents an analysis of the experimental results and important features. Section 5 reports the research results, and Section 6 discusses the contributions as well as the theoretical and practical implications of the present work. Finally, Section 7 presents our conclusions and outlines possible directions for future research. In addition, detailed explanations of the variables and abbreviations used in this study are provided in the Appendix A.

2. Background

2.1. Outbound Telemarketing

Outbound telemarketing methods are based on offering products or services based on a customer database. Therefore, the development of data-based marketing systems through the construction of a customer database and data mining is crucial. Inbound telemarketing relies on Q&A-oriented scripts, whereas outbound telemarketing utilizes marketing scripts that are strategically written according to the products and services offered. Outbound telemarketing requires advanced recommendation skills and operator expertise. Unlike random phone sales, outbound telemarketing requires preliminary preparation for placing calls, and call connection and marketing success are critical. The advantage of outbound telemarketing is that it can maximize the effectiveness of sales efforts by providing only the necessary information to customers and recommending sales within a short period.

Most studies related to telemarketing success prediction have focused on bank telemarketing (term deposit), and various prediction models have been proposed. Published studies that used machine learning or deep learning are the following: Moro et al. [1] proposed an ML model to predict the success of telemarketing for long-term bank deposits. They analyzed 150 features related to bank customers, products, and socioeconomic attributes and selected 22 features from these. Feature extraction by feature selection generally results in information loss, and no standard feature extraction method has been developed to prevent this. However, as the proposed approach applies a deep learning model, the features are self-extracted; therefore, it is preferable to input all the available variables. The authors of the abovementioned work additionally compared four ML models, including logistic regression (LR), a decision tree (DT), artificial neural networks (ANNs), and a support vector machine (SVM) [7] and found that the ANN model yielded the best results. Kim et al. [2] studied a deep convolutional neural network (DCNN) designed to predict the success of bank telemarketing. They analyzed 16 finance-related attributes. Eight numeric attributes included age, balance, duration of the last contact, number of contacts, number of days passed after the last contact, number of contacts before a specified campaign, and day and month of the last contact, while eight nominal attributes included employment, marital status, education, loan default status, housing, loan amount, and communication type (either cellular or telephone). DCNN-based models were examined in various structural experiments considering factors such as the number of layers, learning rate, initial value of nodes, and other parameters. Their proposed model exhibited a higher performance compared to other traditional ML models.

Asare-Frempong et al. [3] compared multilayer perceptron (MLP), DT, LR, and random forest (RF) [8] algorithms to predict the success of bank telemarketing and found that the RF model outperformed other models. In addition, results from a cluster analysis to identify customer characteristics revealed that customers with higher call durations were more likely to subscribe to term deposits. Koumétio et al. [4] proposed a new classification technique computing a specific similarity for each type of feature. The similarity was estimated by calculating the Euclidean distances for each center of the two classes. The classifier used 21 variables and predicted the class of clients more accurately than four ML models, including a naïve Bayes classifier, a DT, an ANN, and an SVM. They revealed the duration of the call as the most important attribute. However, in reality, this variable is a property that can only be known after performing telemarketing; therefore, its practical use is limited. Turkmen [6] used three types of recurrent neural networks to predict bank telemarketing, including a long short-term memory network, a gated recurrent unit, and simple recurrent neural networks. The synthetic minority oversampling technique (SMOTE) [9] approach was also used to obtain more accurate results. Experimental results showed that the long short-term memory model using SMOTE outperformed other models. Ghatasheh et al. [5] proposed an ANN model for bank telemarketing prediction using 16 variables and compared the performance of traditional machine-learning classifiers. They found that, the Type II and Type I errors of their proposed model were higher and lower, respectively, than those of other models. The authors suggested that their proposed approach would be of benefit in decision-making processes in terms of understanding the probability of clients subscribing to term deposits. In comparison, our proposed model showed lower Type II errors compared to other models. We tried to minimize the Type II errors to reduce the frequency of unwanted contacts. Applying the same approach on other real data. Developing self-explanatory decision process systems or algorithms.

Lee [10] proposed a method to improve the performance of methods predicting probable paying customers using a stacked deep network. Additionally, Lee applied hybrid sampling to balance the amount of data between categories. Hosein et al. [11] presented a mathematical model that can increase the success of telemarketing campaigns under limited currency budgets. They reduced marketing costs by determining the optimal number of calls for each chosen customer. In addition to the studies in Table 1, various architectures and methodologies have been developed to predict telemarketing success [12–14].

However, despite many related studies on predicting telemarketing success, several limitations remain. First, most studies on predicting telemarketing success use a Portuguese Bank dataset, a public dataset provided by the University of California Irvine. The Portuguese Banks dataset is a well-organized repository containing 45,147 instances with 17 attributes with no missing values. Because many variables used in actual business operations were not considered, variables with a significant effect on telemarketing predictions in practice could not be provided. In other words, many studies have suggested methodologies related to models, but very few studies have constructed datasets that can be applied meaningfully and universally in practice. Second, many studies related to telemarketing success have low performance for practical use, owing to their use of shallow models. In addition, various feature selection methods were applied to solve the data imbalance problem. In the process of feature selection, information loss occurs because data observations and variables are reduced compared to raw data. Third, many studies mentioned above focused more on performance rather than interpreting important variables. However, our study utilizes high-dimensional data consisting of more than 200 input dimensions without feature selection. In addition, we propose a DL model that emphasizes the explanatory power of key variables that influence the success of outbound telemarketing for insurance policy loans.

2.2. Deep Neural Networks

DNN architectures model a structure similar to human neurons, comprising layers of neurons used to create and train numerous connections. The greater the number of layers of neurons stacked, the more complex the conceptual features that can be found in the data; thus, the performance of DNN model is improved compared to that of shallower networks. DNNs can be divided into three main layers, including an initial layer called the input layer, a final layer called the output layer, and the layer between the input layer and the output layer, called the hidden layer. The input layer refers to the layer in which the data are entered, and the number of input layers equals the number of input variables. The output layer determines the number of nodes based on the data type of the response variable. In contrast to the ANN technique, a deep neural network increases the representation capacity of the model by increasing the number of hidden layers. Hence, it can solve more complex problems with improved performance.

An activation function is used to pass signals from the input layer to the hidden layer and from the hidden layer to the output layer. The activation function of the hidden layer typically applies nonlinear and non-decreasing functions, such as the rectified linear unit (ReLU) and sigmoid functions. A suitable activation function is then used in the output layer. For example, binary classification typically uses logistic and multiclass classification using a softmax function. In addition, the dropout technique is commonly used to prevent overfitting, supporting learning by randomly removing a certain percentage of neurons. Dropout prevents co-adaptation phenomena from moving together with similar weights as if they were a single neuron [15].

2.3. Convolutional Neural Networks

CNN models are among the most popular deep learning methods. DNN learns global patterns in the input feature space using fully connected layers, whereas CNN learns the local patterns within a relatively small window. Fully connected networks need to learn new patterns appearing in new locations, but convolutional networks can be generalized by learning a small number of training samples. The convolutional operation is applied to a 3D tensor called a feature map, which consists of two spatial axes (height and width) and a depth axis.

The convolution operation extracts small patches from the input feature map and applies the same transformation to all these patches to create an output feature map. In the convolution layer, a filter determining the output depth of the feature map is used, and the number of filters is a model hyperparameter. The output height and width can differ from the input's height and width, and padding can be used to obtain an output feature map with the same height and width as the input. Padding adds an appropriate number of rows and columns to the edges of the input feature map. Downsampling using pooling can be performed to prevent overfitting in convolutional networks. Pooling reduces the number of weights of feature maps. The maximum pooling method takes the maximum value for each channel of the input patch, whereas the average pooling method calculates and transforms the average value; both methods are commonly used. Figure 1 illustrates a general CNN architecture.

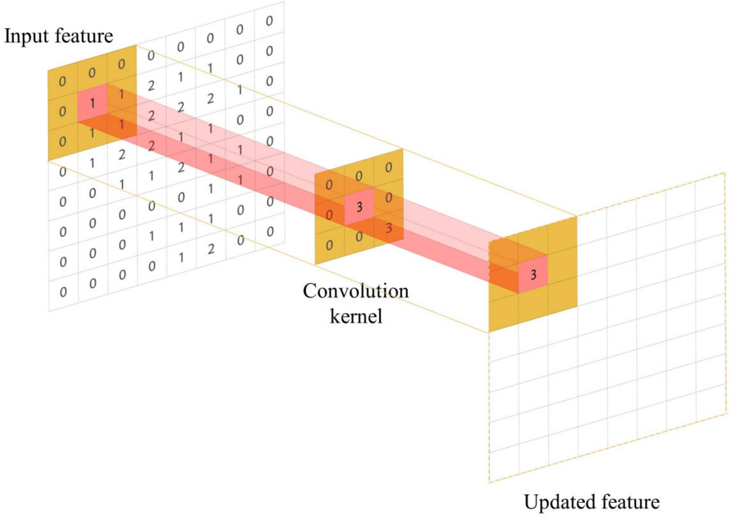

Figure 1. Illustration of convolutional neural network.

Many studies related to CNN have shown excellent performance on unstructured data such as images, video, voice, and audio [16]. Moreover, in many studies using text data, models combining CNN have been developed and have demonstrated suitable performance [17–21]. The proposed model incorporates a CNN model with high-dimensional business tabular data; the related works are described briefly as follows. Neagoe et al. [22] studied a DCNN model versus an MLP model for financial predictions. Their experimental results confirmed the effectiveness of the DCNN model for credit scoring using bank transaction data. The performance of the DCNN model was significantly better than that of an MLP model. Zhang et al. [23] proposed the application of a CNN model to traditional data. Twelve types of traditional tabular data were used, and ML models such as eXtreme gradient boosting (XGBoost) [24], SVM, RF, MLP, and *k*-nearest neighbor clustering were compared with the CNN model. The performance of CNN was demonstrated to be equivalent to that of state-of-the-art XGBoost techniques. The experimental results demonstrated the importance of considering CNN models for classification tasks using traditional data. Kvamme et al. [25] proposed an application of a CNN model to consumer transaction data to predict mortgage defaults. They used time-series data of bank accounts and compared them with LR, MLP, RF, and ensemble models. An ensemble model composed of CNN and RF presented the best results, and the performance increased with the length of the time series examined. De Caigny et al. [26] proposed the incorporation of textual information into customer churn prediction (CCP) models based on a CNN. They used raw data from a financial service provider and confirmed that the inclusion of textual data in a CCP model improved its predictive performance. In addition, the experimental results showed that the CNN model outperformed the current best practices for text mining in CCP. The aforementioned studies achieved high performance using CNN models, following which the proposed approach also uses a CNN model as the base.

