ORIGINAL RESEARCH

A machine learning approach to univariate time series forecasting of quarterly earnings

Jan Alexander Fischer[1] · Philipp Pohl[2] ⓘ **· Dietmar Ratz[2]**

Abstract

We propose our quarterly earnings prediction (QEP_{SVR}) model, which is based on epsilon support vector regression (ε-SVR), as a new univariate model for quarterly earnings forecasting. This follows the recommendations of Lorek (Adv Account 30:315–321, 2014. https://doi.org/10.1016/j.adiac.2014.09.008), who notes that although the model developed by Brown and Rozeff (J Account Res 17:179–189, 1979) (BR ARIMA) is advocated as still being the premier univariate model, it may no longer be suitable for describing recent quarterly earnings series. We conduct empirical studies on recent data to compare the predictive accuracy of the QEP_{SVR} model to that of the BR ARIMA model under a multitude of conditions. Our results show that the predictive accuracy of the QEP_{SVR} model significantly exceeds that of the BR ARIMA model under 24 out of the 28 tested experiment conditions. Furthermore, significance is achieved under all conditions considering short forecast horizons or limited availability of historic data. We therefore advocate the use of the QEP_{SVR} model for firms performing short-term operational planning, for recently founded companies and for firms that have restructured their business model.

Keywords Quarterly earnings forecasting · ARIMA models · Support vector regression · Time-series regression · Machine learning

JEL Classification C22 · C32 · C51 · C52 · C53

✉ Philipp Pohl
pohl@dhbw-karlsruhe.de

Jan Alexander Fischer
fischja10@gmail.com

Dietmar Ratz
ratz@dhbw-karlsruhe.de

[1] University of Zurich, Rämistrasse 71, 8006 Zurich, Switzerland

[2] Baden-Wuerttemberg Cooperative State University Karlsruhe, Erzbergerstrasse 121, 76133 Karlsruhe, Germany

1 Introduction

The quarterly earnings reported by a company is an accounting figure of great significance. Quarterly earnings can be used to track performance in the context of management and debt contracts (Dechow et al. 1998), and are reflective of corporate governance (Chen et al. 2015). Isidro and Dias (2017) also show that earnings are strongly related to stock returns in volatile market conditions, while Zoubi et al. (2016) consider disaggregated earnings to better explain variation in stock returns. Furthermore, differences between forecasted and actual earnings have been used to calculate a firm's market premium (Dopuch et al. 2008).

The prediction of future quarterly earnings using univariate statistical models has been the subject of extensive research. Lorek and Willinger (2011) and Lorek (2014) claim that the autoregressive integrated moving average (ARIMA) model proposed by Brown and Rozeff (1979), denoted by BR ARIMA, is the premier univariate statistical model for the prediction of quarterly earnings. Its functional form is

$$Y_q = Y_{q-4} + \phi\left(Y_{q-1} - Y_{q-5}\right) + \epsilon_q - \theta\left(\epsilon_{q-4}\right),$$

where Y_q is the earnings of a company for a quarter q, ϕ is an autoregressive parameter, θ is a moving average parameter and ϵ_q is the disturbance term at q.

While the BR ARIMA model has been praised for its predictive accuracy on historic data, Lorek (2014) expresses concerns about its ability to describe more recent quarterly earnings. This is because research suggests that the time series properties of quarterly earnings have changed significantly since the development of the BR ARIMA model four decades ago. For instance, Baginski et al. (2003) note an apparent decline in the persistence of quarterly earnings from 1967 to 2001. This could be explained by the increased prevalence of high-tech firms (Kwon and Yin 2015). Lorek and Willinger (2008) also find that for an increasing number of companies, the quarterly earnings series no longer exhibit any significant seasonality. Furthermore, Klein and Marquardt (2006) highlight an increasing frequency of negative quarterly earnings, which according to Hayn (1995) disrupts autocorrelation patterns. Lorek (2014) therefore advocates research towards developing a new univariate model for the prediction of quarterly earnings.

In this paper we introduce a new model based on epsilon support vector regression (ε-SVR) (Smola and Schölkopf 2004; Vapnik 1995), termed the quarterly earnings prediction (QEP$_{SVR}$) model. We choose ε-SVR as a supervised learning algorithm because of (1) the guarantee of finding a globally optimal solution, (2) a sparse solution space, meaning only some training points contribute to the solution, and (3) the use of dot products to facilitate efficient computation of nonlinear solutions (Thissen et al. 2003). Furthermore, ε-SVR has been shown to generally perform well in many real-world applications (Hastie et al. 2009). Initial trials were also conducted using random forests (Breiman 2001), gradient boosting (Friedman 2001), and gaussian processes (Rasmussen and Williams 2006) as base supervised learners for the QEP$_{SVR}$ model. However, utilizing ε-SVR yielded the highest predictive accuracy.

Our QEP$_{SVR}$ model retains the univariate character of the BR ARIMA model, meaning all predictive features are derivable from historic quarterly earnings series. Unlike the BR ARIMA model however, the QEP$_{SVR}$ model is fitted using the historic quarterly earnings of multiple firms. The objective of this paper is to analyze and compare the predictive accuracy of the QEP$_{SVR}$ model to that of the state-of-the-art BR ARIMA model under a multitude of conditions.

The rest of the paper proceeds as follows. Section 2 familiarizes the reader with the basics of ε-SVR. Section 3 then describes the QEP$_{SVR}$ model in detail. This is followed by an explanation of the research method for comparing the QEP$_{SVR}$ model to the BR ARIMA model in Sect. 4. The experimental results are discussed in Sect. 5, while Sect. 6 offers concluding remarks.

2 Epsilon support vector regression (ε-SVR)

ε-SVR (Smola and Schölkopf 2004; Vapnik 1995) is an integral part of the QEP$_{SVR}$ model; its key ideas are introduced in this section. Consider training data of the form $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ is the input space (e.g. $\mathcal{X} = \mathbb{R}^d$), i.e. $x_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$. The goal of ε-SVR regression is to fit a model, $f(x)$, to the training data, such that (1) the maximum deviation of each training point y_i from $f(x_i)$ is ε, and (2) $f(x)$ is as flat as possible. The procedure for achieving this is introduced for cases where $f(x)$ is linear first, followed by an extension to nonlinear models.

