Application of Convolutional Neuronal Networks in Customer Base Analysis

Shahrzad Kurbiel
University Duisburg-Essen

Cite as:
Kurbiel Shahrzad (2024), Application of Convolutional Neuronal Networks in Customer Base Analysis. *Proceedings of the European Marketing Academy*, 52nd, (119378)

Paper from the 53rd Annual EMAC Conference, Bucharest, Romania, May 28-31, 2024

Application of Convolutional Neuronal Networks in Customer Base Analysis

Abstract:

Customer lifetime value is a useful metric in customer relationship marketing, offering insights into distinct customer segments. However, predicting future purchase behavior poses a significant challenge in calculating customer lifetime value, particularly in noncontractual business relationships. In recent decades, various approaches have emerged to predict future customer activities. In the literature of customer lifetime value, probabilistic models represent the most widely used approach, which describe a purchase process based on predefined assumptions. While these assumptions may reflect the inherent features of a purchase process, their applicability cannot be generalized for all purchase processes. To address this, the current study employs the deep learning approach. At its core, a convolutional neuronal network is designed and evaluated on two real-world data sets. Depending on the data, the evaluation of the model shows a better forecast compared to the benchmark models.

Keywords: Customer lifetime value, Convolutional neuronal networks, Time series

Track: Methods, Modelling & Marketing Analytics

1. Introduction

Customer lifetime value (CLV) is a powerful metric used in customer relationship marketing to assess the net financial value of customers and to identify the most profitable customers in a company. For calculating the CLV, various approaches have been proposed in recent decades to explore customers' historical purchasing behavior and to predict their future buying activities in a noncontractual setting (Gupta et al., 2006). The most widely used approach in the literature of CLV are probability models. Probability models view a purchase behavior as realizations of a stochastic process. Some of the most prominent representatives of probability models are Pareto/NBD (Schmittlein, Morrison and Colombo, 1987), BG/NBD (Fader, Hardie and Lee, 2005), and Pareto/GGG (Platzer & Reutterer, 2016).

The major drawback of the probability models lies in the fact that they analyze a purchase process according to predefined assumptions, such as those pertaining to the distribution of interpurchase time. These models often overlook the dynamic evolution of purchase processes over time. While these assumptions can realistically mirror the underlying purchase process in a data set, their performance cannot be generalized for all purchase processes. To overcome this challenge, the deep learning approach has been applied in recent years in order to analyze underlying patterns in purchase behavior of customers (Bauer & Jannach, 2021; Chen, Guitart, Del Rio and Perianez, 2018; Salehinejad & Rahnamayan, 2016; Valendin, Reutterer, Platzer and Kalcher, 2022). Deep learning algorithms offer a flexible framework, which adapts to the conditions of data, instead of fitting the data to the model assumptions. Among all deep learning algorithms in the literature of CLV, the recurrent neural network (RNN) is widely used (Bauer & Jannach, 2021; Salehinejad & Rahnamayan, 2016; Valendin et al, 2022). RNNs are primarily specialized for sequential data, featuring an architecture optimized to effectively model long-term temporal patterns (Rezk, Purnaprajna, Nordstrom and Ul-Abdin, 2020). In a recent study, Valendin et al. (2022) have developed an training model, which represents a special from of RNN named long short-term memory (LSTM). This model learns autoregressively from the input data to predict the individual number of transactions in the next period. A limitation of RNNs lies in their high memory consumption, particularly when dealing with extended input sequences (Rezk et al, 2020).

The current study applies the technique of temporal convolutional neuronal network (CNN) to learn from customers' past purchase behavior and to forecast their future purchase activity. CNNs are a well-established deep learning approach and can, due to their architecture, efficiently learn both spatial and temporal information (Sezer, Gudelek and Ozbayoglu, 2020).

