Gediminas Žylius[*]

Kaunas University of Technology

EVALUATION OF ATM CASH DEMAND PROCESS FACTORS APPLIED FOR FORECASTING WITH CI MODELS

Keywords: computational intelligence, regression, time series forecasting, cash management, data-based forecasting, daily cash flow.

J E L Classification: C45, C53, G21.

Abstract: The purpose of cash management is to optimize distribution of cash. Effective cash management brings savings to retail banks that are related to: dormant cash reduction; reduced replenishment costs; decrease of cash preparation costs; reduction of cash insurance costs.

Optimization of cash distribution for retail banking in ATM and branch networks requires estimation of cash demand/supply in the future. This estimation determines overall cash management efficiency: accurate cash demand estimation reduces bank overall costs. In order to estimate cash demand in the future, cash flow forecasting must be performed that is usually based on historical cash point (ATM or branch) cash flow data.

Many factors that are uncertain and may change in time influence cash supply/demand process for cash point. These may change throughout cash points and are related to location, climate, holiday, celebration day and special event (such as salary days and sale of nearby supermarket) factors. Some factors affect cash demand periodically. Periodical factors form various seasonality in cash flow process: daily (related to intraday factors throughout the day), weekly (mostly related to weekend effects), monthly

---

Date of submission: August 7, 2015: date of acceptance: October 14, 2015.

[*] Contact information: gediminas.zylius@ktu.edu, Department of Automation, Faculty of Electrical and Electronics Engineering, Kaunas University of Technology, Studentų g. 50 - 154, LT - 51367 Kaunas, Lithuania, phone: +370 642 04 808.

(related to payday) and yearly (related to climate seasons, tourist and student arrivals, periodical celebration days such as New Year) seasons. Uncertain (aperiodic) factors are mostly related to celebration days that do not occur periodically (such as Easter), structural break factors that form long term or permanent cash flow shift (new shopping mall near cash point, shift of working hours) and some may be temporal (reconstruction of nearby building that restricts cash point reachability).

Those factors form cash flow process that contains linear or nonlinear trend, mixtures of various seasonal components (intraday, weekly, monthly yearly), level shifts and heteroscedastic uncertainty. So historical data-based forecasting models need to be able to approximate historical cash demand process as accurately as possible properly evaluating these factors and perform forecasting of cash flow in the future based on estimated empirical relationship.

◼◼◼ INTRODUCTION

The aim of this research is to study how cash flow process factors affect cash flow forecasting accuracy in ATM network, using computational intelligence methods as cash flow forecasting models when performing daily aggregated cash flow forecasting. For factor evaluation 8 typical (affected by different factors) ATM cash withdrawal process flows selected from real ATM network are used with historical period of 33 months.

Previous studies of Automatic Tellec Machine (ATM) withdrawal cash flow (Rodrigues, Esteves 2010) show that this process is strongly affected by calendar factors (day-of-the-week, week-of-the-month, month-of-the-year and holidays). Those effects can be used as numerical input values with CI models. Classical neural network regression approach using calendar effects (working day, weekday, holiday effect, salary day effect) as inputs were used by Kumar and Walia (2006) for cash demand process forecasting. However, cash demand varies in time and simply using regression inputs without incorporating recent time series information is usually not enough. Simutis et al. (2007) applied more advanced flexibe neural network approach by incorporating both regression inputs (calendar effects) and time series inputs (such as value of aggregated cash demand of previous several days).

During last decade, support vector machines (SVM) for various computational intelligence task was considered as alternative to neural networks because of benefits such as decision stability, less overfitting and better generalization abilities when smaller training (historical) data is available. However Simutis, Dilijonas and Bastina (2008) show that application of support vector machines to cash demand forecasting process has no superiority to neural networks and is even less accurate when reasonably long historical data period

for model training is available. As an alternative to neural network approach, interval type-2 fuzzy neural network (IT2FNN) was applied for cash demand forecasting (Darwish 2013). This type of model has both on-line structure and parameter learning abilities that lets model automatically adapt to different cash flow processes.

CI models for forecasting are considered to be a more advanced approach. However, time series models are also used for cash flow process forecasting. When comparing classical econometric time series models researches show (Gurgul, Suder 2013) that SARIMA (seasonal autoregressive integrated moving average) models are most accurate. It is shown (Wagner 2010) that SARIMA model even outperform joint forecasting approach using vector time series models. A comparison between time series probability density forecasting models (such as linear, autoregressive and structural and Markov-switching time series models) is made by Brentnall, Crowder and Hand (2010b) and results show that Markov-switching density forecasting model performs best.

