Clustering of Bank Customers using LSTM-based encoder-decoder and Dynamic Time Warping

Abstract

Clustering is an unsupervised data mining technique that can be employed to segment customers. The efficient clustering of customers enables banks to design and make offers based on the features of the target customers. The present study aims to cluster bank customers by an encoder-decoder network and the dynamic time warping (DTW) method. Once the deep neural network had been trained by customer transaction data, a feature vector of each customer was automatically extracted by the encoder. Long short-term memory (LSTM) deep networks were employed since bank transactions are time-series data in nature. Moreover, the distance of pairs of transaction amount sequences was obtained using DTW and stored in a matrix with a length as large as the number of customers. By combining the features obtained from this matrix, encoder-decoder network, and demographic data (e.g. age and gender), a multi-dimensional feature vector was extracted for each customer. The feature vector was introduced as input to different clustering algorithms, comparing the results. The contributions of the present work include automatic feature extraction, chronological transaction importance of features, model flexibility to customer transaction amounts, feature number adjustment, and a hybrid clustering approach. It was found that the clusters obtained from the hybrid approach were more significant than those derived from individual clustering techniques. The neural network layers had a strong effect on the clusters, and the hybrid GRU-LSTM approach yielded the most optimal results. Also, a high network error would not necessarily worsen clustering performance.

Keywords: Customer clustering, Time-series clustering, LSTM encoder-decoder, Dynamic time warping

Introduction

Banks seek to obtain competitive advantages in today's devastating competition and globalization []. It is important for the banking sector to identify advanced big data analysis methods, e.g., data mining techniques, in order to extract valuable information from a huge amount of data and improve strategic management and customer satisfaction [2].

Service marketing research has shown that companies should not offer the same services for all customers in most cases. Therefore, customer clustering and customer relationship management are determinants of business survival [3]. Efficient customer clustering enables the significant segmentation of customers. Clustering classifies customers with similar features and demands into the same group. Through customer clustering, banks can better identify customer behavior and develop more effective marketing strategies. Also, banks take a step toward data-driven decision-making by customer clustering, enhancing their knowledge of customer behavior. Clustering is typically the initial step of customer segmentation. Thus, the present work seeks to extract efficient features to cluster bank customers based on their transactions.

Literature review

Previous studies clustered customers based on customer equity through the k-means and k-medoids techniques, comparing the performances of the two approaches. They found that k-means clustering outperformed k-medoids clustering based on both the average within-cluster (AWC) distance and the Davies-Bouldin index [4]. A relatively recent work employed self-organizing maps and k-means to cluster customers. They divided the variables into demographic, bath consumption, and abstract consumption variables. Customers were clustered based on their three-month consumption and demographic data.

Although earlier works exploited either customer transaction data or demographic data, Davood et al. [6] utilized a combination of transaction and demographic data to obtain more accurate results. Therefore, banks can achieve their business objectives by finding different groups of customers with similar financial behavior. The present study proposes an intelligent model of bank customer clustering based on customer transactions. The proposed model converts customer transactions and static data (e.g., gender and age) into a vector in a latent space. This vector representation of customer data helps cluster customers with similar transaction behavior in the same group. Clusters of higher accuracy can be obtained by using vector representation and customer features of higher optimality.

Theoretical background

Transaction data refer to the dataset of an event such as a financial transaction or online payment. Each transaction involves at least a time dimension and the transition amount [7]. Here, transactions refer to bank transactions, such as payments or money transfers.

An artificial neural network (ANN) is a set of algorithms that attempt to detect basic relationships in a set of data through a human brain-inspired process. An autoencoder is an unsupervised learning method in which ANNs are employed for representation learning. In particular, a neural network architecture is designed to impose a bottleneck to force a compressed representation of the main input. Compression and reconstruction are very difficult. However, a data structure (i.e., a correlation between input features) can be trained and used when forcing through the bottleneck. This group of ANNs is employed to reduce dimensionality and diminish processing time and memory costs. These concepts were introduced by Hinton (11980) and the PDP Research Group. Autoencoders were considered in the form of restricted Boltzmann machines (RBMs) for deep architecture in the 2000s.