2.4. Ensemble Classifier

Ensemble techniques adopt different perspectives on different aspects of the problem, which can be combined to make better-quality decisions. Ensemble classifiers construct a complex model comprised of single models, and generally outperforms single models by integrating the prediction results of all classifiers. Therefore, *k* trained models can be combined to create a newly improved complex model. As shown in Figure 2, a voting strategy is commonly used to combine the predictions of the ensemble classifier to generate new predictions. There are several ways to create multiple classifiers. Different classifiers can be used, and different training data or architectures can be used within the same classifier [27]. In this study, various architectures were constructed to generate multiple classifiers, and the best results were obtained through voting, which was categorized as hard voting or soft voting. Hard voting selects the mode of results presented by single models as a final result, whereas soft voting selects a final result based on the average value of the result probabilities presented by single models. In this study, soft voting based on the average value of the probability of each classifier result was applied to combine the predictions, and the final prediction decision was made using the average value.

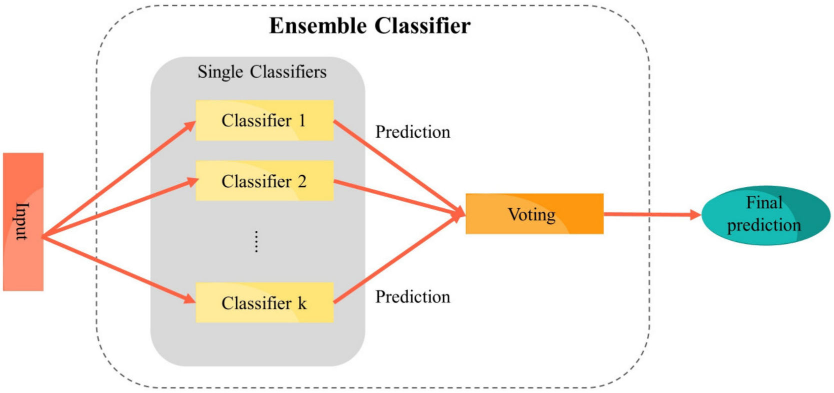

Figure 2. Framework of ensemble classifier.

3. Method

As shown in Figure 3, the experimental procedure of this study consists of data collection, preprocessing, training models, and performance evaluation of the analysis models. Section 3.1 describes in detail the data collection and data preprocessing processes for insurance companies that performed outbound telemarketing of insurance policy loans. Sections 3.2–3.4 explain the data analysis process performed using ML models and DL models, and Section 3.5 discusses an ensemble approach to maximize performance. Finally, Section 3.6 presents the model evaluation criteria. Additionally, we provide a detailed description of the data, such as data distribution, in Table A1 of Appendix A.

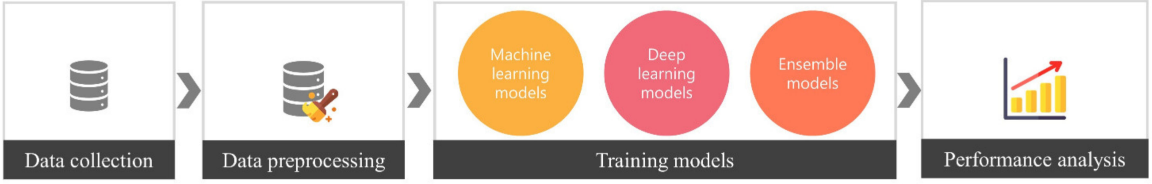

Figure 3. Overview of the analysis procedure.

3.1. Data Description and Preprocessing

In this study, we use the insurance policy loan outbound telemarketing data from a domestic life insurance company that performs outbound telemarketing for insurance customers. The data covered an eight-month period from March to October 2020. The raw data before preprocessing are data collected from 171,424 people allocated for outbound telemarketing of insurance policy loans, as shown in Table 2. Among the marketing targets, 64,359 customers attempted a call, 49,727 customers completed the call, and 45,155 customers completed the insurance policy loan information. Finally, the target variable is whether to execute a loan within one month after the completion of the insurance policy loan guide, and 8530 customers have executed the loan. Among the customers who received loan information, the proportion of customers who executed loans was 18.9%. This refers to the success rate of outbound telemarketing.

Table 2. Number of customers by outbound telemarketing stage.

Stage	Marketing Targets	Attempted Call	Completed Call	Loan Information Completed	Loan Execution [1]
Number of customers	171,424	64,379	49,727	45,155	8530

[1] Loan execution: Number of customers who executed a loan within one month of completion of the loan information.

The dataset consisted of numerical and nominal attributes. The number of numerical variables was 128, and the data range of numerical variables was transformed to a value between [0,1] through min-max scaling. The number of nominal attributes before data preprocessing was 25, including gender, whether e-mail or mobile phones were accepted, whether "Do not call" was registered, whether complaints were received over the past two years, whether the customer had insurance policy loans, and whether they had personal pension tax benefits. The 25 categorical variables were converted to 82 numerical dummy variables using one-hot encoding techniques, and each data representation value was set to either 0 or 1. Therefore, the input of the CNN was 210 data representations, all of which were numerical types.

As shown in Table 3, six types of analysis data were used, namely customer characteristic information, insurance transaction information, insurance policy loan transaction information, general loan transaction information, campaign execution information, and call list information. There are a total of 210 representation dimensions for each variable, and 210 data representations are used in the analysis. The customer characteristic information consisted of 72 variables, such as the customer's age and occupation, and the insurance transaction information includes 55 variables, such as insurance type, payment amount, and withdrawal amount. There are 66 variables, such as loan experience, execution frequency, and limit exhaustion rate, as variables related to insurance policy loans.

Table 3. Descriptions of data category.

Category	Descriptions	Number of Items
Customer characteristics	Age, occupation, region of residence, usage channel, number of complaints, etc.	72
Insurance transaction	Number of contracts, insurance type, payment amount, number of overdue, withdrawal amount, contract channel, insured information, etc.	55
Insurance policy loan	Loan experience, execution frequency, limit exhaustion rate, interest rate, usage period, balance, repayment amount, overdue, loan channel, etc.	66
General loan	Loan balance, number of overdue in the last 3 months, number of overdue in the last 1 year, etc.	3
Campaign execution	Recent campaign experience, number of recent campaign executions, number of campaign executions in the current month, etc.	9
Call list	Mobile experience, ARS experience, existing call list groups, etc.	5
	Total	210

Through data cleansing for missing values and outliers, 44,412 data sets were confirmed, 70% (31,089 cases) of the total data were used as training data, and the remaining 30% (13,323) were used as validation and testing data. In addition, the SMOTE was applied to the training data to solve the imbalance problem between classes of target variables. As shown in Table 4, while maintaining the scale of the number of "Loans not executed" in the major group, the proportion of "Loans executed" in the minor group increased from 19.2% to 50.0%.

Table 4. Comparison before and after SMOTE technique to solve class imbalance problem.

	Loans Not Executed	Loans Executed	Total
Before SMOTE	25,118 (80.8%)	5971 (19.2%)	31,089 (100.0%)
After SMOTE	29,855 (50.0%)	29,855 (50.0%)	59,710 (100.0%)

3.2. Proposed Method

In this study, we propose an explainable multiple-filter CNN architecture (XmCNN) that efficiently extracts useful information from many variables of insurance policy loan outbound telemarketing data. Existing ML models have a curse of dimensionality problem in which the data required for training increases exponentially as the input dimension increases. If the training data are insufficient, the predictive model may not be able to generalize well and may overfit the training data. Therefore, it is essential to select variables with high feature importance when training ML models. However, because deep learning models can have a large capacity, they are suitable for high-dimensional data and can solve more complex problems. Our proposed XmCNN model is also trained using all variables without feature selection because it aims to achieve good performance without feature selection. As shown in Figure 4, our proposed model consists of three parts: the input, feature extractor, and classifier. All three parts were trained using an end-to-end method directly considering the inputs and outputs to optimize the network weights.

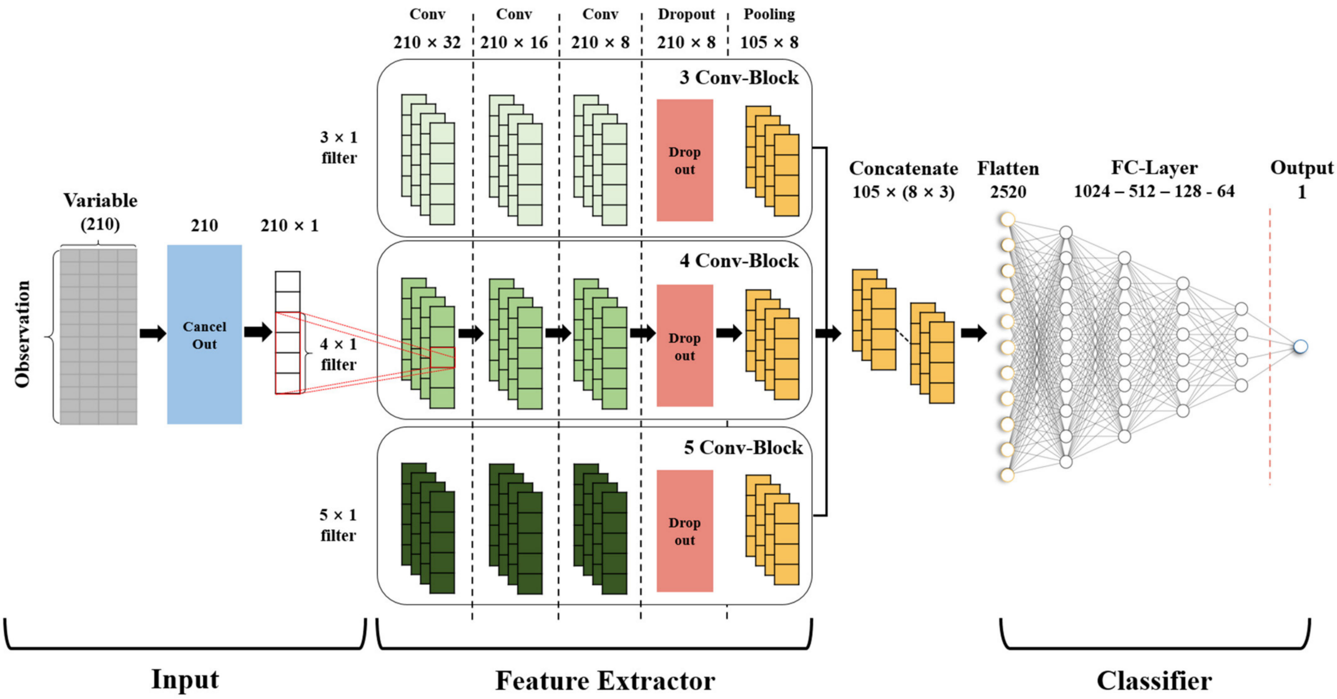

Figure 4. The proposed explainable multiple-filter CNN Architecture (XmCNN).