Deeper investigation of cash demand process factors are also useful for forecasting. Random-effects models (Brentnall, Crowder & Hand 2008, 2010a) are used when modelling of individual client cash withdrawal patterns is made for ATM withdrawal forecasting. Laukaitis (2008) presented research on cash flow forecasting when intraday cash flow time series is treated as random continuous functions projected onto low dimensional subspace and apply functional autoregressive model as predictor of cash flow and intensity of transactions.

Forecasting models

This section presents computational intelligence forecasting models that are applied for daily-aggregated ATM cash flow forecasting.

Support vector regression model. Support vector regression (SVR) is application of support vector machines (SVM) model for regression problems. SVM was originally proposed by Cortes and Vapnik (1995) as a robust linear model for classification problems. The idea behind SVM is to map input space data vectors to output space via high dimensional space called feature space. This mapping is performed using so called kernel trick. By doing so, linear nature of SVM model can be applied for nonlinear function approximation. The nonlinear mapping requires nonlinear kernel function selection. Gaussian kernel functions are usually used because of few parameters. In this paper two types of

SVR models are applied: 1) 1) ν-support vector regression (ν-SVR) and 2) least squares support vector regression (LSSVR).

- ν-*SVR.* Let input data vectors to be $x_i \in R^n$ (*i*-th observation *n*-dimensional vector) and output (cash flow) data to be $y_i R^1$. Model optimization problem for ν-SVR algorithm is formulated by following equations (Chih--Chung & Chih-Jen 2011):

$$\min_{w,\varepsilon,\xi,\xi^*,b} \left\{ \frac{1}{2} w^T w + C \left[v\varepsilon + \frac{1}{l} \sum_{i=1}^{l} \left(\xi_i + \xi_i^* \right) \right] \right\}$$

$$\text{subject to} \quad \begin{cases} \left[w^T \phi(x_i) + b \right] - y_i \leq \varepsilon + \xi_i, \\ y_i - \left[w^T \phi(x_i) + b \right] \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \\ \varepsilon \geq 0. \end{cases}$$

Where $\phi(x_i)$ is kernel function (Gaussian) that performs mapping of input space to high dimensional feature space (the space where linear regression is performed); w is a parameter vector of *n*-dimensional hyperplane; b is hyperplane bias parameter; ξ_i^*, ξ_i are upper and lower training errors (slack variables) subject to ε – insensitive tube; C is a cost parameter, that controls the trade-off between allowing training errors and forcing rigid margins; v is regularization parameter that controls parameter number of support vectors; l – is number of data points (observations). Data points that lie on the boundaries of ε – insensitive tube are called *support vectors*. Graphical illustration of ν-SVR model is depicted in Figure 1.

Figure 1. An illustration of ν-SVR approximation principle
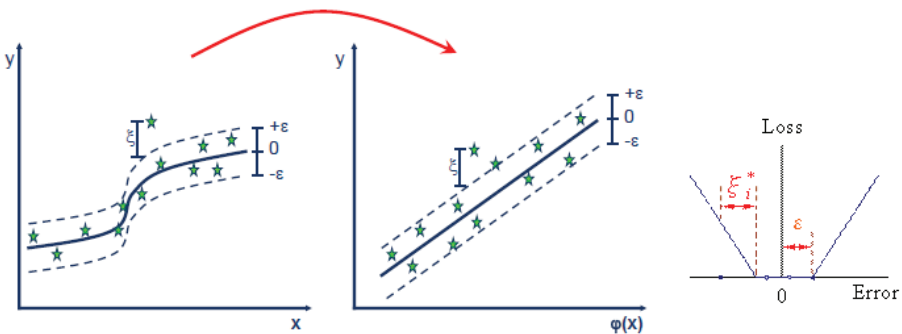
In this research ν-SVR code is used that is implemented in LIBSVM library (see Chih-Chung & Chih-Jen 2011).

- *LS-SVR.* Least squares support vector regression (LS-SVR) model is very similar to ν-SVR. Optimization problem is formulated by following equations (Suykens et al. 2002):

$$\min_{w,b,e}\left\{\frac{1}{2}\,\boldsymbol{w}^T\boldsymbol{w}+\frac{\gamma}{2}\sum_{i=1}^{l}e_i^2\right\}$$

subject to $\begin{cases}\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b+e_i=y_i,\\ \gamma\geq 0.\end{cases}$

Where e_i are error variables and γ is regularization constant.