In the realm of time series, CNNs are well-suited for capturing short-term patterns and dependencies, in contrast to RNNs (Lai, Chang, Yang and Liu, 2018). Their inherent design for translation invariant enables them to recognize patterns regardless of their position in the input. This can be beneficial when the specific timing of a pattern is not crucial in time series (Biscione & Bowers, 2020). Furthermore, the problems of vanishing or exploding gradients are highly reduced in CNNs due to the property of local connectivity and weight sharing (Bai, Kolter and Koltun, 2018; Martens & Sutskever, 2011; O'Shea & Nash, 2015p. 8). Despite these advantages, CNNs has rarely been analyzed in the domain of CLV. Chen et al. (2018) are, to our best of knowledge, the only researchers in the field, who have employed this technique to predict the future expenditure of customers. The current study proposes a CNN-based algorithm to learn the features of purchase processes over time and to forecast the future number of purchases. This model is subsequently evaluated on two real-world data sets.

2. Modeling Approach

At the core of CNN architectures lie the convolutional layers, designed primarily to extract key features from the input data. Figure 1 shows an example of a simple temporal convolutional layer with a univariate time sequence as input data. The input sequence is convolved by performing an element-wise scalar product between the kernel and the input data at each position, followed by a summation. The parameters of the kernel are learnable entities. The outcome is passed through an activation function, determining the output of the convolutional layer (Valueva, Nagornov, Lyakhov, Valuev and Chervyakov, 2020). The most commonly used activation function for a CNN layer is the rectified linear unit (Relu), which allows only the positive inputs to get through (Ide & Kurita, 2017). The result generated by a convolutional layer is referred to as a feature map because it captures the characteristics of the input data.

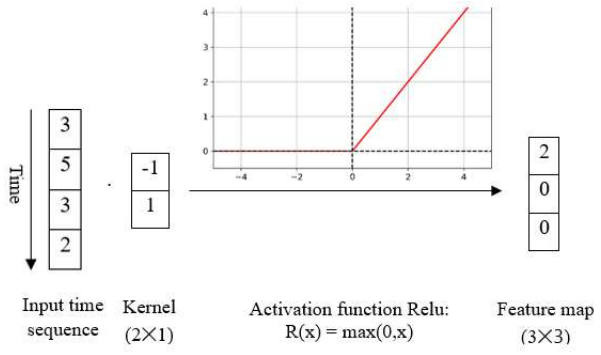

Figure 1. An example of a temporal convolutional layer

Convolutional layers are usually followed by a max-pooling layer, which reduces the size of feature maps by retaining only their local maximum values, thereby decreasing the computational complexity of the model (O'Shea & Nash, 2015p. 8). CNN architectures are constructed by stacking several convolutional layers and pooling layers, to progressively extract more and more complex features of the original input data. The output of the last convolutional layer is flattened and then passed through fully connected (FC) layers. The FC layers transform the features of the input data to a score, which can be interpreted as a probability in classification tasks or as a prediction in sequence modeling (O'Shea & Nash, 2015p. 5). The whole model is trained using a variant of gradient descent, like stochastic gradient descent (SGD). At the end of each iteration, known as epoch, the model parameters are adjusted by minimizing the loss function and backpropagating through all the layers.

Figure 2 gives an overview of the base architecture of the proposed CNN. The hyperparameters of the model are based on the research of Chen et al. (2018) and Livieris, Pintelas and Pintelas (2020). However, the initial hyperparameters in these studies were further adjusted by implementing the random walk algorithm (Matuszyk, Castillo, Kottke and Spiliopoulou, 2016). The batch normalization layer normalizes the output of each layer before proceeding to the next layer. This accelerates the training process and enables a higher learning rate (Santurkar, Tsipras, Ilyas and Madry, 2018). Using a linear activation function, the model forecasts the future number of purchases. The linear activation function is frequently employed when the objective is to predict a continuous output. The CNN model iterates until the loss function for the test set shows no decrease for 20 consecutive epochs.