As shown in Figure 4, the input stage consists of a CancelOut [28] layer to identify important variables and a reshape layer to convert CancelOut output into appropriate inputs of the convolutional layer. To identify variables that significantly impact performance and make the model explainable, we calculate feature importance by adding a CancelOut layer after the input layer.

CancelOut is a new layer for deep neural networks that can be used for Feature Ranking and Feature Selection tasks. The CancelOut layer has only one connection to one particular input and as seen in Equation (1), the CancelOut layer is to update weights so that irrelevant features will be canceled out with a negative weight. In Equation (1), X is the input vector, ⊗ denotes element-wise multiplication, σ is the sigmoid activation function, and $W_{CancelOut}$ is the weight of the CancelOut layer. The weights $W_{CancelOut}$ in the CancelOut layer are initialized to a uniform distribution using an additional β coefficient, as in Equaiton (2), because random initialization is undesirable. In Equation (2), n_X is the size of the input layer, and β is a coefficient to control the initial output value. By adding the CancelOut layer, the value after the activation function of the CancelOut layer indicates the contribution to the output of the corresponding variable, and important variables can be extracted through the trained weight. After the CancelOut layer, the output of the CancelOut layer is reshaped to (210 × 1) to be used as the input of the convolutional layer:

$$CancelOut(X) = X \otimes \sigma(W_{CancelOut}) \quad (1)$$

$$W_{CancelOut} \sim \mathbf{u}\left(-\frac{1}{\sqrt{n_X}} + \beta, \frac{1}{\sqrt{n_X}} + \beta\right) \quad (2)$$

The feature extractor of the CNN extracts features from many variables of the input data. In general, the feature extractor step is divided into a convolutional layer for extracting features and a pooling layer for sub-sampling the extracted features. In particular, the convolutional layer extracts useful features from the input using filters and activation functions. The filter is moved in the height direction using a 1D convolutional layer, which is suitable for the dataset of this study because it expresses local features well regardless of location. However, because our dataset has 210 input dimensions, information could be lost in the process of feature extraction. To solve this problem, we used three filters of different sizes, rather than a single filter. Multiple filters have been used in natural language processing and computer vision, and it has been demonstrated that convolutional layers applying multiple filters and feature maps can have greater capacity [29]. In particular, using multiple filters has the advantage that different kernels can detect various features of a local region [30], and the model performance is improved compared to using only a single filter.

The filter window size of the 1D convolutional layer determines the amount of context information to be extracted from the variable. In this study, we use the window size of each filter as {3,4,5} and extract multiple features from multiple filters. Three 1D convolutional layers with different filter sizes accept input data and extract individual feature maps. Because the number of feature maps is determined by the number of filters, the proposed model compresses features by reducing the number of filters from 32 to 16 and then 8. In addition, padding is applied before convolution. The "same" padding is applied to ensure that the output of the convolution operation has the same length as the original input. Padding refers to the addition of zero values to the edges of the input image matrix to prevent the output values from becoming smaller and lost.

Additionally, the dropout layer is placed after the last convolutional layer. Dropout is a regularization method, and most CNN models use dropout to prevent overfitting. After the dropout layer, the proposed model includes an average pooling layer added to the subsample of the extracted features. Average pooling has the advantage of obtaining invariance, which is advantageous for classification and can reduce the CNN feature dimensions by summarizing spatial information. The process from convolutional layer to average pooling is called a conv-block, and our model consists of three conv-blocks with different filter sizes. Subsequently, features extracted from the three conv-blocks are concatenated as integrated features. This process is similar to the inception module [31] that concatenates the results of each filter, and features extracted from various local regions are combined.

The concatenated feature is transferred to the input of the classifier part, as shown in Figure 4. The classifier in the CNN calculates the probability value of the target label. The classifier consists of five fully connected layers, a dropout layer, and an output layer that calculates the probability values. As shown in Equation (3), the final output value of the output layer is calculated as a value between zero and one using the sigmoid function. Finally, if the output value is greater than 0.5, it is classified as a success of outbound telemarketing (class 1); if the output value is less than 0.5, it is classified as a failure in outbound telemarketing (class 0). In Equation (3), $\boldsymbol{s_i}$ is an element of the input vector for the sigmoid function. The loss function is learned in the direction in which the cross-entropy is minimized, as shown in Equation (4). In Equation (4), $\boldsymbol{t_i}$ is the actual class value, C is the number of classes, and cross-entropy is used to calculate the dissimilarity between the actual and predicted values:

$$\boldsymbol{f(s_i) = \frac{1}{1+e^{-s_i}}} \tag{3}$$

$$\mathbf{CE} = -\sum_{i}^{C} \boldsymbol{t_i} \mathbf{log}(\boldsymbol{f(s_i)}) \tag{4}$$

3.3. Comparative Machine Learning Models

We compared five ML models that are mainly used for business tabular data. As comparative ML models, RF, SVM, gradient boosting machine [32], eXtreme gradient boosting, and light gradient boosting machine [33] were used.

The RF algorithm is a model that generates multiple decision trees and combines the predictions of each tree to make a conclusion. An SVM is a binary linear classification model classifying two groups of data with a p-dimensional space using a p-1 dimensional hyperplane. In other words, SVM is an algorithm that finds a decision boundary with the largest margin. In addition, boosting models commonly used in classification problems were also used. Among them, the gradient boosting machine (GBM) is an ML model that combines several weak models (weak learners) to develop a single strong model (strong learner) with improved accuracy. Similar to RF, this is an ensemble method that combines several decision trees into a single model. Furthermore, XGBoost model and a light gradient boosting machine (LightGBM) were used. Unlike the gradient boosting model, the XGBoost model improves the learning speed through parallel execution and is more robust against overfitting by adding an overfitting regulation function. In addition, XGBoost uses a weighted quantum sketch for efficient proposal calculation and a novel sparsity-aware algorithm for parallel tree learning. Finally, unlike the general GBM tree division method, LightGBM uses a leaf-wise method. LightGBM uses two novel techniques, including gradient-based one-side sampling and exclusive feature bundling.

Each ML model was optimized to improve performance. For optimization, hyperparameters were tuned through grid search and K-fold cross-validation. Grid search is a method used to find optimal parameters by trying all possible combinations of candidate parameters. Grid search has the disadvantage that it requires a long time, but it is widely used because it improves the generalization performance of ML models.

K-fold cross-validation was used to verify the performance of the proposed model and increase the statistical reliability. We divided the entire dataset into five groups using five-fold cross-validation and performed five evaluations. Four subsets were used as training data, and the remaining were used as testing data. Then, the testing set was evaluated while changing without overlapping, and the performance of the model was evaluated by averaging the five evaluation indicators. We used the F1-score as a model evaluation index to select and evaluate the model. Finally, through optimization, a final prediction model was determined by finding an optimal hyperparameter combination for each prediction model. In addition, a fixed seed was set up to compare the results of each model.

3.4. Comparative Deep Learning Models

We generated a comparative deep learning model to verify the performance and effectiveness of the proposed XmCNN model. First, we created a basic deep neural network model with fully connected layers stacked to check the effect of the convolutional layer. The DNN model was configured identically to the classifier part of the proposed CNN architecture. Second, to determine whether using multiple filter sizes instead of a single filter improves performance, we compared the model with a convolutional neural network $\mathrm{CNN_S}$ using a single filter with a filter window size of 3, 4, and 5. The hyperparameters, CancelOut layer setting, and experimental settings of the DNN and $\mathrm{CNN_S}$ models were the same as those of the proposed XmCNN model.

3.5. Ensemble Approaches

We combined the advantages of a single model by building single machine learning models as ensemble models. The soft voting ensemble technique was used, and the final result was calculated and verified based on the average value of the predicted probabilities of the trained single models. To construct an optimal ensemble model, the backward removal method was applied according to the F1-score order of the single-model verification result. A total of 26 ensemble models ($\sum_{r=2}^{5} {}_5C_r$) were created and verified with all combinations of comparative ML models, such as RF, SVM, GBM, XGBoost, and LightGBM.

The DL model also maximized performance by creating an ensemble model. Our DL ensemble model uses a soft voting method that calculates the average probability of all classes obtained from DNN, $\mathrm{CNN_{S(3)}}$, $\mathrm{CNN_{S(4)}}$, $\mathrm{CNN_{S(5)}}$ and XmCNN models and selects the class with the highest average value. A total of 26 ensemble ($\sum_{r=2}^{5} {}_5C_r$) were created by combining five DL models in the same way as the ML ensemble model combination. When the DL model was an ensemble model, each DL model was trained five times. For example, for an ensemble of three DL models, DNN, $\mathrm{CNN_{S(4)}}$, and XmCNN, each model was independently trained five times, and then a total of 15 models were ensembled. Even with the same model, there was a slight difference in performance due to the initial weight value and hyperparameter tuning, so that the diversity of the model can be secured. In general, it is known that the performance of DL ensemble model increases when different DL models are combined. Additionally, it has been empirically demonstrated that constructing DL models as ensemble models can improve the accuracy, uncertainty, and out-of-distribution robustness of each model.