Differently from ν-SVR, LS-SVR doesn't use insensitive tube and is regularized only by parameter γ, so is not as sparse as ν-SVR model (has more support vectors). But least squares loss function brings other flexibility benefits for regression problems. As for ν-SVR model, for LS-SVR model Gaussian kernel functions are used.

In this research LS-SVR code is used from *LS-SVMlab* toolbox presented in website (see Pelckmans et al. 2002).

Relevance vector regression model. Relevance vector regression (Tipping 2001) is model that has same linear functional form as support vector regression:

$$y(\boldsymbol{x};\boldsymbol{w})=\sum_{i=1}^{M}w_iK(\boldsymbol{x},\boldsymbol{x}_i)+w_0.$$

Where $K(\boldsymbol{x},\boldsymbol{x}_i)$ is defined as kernel function and is model weight vector.

RVR uses even less support vectors (is more sparse than ν-SVR), that are called *relevance vectors*. Bayesian inference methodology is used during RVR model parameter and relevance vector determination. RVR uses EM-like (expectation-maximization) learning algorithm and applies *a priori* distributions (because of Bayesian methodology) over parameters. Gaussian kernel as in both LSSVR and ν-SVR cases are also used with RVR.

In this research RVR implemented in *SparseBayes* package for *MATLAB* by RVR author himself (see Tipping 2001) is used.

Feed-forward neural network model. Feed-forward neural network (FFNN) is the most popular artificial neural network architecture for nonlinear function approximation (classification and regression) problems. The architecture of algorithm was inspired by biological neural networks (brains) and models the interconnection of neurons that can compute values from inputs by feeding information through the network. FFNN model architecture is depicted in Figure 2.

Figure 2. An illustration of feed-forward neural network architecture
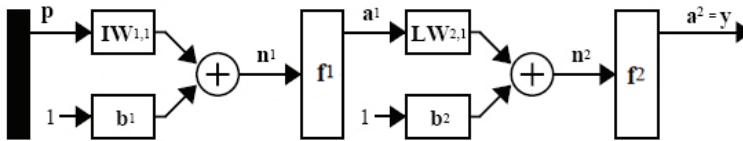
S o u r c e : created by authors.

In this research, logarithmic sigmoid transfer function was used in the hidden layers and linear transfer function in the output layer. Logarithmic sigmoid transfer function output of input variable is equal to:

$$y = \log\left(\frac{1}{1 + e^{-a}}\right).$$

In this work, Levenberg – Marquardt (Hagan & Menhaj 1994) backpropagation (Rumelhart, Hinton & Williams 1986) training algorithm was used for feed-forward neural network model training, which is implemented in *MATLAB Neural Networks Toolbox.*

Generalized regression neural network model. Generalized Regression Neural Network (GRNN) first proposed by Specht (1991) is special case of radial basis function (RBF) neural network. GRNN does not require an iterative training procedure (error back propagation as with other neural network architectures). GRNN training procedure requires specification of RBF spread parameter. It uses those functions to cover input space and approximates function as weighted linear combination of radial basis functions. Number of RBF function is equal to number of observations (number of days in historical daily cash flow). Each RBF is formed for each data point vector that is a center of RBF. RBF transfer function values are calculated according to Euclidean distance from the central point to input vector. In this research GRNN implemented in *MATLAB Neural Network Toolbox* is used.

Adaptive neuro-fuzzy inference system model. Adaptive neuro-fuzzy inference system (ANFIS) (Jang 1993) combines fuzzy inference system and neural network features: neural network training capabilities (backpropagation) with fuzzy input and output formation (Takagi–Sugeno fuzzy inference system). An architecture of ANFIS model that has two membership functions is depicted in Figure 3. This type of model architecture has 5 layers: fuzzy layer (1), product layer (2), normalization layer (3), defuzzification layer (4) and summation layer (5).

Figure 3. Illustration of ANFIS model architecture with two membership functions
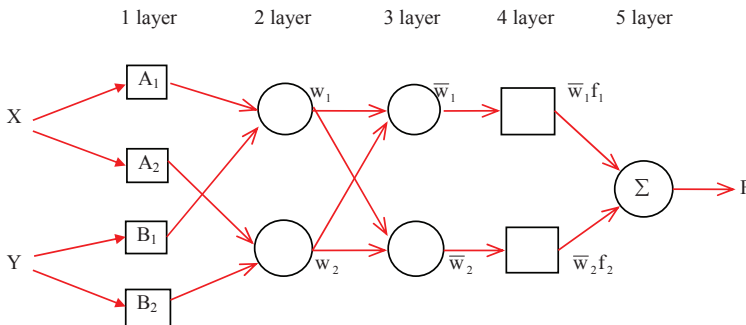
S o u r c e : created by authors.