Convolutional Layers	Parameter
CNN 1	
Number of Kernel	32
Kernel size	7
Activation function	Relu
Padding	Valid
CNN 2	
Number of Kernel	64
Kernel size	4
Activation function	Relu
Padding	Valid
Batch normalization	-
Max pooling	
pool-size	2
Flatten	-

Fully connected Layers	Parameter
FC 1	
Number of nodes	300
Activation function	Relu
FC 2	
Number of nodes	150
Activation function	Relu
FC 3	
Number of nodes	60
Activation function	Relu
Batch normalization	-

Model fitting	
Optimizer	SGD
Loss function	MAE
Activation function	Linear
Epochs	200

Figure 2. Architecture of the proposed CNN model

In the present study, the input data comprises a multivariate time series consisting of the following individual sequences: (Chen et al, 2018; Tran, Nguyen, Van-Ho and Ho, 2021):

- **Recency:** the time of the last purchase in each period
- **Frequency:** the accumulated number of purchases in each period

- **Monetary:** the accumulated amount spent per purchase in each period
- **Transaction rate:** calculated as the reciprocal of the interpurchase time between two consecutive purchases in each period (Allenby, Leone and Jen, 1999)

Table 1 gives an example of the input data of a hypothetical customer over time. In this scenario, the customer makes a purchase during the first, second, fifth, and eighth week, spending €29.33, €13.97, €38.9, and €14.3, respectively. The associated recency values for each purchase opportunity are 0.14, 1.86, 4.86, and 7.16, respectively. The transaction rate from the first to the second transaction is 1, since the interpurchase time between the first and the second transaction is just one week. The interpurchase time from the second to the third transaction, and from the third to the fourth transaction amounts to three weeks each. Hence, the transaction rate in these weeks is equal 0.33.

	Week 1	**Week 2**	**Week 3**	**Week 4**	**Week 5**	**Week 6**	**Week 7**	**Week 8**
Recency	0.14	1.86	1.86	1.86	4.86	4.86	4.86	7.16
Frequency	1	2	2	2	3	3	3	4
Monetary	29.33	43.3	43.3	43.3	82.2	82.2	82.2	96.5
Transaction rate	0	1	0.33	0.33	0.33	0.33	0.33	0.33

Table 1. Input sequence of a hypothetical customer over time

3. Data Sets

The proposed CNN model is applied and evaluated on the following two data sets:

- **CDNOW:** this data set is the most extensively analyzed data set in the literature of CLV (Platzer & Reutterer, 2016) and covers the purchase history of 23,570 customers of an online CD retailer from January 1st, 1997 to June 30th, 1998. The length of the calibration and holdout period is 39 weeks. A random ratio of 90% of this data set is used for the training set and the remaining proportion of 10% for the test set.
- **HandyTicket:** this data set includes the purchase records of 2,454 passengers, who were registered on a German public transport mobile application between January 24th, 2021, and January 22nd, 2023. This application enables users to conveniently buy bus and train tickets whenever necessary. Both the calibration and holdout periods each span 52 weeks. The training set comprises a randomly selected 70% of passengers, while the test set includes the remaining 30%.

Figure 3 shows the weekly aggregate transactions of both data sets. In contrast to the CDNOW data set, the HandyTicket data set exhibits a sudden temporal decline in the number

of transactions in the holdout period.[1] An imbalanced training set could have a detrimental influence on the performance of CNNs, since the models might focus more on the majority class and fail to learn important features related to the minority classes (Buda, Maki and Mazurowski, 2018). Therefore, the unsupervised clustering algorithm of k-means is first applied to gain a deeper understanding of different customer clusters in the training set (Anitha & Patil, 2022). To cluster the customers, the individual indicators of recency, frequency, and monetary value at the end of the calibration period, as well as the mean value of transaction rate over the calibration period are used.