3.6. Evaluation Criteria

We used the false positive rate (FPR), false negative rate (FNR), recall, precision, accuracy, and F1-score as indicators to evaluate the model's performance, and the calculation formula is shown in Table 5. To offer exhaustive evaluations, these classification performance metrics have been comprehensively used in related research [1–6,10–14]. The confusion matrix which is the basis for calculating the performance evaluation index is presented in Table 6. The confusion matrix is the most common way to evaluate performance. In Table 6, each row represents an actual value, and each column represents a predicted value. A true positive (TP) and a true negative (TN) indicate a correct classification, which implies that the predicted class and the actual class match. By contrast, false negative (FN) and false positive (FP) indicate incorrect classification, which implies that an actual positive was predicted as negative and an actual negative was classified as positive.

Table 5. Measures of model performance.

Measures	Formulation
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
False Negative Rate (FNR)	$\frac{FN}{TP+FN}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$
F1-score	$2 \times \frac{Precision \times Recall}{Precision+Recall}$

Table 6. Confusion matrix for positive and negative records.

		Predict	
		Positive	**Negative**
Actual	**Positive**	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

Accuracy is the ratio of predicting the actual loan execution as loan execution, and the actual loan not execution as loan not execution among all targets. Recall is the ratio predicted by loan execution among the actual loan execution targets, and precision is the ratio of the actual loan execution among the targets predicted by loan execution. The F1-score is commonly used to accurately evaluate model performance for imbalanced data and is calculated as the harmonic average of recall and precision. The accuracy of the prediction model is important because the purpose of this study is to predict the success or failure of outbound telemarketing operations.

In addition, for efficient telemarketing operations, the selection performance of marketing to target customers is an important factor. Therefore, recall and precision must also be checked. Finally, the data imbalance problem of the training data was solved using the SMOTE technique. However, because the validation and testing data had a class imbalance problem, it was necessary to measure the F1-score, FPR, and FNR. In particular, the FPR metric in this study indicates the probability of placing calls perceived as spam to customers who do not require information on insurance policy loans. In other words, the FPR metric represents the percentage of customers who perceive telemarketing contacts as spam. Therefore, FPR is an important metric used to evaluate performance. As described above, the performance of predictive models was evaluated using all six indicators, and in particular, recall, precision, and F1-score were calculated as macro averages. Based on the evaluation metrics of our research, a good prediction model should have a high F1-score and Accuracy and low FPR and FNR.

4. Experimental Results

The objective of this study was to construct a predictive model to recommend insurance policy loans to customers with a high probability of successful outbound telemarketing of insurance policy loans. To do so, we collected data for analysis and determined the input variables through preprocessing using oversampling (SMOTE), normalization (min-max scaling), and one-hot encoding. Furthermore, we compared the performance of the proposed model with those of machine learning models, DNN models, CNN models, and ensemble models. The performance of each model was verified using the testing data.

4.1. Comparison of Machine Learning Model Results

Table 7 shows the performance of comparative ML models such as RF, SVM, GBM, XGBoost, and LightGBM. As mentioned in Section 3.2, each model is a final prediction model selected by tuning hyperparameters through grid search and *k*-fold cross-validation. According to the analysis shown in Table 7, it may be observed that LightGBM exhibited the best performance among comparative ML models, with an accuracy of 0.8384. In addition, the F1-score and FPR outperformed other ML models in most aspects. However, owing to the imbalance between classes, the overall F1-score generally showed low performance. In addition, the FNR, which is actually a positive ratio but was predicted to be negative, was mostly high. As a result, ML models tend to focus more on predicting outbound telemarketing failure (class 0) than predicting outbound telemarketing success (class 1).

Table 7. Performance evaluation of machine learning models.

Model	FPR	FNR	Recall	Precision	Accuracy	F1-Score
RF	0.0632	0.7506	0.5931	0.6629	0.8030	0.6073
SVM	0.4658	0.0154	0.7594	0.6656	0.6218	0.5990
GBM	0.1497	0.6369	0.6067	0.6081	0.7554	0.6074
XGBoost	0.1352	0.4710	0.6968	0.6847	0.7993	0.6903
LightGBM	0.0821	0.3264	0.6635	0.7541	0.8384	0.6903

4.2. Performance Analysis of the Proposed Model and Deep Learning Models

In Table 8, we compare the performance of the proposed XmCNN model and the comparative DL models. The comparative DL model was a DNN model and three CNN_S models. As mentioned in Section 3.4, the DNN model was composed of a fully connected layer, and the three CNN_S models implemented convolutional neural networks using only a single filter size of 3, 4, and 5.

Table 8. Performance evaluation of deep learning models.

Model	FPR	FNR	Recall	Precision	Accuracy	F1-Score
DNN	0.1143	0.1870	0.8493	0.7918	0.8715	0.8143
$\text{CNN}_{\text{S}(3)}$	0.0898	0.1941	0.8581	0.8177	0.8899	0.8352
$\text{CNN}_{\text{S}(4)}$	0.0887	0.1902	0.8605	0.8201	0.8915	0.8376
$\text{CNN}_{\text{S}(5)}$	0.0887	0.1889	0.8612	0.8204	0.8918	0.8387
XmCNN(***proposed model***)	0.0756	0.1916	0.8664	0.8366	0.9018	0.8502

DNN: Deep neural networks with fully connected layers. $\text{CNN}_{\text{S}(3)}, \text{CNN}_{\text{S}(4)}, \text{CNN}_{\text{S}(5)}$: Convolutional neural networks using a single filter with a filter window size 3,4,5. XmCNN: Convolutional neural networks using CancelOut and multiple filters with a filter window size {3,4,5}.

As shown in Table 8, the proposed XmCNN model outperformed the DNN model and the three CNN_S models, which are comparative DL models. First, among the CNN_S models, the F1-score of the $\text{CNN}_{\text{S}(5)}$ model was the highest at 0.8387, but the overall performance of the CNN_S models was similar. The accuracy of $\text{CNN}_{\text{S}(5)}$ increased by 2.33% compared to that of the DNN model, and the F1-score improved by 2.92%. This means that when a convolutional layer was added, the performance was improved compared to a DNN composed of only a fully connected layer. In addition, the accuracy of the proposed XmCNN model increased by 1.12%, and the F1-score increased by 1.44% compared to the $\text{CNN}_{\text{S}(5)}$ model using only a single filter size of 5. Therefore, it implies that multiple filters are effective in improving model performance.

The hyperparameters of the proposed model were optimized and selected based on the performance of the validation data. We used the Adam optimizer [34] with an initial learning rate of 0.001 when training the DL models. We set up the learning rate decay scheduling to decrease the learning rate according to the change in validation loss. We halved the learning rate if the validation loss did not improve for 30 epochs. In addition, all DL models were trained with 500 epochs and mini-batch sizes of 64. We utilized TensorFlow 2.4 and Keras on a single NVIDIA GeForce RTX 3080 to perform the experiments.

Additionally, the results of comparison between the performance of the DL model and the ML model are provided in Figure 5. ML models generally exhibited poor performance, owing to the large number of variables. However, DL models have significantly higher accuracy and F1-score compared to comparative ML models. Among them, the proposed XmCNN model outperformed all the comparative ML and DL models. The accuracy and F1-score of the XmCNN model were 0.9018 and 0.8502, respectively, which were 7.56% and 23.16% higher than those of LightGBM, which had the best performance among the ML models. As a result, DL models predicted both positive and negative classes better than ML models, which means that meaningful information was extracted well from numerous features. Compared to the ML approach, the CNN model using the dropout regularization technique and the convolutional layer not only improved the representative capacity but also overcame the curse of dimensionality problem better by preventing overfitting.

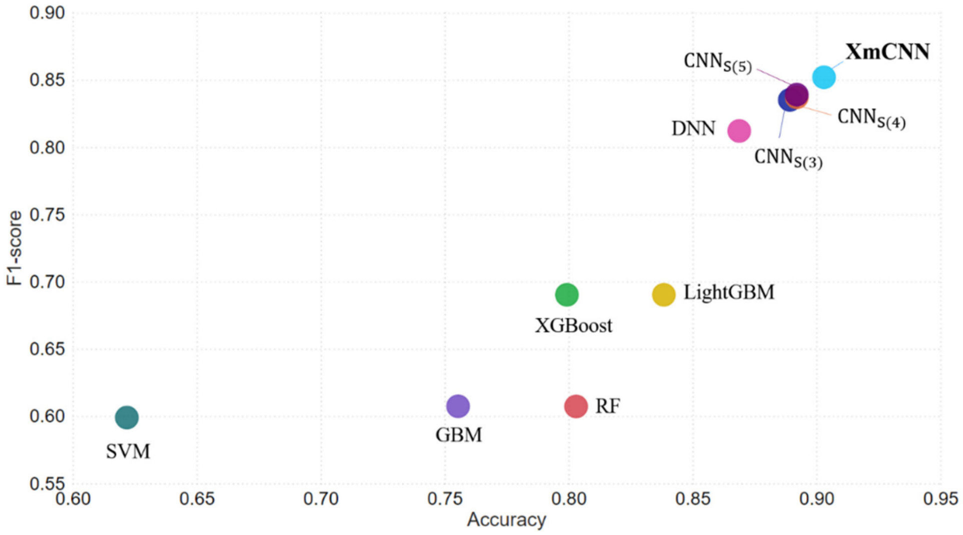

Figure 5. Performance comparison of the overall models.

4.3. Investigation Results of Ensemble Models

As mentioned in Section 3.4, the machine learning ensemble model created and verified 26 ensemble models for all combinations of the five ML models. In addition, the DL ensemble model was composed and 26 ensemble models with five DL models were evaluated. In Table 9, machine learning ensemble models are the top five models based on F1-score among all machine learning ensemble models. We denote the ensemble model as the Ensemble (element: models used in combination) model. Similarly, the DL ensemble shows the top five models based on the F1-score. All five machine learning ensemble models outperformed the individual machine learning models. In particular, the Ensemble (RF, SVM, GBM) model increased the F1-score by 15.32% compared to the LightGBM model, which performed the best among individual ML models. Meanwhile, the results of the Ensemble ($\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(4)}}, \mathrm{CNN_{S(5)}}$, XmCNN) model, which performed the best among the DL ensemble models, showed increased performance in all aspects compared to the machine learning ensemble models. The F1-score was improved by 9.58% compared to the Ensemble (RF, SVM, GBM) model.