For a 1st order of Sugeno fuzzy model, a typical **IF-THEN** rule set can be expressed as:

1) **IF** x is A_1 **AND** y is B_1 **THEN** $f_1 = p_1 x + q_1 y + r_1$;
2) **IF** x is A_2 **AND** y is B_2 **THEN** $f_2 = p_2 x + q_2 y + r_2$.

Further each of five layer functionality is shortly explained:

- 1st layer. Forms output, which determines membership degree in each of membership functions ($\mu_{A1}, \mu_{A2}, \mu_{B1}, \mu_{B2}$):

$$O_{1,i} = \mu_{A_i}(x), i = 1, 2,$$

$$O_{1,i} = \mu_{B_{i-2}}(y), \quad i = 3, 4.$$

- 2nd layer. In this layer each node is fixed and represents weight of particular rule. In this node **AND** operation is performed, which is product of inputs:

$$O_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), i = 1, 2.$$

- 3rd layer. Each node of this layer is also fixed and calculates normalized rule excitation degree:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2.$$

- 4th layer. This layer is not fixed as other and parameters (p_i, q_i, r_i) are estimated during training process. Output of nodes are calculated as:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i)$$

- 5th layer. This is an output layer, where output value is calculated as a sum of all inputs:

$$O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i \overline{w}_i f_i}{\sum_i w_i}.$$

In this research, two types of ANFIS model training algorithms are used: classical gradient steepest descend backpropagation and hybrid training algorithm. Hybrid training combines gradient descend backpropagation and least squares methods. Backpropagation is used to tune input layer membership function parameters, while least squares method is used for output function parameter tuning.

For input layer membership function parameter initialization fuzzy c-means (FCM) clustering algorithms is used, that partitions data of the input space into some number (c) of clusters and use them as input membership function initialization.

In this research ANFIS model that is implemented in *MATLAB Fuzzy Logic Toolbox* is used.

- *Extreme learning machine model.* Extreme learning machines (ELM) (Huang, Zhu & Siew 2006) have the same architecture as single hidden layer feed-forward neural networks. Differently from conventional neural network achitectures, ELM doesn't require backpropagation for parameter tuning. Instead, hidden node weights are chosen randomly and output weights are determined analytically. Main advantage of this type of learning is speed, which is many times faster than conventional iterative tuning (such as backpropagation).

Given N number of observations $(\boldsymbol{x}_i, y_i)$, single layer neural network output with M hidden nodes is modeled as:

$$o_i = \sum_{j=1}^{M} \beta_j f(w_j x_i + b_j).$$

Where $\boldsymbol{x}_i$ is $\boldsymbol{w}_j$ th input vector; is weight vector connecting the jth hidden node and the input nodes; β_j is weight scalar connecting jth hidden node and output node; b_j is bias parameter of jth hidden node.

In this research linear output nodes and sigmoid hidden nodes are used. Above equation can be written in vector form:

$$\boldsymbol{o} = \mathbf{H}\boldsymbol{\beta}.$$

Where $\mathbf{H}$ is N x M hidden layer output matrix and $\mathbf{H}_{i,j} = f(\boldsymbol{w}_j \boldsymbol{x}_i + b_j)$.
The solution of applying ELM theory is simply estimated as:

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\boldsymbol{o}.$$

Where $\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$ is Moore – Penrose generalized inverse (pseudoinverse) matrix.

A *MATLAB* implementation of classical ELM is used in this research, which is available at webpage (see Huang, Zhu & Siew 2006).

Experimental data

In this research 8 different typical ATM daily withdrawal data is used with historical period up to 990 days. Those 8 ATMs were selected from large database and represent typical cash flow factors that occur in cash flow process. Further a short explanation of every ATM is conduced:

- ATM number 1 contains cash flow with strong weekly seasonality factor;
- ATM number 2 contains cash flow with strong yearly and weekly seasonality factors, when yearly seasonality is smooth;
- ATM number 3 contains cash flow with strong monthly seasonality factor;
- ATM number 4 contains cash flow with strong yearly and weekly seasonality factors, when yearly seasonality is not smooth (temporal structural breaks);
- ATM number 5 contains cash flow with weak mixed (weekly and monthly) seasonality, increasing trend and heteroscedasticity factors with

permanent structural break factor (sudden cash flow break when ATM cash flow decreases);

- ATM number 6 contains cash flow with mixed (weekly and monthly) seasonality and permanent structural break factors (sudden cash flow break when ATM cash flow increases);
- ATM number 7 contains cash flow with weekly seasonality and temporal structural break factors (sudden cash flow break when ATM cash flow decreases and again increases to normal cash flow level after some period);
- ATM number 8 contains cash flow with weekly seasonality, increasing trend and heteroscedasticity factors.

