



Stochastics and Statistics

Cash demand forecasting in ATMs by clustering and neural networks

Kamini Venkatesh^a, Vadlamani Ravi^{a,*}, Anita Prinzie^b, Dirk Van den Poel^b^a Institute for Development and Research in Banking Technology (IDRBT), Masab Tank, Hyderabad, Andhra Pradesh 500057, India^b Department of Marketing, Faculty of Economics and Business Administration, Ghent University, Tweeackerkenstraat 2, Ghent, Belgium

ARTICLE INFO

Article history:

Received 29 November 2012

Accepted 19 July 2013

Available online 2 August 2013

Keywords:

Time series

Neural networks

SAM method

Clustering

ATM cash withdrawal forecasting

ABSTRACT

To improve ATMs' cash demand forecasts, this paper advocates the prediction of cash demand for groups of ATMs with similar day-of-the week cash demand patterns. We first clustered ATM centers into ATM clusters having similar day-of-the week withdrawal patterns. To retrieve "day-of-the-week" withdrawal seasonality parameters (effect of a Monday, etc.) we built a time series model for each ATMs. For clustering, the succession of seven continuous daily withdrawal seasonality parameters of ATMs is discretized. Next, the similarity between the different ATMs' discretized daily withdrawal seasonality sequence is measured by the Sequence Alignment Method (SAM). For each cluster of ATMs, four neural networks viz., general regression neural network (GRNN), multi layer feed forward neural network (MLFF), group method of data handling (GMDH) and wavelet neural network (WNN) are built to predict an ATM center's cash demand. The proposed methodology is applied on the NN5 competition dataset. We observed that GRNN yielded the best result of 18.44% symmetric mean absolute percentage error (SMAPE), which is better than the result of Andrawis, Atiya, and El-Shishiny (2011). This is due to clustering followed by a forecasting phase. Further, the proposed approach yielded much smaller SMAPE values than the approach of direct prediction on the entire sample without clustering. From a managerial perspective, the clusterwise cash demand forecast helps the bank's top management to design similar cash replenishment plans for all the ATMs in the same cluster. This cluster-level replenishment plans could result in saving huge operational costs for ATMs operating in a similar geographical region.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The importance of accurate forecasting of the withdrawal amounts in ATMs has the following motivation. Cash demand in ATMs needs to be forecasted accurately similar to other products in vending machines, as an inventory of cash needs to be ordered and replenished for a priori set period of time. If the forecasts are wrong, they induce costs. If the forecast is too high unused cash is stored in the ATM incurring costs to the bank. The bank pays different refilling costs depending on its policy with the money transportation company. In the first policy type, the bank pays a significant fixed fee for the refilling, independently of the amount, plus a small extra cost for each fraction of the transported money amount. In the second policy type, the bank pays a small fixed fee for refilling while the staircase costs are significant (Castro, 2009). According to Simutis, Dilijonas, Bastina, Friman, and Drobinov (2007) such cash-related costs represent about 35–60% of the overall cost of running an ATM. Wagner (2007) estimated a 28% cost saving as a result of improving the inventory policies and cash transportation decisions for an ATM network for a financial

institution ranked among the world top 700 banks. On the other hand, if the ATM runs out of cash, profit is lost and customers are dissatisfied due to bad service (www.neural-forecasting-competition.com/NN5/motivation.htm). Osorio and Toro (2012) minimized the cost of a cash-management system for a Colombian financial services institution without negatively affecting the service level.

It is obvious that daily cash withdrawal amounts are time series. Hence, typical cash demand forecast models will have to use time series prediction methods. Recognizing the need, Lancaster University came up with a NN5 timeseries competition, wherein daily cash withdrawal amounts over 2 years from 111 ATM centers across the UK are posted as the input data sets and several researchers proposed various models for the task (www.neural-forecasting-competition.com/NN5). In this study, the available data from NN5 time series competition (Crone, 2008) is used. For each of the 111 time series we forecast the next cash demands as a trace forecast for a forecasting horizon of 1 week.

This paper advocates the use of clusterwise cash demand prediction as it might have two advantages: (1) improved accuracy of the cash demand forecasts due to reduction in computational complexity when predicting an ATMs daily cash demand for groups of ATM centers with similar day-of-the week cash withdrawal seasonality patterns and (2) potentially huge savings in

* Corresponding author. Tel.: +91 40 23294042; fax: +91 40 23435157.

E-mail address: rav_padma@yahoo.com (V. Ravi).

operational costs as similar cash replenishment models can be used for ATM centers belonging to the same cluster.

To facilitate the cash demand forecasts, the ATM centers are (1) clustered into groups of ATM centers with similar day-of-the-week cash withdrawal patterns followed by (2) a clusterwise prediction of the daily cash demands.

First, each ATM center's withdrawal time series is translated into a "day-of-the-week" cash withdrawal seasonality sequence containing seven day-of-the-week cash withdrawal seasonality parameters. For each ATM center, the continuous seasonality sequence is translated into a discrete cash withdrawal seasonality sequence. This "abstraction" transforms the continuous seasonality sequence into a high-level quality seasonality sequence facilitating the detection of ATM clusters with similar day-of-the-week cash demand patterns. The similarity between the ATMs' discretized daily withdrawal seasonality parameter sequence is measured by calculating the Levenshtein distance using the Sequence Alignment Method (SAM). These distances are further processed by a clustering algorithm to produce groups of ATM centers which are relatively homogeneous with respect to the day-of-the-week cash withdrawal seasonality patterns.

Second, a predictive model is built for each cluster. For each ATM cluster four different neural networks are employed separately for forecasting purpose. We used MLP because it is universally popular in forecasting tasks. We employed GRNN, GMDH and WNN based on our experience and that of other authors (Li, Luo, Zhu, Liu, & Le, 2008; Mohanthy et al., 2010a, 2010b; Rajkiran & Ravi, 2007; Ravisankar & Ravi, 2010; Ravisankar, Ravi, Raghava Rao, & Bose, 2011; Srinivasan, 2008; Vinay Kumar et al., 2008).

The proposed approach is similar to Prinzie and Van den Poel (2006), as we also use SAM to first find ATM center clusters with similar temporal patterns. Our approach differs from Prinzie and Van den Poel (2006) in two major ways. Firstly, the sequences are represented by seven discretized day-of-the-week time series seasonality parameters rather than four discretized relative evolution turnover variables. We believe that the seasonality parameters estimated by time-series models are more precise than the calculation of relative evolution variables. Secondly, whereas Prinzie and Van den Poel (2006) include the cluster indicators as one of the predictors in the churn attrition model, this paper builds a separate cash demand forecasting model per ATM cluster.

The rest of the paper is organized as follows. In Section 2, literature review is presented. In Section 3, an overview of Sequence-alignment method is described. Section 4 presents the proposed methodology: construction of the sequential dimension; method to find effect-of-the-day parameter and its discretization; calculation of SAM distances; and the clustering procedure employing the Taylor-Butina algorithm. Section 5 presents a brief overview of the forecasting methods viz., WNN, GMDH, MLFF and GRNN. Results are discussed in Section 6. Finally, Section 7 concludes the work.