A linear function is stipulated as

$$f(x) = \langle w, x \rangle + b,$$

where $w \in \mathcal{X}, b \in \mathbb{R}$, and $\langle w, x \rangle$ denotes the dot product of w and x. For this function, flatness refers to w being small. ε-SVR fits a linear function to training data by solving the following convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned}$$

where $\|w\|^2 = \langle w, w \rangle$. Since satisfying these constraints may be infeasible, slack variables ξ_i and ξ_i^* are introduced:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ \text{subject to} \quad & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned}$$

Here, deviations only contribute to the total error when $\left| f(x_i) - y_i \right| > \varepsilon$. The trade-off between attaining a flat f and not allowing deviations greater than ε is controlled by the constant $C > 0$. However, the above optimization problem is usually solved in its dual formulation. The involves the construction of a Lagrange function, L:

$$\begin{aligned} L = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) - \sum_{i=1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=1}^{n} \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) - \sum_{i=1}^{n} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b), \end{aligned}$$

where the variables $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$ are Lagrange multipliers. To find the optimal solution, the partial derivatives of L with respect to the variables w, b, ξ_i and ξ_i^* are set to 0:

$$\partial_b L = \sum_{i=1}^{n} \left(\alpha_i^* - \alpha_i \right) = 0$$

$$\partial_w L = w - \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) x_i = 0$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0$$

The substitution of these equations into L gives the dual optimization problem, in which η_i and η_i^* vanish:

$$\text{maximize} \quad -\frac{1}{2} \sum_{i,j=1}^{n} \left(\alpha_i - \alpha_i^* \right) \left(\alpha_j - \alpha_j^* \right) \langle x_i, x_j \rangle - \varepsilon \sum_{i=1}^{n} \left(\alpha_i + \alpha_i^* \right) + \sum_{i=1}^{n} y_i \left(\alpha_i - \alpha_i^* \right)$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

The optimization problem can therefore be stipulated in terms of dot products between the training data. This observation is key for the subsequent extension of ε-SVR to nonlinear $f(x)$. Since rearranging $\partial_w L$ gives

$$w = \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) x_i,$$

it follows that $f(x)$ can be rewritten as a linear combination of the training inputs, x_i:

$$f(x) = \langle w, x \rangle + b = \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) \langle x_i, x \rangle + b.$$

ε-SVR is also capable of fitting nonlinear $f(x)$ to training data. This requires a mapping of the input space $\mathcal{X}$ to another space, $\mathcal{X}^*$. The dual optimization problem specified above could then be solved for the modified training data, $\left\{ \left(x_1^*, y_1 \right), \dots, \left(x_n^*, y_n \right) \right\} \subset \mathcal{X}^* \times \mathbb{R}$. However, computing dot products in $\mathcal{X}^*$ may be computationally infeasible. Instead, a kernel function K can be used to compute the dot products $\langle x_i^*, x_j^* \rangle$ specified in the optimization problem from $\mathcal{X}$ directly—avoiding the explicit transformation $\mathcal{X} \mapsto \mathcal{X}^*$. The dual optimization problem is thus rewritten as

$$\text{maximize} \quad -\frac{1}{2} \sum_{i,j=1}^{n} \left(\alpha_i - \alpha_i^* \right) \left(\alpha_j - \alpha_j^* \right) K(x_i, x_j) - \varepsilon \sum_{i=1}^{n} \left(\alpha_i + \alpha_i^* \right) + \sum_{i=1}^{n} y_i \left(\alpha_i - \alpha_i^* \right)$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) = 0 \\ \alpha_i, \alpha_i^* \in [0, C], \end{cases}$$

where flatness of $f(x)$ is maximized in $\mathcal{X}^*$ rather than in $\mathcal{X}$. The fitted nonlinear function is stipulated as follows:

$$f(x) = \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^* \right) K\left(x_i, x \right) + b.$$

Finally, the variable b can be estimated using the Karush–Kuhn–Tucker conditions (Karush 1939; Kuhn and Tucker 1951), which are not discussed further.

3 The QEP$_{\text{SVR}}$ model

The notation used in the description of the QEP$_{\text{SVR}}$ model is as follows. Firstly, quarterly earnings series are assigned relative integer indices. For example, if $Y_q | q \in \mathbb{Z}$ denotes the fourth quarter (Q4) earnings for 2015 in a series Y, then Y_{q-5} denotes the Q3 earnings for 2014. The unmodified quarterly earnings series are referred to as original (orig) series. However, differenced (diff) and quarterly differenced (qdiff) series are also considered in order to expose quarter-by-quarter and quarter-to-quarter earnings relationships, respectively (Lorek and Willinger 2011). Their derivation from orig series is shown below.

$$\text{diff}_q = \text{orig}_q - \text{orig}_{q-1}$$
$$\text{qdiff}_q = \text{orig}_q - \text{orig}_{q-4}.$$

This means that for a continuous orig series of length n, the derived diff and qdiff series have lengths $n-1$ and $n-4$, respectively.

At a high level, the QEP$_{\text{SVR}}$ model can be thought of as a sequence of data manipulation steps. Given a historic quarterly earnings series, the QEP$_{\text{SVR}}$ model first extracts several explanatory variables (features). Next, a heuristic is used to estimate the predictive power of each feature – only the top features are selected. The retained features are subsequently scaled to normalize their range. An ε-SVR model is then applied to the scaled features, yielding a (scaled) one-step-ahead quarterly earnings prediction. Finally, depending on the configuration of the ε-SVR model, this prediction is scaled back into the range of the original series. This gives the true one-step-ahead quarterly earnings prediction.