Two-component autoencoder structure

Such an autoencoder has an encoder and a decoder. The encoder maps input data into the feature space, while the decoder reconverts the feature data into the original form. In fact, the middle latent layer is the main layer of an autoencoder. It is employed as the extracted feature for clustering, as shown in Fig. 1.

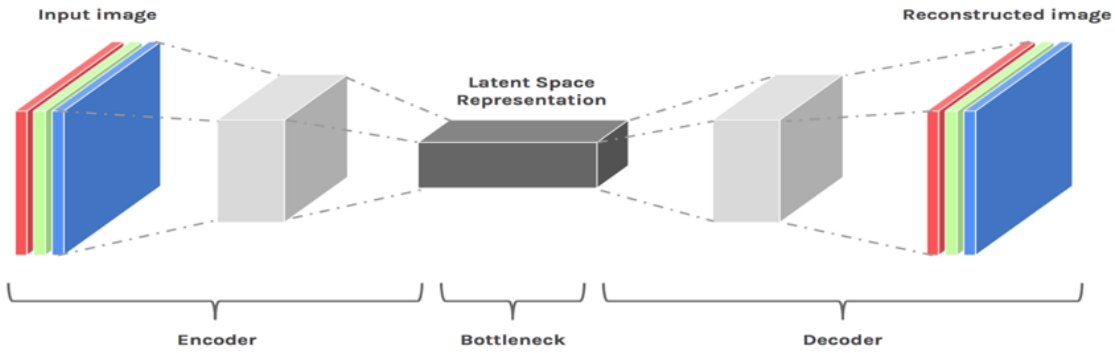


Fig. 1. Encoder and decoder of an autoencoder

Encoder-decoder framework

For a learning pair of (A, B) and model parameters θ, the framework calculates a sequence-to-sequence model of conditional probability P(B|A; θ). This can be performed by estimating probability conditions through the chain rule:

$$P(B|A;\theta) = \prod_{i=1}^{L} P(b_i|b_1, \ldots, b_{i-1}, A; \theta)$$

The model parameters are learned by maximizing the conditional probabilities of training data:

$$\theta^* = arg\max_{\theta} \sum_{(A,B)} (A,B) log P(B|A;\theta)$$

To construct an encoder-decoder network, different layers can be employed, including long short-term memory (LSTM) networks. They prove to be resistant to vanishing and exploding gradients [8]. The architecture of this ANN involves a set of sub-networks that are recurrently interconnected. This set of sub-networks is known as the memory block. Each block includes one or more self-connected memory cells and multiplying input, output, and forget gate layers providing the memory cells with the reading, writing, and reset processes. The LSTM network operates the same as a conventional recurrent neural network (RNN), except that the latent layer of LSTM models a nonlinear process [9].

The present study also evaluated dynamic time warping (DTW). For time series, DTW measures the dependence or similarity of two time sequences that may differ in time. For example, DTW can find the similarity of two walking patterns, even when the waking speeds or accelerations are not the same in time intervals. DTW has analyzed time sequences of audio, video, and image data. In fact, DTW can analyze data that can be obtained in the form of sequences. The DTW algorithm calculates the distance between two sequences as:

Algorithm DTW

Require: $A = < a_1, \ldots, a_S >$

Require: $B = < b_1, \ldots, b_T >$

Let δ be a distance between coordiantes sequences

Let $m[S,T]$ be the matrix of couples (cost, path)