2. Literature review

In the following, we review the literature on modeling and analyzing NN5 competition data (Crone, 2008).

Bontempi and Taieb (2010) discussed the limitations of single-output approaches when the predictor is expected to return a long series of future values, and presents a multi-output approach to long term prediction. They also discussed here a multi-output extension of conventional local modeling approaches, and present and compare three distinct criteria for performing conditionally dependent model selection. Coyle, Prasad, and McGinnity (2010) employed self-organizing fuzzy neural network (SOFNN) to create an accurate and easily calibrated approach to multiple-step-ahead

prediction for the NN5 forecasting competition. Lemke and Gabrys (2010) investigated meta-learning for time series prediction with the aim to link problem-specific knowledge to well performing forecasting methods and apply them in similar situations. A forecasting approach based on Multi-Layer Perceptron (MLP) Artificial Neural Networks (named by the authors MULP) is proposed by Pasero, Raimondo, and Ruffa (2010) for the NN5 111 time series long-term, out of sample forecasting competition. Good results had also been obtained using the ANNs forecaster together with a dimensional reduction of the input features space performed through a Principal Component Analysis (PCA) and a proper information theory based backward selection algorithm. Teddy and Ng (2010) proposed a novel local learning model of the pseudo self-evolving cerebellar model articulation controller (PSECMAC) associative memory network to produce accurate forecasts of ATM cash demands. As a computational model of the human cerebellum, their model can incorporate local learning to effectively model the complex dynamics of heteroskedastic time series. Andrawis et al. (2011) used Forecast combinations of computational intelligence and linear models to solve the problem. The main idea of this model is to utilize the concept of combination of forecasts (ensembling), which has proven to be an effective methodology in the forecasting literature. The models used are neural networks, Gaussian process regression, and a linear model. Wichard (2011) proposed a simple way of predicting time series with recurring seasonal periods. Missing values of the time series are estimated and interpolated in a preprocessing step. He combined several forecasting methods by taking the weighted mean of forecasts that were generated with time-domain models which were validated on left-out parts of the time series. The hybrid model is a combination of a neural network ensemble of nearest trajectory models.

None of the papers modelling NN5 data clustered the time series first on daily withdrawal trends before predicting the ATM center's cash demand. Most authors try to reduce the complexity of the cash demand forecasting by either performing a data reduction of the input features space before prediction (Pasero et al., 2010) or by employing a local learning model (Teddy & Ng, 2010). This paper reduces the forecasting problem complexity by predicting cash demand per ATM cluster containing ATM centres with similar daily cash withdrawal patterns. We thereby follow an approach similar to Prinzie and Van den Poel (2006), who first clustered customers of an International Financial-Services Provider (IFSP) into clusters with similar relative evolution in turnover using SAM and the Taylor-Butina clustering algorithm followed by a binary logistic regression model predicting the customer's churn probability. Rather than including the cluster indicators as additional predictors in the forecasting model as in Prinzie and Van den Poel (2006) we build cash demand forecast models per ATM cluster with similar day-of-the-week cash withdrawal patterns.

3. Overview of sequence-alignment method (SAM)

The ATM centers are compared with respect to their daily cash withdrawal trends. Therefore, each ATM center is represented as a sequence of seven 'day-of-the-week' cash demand seasonality parameters. To facilitate the detection of ATM clusters with similar day-of-the-week cash withdrawal patterns, the "day-of-the-week" seasonality parameters are discretized. Then, the Sequence-Alignment Method is used to calculate the similarity of each pair of ATM centers on this discretized sequential dimension. Subsequently, these SAM distances are input to a clustering algorithm to identify ATM centers with similar day-of-the-week cash withdrawal patterns.

The Sequence-Alignment Method (SAM) (Levenshtein, 1965) was developed in computer science and found applications in text

editing, voice recognition and molecular biology (protein and nucleic acid analysis). A common application in computer science is string correction or string editing (Wagner & Fischer, 1974). The main use of sequence comparison in molecular biology is to detect the homology between macromolecules. If the distance between two macromolecules is small enough, one may conclude that they have a common evolutionary ancestor. Applications of sequence alignment in molecular biology use comparatively simple alphabets (the four nucleotide molecules or the twenty amino acids) but tend to have very long sequences. Besides computer sciences and molecular biology, SAM has applications in social science, transportation research and speech processing. SAM has also been applied in marketing to discover visiting patterns of websites (Hay, Wets, & Vanhoof, 2003), to identify purchase-history sensitive shopper segments (Joh, Timmermans and Popkowski-Leszczyc, 2003) and to identify customer segments with similar trends in turnover. Methodological reference works of SAM include Sankoff and Kruskal (1983) and Gribskov and Devereux (1992).

SAM handles variable length sequences and incorporates sequential information, i.e., the order in which the elements appear in a sequence, into its distance measure (unlike conventional position-based distance measures, like Euclidean, Minkowsky, City block and Hamming distances). The original SAM can be summarized as follows. Suppose we compare sequence a , called the source, having i elements $a = [a_1, \dots, a_i]$ with sequence b , i.e., the target, having j elements $b = [b_1, \dots, b_j]$. In general, the distance or similarity between sequences a and b is expressed by the number of operations (i.e., total amount of effort) necessary to convert sequence a into b . The SAM distance is represented by a score. The higher the score, the more effort it takes to equalize the sequences and the less similar they are. The elementary operations are insertions, deletions and substitutions or replacements. Deletion and insertion operations, often referred to as indel, are applied to elements of the source (first) sequence in order to change the source into the target (second) sequence. Substitution operations include both deletion and insertion. Every elementary operation is given a weight (i.e., cost) greater than or equal to zero. Weights may be tailored to reflect the importance of operations, the similarity of particular elements (element sensitive), the position of elements in the sequence (position sensitive), or the number/type of neighbouring elements or gaps. Different meanings can be given to the distance in sequence comparison. In this paper, we express similarity or distance between ATM centers on their parameter “day-of-the-week” effect by calculating the Levenshtein distance (Levenshtein, 1965) between each possible pair of ATM centers (i.e., pairwise-sequence analysis). The Levenshtein distance defines dissimilarity as the smallest sum of operation-weighting values required to change sequence $a-b$. This way, a distance matrix is constructed and subsequently, used as input for cluster analysis.

4. Proposed methodology

The proposed approach first clusters different ATM centers based on their daily withdrawal trends using the Taylor–Butina’s clustering algorithm and next builds clusterwise cash demand forecasting model using four types of neural network models: general regression neural network (GRNN), multi layer feed forward neural network (MLFF), group method of data handling (GMDH) and wavelet neural network (WNN). The entire data flow and the block diagram for the proposed method are depicted in Fig. 1.