The BR ARIMA model and the QEP$_{\text{SVR}}$ model always return one-step-ahead predictions when evaluated. We therefore obtain multi-step-ahead predictions via a series of one-step-ahead predictions. For example, making a two-step-ahead prediction for a historic series of length $AvailableSeriesLength(ASL) \in \mathbb{N}^+$, $orig_{-ASL}, \dots, orig_{-1}$, requires (1) evaluating a model for $orig_{-ASL}, \dots, orig_{-1}$ to obtain the one-step-ahead prediction of $orig_0$, denoted by $orig_0^*$, then (2) evaluating the same model again for $orig_{-ASL+1}, \dots, orig_0^*$, yielding the desired two-step-ahead prediction of $orig_1$, denoted by $orig_1^*$. Note that the model parameters are the same in (1) and (2), i.e. the BR ARIMA and QEP$_{\text{SVR}}$ models are not refitted during multi-step-ahead predictions.

The QEP$_{\text{SVR}}$ model has two operations: fit and predict. Fitting is the process of estimating the QEP$_{\text{SVR}}$ model's parameters from a set of historic quarterly earnings series, while the predict operation uses the model parameters to make one-step-ahead quarterly earnings forecasts. We describe the QEP$_{\text{SVR}}$ model in the context of these two operations for the remainder of this section, hereby refining the high-level approach outlined in the previous paragraph.

We define both the fitting and prediction operations of the QEP_{SVR} model in pseudocode. The notation $i[c]$ refers to an item i in a collection c. Furthermore, the notation $x \leftarrow e$ denotes the assignment of the value of an expression e to a variable x. All referenced functions are prefixed by the name of the module they belong to. There are four modules, *Extraction*, *Selection*, *Scaling* and *SVR*, each of which encapsulates a specific type of data manipulation. While the predict operation is explained first, some functions and parameters referenced by both operations are explained more closely in the subsequent description of the fit operation.

operation predict($origSeriesCollection, modelParams$) **returns** *predictions*
  $X_1 \leftarrow$ Extraction.fTransform($origSeriesCollection$)
  $X_2 \leftarrow$ Selection.transform($X_1, selectedFeatures[modelParams]$)
  $X_3 \leftarrow$ Scaling.transform($X_2, scaleParamsX[modelParams]$)
  $Y_1 \leftarrow$ SVR.predict($X_3, svrParams[modelParams]$)
  $Y_2 \leftarrow$ Scaling.invTransform($Y_1, scaleParamsY[modelParams]$)
  $Y_3 \leftarrow$ Extraction.invTTransform($Y_2, X_1, targetVar[modelParams]$)
  return Y_3

The input *origSeriesCollection* of the predict operation is a set of m continuous *orig* series. All series in *origSeriesCollection* are of equal length $AvailableSeriesLength(ASL) \in \mathbb{N}^+$ and have the form $orig_{-ASL}, \ldots, orig_{-1}$. The first step of the predict operation is to extract the diff and qdiff series corresponding to each of the m *orig* series in the input. This is done via the *fTransform* function of the *Extraction* module, yielding an output matrix $X_1 \in \mathbb{R}^{m \times (3 \cdot ASL - 5)}$. Each of the m rows of X_1 contains the $3 \cdot ASL - 5$ elements of the *orig*, diff and qdiff series for a single input series. Since the elements of each row are aligned, the column indices of X_1 are the set $\{orig_{-ASL}, \ldots, orig_{-1}\} \cup \{\text{diff}_{-ASL+1}, \ldots, \text{diff}_{-1}\} \cup \{\text{qdiff}_{-ASL+4}, \ldots, \text{qdiff}_{-1}\}$. These indices are also referred to as features.

The next step is to remove columns from X_1 corresponding to features with low predictive power. This is done using the *transform* function of the *Selection* module. It requires a model parameter, *selectedFeatures*, that specifies the set of k features to retain. Note that all model parameters in the *modelParams* collection are determined using the QEP_{SVR} model's fit operation. Subsequently, each of the remaining columns in $X_2 \in \mathbb{R}^{m \times k}$ are scaled by the *transform* function of the *Scaling* module. This produces a scaled matrix, $X_3 \in \mathbb{R}^{m \times k}$. The scaling of each column is controlled by the model parameter *scaleParamsX*.

Next, the *predict* function of the *SVR* module uses the model parameter *svrParams* to estimate the value of a specific target variable for each row of X_3. The result, $Y_1 \in \mathbb{R}^{m \times 1}$, is then subject to inverse scaling by the *invTransform* function of the *Scaling* module to produce $Y_2 \in \mathbb{R}^{m \times 1}$. Note that for a given matrix X and constant scaling parameters P, the application of *Scaling.invTransform* to the output of *Scaling.transform*(X, P) (and vice versa) yields the original matrix, X.

The final step of the predict operation is the application of the *Extraction* module's *invTTransform* function to Y_2. This ensures that the output, $Y_3 \in \mathbb{R}^{m \times 1}$, consists of one-step-ahead predictions for the m input orig series in *origSeriesCollection*, since the *SVR.predict* function may have returned (scaled) predictions for the corresponding

diff or qdiff series instead. The *Extraction.invTTransform* is notified of this through the model parameter *targetVar*, and uses the previously determined $X_1 \in \mathbb{R}^{m \times (3 \cdot ASL - 5)}$ to compute the following:

$$Y_3 = \begin{cases} Y_2, & \text{if } targetVar = orig \\ X_1^{orig_{-1}} + Y_2, & \text{if } targetVar = \text{diff} \\ X_1^{orig_{-4}} + Y_2, & \text{if } targetVar = \text{qdiff}, \end{cases}$$

where the notation $X_1^{orig_{-q}}$ denotes the column of X_1 corresponding to the feature $orig_{-q}$.