$m[S,T] \leftarrow \big(\delta(a_1, b_1), (0,0)\big)$

for $i \leftarrow 2$ to S **do**

$m[i,1] \leftarrow \big(m[i-1,1,1] + \delta(a_i, b_1), (i-1,1)\big)$

---

end for

for $j \leftarrow 2$ to T **do**

$m[1,j] \leftarrow \left(m[1,j-1,1] + \delta(a_1, b_j), (1, j-1)\right)$

end for

for $i \leftarrow 2$ to S **do**

for $j \leftarrow 2$ to T **do**

$minimum \leftarrow minVal(m[i-1,j], m[i,j-1], m[i-1,j-1])$

$m[i,j] \leftarrow (first(minimum) + \delta(a_i, b_j), second(minimum))$

end for

end for

return $m[S,T]$

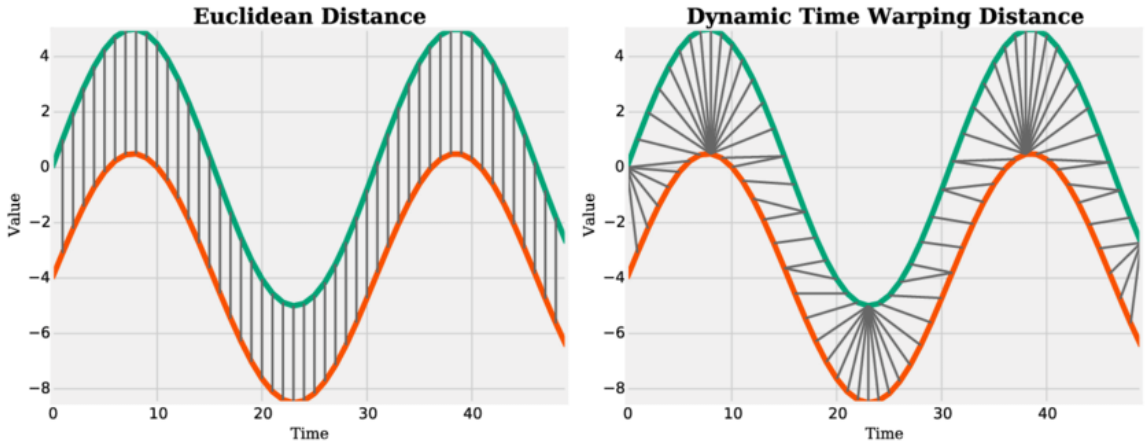


Fig. 2. Comparison of Euclidean and DTW distances [10]

Proposed customer clustering framework

Customer data are typically stored in a raw form in the databases of banks, without labels of valuable or uncreditworthy customers. Thus, unsupervised approaches are more efficient in the extraction of customer features through transaction data.

To extract customer features from customer transactions, the present study adopted a multilayer combination of LSTM and GRU. The input and output are a list of the transaction sequences of customers. The output of the encoder is known as the latent space. Thus, each input transaction has a vector representation in the latent space that contains most characteristics of the transaction sequence. The LSTM layer was employed to help the network better learn the transaction time-series [11].

The performance of this encoder-decoder model was improved by incorporating the attention mechanism [12]. This mechanism is used to tackle initial input sequence element information forgetting in the coded vector when the input sequence is long. In each output step, the last decoder latent state is utilized to generate an attention vector in the encoder for downsizing and disseminating information from the encoder to the decoder in each output step.

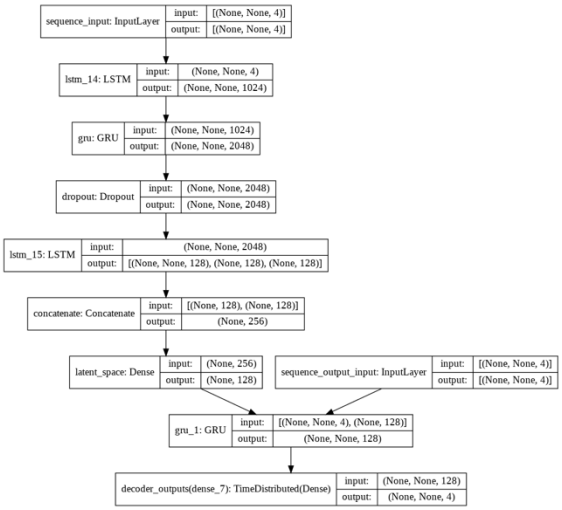


Fig. Architecture of the proposed encoder-decoder neural network

The decoder is an autoregressive model. In such models, however, a possible error in a step would continue to spread since the output of each step is used as the input of the next step [13]. Customer transactions are introduced as inputs in chronological order to the proposed neural network. To this end, transactions are classified based on the account numbers of customers.