- (1) We built one multiplicative time series model for each of the ATM center’s daily withdrawal amounts. We first reordered the data so that the withdrawal amounts are rearranged according to the day of the week. Thus, for each day of the

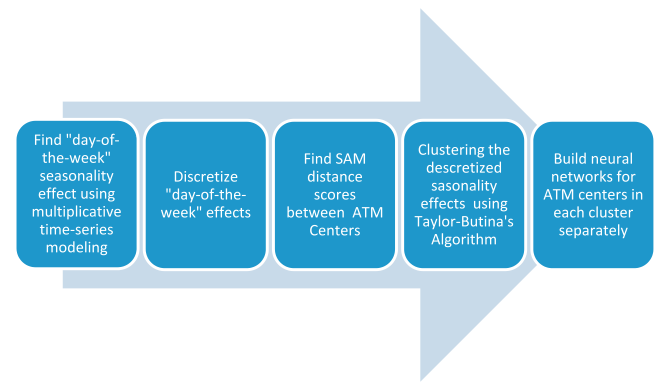


Fig. 1. Block diagram and data flow of the proposed methodology.

week, viz., Monday, Tuesday, ..., and Sunday we fitted the multiplicative time series $Y = T * S * C * I$, where T , S , C and I respectively represent trend, seasonality, cyclic movement and irregular part of the time series. We modeled the trend in the time series by linear regression using the principle of linear least squares. There is no cyclic effect on the time series and irregular part is assumed to be 1. Hence, we divided the original time series Y by the trend values in order to get seasonality effects, which are pure numbers. The seasonality part is estimated as $S = Y/T$. They are treated as the 7 day-of-the-week seasonality parameters (Monday, Tuesday, etc.).

- (2) Each ATM center’s time series of daily withdrawal amounts are replaced by a sequence of 7 day-of-the-week cash withdrawal seasonality parameters. The day-of-the-week seasonality parameter values are (for Monday, Tuesday, ..., Sunday) computed by separately taking average values of every Monday values (1,8,15,23...), Tuesday (2,9,16, 24...) and so on. To facilitate the identification of ATM clusters with similar day-of-the-week cash withdrawal patterns, the sequence of seven continuous daily withdrawal seasonality parameters are discretized. Consequently, for each ATM center, the continuous cash withdrawal seasonality sequence is translated into a discrete cash withdrawal seasonality sequence.
- (3) Then, we calculated the Levenshtein distance score using SAM to express how similar ATM centers are on their discretized daily withdrawal seasonality values.
- (4) Then, by using the Taylor–Butina’s clustering algorithm, we clustered the ATM centers on the SAM distances, thereby identifying ATM clusters with similar daily cash withdrawal seasonality values.
- (5) Then, as a last step, for each of the ATM clusters, four types of neural networks are built separately on the ATM’s daily cash withdrawal time series to forecast the ATM center’s daily cash demand by taking lagged data, as described in Section 4.7.

4.1. Construction of the sequential dimension

We want to facilitate the ATMs’ cash demand forecast by building cash demand prediction models for similar ATM centers. Clustering the ATM centers on their day-of-the-week withdrawal patterns is likely to reduce the complexity of the cash demand prediction task. For example ATM centers in business districts might be characterized by high cash demand during the week and low cash demand during the weekend. On the contrary, ATM centers in shopping districts are more likely to show a reverse trend: low cash demand during the week followed by huge cash demand in the weekend. Predicting an ATM’s cash demand knowing it belongs

to the ‘business district’ ATM cluster might be easier than predicting that same ATM center’s cash demand in absence of this cluster information.

4.2. “Day-of-the-week” seasonality effects parameter

As explained above, we first built a multiplicative time series model for each ATM center to find the seasonality effect on the cash withdrawal amount for each day of the week. The time series model is built on the daily withdrawal amounts for each ATM center, which are treated as the training data as specified by the NN5 competition. That way, we obtained, for each ATM center, the effect of a Monday, Tuesday, ..., and Sunday on cash demand, as seven continuous seasonality parameters. These seven “day-of-the-week” parameters are used to translate each ATM center’s 2 years time series of daily cash withdrawal amount into a seasonality sequence of seven numerical “day-of-the-week” effects. Hence, the length of each ATM center’s sequence would be seven.

4.3. Discretizing the “day-of-the-week” parameter

This paper proposes cash demand forecasting within ATM center clusters with similar day-of-the-week cash withdrawal patterns. In order to identify such ATM center clusters, we will cluster the ATMs’ day-of-the-week seasonality values. To enhance the detection of ATM centers with similar daily cash withdrawal trends with seasonality effects, we discretize each ATM center’s continuous seasonality sequence. We discretize the effect of day-of-the-week seasonality effect and subsequently replace each ATM center’s sequence of continuous daily seasonality parameters by a sequence of discretized daily seasonality parameters. After all, we want to limit the sequence alphabet in order to scale down the differences between the ATM centers’ daily seasonality sequences and facilitate ATM cluster detection. The alphabet size is limited to four by using the quartiles of the “day-of-the-week” parameters as shown in Table 1. Fig. 2 gives an example of how an ATM center’s 2 years cash demand time series is translated into a discrete day-of-the-week cash seasonality sequence.

This way the original 2 years cash demand time series data is transformed into a sequence of seven discretized seasonality effects of “day-of-the-week” per ATM center. Then, we calculated Levenshtein distance score using SAM to compare ATM centers on the similarity on these discrete sequence of “day-of-the-week” parameters.

4.4. Calculation of SAM scores

The elementary operations in the SAM method are insertions, deletions and reordering. Weights may be tailored to reflect the importance of operations and the similarity of particular elements.

The cost of reordering is given as follows:

$$\text{Cost}_{\text{reordering}} = \sum_{r=1}^R \eta * \text{pos}_{\text{reo}}$$

Table 1
Quartiles used to discretize the “day-of-the-week” seasonality parameter.

Day/quartile	1	2	3	4
Monday	<0.69	0.69–0.77	0.77–0.81	>0.81
Tuesday	<0.75	0.75–0.81	0.81–0.85	>0.85
Wednesday	<0.95	0.95–1.03	1.03–1.08	>1.08
Thursday	<1.34	1.34–1.46	1.46–1.55	>1.55
Friday	<1.21	1.21–1.29	1.29–1.38	>1.38
Saturday	<0.65	0.65–0.73	0.73–0.8	>0.8
Sunday	<0.77	0.77–0.83	0.83–0.87	>0.87

where R is the number of reorderings and η is the reordering weight and pos_{reo} is the absolute position of r th reordered element in the source.

The cost of deletion is given by

$$\text{Cost}_{\text{deletion}} = \sum_{d=1}^D W_d * c_{d,e}$$

where W_d is the weight for deletion and $c_{d,e}$ is cost for deletion of d th element with a certain value (i.e., element cost).

Finally, the cost of insertion is given by

$$\text{Cost}_{\text{insertion}} = \sum_{i=1}^I W_i * c_{i,e}$$

where W_i is the weight for insertion; $c_{i,e}$ is the cost for insertion of i th element with a certain value (i.e., element cost).

Then, the total SAM distance is given as follows:

$$\text{SAM}_{\text{dist}} = \min \left[\left(\sum_{r=1}^R \eta * \text{pos}_{\text{reo}} \right) + \left(\sum_{i=1}^I W_i * c_{i,e} \right) + \left(\sum_{d=1}^D W_d * c_{d,e} \right) \right]$$

In this paper, we set the operational cost in line with standard practice: the insertion and deletion operational costs are 1 and the reordering cost is 2. To maximize the SAM distance variance, we calculate the similarity between a pair of ATM discretized day-of-the-week trend sequences using element-sensitive deletion and insertion costs. As the day-of-the-week parameters are discretized in the ascending order of 1–4, their significance is reflected in the deletion/insertion costs in the ascending order of 0.25–1.