The fit operation determines model parameters for a specified *orig* series length, *ASL*, for which one-step-ahead forecasts are to subsequently be made via the predict operation. Fitting is controlled by hyperparameters (*hyperParams*). As for the predict operation, the input *origSeriesCollection* is a set of continuous *orig* series. For the fit operation however, each series must have a length of at least $ASL + 1$ and lengths are not required to be equal.

```
operation fit(origSeriesCollection, ASL, hyperParams) returns modelParams
    X₁, Y₁ ← Extraction. windowTransform(origSeriesCollection, ASL)
    X₂ ← Extraction. fTransform(X₁)
    Y₂ ← Extraction. tTransform(X₂, Y₁, targetVar[hyperParams])
    selectedFeatures ← Selection. fit(X₂, Y₂, k[hyperParams])
    X₃ ← Selection. transform(X₂, selectedFeatures)
    scaleParamsX ← Scaling. fit(X₃, scaleTypeX[hyperParams])
    scaleParamsY ← Scaling. fit(Y₂, scaleTypeY[hyperParams])
    X₄ ← Scaling. transform(X₃, scaleParamsX)
    Y₃ ← Scaling. transform(Y₂, scaleParamsY)
    svrParams ← SVR. fit(X₄, Y₃, ε[hyperParams], C[hyperParams], γ[hyperParams])
    modelParams ← {selectedFeatures, scaleParamsX, scaleParamsY, svrParams, targetVar[hyperParams]}
    return modelParams
```

The first step of the fit operation is to decompose the series in *origSeriesCollection*. Conceptually, this is done by sliding a window of length $ASL + 1$ along each input series, one step of a time. For example, given $ASL = 6$, a single input series $orig_1 \ldots, orig_8$ would be split into two series: $orig_1, \ldots, orig_7$ and $orig_2, \ldots, orig_8$. Assume this yields a total of s series, each of the form $orig_{-ASL}, \ldots, orig_0$. Next, the first *ASL* elements of each of these series are assigned to a row of the matrix $X_1 \in \mathbb{R}^{s \times ASL}$, while the last element of each series is assigned to the corresponding row in $Y_1 \in \mathbb{R}^{s \times 1}$. This decomposition is performed by the *Extraction.windowTransform* function.

The *fTransform* function of the *Extraction* module also accepts *orig* series in matrix form (in addition to set form, as in the predict operation). It uses X_1 to create a feature matrix $X_2 \in \mathbb{R}^{s \times (3 \cdot ASL - 5)}$. The target matrix $Y_2 \in \mathbb{R}^{s \times 1}$ is computed by the *Extraction.tTransform* function using X_2, Y_1 and the hyperparameter *targetVar* as follows

$$Y_2 = \begin{cases} Y_1, & \text{if } targetVar = orig \\ Y_1 - X_2^{orig_{-1}}, & \text{if } targetVar = \text{diff} \\ Y_1 - X_2^{orig_{-4}}, & \text{if } targetVar = \text{qdiff}, \end{cases}$$

where $X_2^{orig_{-q}}$ denotes the column of X_2 corresponding to the feature $orig_{-q}$. Note that since *targetVar* is also required as a model parameter by the predict operation, it is added to the *modelParams* collection returned by the fit operation.

The next step is to choose the $k \in \mathbb{N}^+$ features in X_2 with the highest estimated explanatory power for predicting Y_2. This is performed by the *fit* function of the *Selection* module. A heuristic for explanatory power is the mutual information score (Cover and Thomas 1991) of a feature with the target variable. It is defined as

$$I(A;B) = \int\limits_{B} \int\limits_{A} P_{(A,B)}(a,b) \log_2 \frac{P_{(A,B)}(a,b)}{P_{(A)}(a)P_{(B)}(b)} da db,$$

where A and B are continuous random variables, $P_{(A,B)}$ denotes the joint probability density function of A and B, and $P_{(A)}$ and $P_{(B)}$ are the marginal probability density functions of A and B, respectively. Intuitively, mutual information describes the degree of uncertainty reduction in A through knowledge of B. A higher mutual information score therefore suggests greater explanatory power. Once the set of features *selectedFeatures* is determined, *Selection.transform* removes all unimportant features from X_2 to produce $X_3 \in \mathbb{R}^{s \times k}$.

Subsequently, the columns of X_3 and Y_2 are scaled. The type of scaling is determined by the *scaleTypeX* and *scaleTypeY* hyperparameters, respectively. These are either set to none, Gaussian or quantile Gaussian. The *fit* function of the *Scaling* module determines the scaling parameters *scaleParamsX* and *scaleParamsY*, which are specific to the chosen scaling type (*scaleParamsX* contains scaling parameters for each of the k columns of X_3). The *Scaling* module's *transform* function infers the scaling type from the scaling parameters and applies the scaling to produce $X_4 \in \mathbb{R}^{s \times k}$ and $Y_3 \in \mathbb{R}^{s \times 1}$.

If the scaling type is none, the scaling parameters are an empty set. Applying this type of scaling has no effect on the input matrix. In the case of Gaussian scaling, the scaling parameters are the sample mean and sample standard deviation of each column. The *transform* function of the *Scaling* module uses these parameters to replace each element of an input matrix X, with the z-score

$$z_{i,j} = \frac{x_{i,j} - \bar{x}_j}{s_j},$$

where $x_{i,j}$ is the element in the ith row and jth column of X, and $\bar{x}_j$ and s_j are the sample mean and sample standard deviation of the jth column of X, respectively. Conversely, calling the *Scaling.invTransform* function (referenced in the predict operation) with Gaussian scaling parameters has the effect of replacing each element $x_{i,j}$ with $s_j x_{i,j} + \bar{x}_j$.

Since Gaussian scaling depends on sample means, it is susceptible to outliers. Quantile Gaussian scaling uses a deterministic rank-based inverse normal transformation (Beasley et al. 2009), which is much more robust in the presence of outliers. The scaling parameters are a set of functions, $\{Q_j\}$, where $Q_j : [0, 1] \mapsto \mathbb{R}$ is a quantile function for the jth column of the matrix to be scaled. In this case, applying *Scaling.transform* replaces each element of an input matrix X, $x_{i,j}$, with $\Phi^{-1}(Q_j^{-1}(x_{i,j}))$, where Q_j^{-1} is the inverse quantile function for the jth column and Φ^{-1} is the inverse cumulative distribution function of a standard normal distribution. The application of *Scaling.invTransform* replaces each $x_{i,j}$ with $Q_j(\Phi(x_{i,j}))$.