Table 9. Performance evaluation of ensemble models.

Ensemble Model	FPR	FNR	Recall	Precision	Accuracy	F1-Score
RF, SVM, GBM, XGBoost, LightGBM	0.0831	0.3631	0.7769	0.7810	0.8624	0.7789
SVM, GBM, XGBoost, LightGBM	0.0739	0.3714	0.7773	0.7921	0.8682	0.7843
SVM, GBM, LightGBM	0.0584	0.3856	0.7780	0.8138	0.8779	0.7938
SVM, GBM, XGBoost	0.1073	0.2622	0.8152	0.7790	0.8625	0.7945
RF, SVM, GBM	0.0840	0.3168	0.7996	0.7928	0.8707	0.7961
$\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(4)}}, \mathrm{CNN_{S(5)}}$	0.0537	0.1992	0.8735	0.8671	0.9179	0.8703
$\mathrm{CNN_{S(5)}}$, XmCNN	0.0519	0.2031	0.8725	0.8693	0.9187	0.8709
$\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(5)}}$, XmCNN	0.0527	0.1992	0.8741	0.8689	0.9188	0.8714
DNN, $\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(4)}}, \mathrm{CNN_{S(5)}}$, XmCNN	0.0525	0.1999	0.8738	0.8690	0.9188	0.8714
$\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(4)}}, \mathrm{CNN_{S(5)}}$, XmCNN	0.0517	0.1992	0.8745	0.8704	0.9196	0.8724

In addition, the Ensemble ($\mathrm{CNN_{S(3)}}, \mathrm{CNN_{S(4)}}, \mathrm{CNN_{S(5)}}$, XmCNN) model outperformed our proposed XmCNN model. Compared to the precision value of the XmCNN model, the increase was 4.04%, indicating that the ratio of actual loan executors among the targets predicted by loan execution increases. The experiment results confirmed that the ensemble model was robust to data with class imbalance problems, such as our outbound telemarketing dataset. Additionally, we verified that the performance was significantly better than that of the machine learning ensemble model, even if all of the variables were used without feature selection.

4.4. Feature Importance

We conducted additional experiments to identify and compare important variables of the ML and DL models. The relative importance of 210 independent variables for the four ML models (RF, GBM, XGBoost, LightGBM) was calculated using the permutation importance module of the eli5 package. Variables with positive feature importance values were considered important variables because they significantly influence the predictive model. Figure 6 shows the top 10 important variables based on the average of the feature importance of the four ML models. The most important variable in the ML model was "Days of application for loan execution in the last year". Meanwhile, as mentioned in Section 3.2, the feature importance of the DL model was calculated by adding a CancelOut layer after the input layer of the XmCNN model. To calculate the final feature importance of the DL model, the XmCNN model to which the CancelOut layer was added was independently trained 10 times, and the weight values of the CancelOut layer were averaged. Figure 7 shows the top 10 important variables based on the feature importance of the DL model. In contrast ML models, the most important variable in the DL model was the "Percentage of one-time loan execution in the last year". The important variable sets of the ML model and DL model were different.

As shown in Figure 8, the intersection of ML and DL important features increased with increasing top N percent of feature importance. In the top 20% of the feature importance criteria, 9 out of 42 variables (see Table 10) were considered important simultaneously in ML and DL, which corresponded to 4.28% of the total features. The intersection of the important features of ML and DL increased rapidly, but the intersection of the upper intervals according to the important features was relatively small. In other words, when ML and DL models were trained, the features they focused on were different.

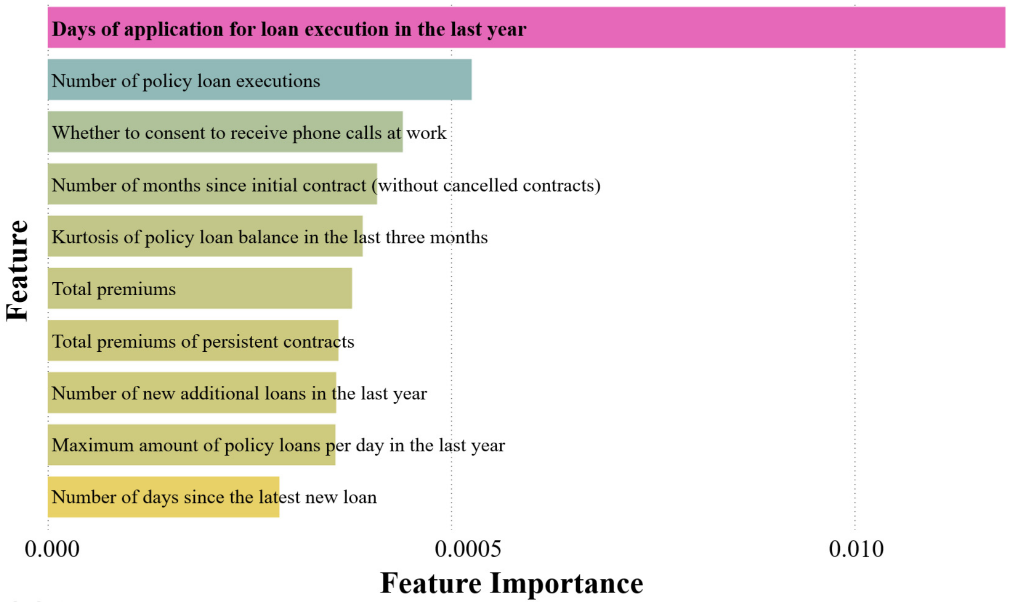

Figure 6. Top 10 important features in machine learning models.

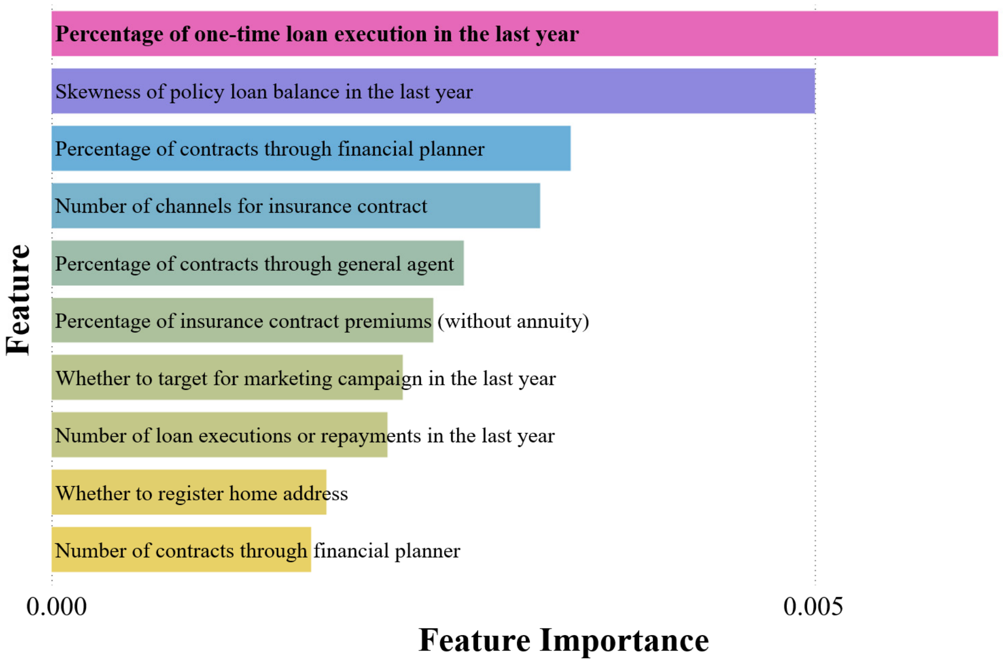

Figure 7. Top 10 important features in deep learning model.

Finally, as shown in Table 11, we investigated the performance of ML and DL models according to the feature selection used. The features corresponding to the top N percent based on feature importance of ML and DL were selected and used for modeling. We experimented with XGBoost, LightGBM, and the proposed XmCNN model. The XGBoost model performed the best when using only the top 60% of features based on feature importance, while the LightGBM model and XmCNN model performed the best when using all features. In particular, the proposed XmCNN model outperformed ML models trained with all features, even if only the top 20% based on feature importance were used.

The XmCNN model was able to achieve high performance with few features as well as reduce the overfitting problem and maximize performance even when many features were used without feature selection.

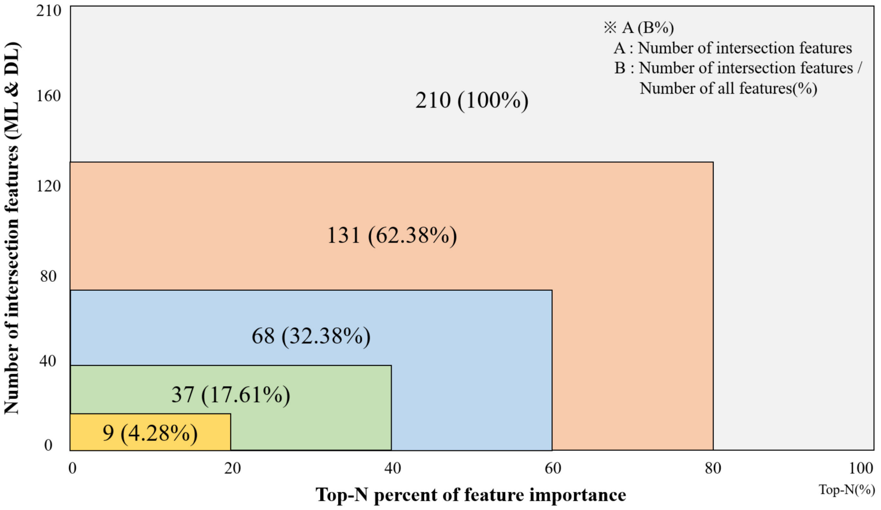

Figure 8. The number and percentage of intersections important features of ML and DL models according to the top N percent of feature importance.

Table 10. List of important variables in both ML and DL based on top 20% of feature importance.