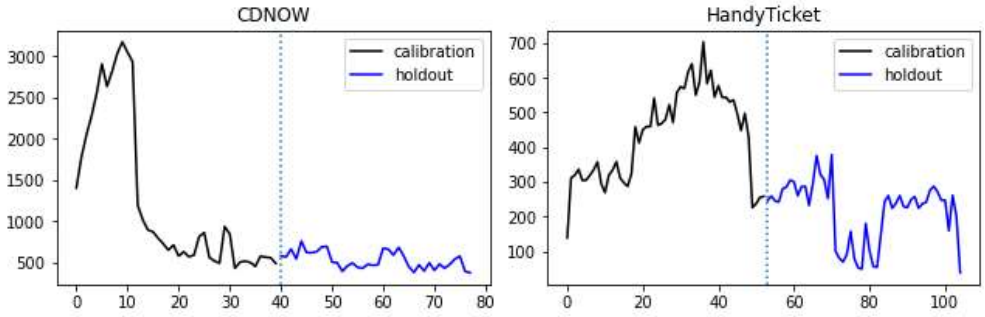

Figure 3. Weekly aggregate transactions

Figure 4 illustrates the optimal number of clusters determined through the elbow criterion. According to this analysis, the CDNOW data set exhibits an optimal number of 5 clusters, while the HandyTicket data set shows an optimal number of 7 clusters. Silhouette score provides a measure to interpret how well the data points fit the cluster that they are assigned to (Anitha & Patil, 2022). A silhouette score above 0.65 denotes a well clustered data set (Lovmar, Ahlford, Jonsson and Syvänen, 2005). The silhouette score shows a value of 0.77 for the CDNOW data set and of 0.75 for the HandyTicket data set.

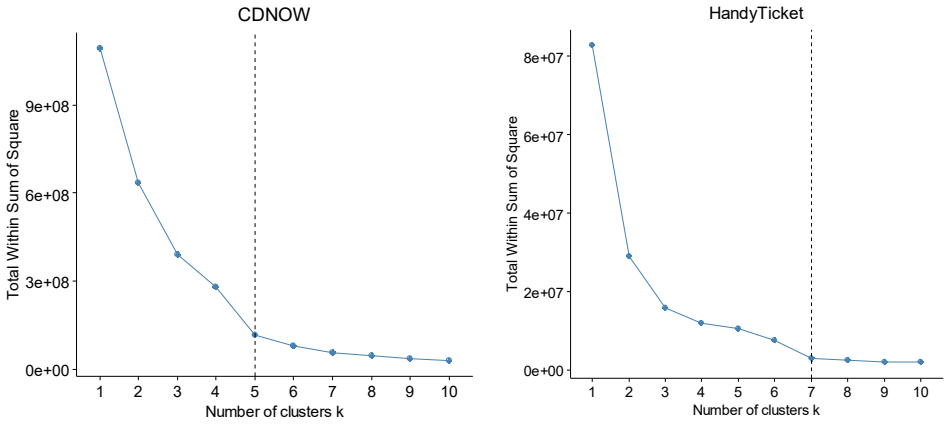

Figure 4. Elbow criteria

[1] It is due to the promotional period of the €9 ticket from May 1st, 2022 to August 31st, 2022. During this period, everyone in Germany could use buses and trains for just €9 per month.

In Table 2, the characteristics of each cluster in both data sets are summarized. As can be seen, both data sets are highly imbalanced. The imbalance ratio (the number of data points in the minority class divided by the number of data points in the majority class) ranges from 0.005% in the CDNOW data set to almost 0.1% in the HandyTicket data set. To tackle the issue of imbalanced data, the current research utilizes the hybrid resampling technique (Wongvorachan, He and Bulut, 2023). In this method, the minority clusters are oversampled to achieve a selected desired ratio relative to the majority cluster, while the majority cluster is randomly undersampled to meet the desired ratio. The selection of the desired ratio depends on the size of the data set and the cluster proportions (Chawla, Bowyer, Hall and Kegelmeyer, 2002). The desired ratio for the CDNOW data has been set at 20%, and for the HandyTicket data set, it is 80% of the majority class (Chawla et al, 2002). Furthermore, the input and output data are standardized to improve and accelerate the training process (Wibawa et al., 2022).