Each transaction involves four aspects: (1) type, (2) timestamp, (3) amount, and (4) account balance. To equalize the dimensions of the transactions, two-dimensional zero arrays were added to the lists of transactions that were fewer than the maximum transaction number. A matrix of the transactions of all customers was obtained. Eqs. (1-3) represent a transaction feature vector, the feature matrix of a customer, and the feature vector of all customers:

$$(1)[type(p_0) \quad amount(m_0) \quad balance(b_0) \quad timestamp(t_0)]$$

$$(2)\begin{bmatrix} p_0 & m_0 & b_0 & t_0 \\ p_1 & m_1 & b_1 & t_1 \\ \vdots & \vdots & \vdots & \vdots \\ p_{i-2} & m_{i-2} & b_{i-2} & t_{i-2} \\ p_{i-1} & m_{i-1} & b_{i-1} & t_{i-1} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\begin{bmatrix} p_{0,0} & m_{0,0} & b_{0,0} & t_{0,0} \\ p_{1,0} & m_{1,0} & b_{1,0} & t_{1,0} \\ \vdots & \vdots & \vdots & \vdots \\ p_{i-1,0} & m_{i-1,0} & b_{i-1,0} & t_{i-1,0} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \cdots \quad \begin{bmatrix} p_{0,j} & m_{0,j} & b_{0,j} & t_{0,j} \\ p_{1,j} & m_{1,j} & b_{1,j} & t_{1,j} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k-1,j} & m_{k-1,j} & b_{k-1,j} & t_{k-1,j} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}\right] \qquad (3)$$

The data should be normalized prior to training. Normalization was carried out using the z-score as:

$$Z = \frac{x - \mu}{\sigma}$$

where Z is the final value, x is the initial value, μ is the mean, and σ is the standard deviation of the data. The average of the z-scores is zero. A Start-of-Sequence (SOS) label is applied to indicate the start of the transaction sequence so that the teacher forcing model is used. Teacher forcing is a strategy of training recurrent neural networks in which the output of a time step is used as the input of the next time step. The input of the decoder has only a start, and the output of the decoder has only an end. Thus, the input is shifted by a time step. Thus, it is required to use $[-1 \quad -1 \quad -1 \quad -1]$ for the start of the sequence. Also, $[-2 \quad -2 \quad -2 \quad -2]$ is used for the end of the sequence. For example,

$$Input = \begin{bmatrix} -1 & -1 & -1 & -1 \\ p_0 & m_0 & b_0 & t_0 \\ p_1 & m_1 & b_1 & t_1 \\ \vdots & \vdots & \vdots & \vdots \\ p_{i-1} & m_{i-1} & b_{i-1} & t_{i-1} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Output = \begin{bmatrix} p_0 & m_0 & b_0 & t_0 \\ p_1 & m_1 & b_1 & t_1 \\ \vdots & \vdots & \vdots & \vdots \\ p_{i-1} & m_{i-1} & b_{i-1} & t_{i-1} \\ -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Once training has been completed, it is required to define the decoder as a distinct model to receive customer transaction inputs and assign a feature vector to each customer to represent useful information on the transactions of the customer. The dimensionality of the latent space is recommended to be lower than the maximum number of transactions. The output of the decoder for each customer is a feature vector as:

$$Custmer\ Feature\ Vector = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_l \end{bmatrix}$$