Number	Feature
1	Percentage of one-time loan execution in the last year
2	Percentage of contracts through financial planner
3	Number of channels for insurance contract
4	Percentage of insurance contract premiums (without annuity)
5	Maximum duration of policy loan in the last three years
6	Minimum rate of policy loans
7	Total premium
8	Maximum amount of policy loans per day in the last year
9	Number of call center uses in the last year

Table 11. Model performance according to feature selection.

Feature Importance Top-N (%)	Model					
	XGBoost		LightGBM		XmCNN	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
20	0.6741	0.6055	0.8059	0.5756	0.8702	0.8044
40	0.7953	0.6746	0.8290	0.6702	0.8890	0.8338
60	0.8074	0.6934	0.8176	0.6624	0.8892	0.8344
80	0.7690	0.6653	0.8306	0.6896	0.8922	0.8395
100	0.7993	0.6903	0.8384	0.6903	0.9018	0.8502

5. Discussion

In this study, we investigated deep learning-based models to predict the success of outbound telemarketing for insurance policy loans, and we proposed an explainable multiple-filter CNN model named XmCNN. For the analysis, we extracted and refined the data of 171,424 customers from the outbound telemarketing raw data of a Korean life insurance company. After data preprocessing, an analysis dataset containing 44,412 observations was obtained. We compared the performance of the proposed model with traditional ML models and basic deep learning models, which were mainly used in previous studies.

In addition, we constructed an ensemble model composed of a CNN model and a basic DNN model to improve the model performance. Finally, we identified and compared the important variables of the ML model and the DL model.

Figure 9 shows the F1-score of the proposed model and the comparative models. F1-score can accurately evaluate model performance for imbalanced data and is calculated as the harmonic average of recall and precision. The F1-score of the proposed XmCNN model significantly outperformed the F1-score of the DNN model and the comparative ML models. We additionally confirmed that the ensemble approach, which combines DL models, was effective in maximizing the performance of DL models.

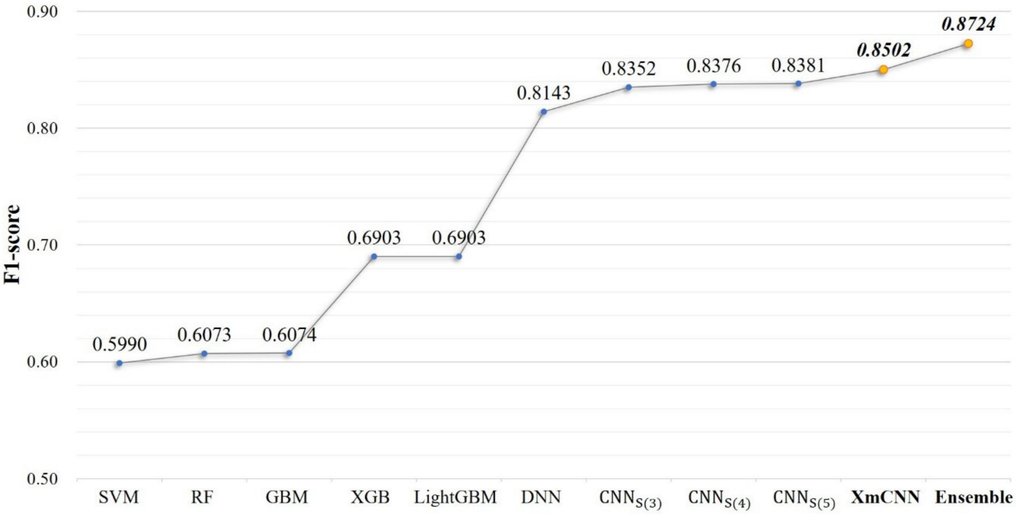

Figure 9. F1-score comparison of the overall models. Ensemble indicates Ensemble ($\mathrm{CNN}_{\mathrm{S}(3)}$, $\mathrm{CNN}_{\mathrm{S}(4)}$, $\mathrm{CNN}_{\mathrm{S}(5)}$, XmCNN) model.

As shown in Figure 10, the Ensemble ($\mathrm{CNN}_{\mathrm{S}(3)}$, $\mathrm{CNN}_{\mathrm{S}(4)}$, $\mathrm{CNN}_{\mathrm{S}(5)}$, XmCNN) model presented the lowest FPR among all the models. In this study, the FPR represents the probability of incorrectly calling customers who do not require information regarding insurance policy loans. Outbound telemarketing is an effective promotion for potential customers but can also be considered advertising spam for uninterested customers. If a company continues to send outbound telemarketing to the incorrectly selected target audience, it can cause customer dissatisfaction and even damage the corporate image and associated brand perception. Therefore, successful targeting for outbound telemarketing is very important. Our proposed XmCNN model and the ensemble model not only contribute to improving the accuracy of outbound telemarketing predictions but also imply that the spam problem can be reduced by minimizing the FPR.

The experimental results on the feature importance were different in the ML model and DL model. In the ML model, variables related to the past transaction amount or period appeared to be important. Because the reuse rate of insurance policy loans is high, past transaction patterns seem to be important. This result can be predicted to some extent through domain knowledge. On the other hand, the DL model yielded unexpected results from domain knowledge. The variables related to the channels for insurance contracts appeared to be important. In particular, variables related to the contract by financial planner or general agent were important. The financial planner is affiliated with only one insurance company and can advise only that company's products to customers. On the other hand, a general agent can partner with several insurance companies to advise products from various companies to customers. Compared to other channels, financial planners or general agents seems to perform well at guiding insurance policy loans when making insurance contracts.

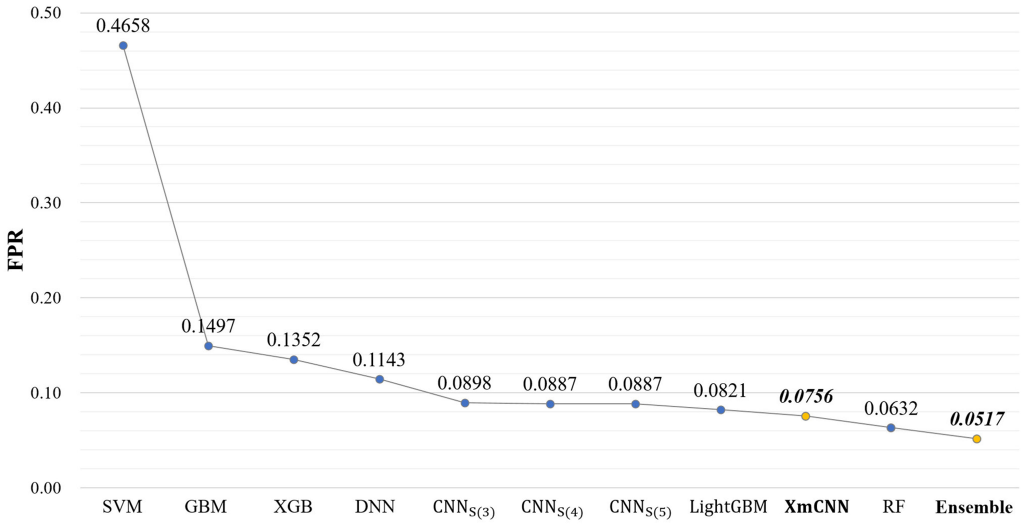

Figure 10. FPR comparison of the overall models. Ensemble indicates Ensemble (CNN$_{S(3)}$, CNN$_{S(4)}$, CNN$_{S(5)}$, XmCNN) model.

6. Implications

Most of the existing research on outbound telemarketing used Portuguese bank telemarketing data studied by Moro et al. [1], making it difficult for insurance companies with different ecosystems to utilize existing studies. On the other hand, our study can be applied to the actual insurance policy loan outbound telemarketing business because we collected and investigated actual business data conducted by insurance companies to customers. In particular, 153 of the actual transaction data of insurance customers, such as insurance transaction information and loan transaction information, were analyzed using deep learning models; through this analysis, previously unknown variables affecting outbound telemarketing success prediction were newly discovered and visualized.

6.1. Practical Implications

First, the proposed model can increase time and cost efficiency by prioritizing calls from outbound telemarketing target customers for insurance policy loans. Because the number of customers that one telemarketer can call per day is limited, it is very important to achieve maximum efficiency within a given range. Therefore, if telephone numbers are dialed in the order of the highest predicted probability of success using the proposed model, the time and cost constraints of telemarketers can be mitigated.

Second, the proposed method is expected to be of benefit to companies in improving marketing sales and increasing customers by broadening the scope of telemarketing target selection. In the current practice of outbound telemarketing, the selection of marketing targets relies on data regarding whether customers have used insurance policy loans in the past and the subjective judgment of telemarketers. In particular, due to the high reuse rate of insurance policy loans, the company from which the dataset was obtained is mainly marketing to customers who have used insurance policy loans in the past. As a result, outbound telemarketing performance is maintained steadily, but total outbound telemarketing sales do not increase. The model proposed in this work was demonstrated to be effective in expanding customers and improving telemarketing success rates; customers who did not use insurance policy loans in the past can also be included in the target set, because the model judges success predictions based on various variables.

Finally, the proposal model can alleviate the problem of customer experience degradation due to incorrect targeting in terms of marketing ethics. In the case of outbound marketing for insurance policy loans, target selection can be very sensitive, as there are many customers who have never used insurance policy loans or are not familiar with them. In particular, incorrect target customer selection can be tantamount to spam that adversely affects society, and customers who experience it may develop an antipathy towards the company. Therefore, it is very important, in practice, to classify customers who are predicted to need insurance policy loans and those who do not, even for customers who have not used insurance policy loans in the past. The FPR used as a model performance indicator in this study is the rate that customers would recognize the calls as spam; the FPR of the proposed model was 0.07, indicating very good performance compared to the comparative ML models. Accordingly, we believe that the proposed model would contribute not only to improving the efficiency of outbound telemarketing for insurance policy loans but also to address the ethical issues involved in outbound telemarketing.

6.2. Academic Implications

First, we proposed an explainable deep learning model based on CNN. We validated that the proposed XmCNN model performed well for predicting the success of outbound telemarketing with insurance policy loan data. The deep learning models exhibited superior performance compared to the comparative ML models. In particular, the ensemble model built with the proposed model showed the lowest FPR and the highest F1-score. Most of the marketing response predictions used in the field are conservatively using traditional ML models; however, to improve the prediction accuracy, it is necessary to use deep learning-based models actively.