Data set	Cluster	Number of data point	Mean of recency	Mean of frequency	Mean of monetary	Mean of transaction rate
CDNOW	1	18,907	11.22	1.57	43.64	0.047
	2	2,074	26.82	5.33	276.61	0.14
	3	221	31.89	12.18	1,017.31	0.28
	4	10	33.55	43.8	5,024.19	0.61
	5	1	38.71	107	20,895.27	0.97
HandyTicket	1	1,047	30.59	3.48	20.78	0.08
	2	370	39.35	12.24	101.62	0.19
	3	150	42.32	21.74	242.95	0.30
	4	90	44.9	28.53	464.26	0.34
	5	42	46.85	36.66	832.95	0.35
	6	19	48.36	39.31	1,227.76	0.37
	7	1	51.71	90	3,530.99	0.22

Table 2. Cluster characteristics

4. Results and Comparison

The implementation of the model and the prediction of the future purchase numbers are carried out in Python (version 3.8) using open-source libraries Keras and TensorFlow. Because of the stochastic nature of the algorithm, the results of neural networks may exhibit variations in numerical precision. Therefore, the training of the proposed CNN is performed five times. The prediction of individual purchases in each period is based on the average outcome. To evaluate the forecast ability of the proposed CNN over time, the metric of mean absolute percent error (MAPE) is computed, since MAPE remains unaffected by the outliers over the forecast period (Valendin et al, 2022). As Valendin et al. (2022) suggest, the metric of root mean squared error (RMSE) is reported for evaluating the performance of the model at the individual level.

The following probability models are considered as comparison models: Pareto/GGG (Platzer & Reutterer, 2016), Pareto/NBD (Schmittlein et al, 1987), and BG/NBD (Fader et al, 2005). Additionally, the proposed LSTM algorithm by Valendin et al. (2022) is also used for benchmarking. Table 3 gives an overview of the forecast ability of all models over time (MAPE) and at individual level (RMSE). Regarding the CDNOW data set, all probability models provide a better forecast ability than the neuronal networks. The exceedingly high imbalance ratio makes this data set a challenge for neural networks. In the HandyTicket data set, the proposed CNN model effectively applies the features extracted from the input sequences to predict the future number of purchases. The prognostic efficacy of the CNN model surpasses that of benchmark models both temporally and across customer profiles.

Data set	Model	RMSE	MAPE
CDNOW	Proposed CNN	2.33	12.56%
	LSTM von Valendin et al. (2022)	2.30	21.04%
	Pareto/GGG	1.603	12.07%
	Pareto/NBD	**1.602**	**11.76%**
	BG/NBD	1.607	12.08%
HandyTicket	Proposed CNN	**8.54**	**5.74%**
	LSTM von Valendin et al. (2022)	18.75	33.49%
	Pareto/GGG	14.24	7.86%
	Pareto/NBD	13.89	10.51%
	BG/NBD	13.54	11.01%

Table 3. Summary of results for all models

5. Summary and Implications

This paper introduces a CNN-based algorithm for predicting the number of individual purchases in noncontractual business relationships. The algorithm is applied to the CDNOW and HandyTicket data sets. To address the potential negative impact of an imbalanced training set on CNN performance, the k-means clustering algorithm is initially employed. This results in the creation of 5 clusters for the CDNOW data set and 7 clusters for the HandyTicket data set. Addressing the existing imbalance issue, the hybrid resampling method is implemented. In case of the HandyTicket data set, the proposed CNN provides substantially better predictions to the benchmark models. These findings enable the provider of the HandyTicket application to take suitable actions based on the forecasted behavior of his customers within each cluster. However, for the CDNOW data set, the probabilistic models outperform the known deep learning algorithms. Applying neuronal networks for this data set requires exploring alternatives such as transfer learning or data augmentation techniques to overcome the

challenges arising from the high imbalance ratio. Furthermore, the suggested CNN model can be enhanced by incorporating additional input sequences, e. g. marketing appeals.

References

Allenby, G., Leone, R. P., & Jen, L. (1999). A Dynamic Model of Purchase Timing with Application to Direct Marketing. *Journal of the American Statistical Association*, 94(446), 365–374. https://doi.org/10.1080/01621459.1999.10474127

Anitha, P., & Patil, M. M. (2022). RFM model for customer purchase behavior using K-Means algorithm. *Journal of King Saud University-Computer and Information Sciences*, 34(5), 1785–1792. https://doi.org/10.1016/j.jksuci.2019.12.011

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *ArXiv 2018. ArXiv Preprint ArXiv:1803.01271*. https://arxiv.org/pdf/1803.01271.pdf