After the completion of scaling, the *svrParams* model parameter is determined using the *fit* function of the *SVR* module. The *SVR* module performs ε-SVR, which was introduced in Sect. 2. In this case X_4 and Y_3 are considered as training data of the form $\left\{\left(X_4^{(1)}, Y_3^{(1)}\right), \ldots, \left(X_4^{(s)}, Y_3^{(s)}\right)\right\} \subset \mathbb{R}^k \times \mathbb{R}$, where the notation $X_4^{(i)}$ denotes the ith row of X_4. ε-SVR fits a model, $f : \mathbb{R}^k \mapsto \mathbb{R}$, to this training data, i.e. it learns the mapping from each row of X_4 to the corresponding element of Y_3. In the context of the QEP$_{\text{SVR}}$ model, f has the functional form

$$f(X) = \sum_{i=1}^{s} \left(\alpha_i - \alpha_i^* \right) K\left(X_4^{(i)}, X \right) + b,$$

where $X \in \mathbb{R}^k$, α_i and α_i^* are Lagrange multipliers associated with the ith training point and b is a constant. The kernel function K is chosen to be the squared exponential (Rasmussen and Williams 2006), stipulated as

$$K\left(X, X' \right) = e^{-\gamma \|X - X'\|^2},$$

where $X, X' \in \mathbb{R}^k$, $\|X - X'\|$ denotes the Euclidean norm of $X - X'$, and the hyperparameter $\gamma \in \mathbb{R}$ is included in the *hyperParams* collection provided as an input to the fit operation. As specified in Sect. 2, the value of b can be estimated using the Karush–Kuhn–Tucker conditions (Karush 1939; Kuhn and Tucker 1951), while the Lagrange multipliers are found by solving the reformulated dual optimization problem:

$$maximize \quad -\frac{1}{2} \sum_{i,j=1}^{s} \left(\alpha_i - \alpha_i^* \right)\left(\alpha_j - \alpha_j^* \right) K\left(X_4^{(i)}, X_4^{(j)} \right) - \varepsilon \sum_{i=1}^{s} \left(\alpha_i + \alpha_i^* \right) + \sum_{i=1}^{s} Y_3^{(i)}\left(\alpha_i - \alpha_i^* \right)$$

$$subject\ to \quad \begin{cases} \sum_{i=1}^{s} \left(\alpha_i - \alpha_i^* \right) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

Recall that the constant $C > 0$ controls the trade-off between attaining a flat f and not allowing deviations greater than ε. Both ε and C are hyperparameters that must be passed to the fit operation. The set of Lagrange multipliers $\left\{ \left(\alpha_i, \alpha_i^* \right) | i = 1, 2, \dots s \right\}$, the training points $\{ X_4^{(i)} | i = 1, 2, \dots s \}$ and the constant b are assigned to the collection *svrParams*.

Finally, *selectedFeatures*, *scaleParamsX*, *scaleParamsY*, *svrParams* as well as the hyperparameter *targetVar* are returned by the fit operation as the set of model parameters required by the predict operation of the QEP$_{SVR}$ model.

4 Research method

At any point during their practical application, the BR ARIMA model and the QEP$_{SVR}$ model are in one of two phases: development or operation. When in operation, a model can be evaluated. Evaluating a model means passing it a continuous historic *orig* series, $orig_{-ASL}, \dots, orig_{-1}$, for which it returns a one-step-ahead prediction of $orig_0$, denoted by $orig_0^*$. As mentioned previously, both models consider an iterative approach to making multi-step-ahead predictions, during which a model is not refitted. Model parameters therefore cannot be modified during operation.

The development phase consists of any activities that prepare a model for operation. In the case of the BR ARIMA model, the values of ϕ (the autoregressive parameter) and θ (the moving average parameter) are firm-specific. This means that they are estimated from the single *orig* series for which predictions are to be made (Lorek and Willinger 2011). Once ϕ and θ are determined, the BR ARIMA model is immediately evaluated to obtain predictions. Conversely, the hyperparameters and model parameters of the QEP$_{SVR}$ model are estimated from the historic data of a collection of firms. The model can then be evaluated for multiple firms before entering another development phase.

Table 1 Distribution of quarterly earnings and book value of total assets

	Mean	1st quartile	Median	3rd quartile
Quarterly earnings	104.1	5.3	20.3	62.0
Yearly book value of total assets	34,785.8	941.9	2288.5	7792.8

Table 2 Industries of companies in the experiment data

ISIC section	Number of companies
Manufacturing	51
Information and communication	16
Financial and insurance activities	13
Real estate activities	8
Wholesale and retail trade	8
Transportation and storage	6
Professional, scientific and technical activities	5
Other (sections with less than 5 companies)	10
Total	117

The predictive accuracy of the QEP_{SVR} model is compared to that of the BR ARIMA model under different experiment conditions. A condition is described by two variables: available series length (*ASL*) and prediction steps (*PS*). *ASL* is the length of the series for which predictions are made during operation, while *PS* specifies how many quarters into the future earnings should be predicted (i.e. given $orig_{-ASL}, \ldots, orig_{-1}$, the element $orig_{PS-1}$ is to be predicted). A total of 28 experiment conditions, $(ASL, PS) \in \{6, \ldots, 12\} \times \{1, \ldots, 4\}$, are considered. The chosen measure of predictive accuracy is mean absolute percentage error (MAPE) (Makridakis et al. 1982). It is defined as

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right|,$$

where *n* is the number of predictions, and *F* and *A* denote the forecasted and actual quarterly earnings. As in Lorek and Willinger (2011), prediction errors exceeding 100 percent are truncated to 100 percent. This is done to avoid the effects of explosive prediction errors.

The data used for the experiment consists of the *orig* quarterly earnings series of 117 companies across the German DAX, MDAX, SDAX and TecDAX stock market indices.[1] Each series contains 24 consecutive quarterly earnings from Q1 2012 to Q4 2017, giving a total of 2808 quarterly earnings. Table 1 summarizes the distribution of quarterly earnings and yearly book value of total assets of the companies in the experiment data. All values are in millions of Euros, to the nearest hundred thousand.