Element-based costs

Element	Del/Ins cost
1	0.25
2	0.5
3	0.75
4	1

The SAM distance is calculated using the algorithm by Hay et al. (2003). Here is an example of how to calculate the SAM distance for a pair of ATM centers. In the first step the longest common substring is defined.

ATM Center 1 (target)	1 4 4 3 1 4 2
ATM Center 2 (source)	1 3 4 1 1 4 4
ATM Center 1	1 4 4 3 1 4 2
ATM Center 2	1 3 4 1 1 4 4

In a second step, common elements not appearing in the substring are identified as elements that need to be reordered: elements 3 and 4.

$$\text{Cost}_{\text{reordering}} = 2 * 6 + 2 * 2 = 16 \text{ for } 3, 4.$$

where 2 is the reordering weight

ATM Center 1	1 4 4 3 1 4 2
ATM Center 2	1 3 4 1 1 4 4

In the third step unique elements in the source are deleted and unique elements in target are inserted. Element 1 is to be deleted from the source ATM center 2 and element 2 has to be inserted.

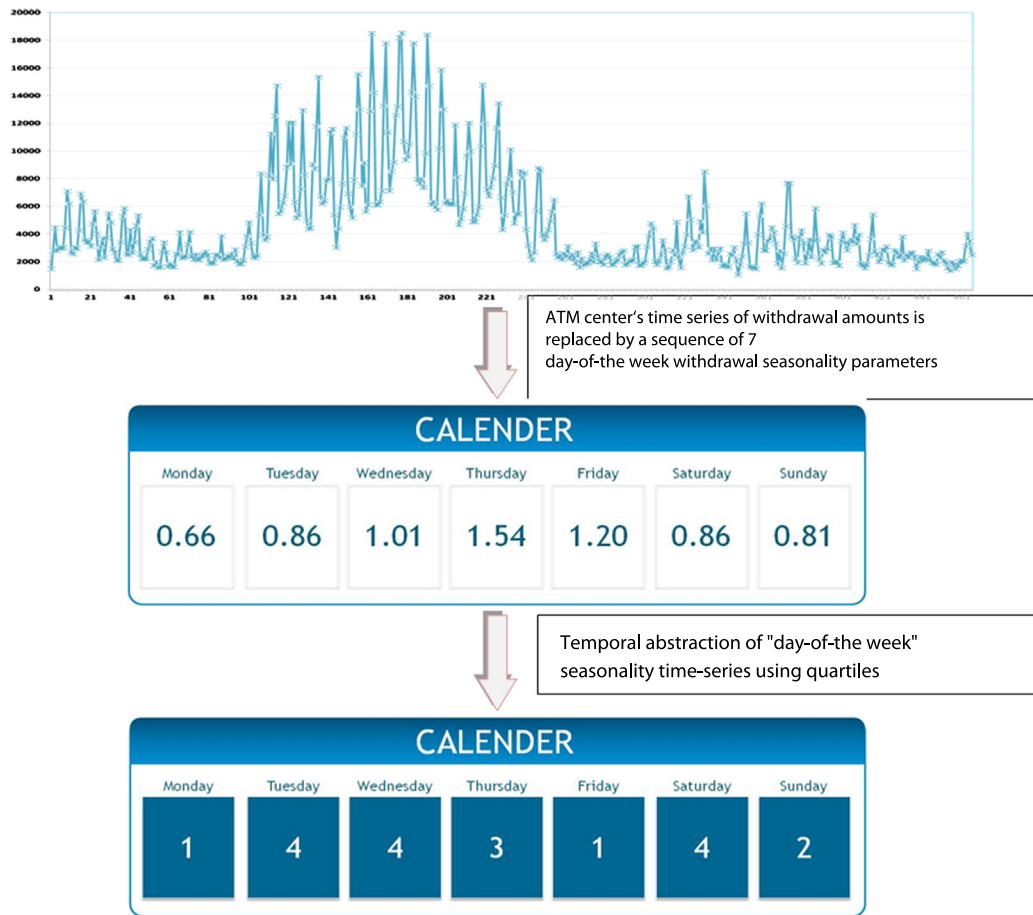


Fig. 2. Schematic diagram of how each ATM center's continuous daily cash demand time series is translated into a discrete "day-of-the-week" seasonality sequence.

$$\text{Cost}_{\text{deletion}} = 1 * (0.25) = 0.25$$

$$\text{Cost}_{\text{insertion}} = 1 * (0.5) = 0.5$$

$$\begin{aligned} \text{Then, total Cost} &= \text{Cost}_{\text{reordering}} + \text{Cost}_{\text{deletion}} + \text{Cost}_{\text{insertion}} \\ &= 16 + 0.25 + 0.5 = 16.75 \end{aligned}$$

4.5. Cluster analysis of Levenshtein SAM distances

We cluster the ATM centers on the Levenshtein SAM distances, expressing how similar the ATM centers are with respect to their day-of-the-week cash withdrawal seasonality patterns. For each of the clusters obtained, different neural networks are built separately for predicting the ATM withdrawal. While training each neural network architecture for each of the clusters, the input variables are the withdrawal amounts of past seven days (i.e. seven lags). We hypothesize that a prediction model for each cluster center based on SAM distances will outperform a similar model applied on the original data. A distance matrix holding the pairwise Levenshtein distances between ATM centers sequences is used as a distance measure for clustering. We apply the Taylor–Butina cluster algorithm on this Levenshtein distance matrix to cluster ATM centers on "day-of-the-week" seasonality information. MacCuish, Nicolaou, and MacCuish (2001) converted the Taylor–Butina exclusion region grouping algorithms into a real clustering algorithm, which can be used for both disjoint or non-disjoint (overlapping), either symmetric or asymmetric clustering. Although this algorithm is designed for clustering compounds (i.e., the chemi-informatics field with applications like compound acquisition and lead optimization in high-throughput

screening), in this paper it is employed to cluster ATM centers on "day-of-the-week" effect information.

4.6. Taylor–Butina algorithm

The Taylor–Butina algorithm is a five-step procedure described as follows:

1. Construct the threshold nearest-neighbour table using similarities in both directions.
2. Find true singletons, i.e., data points (in our case ATM centers) with an empty nearest-neighbour list. Those elements do not fall into any cluster.
3. Find the data point with the largest nearest neighbour list. This point tends to be in the center of the k th (k clusters) most densely occupied region of the data space. The data point together with all its neighbours within its exclusion region, constitutes a cluster. The data point itself becomes the representative data point for the cluster. Remove all elements in the cluster from all nearest-neighbour lists. This process can be seen as putting an 'exclusion sphere' around the newly formed cluster (Butina, 1999).
4. Repeat step 3 until no data points exist with a nonempty nearest-neighbour list.
5. Assign remaining data points, i.e., false singletons, to the group that contains their most similar nearest neighbour, but identify them as false singletons. These elements have neighbours at the given similarity threshold criterion (e.g. all elements with a

dissimilarity measure smaller than 0.3 are deemed similar), but a stronger cluster representative i.e., one with more neighbours in the list, excluding those neighbours (cluster criterion).