Cash flow pictures together with autocorrelation functions are presented in Appendix for every ATM.

METHODOLOGY

In order to evaluate forecasting accuracy a specific metric is needed. In this research forecasting accuracy metric called symmetric mean absolute percentage error (SMAPE) is used which is calculated with following formula:

$$SMAPE = \frac{100}{N} \sum_{t=1}^{N} \frac{|\hat{y}_t - y_t|}{\frac{1}{2}(\hat{y}_t + y_t)}.$$

Where $\hat{y}_t$ is forecasted cash flow value and y_t is real cash flow value.

Forecasting is performed by *training* model with one part of historical cash flow dataset once and *testing* (forecasting accuracy evaluation) is done with another part of historical cash flow dataset. Four cases of training for every ATM are performed (Figure 4): 1) models are trained with 2 year (or 800 days for some ATMs) historical period; 2) models are trained with 1.5 year (or 400 days for some ATMs) historical period; 3) models are trained with 1 year (or 200 days for some ATMs) historical period; 4) models are trained with 0.5 year (or 100 days for some ATMs) historical period. But for testing, same amount of data is used for every training case.

Figure 4. Illustration of ATM cash flow data division
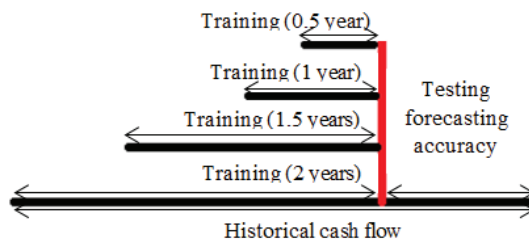
In order to train model for forecasting, some inputs *features* that represent cash flow factors must be constructed. In this research 9 features are constructed as inputs for every forecasting model (*i* represents forecasting day index):

- 1st input. Month number (numbers 1–12);
- 2nd input. Day of the month (numbers 1–31);
- 3rd input. Day of the week (numbers 1–7);
- 4th input. Withdrawal amount of *i*-1 day (previous day);
- 5th input. Withdrawal amount of *i*-7 day (previous week day);
- 6th input. Withdrawal amount of *i*-14 day (previous 2nd week day);
- 7th input. Withdrawal amount of *i*-21 day (previous 3rd week day);
- 8th input. Withdrawal amount of *i*-28 day (previous 4th week day);
- 9th input. Sum of withdrawals of last 5 days (*i*-5 to *i*-1).

Those features were selected after experimental studies. In order to evaluate what features (factors) are best for each particular type of ATM cash flow forecasting, one must perform forecasting with all combinations of features which using binomial formula is number of training cases. However, using knowledge from previous experimental studies features from 5 to 8 are considered as one set. So the combination set reduces from 9 to 6 and so number of combinations reduces from 511 to 63.

CI forecasting models have specific external parameters that need to be adjusted properly during training phase. In order to statistically minimize CI forecasting model overfitting and underfitting during training phase, a 10-fold cross-validation procedure is used with every forecasting model that needs parameter tuning. Tuned parameters during training phase are used for testing phase.

Results

This section presents forecasting accuracy results. Forecasting accuracy results averaged over all 7 forecasting models are presented in Table 1. The first row in the table represents ATM numbers according to their type, which is described in section 3. Second to fifth rows show average SMAPE (averaged over all model forecasts) for particular training type when inputs that yield best average SMAPE amongst all forecasting models and training types (averaged over all seven models and all four training types) are selected. Sixth row show those best average input numbers according to their type as described in section 4. Seventh row show best averaged over all models SMAPE when best input set for particular training type (not for all training types) is selected. Eighth row show those best input numbers for particular training type. Other rows show same kind of results for other training types. Forecasting results averaged for all forecasting models and sorted according to average of all forecasting models and training types for each particular input type (sorted 63 inputs) are depicted in Appendix. Further an explanation of forecasting results for each ATM is made:

- ATM number 1: worst accuracy is achieved with shortest training history (0.5 year) mostly for all training inputs (see figure in Appendix). Training with long history (2 years) doesn't improve average forecasting results and best results are achieved with 1.5 year training – a trade-off between overfitting because of too short history and overfitting because of learning patterns that do not reoccur in the future. Best average input set is weekly values of cash flow (1–4 weeks before, see Table 1). However week number wasn't selected as best input over all models, this shows that cash flow is non-stationary and recent history update of cash flow pattern improves forecasting even for strong weekly seasonality.