Second, we used high-dimensional insurance policy loan data consisting of more than 200 input dimensions without feature selection. We confirmed that using various transaction data related to insurance customers, such as customer characteristics, insurance transactions, and insurance policy loan transactions, contributes to the prediction of the success of insurance policy loans outbound telemarketing. However, the business tabular data analyzed in this study are different from the unstructured data, such as image, video, and audio data; therefore, to extract various features using a deep learning-based model, it is necessary to discover and add more related variables.

Third, our study has implications as an early work in the field of outbound telemarketing of insurance policy loans. In constructing the proposed XmCNN model, the results of various experiments, such as the configuration of the architecture and the selection of hyperparameters, provide useful information for future research. In particular, by presenting 10 important variables affecting ML and DL models designed for insurance policy loan prediction, variables to be considered in practice have been established.

7. Conclusions

Outbound telemarketing is often criticized as an unethical marketing method owing to the perception of high-pressure sales during unsolicited calls. It could additionally be considered an annoyance, especially during specific times in the day. Hence, predictive models for outbound telemarketing might be considered relatively important in reducing customer complaints and social problems. However, most of the existing studies related to prediction models for outbound telemarketing have focused on improving the predictive accuracy of marketing success. Therefore, models should be developed for improving the accuracy of marketing success prediction and for reducing the FPR. Through this study, we have proposed a model with the lowest FPR (4.92%) and the highest F1-score (87.47%), compared with prior works, and revealed important variables affecting the predictive power of a model considering its practical use. However, despite the importance of this study and the academic and practical implications described in Section 6, some limitations may be noted. First, as in this study, it is difficult to obtain insurance policy loan-related data that includes a large number of variables. In particular, it is not easy for an individual to obtain such data independently, as customer-related information is sensitive and generally restricted to personnel with authorized access. Second, it might be difficult to achieve the level of accuracy demonstrated in this study if the data and the number or types of variables differ from those used herein. Because there is no standardized collection format with respect to data on insurance policy loans, to use the framework presented in this study, it is necessary to retrain the model according to the data and optimize the architecture and hyperparameters. Nevertheless, the present work is meaningful in that it uncovered important variables in outbound telemarketing of insurance policy loans that had not been revealed thus so far and proposed the first framework for this. Some possible directions for future research could include the diversity of CNN filters and feature maps to further improve the prediction performance or the application of another deep learning technique, such as an attention mechanism or TabNet [35].

Author Contributions: Conceptualization, H.K., J.G. and J.N.; Data curation, J.N.; Formal analysis, H.K., J.G. and J.P.; Funding acquisition, H.K.; Investigation, J.G., J.N. and J.P.; methodology, H.K. and J.G.; Project administration, H.K.; Resources, H.K., J.G., J.N. and J.P.; Software, J.G.; Supervision, H.K.; Validation, J.G., J.N. and J.P.; Visualization, J.G and J.P. writing—original draft preparation, J.G. and J.N.; writing—review and editing, J.G. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean Government (MOTIE) (20202020800030, Development of Smart Hybrid Envelope Systems for Zero Energy Buildings through Holistic Performance Test and Evaluation Methods and Fields Verifications).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to express our appreciation to KYOBO Life Insurance Company, who provided us with the insurance policy loan outbound telemarketing dataset.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Description
ANN	Artificial Neural Network
CCP	Customer Churn Prediction
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network with fully connected layers.
DT	Decision Tree
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GBM	Gradient Boosting Machine
LightGBM	Light Gradient Boosting Machine
LR	Logistic Regression
ML	Machine Learning
MLP	Multi-layer Perceptron
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
XGBoost	eXtreme gradient boosting
XmCNN	eXplainable Multiple-filter Convolutional Neural Network

Appendix A

Table A1. The data distributions for continuous variables.

Num	Variable	Mean	Std. Dev	Skewness	Kurtosis
1	Number of months since security card issuance	0.3006	0.2879	0.53	−0.97
2	Number of lapsed contracts	0.0054	0.0361	10.65	161.20
3	Number of persistent contracts	0.0675	0.0618	2.88	14.33
4	Numbers of cancelled contracts	0.0171	0.0407	5.86	61.69
5	Number of total contracts (without cancelled contracts)	0.0404	0.0587	3.44	21.47
6	Number of total contracts (including cancelled contracts)	0.0434	0.0626	2.77	12.91
7	Number of persistent contracts applied in the last year	0.0094	0.0348	6.69	80.34
8	Number of months since initial contract (including cancelled contracts)	0.3475	0.1513	0.10	−0.66
9	Number of months since initial contract (without cancelled contracts)	0.3257	0.1521	0.11	−0.64
10	Number of lapse in the last year	0.0005	0.0148	37.34	1809.45
11	Number of reinstatement in the last year	0.0008	0.0200	31.49	1195.51
12	Percentage of insurance contracts (without annuity)	0.7621	0.3573	−1.27	0.12
13	Percentage of annuity contracts	0.2372	0.3568	1.27	0.13
14	Number of annuity contracts	0.0229	0.0366	3.36	32.74
15	Number of insurance contracts (without annuity)	0.0597	0.0618	2.56	11.38
16	Fractional premiums of contracts applied in the last year (monthly payment)	0.0013	0.0168	32.62	1362.61
17	Fractional premiums of contracts applied in the last year (total payment)	0.0010	0.0146	37.06	1796.87
18	Lump sum premium of contracts applied in the last year	0.0002	0.0117	68.02	5020.47
19	Total premiums of contracts applied in the last year	0.0011	0.0147	36.28	1738.72
20	Fractional premiums of persistent contracts (monthly payment)	0.0047	0.0196	19.74	589.15
21	Fractional premiums of persistent contracts (total payment)	0.0097	0.0237	11.65	261.33
22	Lump sum premium of persistent contracts	0.0009	0.0132	37.00	2034.87
23	Total premiums of persistent contracts	0.0098	0.0239	11.49	254.66
24	Number of overdue premiums in the last year	0.0259	0.0483	4.22	34.69
25	Total premiums	0.0098	0.0239	11.49	254.66
26	Premiums of annuity contracts	0.0026	0.0159	28.51	1143.59
27	Premiums of insurance contracts (without annuity)	0.0155	0.0367	6.35	62.98
28	Percentage of insurance contract premiums (without annuity)	0.7417	0.3828	−1.09	−0.51
29	Percentage of annuity contract premiums	0.2577	0.3824	1.09	−0.51
30	Amount of withdrawals in the last year	0.0002	0.0060	148.51	24,342.78
31	Amount of withdrawals in the last three months	0.0001	0.0066	123.16	17,663.96
32	Number of withdrawals in the last three months	0.0011	0.0193	28.26	1012.06
33	Number of withdrawals in the last year	0.0017	0.0188	22.24	719.69
34	Total amount of withdrawals	0.0002	0.0061	146.03	23,787.52
35	Total number of withdrawals	0.0040	0.0233	17.43	493.54
36	Minimum amount of withdrawals	0.0025	0.0209	26.97	972.92
37	Maximum amount of withdrawals	0.0033	0.0248	23.23	719.68
38	Average amount of withdrawals	0.0024	0.0154	23.72	999.05
39	Total number of the insured	0.0914	0.1306	1.46	1.76
40	Total number of contracts for the insured	0.0676	0.0463	2.93	19.38
41	The difference between the number of contracts and the number of the insured	0.1230	0.0452	3.41	22.70
42	Number of days since the latest new loan	0.1844	0.1708	0.62	−0.54
43	Average duration of policy loans in the last year	0.0236	0.0884	5.31	33.03
44	Minimum duration of policy loans in the last year	0.0176	0.0808	6.38	46.36
45	Maximum duration of policy loans in the last year	0.0326	0.1114	4.34	20.66
46	Number of new loans in the last year	0.0038	0.0144	18.65	867.61
47	Number of new additional loans in the last year	0.0067	0.0225	11.53	273.93
48	Number of new loans in the last three months	0.0025	0.0145	20.93	979.93
49	Number of new additional loans in the last three months	0.0051	0.0222	11.10	240.79
50	Average amount of policy loans per day in the last year	0.0023	0.0099	45.47	3846.50
51	Maximum amount of policy loans per day in the last year	0.0020	0.0097	57.96	5269.69
52	Minimum amount of policy loans per day in the last year	0.0016	0.0130	34.13	1995.08
53	Days of application for loan execution in the last year	0.0110	0.0345	9.10	129.16
54	Number of loan executions or repayments in the last year	0.0035	0.0147	30.21	1651.34