The main industries of companies in the experiment data are highlighted in Table 2. Industry classification is performed according to the International Standard Industrial Classification of All Economic Activities (ISIC) system (United Nations, 2008).

The MAPE values of the QEP_{SVR} model are calculated for all considered experiment conditions by executing *GetQEPErrors*. This operation implements tenfold cross-validation. The input *S* is therefore a set of 10 disjoint subsets (folds), $\{S_1, \ldots, S_{10}\}$, of the 117

---

[1] All data is obtained from publicly available sources. The data and code for the experiments is accessible at the following GitHub repository: https://github.com/fischja/quarterly-earnings-predictions.
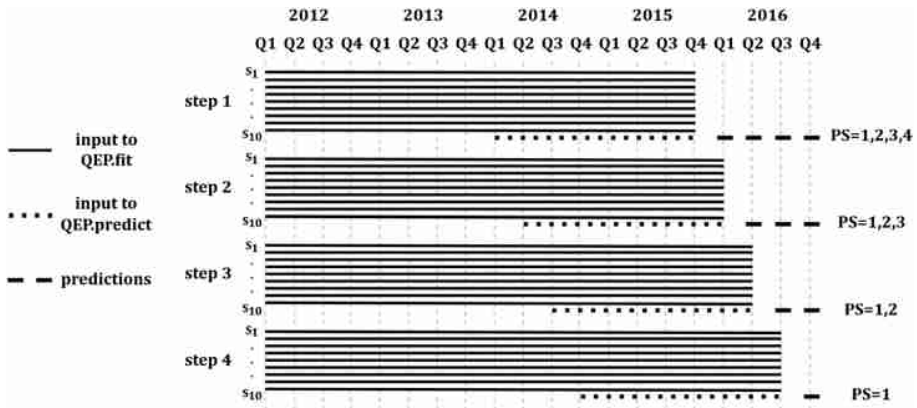
Fig. 1 How the *GetQepErrors* operation makes predictions

series in the experiment data. Each fold is of roughly equal size. The superscript notation $(a : b)$ slices all series in a set by removing earnings at quarters before a and after b. All MAPE values are calculated for predictions of quarterly earnings in *predYear* only.

operation $GetQEPErrors(S, predYear, hyperParams)$ **returns** $errs$

> $errs$ = error collection
> **for** ASL **in** $\{6, \dots, 12\}$
> > **for** $fold$ **in** $\{1, \dots, 10\}$
> > > **for** j **in** $\{0,1,2,3\}$
> > > > $firstFitQuarter$ = Q1 2012
> > > > $lastFitQuarter$ = j quarters after Q4 $(predYear - 1)$
> > > > $train = \{S \setminus S_{fold}\}^{(firstFitQuarter : lastFitQuarter)}$
> > > > $modelParams$ = QEP.fit($train, ASL, hyperParams$)
> > > > **for** k **in** $\{1, \dots, 4 - j\}$
> > > > > $firstValQuarter = (ASL - k)$ quarters before $lastFitQuarter$
> > > > > $lastValQuarter = (ASL - 1)$ quarters after $firstValQuarter$
> > > > > $val = S_{fold}^{(firstValQuarter : lastValQuarter)}$
> > > > > $predictions$ = QEP.predict($val, modelParams$)
> > > > > calculate MAPE values using $predictions$, add to $errs$
> > > > > use $predictions$ to temporarily update S_{fold}
> **return** $errs$

Figure 1 illustrates four steps of how *GetQepErrors* makes predictions for *predYear* = 2016 at $ASL = 8$ and *fold* = 10, assuming each fold S_i were to consist of a single series. In the first step, one to four-step-ahead predictions are made for S_{10}, while only a one-step-ahead prediction can be made for S_{10} in the fourth step.

The hyperparameters required by the fit operation of the QEP_{SVR} model are determined once for all experiment conditions. This is done by minimizing the MAPE values for predictions of quarterly earnings in 2016:

$$optimalHyperParams \leftarrow \underset{hyperParams}{\text{argmin}}\ GetQepErrors(S, 2016, hyperParams).$$

The set of optimal (lowest) MAPE values found during hyperparameter optimization are referred to as validation errors:

$$validationErrs \leftarrow GetQepErrors(S, 2016, optimalHyperParams)$$

However, a more accurate estimate of the QEP_{SVR} model's predictive accuracy is obtained by using the optimized hyperparameters to make predictions for quarterly earnings in 2017. This is because the data for 2017 is not observed during hyperparameter optimization. These errors are referred to as testing errors and are obtained as follows:

$$testingErrs \leftarrow GetQepErrors(S, 2017, optimalHyperParams)$$

The MAPE values for the BR ARIMA model are also calculated for predictions of quarterly earnings in 2016 and 2017. These are obtained by calling *GetBrArimaErrors*(*experimentData*, 2016) and *GetBrArimaErrors*(*experimentData*, 2017), where *experimentData* denotes the set of all 117 *orig* series. The *BR.fit* function determines the values of ϕ and θ by minimizing the sum of squared disturbance terms.

operation $GetBrArimaErrors(experimentData, predYear)$ **returns** $errs$
    $errs$ = error collection
    for ASL **in** $\{6, \dots, 12\}$
        for $series$ **in** $experimentData$
            for j **in** $\{0,1,2,3\}$
                $lastFitQuarter = j$ quarters after Q4 $(predYear - 1)$
                $firstFitQuarter = (ASL - 1)$ quarters before $lastFitQuarter$
                $train = series^{(firstFitQuarter : lastFitQuarter)}$
                $modelParams = BR.fit(train)$
                for k **in** $\{0, \dots, 3 - j\}$
                    $firstValQuarter = k$ quarters after $firstFitQuarter$
                    $lastValQuarter = k$ quarters after $lastFitQuarter$
                    $val = series^{(firstValQuarter : lastValQuarter)}$
                    $predictions = BR.predict(val)$
                    calculate MAPE values using $predictions$, add to $errs$
                    use $predictions$ to temporarily update $series$
    return $errs$

MAPE values are compared on a per-company basis. This means that the errors returned by *GetBrArimaErrors* and *GetQepErrors* are aggregated such that there are 117 MAPE values for the BR ARIMA model and the QEP_{SVR} model under each of the 28 experiment conditions. Hypothesis testing is then performed to asses if the 117 MAPE values

Table 3 Optimal hyperparameter values

Hyperparameter	Optimal value
targetVar	qdiff
k	4
scaleTypeX	Quantile Gaussian
scaleTypeY	Quantile Gaussian
ε	0.04
C	0.2
γ	0.25

calculated for the QEP_{SVR} model are significantly lower than those calculated for the BR ARIMA under each condition, i.e. if the QEP_{SVR} model has a significantly higher predictive accuracy than the BR ARIMA model. Since this involves paired samples, the paired t test (Kim 2015) and the Wilcoxon signed-rank test (Wilcoxon 1945) are used. In both cases, the one-tailed test is considered.