Where f_i denotes feature i in the latent representation and dimension l is the latent layer. The final decoder output, which contains the features of all the customers, is obtained as a matrix:

$$Decoder\ Output: \begin{bmatrix} \begin{bmatrix} f_{0,j} \\ f_{1,j} \\ f_{2,j} \\ \vdots \\ f_{l,j} \end{bmatrix} & \cdots & \begin{bmatrix} f_{0,j} \\ f_{1,j} \\ f_{2,j} \\ \vdots \\ f_{l,j} \end{bmatrix} \end{bmatrix}$$

Additionally, to extract customer features, the DTW distance was employed to measure the similarity of pairs of customers. The distance between the transaction amounts of the two customers was obtained by DTW. The transaction amount and time of each customer were extracted and converted into a two-dimensional sequence in chronological order. Then, DTW was applied to measure the distance between the two transaction sequences. It is the minimum difference between the two transaction sequences under certain conditions. The transaction sequences of the customers were stored in a matrix whose rows and columns are bank transactions, and each entry denotes the DTW distance of the two transactions.

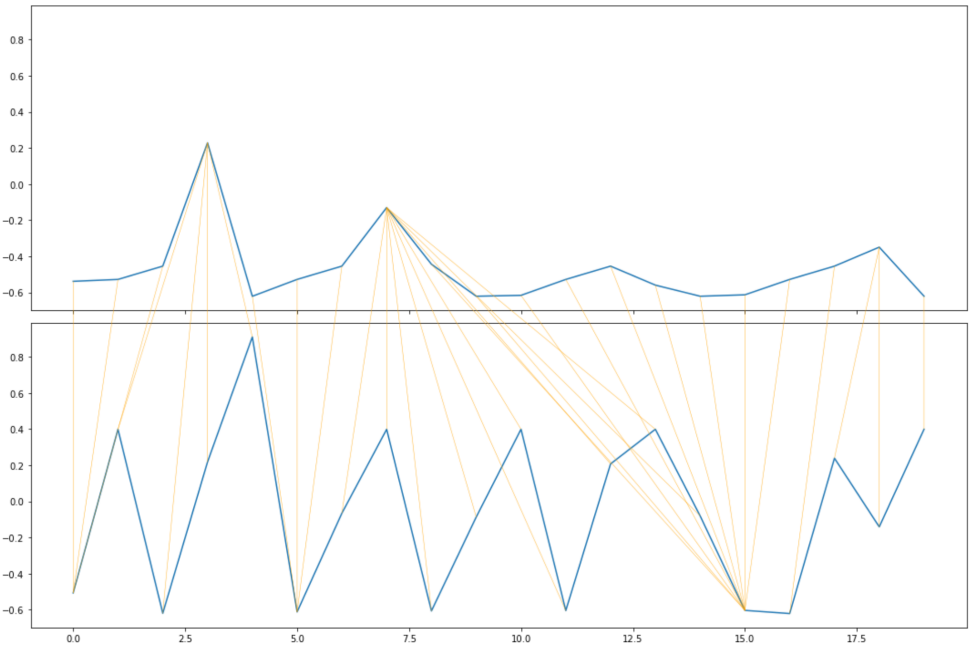


Fig. 3. Optimal DTW path to calculate the distance between two customer transaction amount series

There are as many distances as customers. These distances can be considered as a new feature vector. Then, the DWT features and LSTM features are concatenated. Moreover, the demographic features of the customers (e.g., age, gender, longitude, and latitude) are included. Thus, three feature matrices are concatenated to construct a new matrix. The concatenated matrix has a feature with a size of m+n+d for each customer, in which *m* denotes the LSTM feature length, *n* denotes the DTW distance, and *d* stands for the demographic data.

Indeed, it is required to reduce the dimensionality of the feature vector to improve clustering performance and visualize the results. To reduce feature vector dimensionality, the present study employed Kernel principal component analysis (KernelPCA), Isomap, and t-distributed stochastic neighbor embedding (t-SNE).