4.7. Forecasting using neural networks

Once the ATM centers are clustered as described above, four neural network architectures viz., MLFF, WNN, GRNN and GMDH are employed on the original time series data, which is rearranged with a time lag of seven days. Thus, we wanted to test the influence of the cash withdrawals performed during the last week on the present day's cash withdrawal. Therefore, past 7 days' cash withdrawal amounts are the input variables to predict the cash withdrawal of the present day as follows:

$$Y_i = f(Y_{i-1}, Y_{i-2}, Y_{i-3}, Y_{i-4}, Y_{i-5}, Y_{i-6}, Y_{i-7}),$$

where Y_i is the cash withdrawal on the i th day.

To compare the performance of the neural networks, we did not use mean absolute percentage error (MAPE), because MAPE has the disadvantage of becoming infinite if there are zero values in a series. Instead, we used the Symmetric Mean Absolute Percentage Error (SMAPE) as the error measure, as this is the measure considered in the NN5 competition by many competitors including Andrawis et al. (2011). It is defined as:

$$\text{SMAPE} = \frac{1}{M} \sum_{m=1}^M \frac{|Y_m' - Y_m|}{\left(\frac{|Y_m'| + |Y_m|}{2}\right)}$$

where Y_m is the actual time series value, Y_m' is the forecast, and M is the size of the test period.

5. Neural networks employed

5.1. Wavelet neural networks

The word wavelet is due to Grossmann and Morlet (1984). Wavelets are a class of functions used to localize a given function in both space and scale (<http://mathworld.wolfram.com/wavelet.html>). A family of wavelets can be constructed from a function $\psi(x)$, sometimes known as a “mother wavelet,” which is confined in a finite interval “Daughter wavelets” $\psi^{a,b}(x)$ are then formed by translation (b) and dilation (a). Wavelets are especially useful for compressing image data, since a wavelet transform has properties that are in some ways superior to a conventional Fourier transform. Recently, due to the similarity between the discrete inverse wavelet transform and a one-hidden-layer neural network, the idea of combining both wavelets and neural networks has emerged. This resulted in the wavelet neural network (WNN) – a feed forward neural network with one hidden layer. Based on the use of activation functions in the hidden nodes, there are two variants of WNN that are implemented here. They are Morlet wavelet function and Gaussian wavelet function. Wavelet networks employ activation functions that are dilated and translated versions of a single function $\psi: \mathbb{R}^d \rightarrow \mathbb{R}$, where d is the input dimension as stated in Zhang and Benveniste (1992) and Zhang (1997). It can dramatically increase convergence speed as stated in Zhang et al. (2001). WNN found many applications including software cost estimation (Vinaykumar et al., 2008) and software reliability prediction (Rajkiran & Ravi, 2007).

5.2. Group method of data handling (GMDH)

The Group Method of Data Handling (GMDH), introduced by Ivakhnenko (1966), is a self-organizing approach based on sorting-out of gradually complicated models and evaluation of them

by external criterion on separate part of data sample. GMDH has influenced the development of several techniques for synthesizing (or “self-organizing”) networks of polynomial nodes. GMDH has been applied in many fields such as predicting energy demand (Srinivasan, 2008), predicting software reliability (Mohanty et al., 2010b), web-services classification (Mohanty et al., 2010a), fraud detection (Ravisankar et al., 2011) and bankruptcy prediction (Ravisankar & Ravi, 2010). There is little advantage in precisely estimating the parameters of a model if its basic structure (the input variables, and their transformations and interactions) is rather tentative. The GMDH offers a hierarchic solution to this problem, by trying many simple models, retaining the best, and building on them iteratively, to obtain a composition of functions as the model. The building blocks (polynomial nodes) have the quadratic form

$$Z = W_0 + W_1X_1 + W_2X_2 + W_3X_1^2 + W_4X_2^2 + W_5X_1X_2$$

for inputs x_1 and x_2 , coefficient (or weight) vector w , and node output, z . The coefficients are found by solving the Linear Regression (LR) equations with $z = y$, the response vector.

The GMDH neural network develops on a data set. The data set including independent variables (x_1, x_2, \dots, x_n) and one dependent variable y is split into a training and testing set. During the learning process a forward multilayer neural network is developed in a series of steps.

In these networks, the most important input variables, number of layers, neurons in hidden layers and optimal model structure are determined automatically. These polynomial terms are created by using linear and non-linear regression. The initial layer is simply the input layer. The first layer created is made by computing regressions of the input variables and then choosing the best ones. The second layer is created by computing regressions of the values in the first layer along with the input variables. This means that the algorithm essentially builds polynomials of polynomials. Again, only the best are chosen by the algorithm. These are called survivors. This process continues until a pre-specified selection criterion is met.

5.3. Multi-layer feed forward neural network (MLFF)

MLFF is one of the most common NN structures, as they are simple and effective, and have been employed in a wide assortment of machine learning applications. MLFF starts as a network of nodes arranged in three layers—the input, hidden, and output layers. The input and output layers serve as nodes to buffer input and output for the model, respectively, and the hidden layer serves to provide a means for input relations to be represented in the output. Before any data is passed to the network, the weights for the nodes are random, which has the effect of making the network much like a newborn's brain—developed but without knowledge. MLFF are feed-forward NN trained with the standard back-propagation algorithm. They are supervised networks so they require a desired response to be trained. They learn how to transform input data into a desired response. So they are widely used for pattern classification and prediction. A multi-layer perceptron is made up of several layers of neurons. Each layer is fully connected to the next one. With one or two hidden layers, they can approximate virtually any input–output map. They have been shown to yield accurate predictions in difficult problems (Rumelhart, 1986). Simutis et al. (2007) successfully used a feed-forward neural network to predict the cash demand for n -subsequent days per ATM on simulated data.

5.4. Generalized regression neural network (GRNN)

Specht (1991) introduced GRNN. It can be thought of as a normalized Radial Basis Function (RBF) network in which there is a

hidden unit centered at every training case. These RBF units are called “kernels” and are usually probability density functions such as the Gaussian. GRNN is indeed a nonparametric regression model discretized with the help of Parzen windows approach and implemented as a feed forward neural network. The hidden-to-output weights are just the target values, so the output is simply a weighted average of the target values of training cases close to the given input case. The only quantities that need to be learned are the widths of the RBF units. These widths (often a single common width is used) are called “smoothing parameters” or “bandwidths” and are usually chosen by cross-validation or iterative genetic algorithm. However, gradient descent is not used. GRNN is a universal approximator for smooth functions, so it should be able to solve any smooth function–approximation problem given enough data. The main drawback of GRNN is that, like kernel methods in general, it suffers badly from the curse of dimensionality. GRNN cannot ignore irrelevant inputs without major modifications to the basic algorithm.