For a given condition, let μ_{BR} and μ_{QEP} denote the mean of the 117 MAPE values calculated for the BR ARIMA and QEP_{SVR} models, respectively. Similarly, let M_{BR} and M_{QEP} denote the median of the 117 MAPE values for each model. The null and alternative hypotheses of the paired t-test, H_{T_0} and H_{T_1}, are stated as

$$H_{T_0} : \mu_{QEP} \geq \mu_{BR} \quad H_{T_1} : \mu_{QEP} < \mu_{BR},$$

while the null and alternative hypotheses of the Wilcoxon signed-rank test, H_{W_0} and H_{W_1}, are

$$H_{W_0} : M_{QEP} \geq M_{BR} \quad H_{W_1} : M_{QEP} < M_{BR}.$$

5 Results

The hyperparameter values that minimize the validation errors are shown in Table 3. Table 4 shows the results of comparing the validation and testing errors of the QEP_{SVR} model to the corresponding prediction errors of the BR ARIMA model under each of the 28 experiment conditions.

Table 4 shows that 51 out of the 56 p-values calculated using the paired t-test lie below a significance level of 0.05, while 52 out of the 56 p-values calculated using the Wilcoxon signed-rank are below 0.05. We assume statistical significance under a condition if the p-values for a statistical test lie below 0.05 for predictions in both 2016 and 2017. The results in Table 4 therefore provide evidence for the rejection of H_{T_0} and H_{W_0} in favor of H_{T_1} and H_{W_1}, respectively, under 24 of the 28 experiment conditions. The four conditions, (*ASL*, *PS*), with insufficient evidence for rejection are (9, 3), (9, 4), (11, 4) and (12, 4).

The 24 significant conditions include all those where $PS \in \{1, 2\}$ (i.e. having short forecast horizons). This leads to the first result: The predictive accuracy of the QEP_{SVR} model significantly exceeds that of the BR ARIMA model for short forecast horizons. This means the QEP_{SVR} model is particularly suitable for companies considering short-term operational planning.

Table 4 Results of comparing the predictive accuracy of the QEP_{SVR} model to that of the BR ARIMA model

| Conditions | | Validation (2016) | | | | Testing (2017) | | | |
| ASL | PS | p value | | Mean MAPE | | p value | | Mean MAPE | |
		t test	Wilcoxon	QEP_{SVR}	BR ARIMA	t test	Wilcoxon	QEP_{SVR}	BR ARIMA
6	1	0.000	0.000	0.447	0.587	0.000	0.000	0.438	0.594
6	2	0.000	0.000	0.440	0.615	0.000	0.000	0.453	0.617
6	3	0.000	0.000	0.460	0.665	0.000	0.000	0.485	0.672
6	4	0.000	0.000	0.477	0.653	0.000	0.000	0.535	0.725
7	1	0.000	0.000	0.443	0.514	0.000	0.000	0.438	0.530
7	2	0.000	0.000	0.439	0.526	0.000	0.000	0.454	0.527
7	3	0.000	0.000	0.458	0.548	0.000	0.000	0.487	0.583
7	4	0.000	0.000	0.485	0.591	0.011	0.005	0.540	0.608
8	1	0.000	0.000	0.434	0.508	0.000	0.000	0.429	0.500
8	2	0.000	0.000	0.423	0.491	0.000	0.000	0.441	0.512
8	3	0.000	0.000	0.445	0.523	0.000	0.000	0.473	0.559
8	4	0.002	0.001	0.486	0.571	0.008	0.004	0.536	0.597
9	1	0.005	0.014	0.442	0.472	0.001	0.000	0.429	0.468
9	2	0.013	0.028	0.429	0.462	0.019	0.023	0.437	0.468
9	3	0.079	0.105	0.451	0.476	0.005	0.002	0.469	0.514
9	4	0.158	0.097	0.494	0.522	0.088	0.018	0.543	0.567
10	1	0.000	0.000	0.433	0.575	0.000	0.000	0.430	0.524
10	2	0.000	0.000	0.444	0.587	0.000	0.000	0.440	0.554
10	3	0.000	0.000	0.483	0.566	0.000	0.000	0.465	0.581
10	4	0.001	0.000	0.439	0.568	0.045	0.045	0.546	0.586
11	1	0.000	0.000	0.434	0.549	0.000	0.000	0.425	0.506
11	2	0.000	0.000	0.444	0.532	0.000	0.000	0.434	0.517
11	3	0.000	0.000	0.483	0.556	0.001	0.000	0.468	0.530
11	4	0.000	0.000	0.438	0.575	0.055	0.056	0.540	0.577
12	1	0.000	0.000	0.438	0.507	0.000	0.000	0.431	0.491
12	2	0.000	0.001	0.439	0.507	0.000	0.000	0.438	0.503
12	3	0.000	0.000	0.455	0.539	0.001	0.000	0.471	0.535
12	4	0.018	0.001	0.481	0.539	0.095	0.106	0.543	0.575

Furthermore, the 24 significant conditions also include all those where $ASL \in \{6, 7, 8\}$ (i.e. having limited historic data availability). The second result is thus stated as follows: The predictive accuracy of the QEP_{SVR} model significantly exceeds that of the BR ARIMA model when only limited historic quarterly earnings data is available. Situations of limited data availability arise in start-ups, as well as in companies that have recently made structural changes to their business model.