- ATM number 2: as for ATM 1, worst results are achieved with shortest history training. Now long history is more significant than for ATM 1 and best results are achieved with 2 year training (mostly over all input sets, see figure in Appendix). However best average input set doesn't include month number and day of month in best input set. Week number, 1–4 previous weeks cash flows are related to strong weekly seasonality and moving average (9th input) lets model more adaptively react to recent cash flow trend variations because of yearly seasonality (see Table 1).

▪ ATM number 3: results are not so affected by history period length as for ATM 2 case, but worst results are obtained using 0.5year training also. Month and day of the week numbers weren't included in best input set, this shows that no significant weekly or yearly seasonality that may affect forecasting took place. Other included inputs are related to monthly periodicity and help to forecast changing monthly pattern.

▪ ATM number 4: surprisingly as for yearly seasonality in ATM 2, month number inputs weren't included in best input set, even though this type of ATM has not so smooth transition of yearly pattern. This might be explained by more strong recent cash flow value impacts (moving average, cash flow day before, or 1–4 previous week cash flows) for sudden change in cash flow than yearly-related (month number, day of month) regression input effects even though yearly seasonality is strong, it has a quasi-periodic structure and values of last year only partially reoccur in current year on the same day. However as for ATM number 2, long history is more necessary in order to forecast more accurately.

▪ ATM number 5: interesting results are obtained for this kind of ATM. As expected, most accurate results for most of the inputs (see figure in Appendix) are achieved when shortest history (after sudden cash flow process structural break occurs) is used. This is because cash flow training data after structural break concludes relatively larger part of training examples than for longer period training data. So model adapts to recent break. But it is interesting that this affect is only partial: the results are worse for 1.5 year training, but for 2 year training accuracy increases. This is explained integral part of ATM cash flow (increasing trend) and at the beginning model learns small level cash flow relationship, which is similar to that after the structural break. As expected, for this kind of ATMs with structural break, inputs that encode recent past are more effective than regression inputs.

▪ ATM number 6: even though this ATM has structural break as ATM number 5, the structural break impact is less obvious. This might be related to the direction of structural break and break level difference. An explanation of forecasting results is more difficult: shortest history learning is not the best (but looking at figure in Appendix seems worst) because of overfitting (even though it contains history without structural break) because of too short history training. Long history contains structural break so this disturbs model training and overfittig also takes place. Day

of month input is included in best input set because of monthly seaso-
nality, and 1–4 previous week inputs add to both weekly and monthly
(because of 4 weeks) pattern forecasting. But moving average and pre-
vious day cash flow inputs were not included. This might be because of
too small structural break. However 1–4 previous week cash flow inputs
contain recent past information together encoding weekly and monthly
seasonality patterns.

- ATM number 7: because temporal structural break was relatively short,
2 year training still performed best. 1 year training performed worst
because full temporal break was included and concluded biggest part of
training data compared to 2 or 1.5 year training cases and this lead to
model overfitting. 0.5 year training didn't include temporal structural
break part, but didn't perform best because of too short training history.
Because of weekly seasonality, best input set included related inputs.

- ATM number 8: looking at figure in Appendix it is seen that using 2 year
history training is worst case for most input sets, so this means that if
ATM has growing tendency, forgetting the past is good, but too few tra-
ining values will not train the model with maximum accuracy, so proper
decision for model training is needed because of short and long history
training trade-off. Moving average input is selected to best input set and
helps when with trend forecasting, weekly-related inputs correspond to
weekly seasonality factors in ATM cash flow process.