5.5. Neural networks based time series analysis on the entire data without clustering

We compare the results of our present work with those of direct neural network based time series analysis of the original data without resorting to clustering the ATM centers. In this procedure, we employed three different neural networks viz., MLFF, wavelet neural network (WNN) and GRNN and the traditional ARIMA separately on the entire sample without resorting to clustering. The procedure employed is as follows:

1. The NN5 dataset used in the experiments consists of a few missing values. The missing values are imputed using mean imputation technique.
2. Then, median-based deseasonalization method (Andrawis et al., 2011) is employed to deseasonalize the data.
3. Later, four forecasting techniques Auto Regressive Integrate Moving Averages (ARIMA) (Box and Jenkins, 1976), MLFF, WNN and GRNN are applied to the deseasonalized data with a time lag of 7 days.

6. Results and discussions

In this work, we have used available data from NN5 Forecasting competition, consisting of cash withdrawal of 2 years at various ATMs at different locations in England (see Crone; 2008, and <http://www.neural-forecasting-competition.com/NN5/datasets.htm>). In the NN5 competition, the task was to forecast 111 daily time series, representing daily cash withdrawal amounts at ATMs in various cities in the UK. All time series have a length of 2 years, and the forecast horizon is 56 days. There are few missing values in the data of ATM cash demand sequences. These are replaced by mean imputation, i.e. the mean values of the entire series. While employing four different neural network architectures, we tested on a range of parameters for each of the neural network and obtained the best results for the combination of parameters mentioned below. For MLP the best range of parameters are with learning rate 0.5–0.7; momentum rate 0.02–0.06; one hidden layer and 3–5 number of hidden nodes and maximum number of iterations as 500. For GRNN the range tested for the smoothing parameter is 0.2–0.3. Whereas, for WNN, the best dilation and translation parameter range is 0.5–1 and 2–4 respectively.

The objective is to cluster different ATM centers using the Taylor–Butina’s cluster algorithm and build cluster wise forecasting algorithms using various neural network models. We first built time series model per each ATM center to find the parameter effect of “day-of-the-week” (Monday, Tuesday, and Wednesday). This is used to build the sequence for each ATM center by taking time series seasonality parameters as sequence elements. After that, we

discretized the effect of “day-of-the-week”. Then, we created a distance score using SAM to compare this “day-of-the-week” seasonality parameter. Later, using SAM method, a distance matrix is constructed and consequently used as input for a cluster analysis. Eventually, we ended up with four clusters after applying the Taylor Butina’s clustering algorithm. The Taylor Butina’s algorithm automatically determines the number of clusters unlike other algorithms. Thereafter, for each of the clusters, different neural networks are built for the ATM withdrawal series prediction. Four forecasting neural networks namely GRNN, MLFF, GMDH and WNN are applied for analyzing this data.

For this method, the average SMAPE values over all clusters obtained by different NN techniques are presented in Table 2. We noticed very good performance from GRNN (presented in bold face in Table 2), which yielded an average SMAPE of 18.44. The best result of the study is yielded by GRNN with an average value of 17.67% SMAPE for one of the clusters centers, whereas, it yielded average SMAPE values of 18.83%, 19.38% and 17.89% for clusters 2, 3 and 4 respectively. GMDH yielded average SMAPE values of 19.56%, 20.24%, 21.63% and 19.86% for each of the clusters. As regards MLFF, it yielded average SMAPE values of 19.81%, 21.67%, 22.6% and 20.33% for each of the clusters. As regards WNN, it yielded average SMAPE values of 19.63%, 20.79%, 21.93% and 20.07% for each of the clusters. On an average for clusters, GMDH, MLFF and WNN yielded average SMAPE values of 20.32%, 21.10% and 20.60% respectively. However, GRNN yielded an average SMAPE value of 18.44% which is smaller than that of Andrawis et al. (2011). We also noticed that the centers in the cluster 1 and 4 yielded smaller SMAPE values when compared to that of other two clusters.

6.1. Evaluation and profiling of the ATM clusters

Clustering the ATM centers on their evolution of day-of-the week withdrawal seasonality values identified four ATM clusters. The clusterwise distribution of ATM centers as found by the Taylor–Butina algorithm is presented in Table 4. Cluster 1 has the highest number of ATM centers in it and the rest have almost the same number of ATM centers in them.

The statistical quality of the clustering is reflected by the lower standard deviation for the day-of-the week seasonality parameters within the clusters (row 5 onwards in Table 5) than for the total sample of ATM centers (row 2 in Table 5). Only for ATM cluster 2 the standard deviation on the Wednesday to Saturday seasonality parameters exceeds the total sample standard deviation on these seasonality parameters. In general, clustering the ATM centers on their evolution of day-of-the week withdrawal seasonality patterns reduces the variance on the withdrawal seasonality parameters. The smaller within-cluster variance signals the benefit of predicting the withdrawal amount within each ATM cluster as in our research approach.

Before profiling the different ATM clusters, we first investigate the evolution of the withdrawal seasonality parameters for all ATM machines. Fig. 3 shows the median day-of-the week cash withdrawal seasonality parameters for all ATM machines and for each ATM cluster using the continuous trend data. All ATM centers start with a lower

Table 2
Results of the proposed approach.

Cluster	MLFF	WNN	GRNN	GMDH
Cluster 1 (45)	19.81	19.63	17.67	19.56
Cluster 2 (21)	21.67	20.79	18.83	20.24
Cluster 3 (22)	22.6	21.93	19.38	21.63
Cluster 4 (23)	20.33	20.07	17.89	19.86
Average	21.10	20.60	18.44	20.32

withdrawal amount on Monday, cash demand increases slightly on Tuesday and Wednesday, with a maximum cash demand on Thursday, followed by a slight decrease in demand on Friday, dropping further to lower cash demand similar to Mondays and Tuesdays on Saturday and Sunday. The withdrawal amount on Wednesday represents an average day. The cash withdrawal amount is lower on Mondays, Tuesdays, Saturdays and Sundays. Higher withdrawal amounts are observed for Thursday and Friday. The variation in cash withdrawal seasonality value is highest on Thursday, Friday and Saturday (see bold faced numbers in Table 5).

The four ATM clusters show a pattern of the median day-of-the-week seasonality values that is similar to that of the total sample. Especially ATM cluster 1 closely follows the total sample pattern. However, the four ATM clusters do differ from the total sample in two ways. First, ATM cluster 3 does not have a peak in cash withdrawal on Thursday but on Friday. Second, the four ATM clusters do deviate from the total sample in the absolute size of the seasonality parameters. The latter is clearly reflected in Fig. 3 by the larger spread of the cluster seasonality parameters around the total sample parameter on Thursday, Friday and Saturday. The clustering handles the larger deviation on Thursday, Friday and Saturday and hence the clusterwise cash demand predictive model actually resulted in better predictions than a total sample predictive model. This is evident from Table 3, where the entire sample was analyzed using three neural networks viz., MLFF, WNN and GRNN and the traditional ARIMA approach. The best SMAPE value of the approach is 23.16 yielded by GRNN (presented in bold face in Table 3), much higher than 18.44, which again is yielded by GRNN in the proposed approach. Moreover, clusterwise predictions did yield considerably smaller SMAPE values in the case of MLFF and WNN also compared to the total sample approach. ARIMA approach did yield the worst result in the traditional approach of direct prediction on the entire sample without clustering.