Bauer, J., & Jannach, D. (2021). Improved Customer Lifetime Value Prediction with Sequence-To-Sequence Learning and Feature-Based Models. *ACM Transactions on Knowledge Discovery from Data*, 15(5), 1–37. https://doi.org/10.1145/3441444

Biscione, V., & Bowers, J. (2020). Learning Translation Invariance in CNNs. *ArXiv Preprint ArXiv:2011.11757*,

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks : The Official Journal of the International Neural Network Society*, 106, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953

Chen, P., Guitart, A., Del Rio, A. F., & Perianez, A. (2018). Customer Lifetime Value in Video Games Using Deep Learning and Parametric Models. In *2018 IEEE international conference on big data (big data)* (2134-2140).

Fader, P., Hardie, B., & Lee, K. L. (2005). "Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model. *Marketing Science*, 24(2), 275–284.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., & Sriram, N. (2006). Modeling Customer Lifetime Value. *Journal of Service Research*, 9(2), 139–155. https://doi.org/10.1177/1094670506293810

Ide, H., & Kurita, T. (2017). Improvement of learning for CNN with ReLU activation by sparse regularization. In *2017 international joint conference on neural networks (IJCNN)* (2684-2691).

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (95-104).

Livieris, I., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32(23), 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

Lovmar, L., Ahlford, A., Jonsson, M., & Syvänen, A.-C. (2005). Silhouette scores for assessment of SNP genotype clusters. *BMC Genomics*, 6(1), 35. https://doi.org/10.1186/1471-2164-6-35

Martens, J., & Sutskever, I. (2011). Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (1033-1040). https://www.cs.toronto.edu/~jmartens/docs/rnn_hf.pdf

Matuszyk, P., Castillo, R. T., Kottke, D., & Spiliopoulou, M. (2016). A comparative study on hyperparameter optimization for recommender systems. *Workshop on Recommender Systems and Big Data Analytics (RS-BDA'16)@ IKNOW*, (2016). https://kmd.cs.ovgu.de/pub/matuszyk/hyperparameters-for-rs.pdf

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *ArXiv Preprint ArXiv:1511.08458*,

Platzer, M., & Reutterer, T. (2016). Ticking Away the Moments: Timing Regularity Helps to Better Predict Customer Activity. *Marketing Science*, 35(5), 779–799.

Rezk, N. M., Purnaprajna, M., Nordstrom, T., & Ul-Abdin, Z. (2020). Recurrent Neural Networks: An Embedded Computing Perspective. *IEEE Access*, 8, 57967–57996. https://doi.org/10.1109/access.2020.2982416

Salehinejad, H., & Rahnamayan, S. (2016). Customer shopping pattern prediction: A recurrent neural network approach. In *2016 IEEE symposium series on computational intelligence (SSCI)* (1-6).

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization. In Bengio, S. et al. (eds.), *Advances in neural information processing systems*.

Schmittlein, D. C., Morrison, D. G., & Colombo, R. (1987). Counting Your Customers: Who Are They and What Will They Do Next? *Management Science*, 33(1), 1–24. https://doi.org/10.1287/mnsc.33.1.1

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181. https://doi.org/10.1016/j.asoc.2020.106181

Tran, K.-G., Nguyen, Van-Ho, & Ho, T. (2021). Customer segmentation analysis and customer lifetime value prediction using Pareto/NBD and Gamma-Gamma model. In *The 4th International Conference on Business* (296 - 311).

Valendin, J., Reutterer, T., Platzer, M., & Kalcher, K. (2022). Customer base analysis with recurrent neural networks. *International Journal of Research in Marketing*, 39(4), 988–1018. https://doi.org/10.1016/j.ijresmar.2022.02.007

Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 232–243.

Wibawa, A. P., Utama, A. B. P., Elmunsyah, H., Pujianto, U., Dwiyanto, F., & Hernandez, L. (2022). Time-series analysis with smoothed Convolutional Neural Network. *Journal of Big Data*, 9(1), 44. https://doi.org/10.1186/s40537-022-00599-y

Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, 14(1), 54. https://doi.org/10.3390/info14010054