Table 1. Forecasting results (in SMAPE, %) over all forecasting models for every ATM

ATM Number	1	2	3	4	5	6	7	8
2 Year/800 Days Training (Best Avg. Inputs)	40.64	59.84	66.30	55.00	83.27	39.30	50.94	43.04
1.5 Year/400 Days Training (Best Avg. Inputs)	39.88	61.53	67.02	56.70	85.38	40.21	51.16	43.09
1 Year/200 Days Training (Best Avg. Inputs)	40.55	64.66	70.05	57.35	80.89	39.76	51.80	42.60
0.5 Year/100 Days Training (Best Avg. Inputs)	40.95	67,07	69.96	57.27	82.48	42.61	51.24	43.26
Best Avg. Inputs	5 6 7 8	3 5 6 7 8 9	2 4 5 6 7 8 9	3 4 9	5 6 7 8 9	2 3 5 6 7 8	3 5 6 7 8	3 5 6 7 8 9
2 Year/800 Days Training (Best 2 Year/800 Days Inputs)	40.64	59.84	66.30	55.00	82.40	39.30	50.71	43.04

ATM Number	1	2	3	4	5	6	7	8
Best 2 Year/800 Days Inputs	5 6 7 8	3 5 6 7 8 9	2 4 5 6 7 8 9	3 4 9	9	2 3 5 6 7 8	2 3 5 6 7 8	3 5 6 7 8 9
1.5 Year/400 Days Training (Best 1.5 Year/400 Days Inputs)	39.42	61.53	66.92	55.71	83.31	40.21	51.16	43.09
Best 1.5 Year/400 Days Inputs	3 5 6 7 8	3 5 6 7 8 9	2 4 5 6 7 8	3 9	9	2 3 5 6 7 8	3 5 6 7 8	3 5 6 7 8 9
1 Year/200 Days Training (Best 1 Year/200 Days Inputs)	40.55	64.66	67.24	56.72	80.89	39.76	51.74	42.60
Best 1 Year/200 Days Inputs	5 6 7 8	3 5 6 7 8 9	1 2 4 5 6 7 8	1 3 4 5 6 7 8 9	5 6 7 8 9	2 3 5 6 7 8	2 3 4 5 6 7 8	3 5 6 7 8 9
0.5 Year/100 Days Training (Best 0.5 Year/100 Days Inputs)	40.95	66.88	67.17	57.27	75.45	41.48	51.08	42.52
Best 0.5 Year/100 Days Inputs	5 6 7 8	3 4 5 6 7 8 9	2 3 4	3 4 9	1 2 3	2 4 5 6 7 8	2 3 5 6 7 8	3 4 5 6 7 8

S o u r c e : own studies.

■■■ CONCLUSIONS AND FUTURE WORKS

After obtaining forecasting results for various types of ATM cash flows two important conclusions can be made:

- Choosing model training history is very important. Using long history is useful if yearly seasonality factors are relatively important in ATM cash flow process. If cash flow is relatively stationary (as ATM number 1 and ATM number 3) and has strong monthly or weekly seasonality, using too much history will not increase forecasting accuracy. But if ATM cash flow process has structural breaks or various trend components, training history length must be selected carefully in order to avoid overfitting.
- Proper input selection is even more important than training history period selection. Time-series inputs such as previous day cash flow value, moving average and 1–4 previous week cash flow values proved to be more efficient input features than regression inputs and contain time-varying information with recent history encoding. So calendar effect forecasting inputs are not as important as time-series feature inputs. Purely regression inputs are effective only when stationarity of cash flow could be assumed. Combination of both may increase forecasting accuracy.

In future works an investigation of multiple-day-ahead forecasting will be performed, which is more practical for cash flow forecasting. However, multiple-day-forecasting is far more challenging research object with CI models, because of various factors related to error accumulation, conditional probability estimation. Also, automatic statistical methods that would be useful for ATM type identification for input and history period selection will be investigated.

▮▮▮ References

Brentnall, A.R., Crowder, M.J., & Hand, D.J. (2008). A statistical model for the temporal pattern of individual automated teller machine withdrawals. Journal of the Royal Statistical Society: Series C (Applied Statistics). 57(1). 43–59. http://dx.doi.org/10.1111/j.1467-9876.2007.00599.x.

Brentnall, A.R., Crowder, M.J., & Hand, D.J. (2010a). Predicting the amount individuals withdraw at cash machines using a random effects multinomial model. Statistical Modeling. 10(2). 197–214. http://dx.doi.org/10.1177/1471082X0801000205.

Brentnall, A.R., Crowder, M.J., & Hand, D.J. (2010b). Predictive-sequential forecasting system developement for cash machine stocking. International Journal of Forecasting. 26. 764–776.

Chih-Chung, C., & Chih-Jen, L. (2011). LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2(27). 1–27. http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed: 29.03.2015).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning. 20(3). 273––297. http://dx.doi.org/10.1007/BF00994018.