Table 6 and Fig. 4 are used to profile the different ATM clusters. Table 6 shows which ATM cluster has the lowest median seasonality parameter and which ATM cluster has the highest median seasonality parameter. The numbers in parentheses indicate how much lower (row 3) or higher (row 4) the cluster's median day-of-the-week seasonality parameter is compared to the total sample's median seasonality parameter (row 2). For example, ATM cluster 2's Monday seasonality parameter is the highest, being 13% larger than the total sample's Monday seasonality parameter. Fig. 4 represents a correspondence map on the median day-of-the-week seasonality parameters for the four ATM clusters. The map shows the relationship between the ATM clusters and the effect of a particular day-of-the week on the withdrawal behavior. Monday, Tuesday and Sunday are close to each other in line with the shared lower withdrawal trends for these days. ATM cluster 2 is close to these days and to Saturday too, corresponding to the maximum day-of-the-week seasonality parameters for cluster 2 for

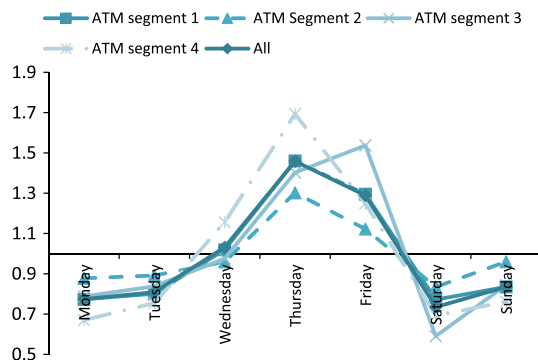


Fig. 3. Median day-of the week seasonality parameters for the four ATM clusters and for the total sample.

Table 3

Results of the neural network based time series analysis on the entire sample.

Model	SMAPE%
ARIMA	27.76
MLFF	23.87
WNN	23.33
GRNN	23.16

Monday, Tuesday, Saturday and Sunday. ATM cluster 3 is close to Friday, due to cluster 3 being the only segment with a peak in withdrawal trend on Friday rather than Thursday. ATM cluster 4 is close to Wednesday and Thursday as this cluster has the highest withdrawal trends for these two days.

Other interesting observations include:

- (1) All ATM clusters follow the total-sample succession of day-of-the-week seasonality parameters. We do not really find a cluster that only has large withdrawal amounts in the weekend and another ATM cluster only having large withdrawal amounts during the week.
- (2) All ATM clusters have small day-of-the week seasonality parameters on Saturday.
- (3) The SAM + clustering reduces the variation on the seasonality parameters which probably leads to better cash demand predictions for Thursday, Friday and Saturday (largest standard deviation for these days in total sample) than a total sample model.

Table 4

Cluster distribution of the ATM centers.

Cluster #	ATM Center #	Cluster #	ATM Center #	Cluster #	ATM Center #	Cluster #	ATM Center #
1	1	1	80	3	6	4	3
1	7	1	96	3	8	4	4
1	10	1	99	3	17	4	5
1	11	1	100	3	26	4	15
1	12	1	105	3	36	4	16
1	13	1	108	3	56	4	18
1	14	1	74	3	57	4	19
1	21	1	76	3	58	4	23
1	22			3	59	4	29
1	24	2	2	3	61	4	30
1	25	2	20	3	69	4	33
1	27	2	60	3	73	4	35
1	28	2	67	3	75	4	41
1	31	2	68	3	81	4	42
1	32	2	71	3	85	4	52
1	34	2	77	3	86	4	79
1	38	2	78	3	91	4	82
1	39	2	84	3	92	4	83
1	40	2	93	3	98	4	87
1	43	2	94	3	102	4	90
1	44	2	95	3	106	4	104
1	45	2	97	3	107	4	109
1	46	2	103			4	111
1	47	2	110				
1	48	2	9				
1	49	2	37				
1	50	2	64				
1	51	2	88				
1	53	2	89				
1	54	2	101				
1	55						
1	62						
1	63						
1	65						
1	66						
1	70						
1	72						

Table 5

The ATM clusters have smaller standard deviation on the day-of-the week seasonality parameters except for cluster 2 Wednesday to Saturday (shown in *italics*).

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
median	0.77	0.81	1.03	1.46	1.29	0.73	0.84
std	0.10	0.08	0.10	0.16	0.19	0.15	0.10
rank	5	7	4	2	1	3	6
ATM cluster 1	0.05	0.06	0.08	0.08	0.07	0.08	0.05
ATM cluster 2	0.08	0.08	0.12	0.17	0.23	0.23	0.09
ATM cluster 3	0.07	0.06	0.06	0.10	0.07	0.10	0.04
ATM cluster 4	0.05	0.07	0.08	0.07	0.08	0.10	0.04

Table 6

Profiling ATM clusters based on median seasonality parameters^a.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Overall	6	5	3	1	2	7	4
Median	0.78	0.82	1.05	1.48	1.30	0.74	0.85
Lowest	Cluster4 (−14%)	Cluster4 (−8%)	Cluster 3 (−7%)	Cluster 2 (−12%)	Cluster 2 (−13%)	Cluster 3 (−20%)	Cluster 4 (−11%)
Highest	Cluster 2 (13%)	Cluster 2 (+9%)	Cluster 4 (+10%)	Cluster 4 (+13%)	Cluster 3 (+19%)	Cluster 2 (+13%)	Cluster 2 (+13%)

^a For example, ATM cluster 2's median Monday seasonality parameter is the highest, being 13% larger than the total sample's Monday seasonality parameter.

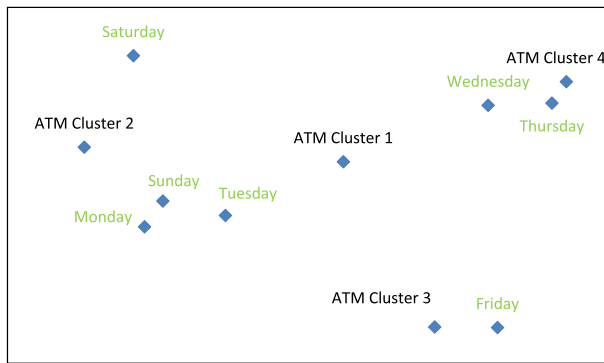


Fig. 4. Correspondence map showing the relationship between ATM clusters and median day-of-the week seasonality parameters. (The map shows the relationship between the ATM clusters and the days of the week. For example, ATM cluster 3 is close to Friday, due to cluster 3 being the only cluster with a peak in withdrawal on Friday rather than Thursday.)

Thus, the chief advantage of clustering the ATM centers first is to look for groups of ATM centers which are similar in the withdrawal pattern. The initial phase of clustering the similar ATM centers reduces the computational task in forecasting phase and also helps the bank's top management to design similar cash replenishment plans for all the ATM centers falling under the same cluster, thereby potentially saving huge operational costs. Thus, separate models need not be developed for each of the ATM centers. This is a significant outcome of the present study.