Table A1. *Cont.*

Num	Variable	Mean	Std. Dev	Skewness	Kurtosis
55	Number of loan executions or repayments in the last three months	0.0035	0.0162	26.65	1310.40
56	Number of overdues for credit or mortgage loans in the last year	0.0018	0.0258	22.76	638.32
57	Number of overdues for credit or mortgage loans in the last three months	0.0019	0.0333	21.18	508.23
58	Number of overdues for policy loans in the last year	0.0028	0.0205	17.69	506.66
59	Number of overdues for policy loans in the last three months	0.0018	0.0158	20.52	765.31
60	Average amount of repayments per day in the last year	0.0023	0.0099	43.40	3790.84
61	Maximum amount of repayments per day in the last year	0.0041	0.0154	22.17	1025.47
62	Minimum amount of repayments per day in the last year	0.0025	0.0197	16.02	412.54
63	Days of application for loan execution or repayment in the last year	0.0052	0.0186	20.63	858.26
64	Number of loan executions in the call center in the last year	0.0039	0.0215	15.33	395.71
65	Number of loan executions in the customer center in the last year	0.0003	0.0065	120.69	18,230.18
66	Number of loan executions through ARS in the last year	0.0010	0.0146	34.22	1688.28
67	Number of loan executions through ATM in the last year	0.0004	0.0094	55.44	4633.79
68	Number of loan executions through mobile in the last year	0.0041	0.0189	16.26	532.14
69	Number of loan executions through PC in the last year	0.0032	0.0236	17.84	452.49
70	Balance of policy loans	0.0053	0.0151	19.24	929.07
71	Variance of policy loan balance in the last year	0.0005	0.0077	88.57	10,180.49
72	Mean of policy loan balance in the last year	0.0113	0.0256	7.29	132.88
73	Skewness of policy loan balance in the last year	0.2377	0.0522	3.93	24.15
74	Kurtosis of policy loan balance in the last year	0.0140	0.0284	11.52	205.91
75	Maximum of policy loan balance in the last year	0.0084	0.0184	12.99	499.93
76	Minimum of policy loan balance in the last year	0.0037	0.0177	15.61	540.52
77	Variance of policy loan balance in the last three months	0.0006	0.0094	80.79	8158.41
78	Mean of policy loan balance in the last three months	0.0108	0.0266	7.68	132.74
79	Skewness of policy loan balance in the last three months	0.2111	0.0447	5.08	36.07
80	Kurtosis of policy loan balance in the last three months	0.0273	0.0243	12.64	279.84
81	Maximum of policy loan balance in the last three months	0.0077	0.0183	11.03	376.43
82	Minimum of policy loan balance in the last three months	0.0047	0.0209	12.71	330.79
83	Sum of credit or mortgage loan balance	0.0055	0.0305	14.42	277.35
84	Number of loan executions for 2 times in the current month	0.0009	0.0174	29.26	1087.64
85	Number of loan executions for 3 times in the current month	0.0058	0.0345	11.19	181.01
86	Number of loan executions for 1 time in the last year	0.0057	0.0229	10.44	221.56
87	Number of loan executions for 2 times in the last year	0.0051	0.0209	12.43	329.46
88	Number of loan executions for 3 times in the last year	0.0069	0.0238	10.42	210.37
89	Number of call center uses in the last year	0.0216	0.0309	4.97	67.76
90	Number of call center uses in the last three months	0.0099	0.0263	6.71	109.37
91	Number of mobile uses in the last three months	0.0156	0.0395	7.55	94.47
92	Number of website uses in the last three months	0.0048	0.0253	14.25	319.68
93	Number of channels for insurance contract	0.2798	0.0986	1.91	5.92
94	Number of contracts through financial planner	0.0453	0.0585	3.40	21.12
95	Number of contracts through general agent	0.0052	0.0200	11.34	299.56
96	Number of contracts through bank (bancassurance)	0.0013	0.0108	33.33	2540.77
97	Number of contracts through direct marketing	0.0093	0.0327	6.29	71.21
98	Number of contracts through other channels	0.0043	0.0290	12.13	219.39
99	Percentage of contracts through financial planner	0.7483	0.4058	−1.15	−0.51
100	Percentage of contracts through general agent	0.0977	0.2784	2.73	5.78
101	Percentage of contracts through bank (bancassurance)	0.0097	0.0665	7.77	66.60
102	Percentage of contracts through direct marketing	0.1006	0.2823	2.65	5.33
103	Percentage of contracts through other channels	0.0239	0.1370	6.21	38.79
104	Percentage of contracts through face−to−face	0.8796	0.3066	−2.33	3.67
105	Average duration of policy loans in the last three years	0.0280	0.0903	5.02	30.08
106	Minimum duration of policy loans in the last three years	0.0198	0.0821	6.14	43.37
107	Maximum duration of policy loans in the last three years	0.0400	0.1147	3.89	16.92
108	Average amoun of repayments with other services per day in the last year	0.0013	0.0158	30.74	1303.93
109	Maximum amount of repayments with other services per day in the last year	0.0019	0.0182	23.16	797.77

Table A1. *Cont.*

Num	Variable	Mean	Std. Dev	Skewness	Kurtosis
110	Minimum amount of repayments with other services per day in the last year	0.0007	0.0148	41.28	2146.83
111	Number of applications for repayment with other services per day in the last year	0.0047	0.0259	12.00	283.97
112	Average rate of policy loans	0.5867	0.1574	0.86	0.18
113	Maximum rate of policy loans	0.6526	0.1854	0.46	−0.78
114	Minimum rate of policy loans	0.5295	0.1681	1.14	0.62
115	Loan limit exhaustion rate	0.2471	0.3669	1.04	−0.66
116	Number of loan completion in the last year	0.0038	0.0145	18.51	855.27
117	Number of policy loan executions	0.0041	0.0125	22.04	1370.04
118	Average days since new additional loans in the last year	0.0139	0.0625	6.85	57.88
119	Percentage of consecutive loan executions in the last year	0.0634	0.2084	3.42	10.77
120	Percentage of consecutive loan executions and repayments in the last year	0.0618	0.2067	3.56	11.83
121	Percentage of recurring loan repayments in the last year	0.0727	0.2194	3.29	9.98
122	Percentage of one−time loan execution in the last year	0.0653	0.2363	3.57	11.07
123	Percentage of one−time loan repayment in the last year	0.0445	0.1990	4.47	18.31
124	Percentage of one−time loan repayment with other services in the last year	0.0083	0.0852	11.02	122.82
125	Number of marketing campaigns in the current month	0.0007	0.0159	29.07	1053.85
126	Number of marketing campaigns in the last three months	0.0021	0.0261	16.03	326.40
127	Number of marketing campaigns in the last year	0.0059	0.0408	10.11	138.04
128	Age	46.1099	8.6803	0.00	−0.37

References

1. Moro, S.; Cortez, P.; Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.* **2014**, *62*, 22–31. [CrossRef]
2. Kim, K.H.; Lee, C.S.; Jo, S.M.; Cho, S.B. Predicting the success of bank telemarketing using deep convolutional neural network. In Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Fukuoka, Japan, 13–15 November 2015; pp. 314–317.
3. Frempong, A.J.; Jayabalan, M. Predicting customer response to bank direct telemarketing campaign. In Proceedings of the 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), Kuala Lumpur, Malaysia, 18–20 September 2017; pp. 1–4.
4. Koumétio, C.S.T.; Cherif, W.; Hassan, S. Optimizing the prediction of telemarketing target calls by a classification technique. In Proceedings of the 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), Marrakesh, Morocco, 16–19 October 2018; pp. 1–6.
5. Ghatasheh, N.; Faris, H.; AlTaharwa, I.; Harb, Y.; Harb, A. Business Analytics in Telemarketing: Cost-Sensitive Analysis of Bank Campaigns Using Artificial Neural Networks. *Appl. Sci.* **2020**, *10*, 2581. [CrossRef]
6. Turkmen, E. Deep Learning Based Methods for Processing Data in Telemarketing-Success Prediction. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 1161–1166.
7. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
8. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
9. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
10. LEE, H. A Method of Bank Telemarketing Customer Prediction based on Hybrid Sampling and Stacked Deep Networks. *J. Korea Soc. Digit. Ind. Inf. Manag.* **2019**, *15*, 197–206.
11. Hosein, P.; Ramoudith, S.; Rahaman, I. On the Optimal Allocation of Resources for a Marketing Campaign. In Proceedings of the 10th International Conference on Operations Research and Enterprise Systems (ICORES 2021), Vienna, Austria, 4–6 February 2021; pp. 169–176.
12. Hosseini, S. A decision support system based on machined learned Bayesian network for predicting successful direct sales marketing. *J. Manag. Anal.* **2021**, *8*, 295–315.
13. Krishna, C.L.; Reddy, P.V.S. Deep Neural Networks for the Classification of Bank Marketing Data using Data Reduction Techniques. *Int. J. Recent Technol. Eng. (IJRTE)* **2019**, *8*, 3.
14. Moro, S.; Cortez, P.; Rita, P. A divide-and-conquer strategy using feature relevance and expert knowledge for enhancing a data mining approach to bank telemarketing. *Expert Syst.* **2018**, *35*, e12253. [CrossRef]

15. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
16. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
17. Arenas-Márquez, F.J.; Martinez-Torres, R.; Toral, S. Convolutional neural encoding of online reviews for the identification of travel group type topics on TripAdvisor. *Inf. Process. Manag.* **2021**, *58*, 102645. [CrossRef]
18. Behera, R.K.; Jena, M.; Rath, S.K.; Misra, S. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Inf. Process. Manag.* **2021**, *58*, 102435. [CrossRef]
19. Fu, X.; Ouyang, T.; Chen, J.; Luo, X. Listening to the investors: A novel framework for online lending default prediction using deep learning neural networks. *Inf. Process. Manag.* **2020**, *57*, 102236. [CrossRef]
20. Goldani, M.H.; Safabakhsh, R.; Momtazi, S. Convolutional neural network with margin loss for fake news detection. *Inf. Process. Manag.* **2021**, *58*, 102418. [CrossRef]
21. Song, C.; Ning, N.; Zhang, Y.; Wu, B. A multimodal fake news detection model based on crossmodal attention residual and multichannel convolutional neural networks. *Inf. Process. Manag.* **2021**, *58*, 102437. [CrossRef]
22. Neagoe, V.E.; Ciotec, A.D.; Cucu, G.S. Deep convolutional neural networks versus multilayer perceptron for financial prediction. In Proceedings of the 2018 International Conference on Communications (COMM), Bucharest, Romania, 14–16 June 2018; pp. 201–206.
23. Zhang, X.; Wu, F.; Li, Z. Application of convolutional neural network to traditional data. *Expert Syst. Appl.* **2021**, *168*, 114185. [CrossRef]
24. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2016; pp. 785–794.
25. Kvamme, H.; Sellereite, N.; Aas, K.; Sjursen, S. Predicting mortgage default using convolutional neural networks. *Expert Syst. Appl.* **2018**, *102*, 207–217. [CrossRef]
26. De Caigny, A.; Coussement, K.; De Bock, K.W.; Lessmann, S. Incorporating textual information in customer churn prediction models based on a convolutional neural network. *Int. J. Forecast.* **2020**, *36*, 1563–1578. [CrossRef]
27. Weng, C.H.; Huang, T.C.K.; Han, R.P. Disease prediction with different types of neural network classifiers. *Telemat. Inform.* **2016**, *33*, 277–292. [CrossRef]
28. Borisov, V.; Haug, J.; Kasneci, G. Cancelout: A layer for feature selection in deep neural networks. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 72–83.
29. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
30. Yin, W.; Schütze, H. Multichannel Variable-size Convolution for Sentence Classification. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China, 30–31 July 2015; pp. 204–214.
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
33. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3146–3154.
34. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.
35. Arik, S.O.; Pfister, T. Tabnet: Attentive interpretable tabular learning. *arXiv* **2019**, arXiv:1908.07442.