Darwish, S.M. (2013). A methodology to improve cash demand forecasting for ATM network. International Journal of Computer and Electrical Engineering. 5(4). 405–409. http://dx.doi.org/10.7763/IJCEE.2013.V5.741.

Ekinci, Y., Lu, J.C., & Duman, E. (2015). Optimization of ATM cash replenishment with group-demand forecasts. Expert Systems with Applications. 42(7). 3480–3490. http://dx.doi.org/10.1016/j.eswa.2014.12.011.

Gurgul, H., & Suder, M. (2013). Modeling of withdrawals from selected ATMs of the Euronet network. AGH Managerial Economics. 13. 65–82. http://dx.doi.org/10.7494/manage.2013.13.65.

Hagan, M.T., & Menhaj, M. (1994). Training feed-forward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks. 5(6). 989–993.

Huang, G.B., Zhu, Q.Y., & Siew, C.K. (2006). Extreme learning machine: theory and applications. Neurocomputing. 70. 489–501. http://www.ntu.edu.sg/home/egbhuang/elm_codes.html (accessed: 29.03.2015).

Jang, J.S.R. (1993). ANFIS: adaptive-network-based fuzzy inference systems. IEEE Transactions on Systems, Man, and Cybernetics. 23(3). 665–685. http://dx.doi.org/10.1109/21.256541.

Kumar, P., & Walia, E. (2006). Cash forecasting: An application of artificial neural networks in finance. International Journal of Computer Science & Applications. 3(1). 61–77.

Laukaitis, A. (2008). Functional data analysis for cash flow and transactions intensity continuous-time prediction using Hilbert-valued autoregressive processes. European Journal of Operational Research. 185(3). 1607–1614. http://dx.doi.org/10.1016/j.ejor.2006.08.030.

Pelckmans, K., Suykens, J.A.K., Van Gestel, T., De Brabanter, J., Lukas, L., Hamers, B., De Moor, B., & Vandewalle, J. (2002). LS-SVMlab : a Matlab/C toolbox for Least Squares Support Vector Machines. Internal Report 02-44, ESAT-SISTA, KU Leuven, Leuven, Belgium, viewed 14 May 2015. http://www.esat.kuleuven.be/sista/lssvmlab/old/lssvmlab_paper0.pdf (accessed: 29.03.2015).

Rodrigues, P., & Esteves, P. (2010). Calendar effects in daily ATM withdrawals. Economics Bulletin. 30(4). 2587–2597.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagation errors. Nature. 323. 533–536.

Simutis, R., Dilijonas, D., & Bastina, L. (2008). Cash demand forecasting for ATM using neural networks and support vector regression algorithms. Proceedings of the twentieth EURO mini conference on continuous optimization and knowledge-based technologies (EurOPT-2008). Neringa. Lithuania. 416–421.

Simutis, R., Dilijonas, D., Bastina, L., & Friman, J. (2007). A flexible neural network for ATM cash demand forecasting. Proceedings of the sixth WSEAS international conference on computational intelligence, man-machine systems and cybernetics (CIMMACS 07). 162–165.

Specht, D.F. (1991). A general regression neural network. IEEE Transactions on Neural Networks. 2(6). 568–576.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). Least Squares Support Vector Machines. World Scientific. Singapore.

Tipping M.E. (2001). Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research. 1. 211–244. http://www.miketipping.com/sparsebayes.htm (accessed: 29.03.2015).

Wagner M. (2010). Forecasting daily demand in cash supply chain. American Journal of Economics and Business Administration. 2(4). 377–383. http://dx.doi.org/10.3844/ajebasp.2010.377.383.

APPENDIX

ATM Number 1 Cash Flow

Sample Autocorrelation Function

ATM Number 1 Forecasting Results

ATM Number 2 Cash Flow

Sample Autocorrelation Function

ATM Number 2 Forecasting Results

ATM Number 3 Cash Flow

Sample Autocorrelation Function

ATM Number 3 Forecasting Results

ATM Number 4 Cash Flow

Sample Autocorrelation Function

ATM Number 4 Forecasting Results

ATM Number 6 Cash Flow

Sample Autocorrelation Function

ATM Number 6 Forecasting Results

ATM Number 7 Cash Flow

Sample Autocorrelation Function

ATM Number 7 Forecasting Results

ATM Number 8 Cash Flow

Sample Autocorrelation Function

ATM Number 8 Forecasting Results

Legend:
- 2 year training
- 1.5 year training
- 1 year training
- 0.5 year training