Finally, the dimension-reduced feature matrix was clustered using the k-means algorithm in light of its satisfactory performance for a large amount of data [14]. Although different augmentations of K-means clustering have been introduced, the present study adopted the elbow method to find the efficient number of customer clusters.

Results and discussion

A shortage of publically available data due to customer privacy protection reasons was a significant challenge. The present study employed an enhanced variant of the Beka Dataset

- the original database was published by Berka (2000). The Beka Database is the financial dataset of a bank in the Czech Republic. It contains data of over 5300 customers with nearly 1,000,000 transactions. Also, the bank granted 700 loans and issued approximately 900 credit cards (provided in the dataset) [15]. This study focuses on transaction and customer tables, as shown in Table 1.

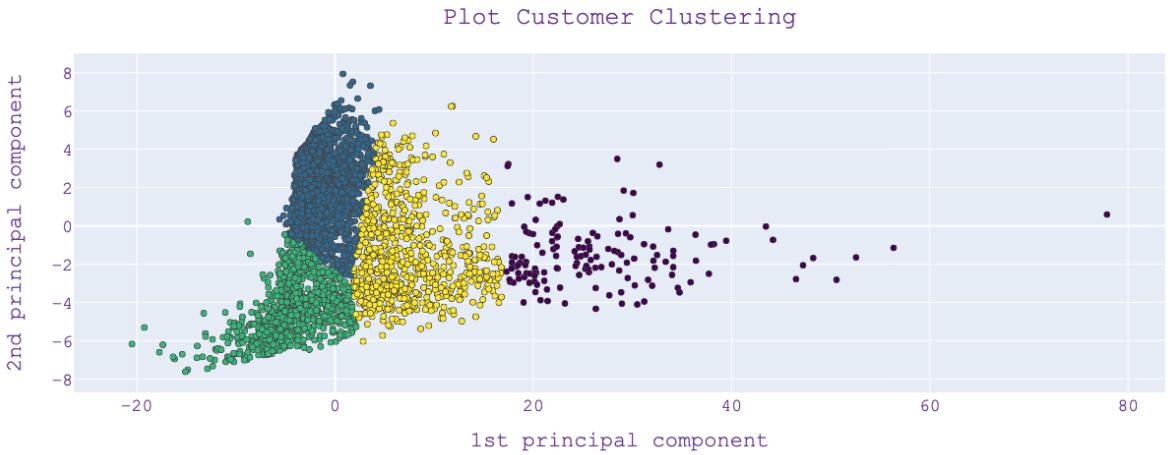
Table 1. Customer transactions

Trans_ID	Account_ID	Type	Amount	Balance	Timestamp
T00695247	A00002378	Credit	700.0	700.0	1356998400
T00171812	A00000576	Credit	900.0	900.0	1356998400
T01117247	A00003818	Credit	600.0	600.0	1356998400
T00579373	A00001972	Credit	400.0	400.0	1357084800
...	...	...	...	...	...

Table 2. Customer features

Customer_ID	Account_ID	Gender	Age	Latitude	Longitude
C00000001	A00000001	0.0	29	35.08449	-106.65114
C00000002	A00000002	1.0	54	40.71427	-74.00597
C00000004	A00000003	1.0	43	39.76838	-86.15804
...	...	...	...	...	...

The present study utilized 70% of the data as the training dataset, 20% as the test dataset, and the remaining 10% as the validation dataset for the encoder-decoder neural network. Both LSTM and GRU layers were employed in the neural network. Also, ReLU and sigmoid activation functions were utilized. The average loss was calculated to be 0.6395.



Then, customer clustering was implemented based on the customer distance matrix through DTW. The customers were divided into three clusters using the k-means method. A customer was randomly selected from each cluster by searching the dataset, plotting their transaction sequences. It was found that Customer Red had lower transaction amounts than Customer Blue. Moreover, Customer Yellow had a larger distance in their transactions. Therefore, it can be said that these clusters were significantly distinct.