7. Conclusions

The primary objective of the paper is to cluster different ATM centers using the Taylor–Butina's clustering algorithm and build cluster-wise forecasting models using neural networks. The best result of the study, an average SMAPE value of 18.44% is yielded by GRNN. This result is better than the result of Andrawis et al. (2011). The initial phase of clustering the similar ATM centers reduces the computational task in the forecasting phase thereby improving cash demand predictions. Further, the proposed approach of clustering followed by prediction yielded much smaller SMAPE values than the traditional approach of direct prediction on the entire sample without clustering. From a managerial

perspective, identifying ATM clusters with similar daily cash demand trends helps the bank's top management to design similar cash replenishment plans for all the ATM centers falling under the same cluster. This segment-level replenishment plans could result in saving huge operational costs for ATM centers operating in a similar geographical region. For a financial-services institution having ATM centers operating in very different geographical regions, such huge cost savings could also be realized if the ATM segment-level cash replenishment plan is applied within operationally meaningful regions. That is ATM clusters with similar day-of-the week cash demand patterns are defined within each operationally meaningful region.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments which greatly improved the presentation of the manuscript.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.ejor.2013.07.027>.

References

- Andrawis, R.R., Atiya, A.F., El-Shishiny, H., 2011. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. *International Journal of Forecasting* 27, 672–688.
- Bontempi, G., Taieb, S.B., 2010. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. *International Journal of Forecasting* 27, 689–699.
- Box, G.E.P., Jenkins, G.M., 1976. *Time series analysis: Forecasting and control*. Holden-Day, San Francisco.
- Butina, D., 1999. Unsupervised data base clustering based on Daylight's fingerprint, Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical* 39 (4), 747–750.
- Castro, J., 2009. A stochastic programming approach to cash management in banking. *European Journal of Operational Research* 192 (3), 963–974.
- Coyle, D., Prasad, G., McGinnity, T., 2010. On utilizing self-organizing fuzzy neural networks for financial forecasts in the NN5 forecasting competition. *IJCNN*, 1–8.
- Crone, S., 2008. Results of the NN5 time series forecasting competition. In *Presentation at the IEEE world congress on computational intelligence. WCCI2008*. Hong Kong.
- Gribnikov, M., Devoreux, J., 1992. *Sequence analysis primer*. Oxford University Press, New York, USA.

- Grossmann, A., Morlet, J., 1984. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal of Mathematical Analysis* 15, 725–736.
- Hay, B., Wets, G., Vanhoof, K., 2003. Segmentation of visiting patterns on web sites using a sequence alignment method. *Journal of Retailing and Consumer Services* 10 (3), 145–153.
- Ivakhnenko, A.G., 1966. Group method of data handling – a rival of the method of stochastic approximation. *Soviet Automation Control* 13, 43–71.
- Joh, C.H., Timmermans, H.J.P., Popkowski-Leszczyk, P.T.L., 2003. Identifying purchase-history sensitive shopper segments using scanner panel data and sequence alignment. *Methods* 10 (3), 135–144.
- Lemke, C., Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. *Neurocomputing* 73, 2006–2016.
- Levenshtein, V.I., 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory* 10 (8), 707–710.
- Li, W., Luo, Y., Zhu, Q., Liu, J., Le, J., 2008. Applications of AR^{*}-GRNN model for financial time series forecasting. *Neural Computing and Applications* 17 (5–6), 441–448.
- MacCuish, J., Nicolaou, C., MacCuish, N.E., 2001. Ties in proximity and clustering. *Compounds. Journal of Chemical Information and Computer Sciences* 41 (1), 134–146.
- Mohanty, R., Ravi, V., Patra, M.R., 2010a. Web services classification using intelligent techniques. *Expert Systems with Applications* 37 (7), 5484–5490.
- Mohanty, R., Ravi, V., Patra, M.R., 2010b. Application of machine learning techniques to predict software reliability. *International Journal of Applied Evolutionary Computation* 1 (3), 70–86.
- Osorio, A.F., Toro, H.H., 2012. An MIP model to optimize a Colombian cash supply chain. *International Transactions in Operational Research*, 659–673.
- Pasero, E., Raimondo, G., Ruffa, S., 2010. MLP: A multi-layer perceptron application to long-term, out-of-sample time series prediction. *ISNN* 2, 566–575.
- Prinzie, A., Van den Poel, D., 2006. Incorporating sequential information into traditional classification models by using an element/position-sensitive SAM. *Decision Support Systems* 42, 508–526.
- Rajkiran, N., & Ravi, V., 2007. Software reliability prediction using wavelet neural networks. In *International conference on computational intelligence and multimedia applications*, Sivakasi, Tamilnadu, India.
- Ravisankar, P., Ravi, V., 2010. Financial distress prediction in banks using Group Method of Data Handling neural network, counter propagation neural network and fuzzy ARTMAP. *Knowledge Based System* 23 (8), 823–831.
- Ravisankar, P., Ravi, V., Raghava Rao, G., Bose, I., 2011. Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems* 50 (2), 491–500.
- Rumelhart, G.E., Hinton, G.E., Williams, R.J., 1986. *Learning internal representations by error propagation*. MIT Press, Cambridge, MA.
- Sankoff, D., Kruskal, J., 1983. *Time warps, string editions, and macromolecules. The Theory and Practice of Sequence Comparison*, Addison-Wesley Publications, Advanced Book Program, Mass.
- Simutis, R., Dilijonas, D., Bastina, L., Friman, J., Drobinov, P., 2007. Optimization of cash management for ATM network. *Information Technology and Control* 36, 117–121.
- Specht, D.F., 1991. A general regression neural network. *IEEE Transactions on Neural Networks* 2, 568–576.
- Srinivasan, D., 2008. Energy demand prediction using GMDH networks. *Neurocomputing* 72 (1–3), 625–629.
- Teddy, S.D., Ng, S.K., 2010. Forecasting ATM cash demands using a local learning model of cerebellar associative memory network. *International Journal of Forecasting* 27, 760–776.
- Vinaykumar, K., Ravi, V., Carr, M., Rajkiran, N., 2008. Software development cost estimation using wavelet neural networks. *Journal of Systems and Software* 81 (11), 1853–1867.
- Wagner, M., 2007. *The optimal cash deployment strategy – modeling a network of ATMs*. M.Sc. Thesis. Finland: Swedish School of Economics and Business Administration.
- Wagner, R.A., Fischer, M.J., 1974. The string-to-string correction problem. *Journal of the Association for Computing Machinery* 21, 168–173.
- Wichard, J.D., 2011. Forecasting the NN5 time series with hybrid models. *International Journal of Forecasting* 27, 700–707.
- Zhang, Q., 1997. Using wavelet network in nonparametric estimation. *IEEE Transactions on Neural Networks* 8 (2), 227–236.
- Zhang, Q., Benviste, A., 1992. Wavelet networks. *IEEE Transactions on Neural Networks* 3 (6), 889–898.
- Zhang, X., Qi, J., Zhang, R., Liu, M., Hu, Z., Xue, H., Fan, B., 2001. Prediction of programmed temperature retention values of naphthas by wavelet neural networks. *Computers and Chemistry* 25, 125–133.