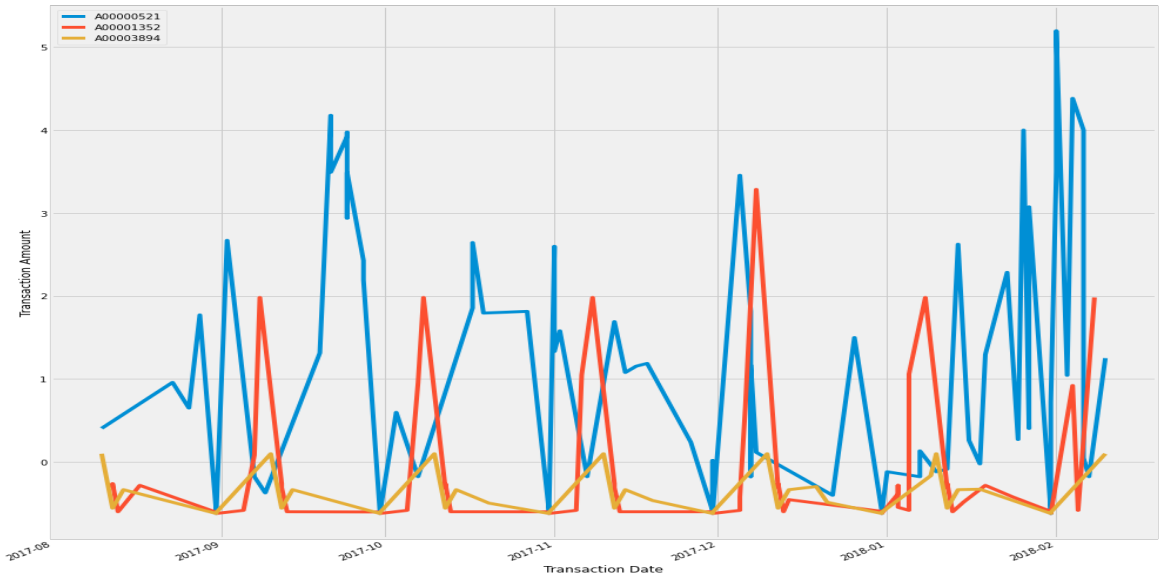


Fig. 4. Transaction sequences of three customers in three different clusters

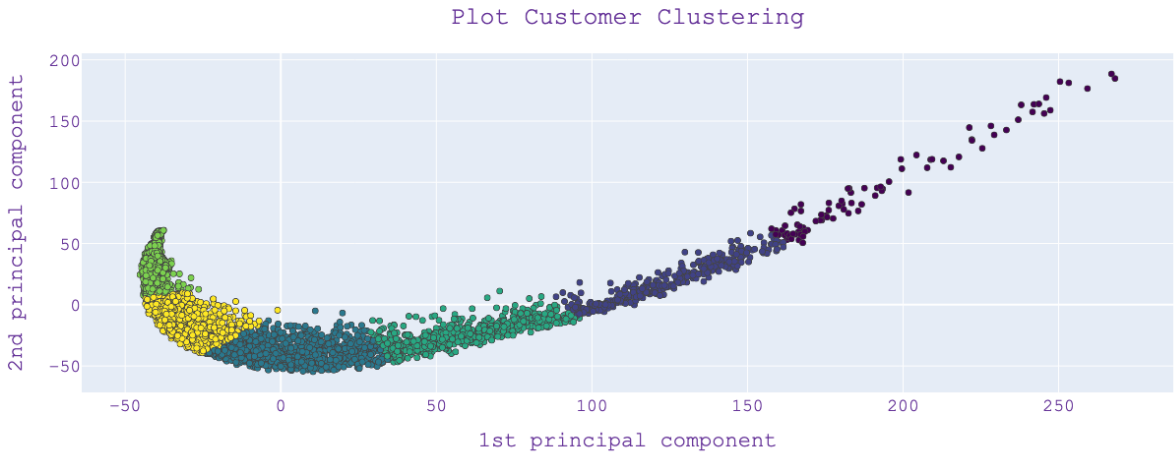


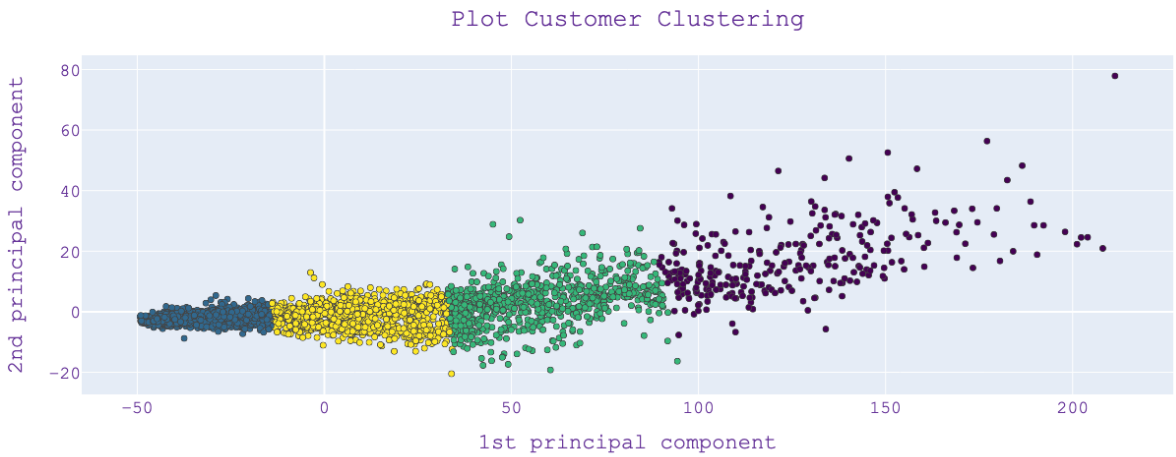
Fig. 5. K-means clustering of DTW features

Once neural network learning and DTW customer distance calculations had been completed, the customers were clustered in three scenarios: (1) LSTM features, (2) DTW distances, and (3) hybrid LSTM-DTW clustering.

Table X compares the clustering approaches. To evaluate the clusters, the Silhouette Coefficient (SC) and Davies-Bouldin Index (DBI) were utilized.

Table X. Comparison of the clustering approaches

No. of Clusters	K-means Clustering					
	Encoder-Decoder LSTM		Dynamic Time Warping		Hybrid Approach	
	DBI	SC	DBI	SC	DBI	SC
2	0.8180	0.4590	0.7061	0.5863	0.5011	0.7261
3	0.5601	0.5936	0.6002	0.5593	0.3823	0.7605
4	0.8067	0.4667	0.5604	0.5720	0.5395	0.7055
5	0.7699	0.4448	0.6033	0.5465	0.5991	0.6814
6	0.8226	0.4349	0.6144	0.5332	0.6733	0.6450



Conclusion

The present study clustered bank customers using LSTM and DWT. The results can be summarized as:

- The simultaneous use of LSTM and GRU layers in the encoder-decoder neural network improved model learning;
- The attention mechanism enhanced the accuracy of the proposed neural network and this clustering performance;
- A low error does not necessarily lead to high clustering performance – the opposite was the case with most cases.
- The training and testing of the model showed that the final clusters would be more unrealistic when the dimensionality of the latent space was larger than the maximum number of transactions of a customer.
- The DTW-extracted features had high continuity. Therefore, the individual DTW approach did not yield significant clustering performance.
- The proposed hybrid model was found to have higher performance evaluation indices than the two individual approaches in most cases.
- Pre-concatenation dimensionality reduction led to higher clustering performance than post-concatenation dimensionality reduction.