The selection of predictive features by the QEP_{SVR} model is also investigated. As shown in Table 3, the optimal value for the hyperparameter k is found to be 4. This means the fit operation of the QEP_{SVR} model selects the four predictive features which have the highest mutual information score with the chosen target variable ($qdiff_0$). Table 5 shows

Table 5 Feature selection probabilities

ASL	Selection probability						
	orig_{-4}	diff_{-3}	diff_{-4}	qdiff_{-4}	qdiff_{-1}	orig_{-2}	orig_{-1}
6	1.00	1.00	0.91	0.00	0.58	0.43	0.09
7	1.00	1.00	0.95	0.00	0.44	0.40	0.16
8	1.00	1.00	0.83	1.00	0.05	0.03	0.10
9	0.99	1.00	0.65	1.00	0.20	0.00	0.16
10	0.96	0.95	0.84	1.00	0.00	0.00	0.00
11	0.98	0.95	0.83	0.96	0.00	0.05	0.06
12	0.98	0.96	0.80	0.88	0.00	0.08	0.20
Mean	**0.99**	**0.98**	**0.83**	**0.69**	**0.18**	**0.14**	**0.11**

the probability of features being selected (i.e. a probability of 1 implies that a feature is always selected) across all predictions of quarterly earnings in 2016 and 2017. Features are arranged from left to right in descending order of mean selection probability across all *ASL* values.

The features orig_{-4} and diff_{-3} have the highest mean selection probabilities and have a selection probability of at least 0.95 across all conditions. After becoming available for selection at $ASL = 8$, the feature qdiff_{-4} is selected with a probability of at least 0.88 under all conditions with $ASL \geq 8$. The feature qdiff_{-1} is never selected for $ASL \geq 10$. Only 7 different features have selection probabilities exceeding 0.1, despite the pool of available features increasing from 13 (when $ASL = 6$) to 31 (when $ASL = 12$). The fact that 6 out of these 7 features are already available for selection at $ASL = 6$ suggests that increasing *ASL* may have little effect on the predictive accuracy of the $\mathrm{QEP}_{\mathrm{SVR}}$ model.

6 Conclusion

Following the recommendations of Lorek (2014), we propose our $\mathrm{QEP}_{\mathrm{SVR}}$ model as a new univariate statistical model for the prediction of quarterly earnings. Empirical evidence shows that under 24 out of 28 tested conditions, the predictive accuracy of the $\mathrm{QEP}_{\mathrm{SVR}}$ model is significantly higher than that of the state-of-the-art BR ARIMA model. Furthermore, the significant conditions include all those considering one and two-step-ahead predictions ($PS \in \{1, 2\}$), as well as those for which only limited historic data is available ($ASL \in \{6, 7, 8\}$). The experimental results therefore advocate using the $\mathrm{QEP}_{\mathrm{SVR}}$ model instead of the BR ARIMA model for short-term operational planning, for recently founded companies and for companies that have recently made fundamental changes to their business model.

Since the hyperparameters and model parameters of the $\mathrm{QEP}_{\mathrm{SVR}}$ model are determined from the historic data of multiple companies, further research is needed to understand how the choice of these companies affects predictive accuracy. Factors of interest include the industry, size, and diversity of companies, as well as their relationship to those companies for which predictions are to be made. Other areas of research include studying the effect of condition-specific hyperparameter optimization and exploring methods for combining the $\mathrm{QEP}_{\mathrm{SVR}}$ model with other forecasting methods. As an example, predictions of the $\mathrm{QEP}_{\mathrm{SVR}}$

model could be combined with analysts' forecasts, building upon the research of Elgers et al. (2016).

References

Baginski SP, Branson BC, Lorek KS, Willinger GL (2003) A time-series approach to measuring the decline in quarterly earnings persistence. Adv Account 20:23–42. https://doi.org/10.1016/S0882-6110(03)20002-X

Beasley TM, Erickson S, Allison DB (2009) Rank-based inverse normal transformations are increasingly used, but are they merited? Behav Genet 39:580–595. https://doi.org/10.1007/s10519-009-9281-0

Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324

Brown LD, Rozeff MS (1979) Univariate time-series models of quarterly accounting earnings per share: a proposed model. J Account Res 17:179–189

Chen Y-C, Lee C-H, Chou P-I (2015) Stock-based compensation and earnings management behaviors. Rev Pac Basin Finan Mark Pol 18:1550008. https://doi.org/10.1142/S0219091515500083

Cover TM, Thomas JA (1991) Elements of information theory. Wiley series in telecommunications. Wiley, New York

Dechow PM, Kothari SP, Watts RL (1998) The relation between earnings and cash flows. J Account Econ 25:133–168. https://doi.org/10.1016/S0165-4101(98)00020-2

Dopuch N, Seethamraju C, Xu W (2008) An empirical assessment of the premium associated with meeting or beating both time-series earnings expectations and analysts' forecasts. Rev Quant Finan Acc 31:147–166. https://doi.org/10.1007/s11156-007-0075-2

Elgers PT, Lo MH, Xie W, Le Xu E (2016) A contextual evaluation of composite forecasts of annual earnings. Rev Pac Basin Finan Mark Pol 19:1650014. https://doi.org/10.1142/S0219091516500144

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Statist 29:1189–1232. https://doi.org/10.1214/aos/1013203451

Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: Data mining, inference, and prediction, 2nd edn. Springer series in statistics. Springer, New York

Hayn C (1995) The information content of losses. J Account Econ 20:125–153. https://doi.org/10.1016/0165-4101(95)00397-2

Isidro H, Dias JG (2017) Earnings quality and the heterogeneous relation between earnings and stock returns. Rev Quant Finan Acc 49:1143–1165. https://doi.org/10.1007/s11156-017-0619-z

Karush W (1939) Minima of functions of several variables with inequalities as side constraints. Master's thesis, University of Chicago

Kim TK (2015) T test as a parametric statistic. Korean J Anesthesiol 68:540–546. https://doi.org/10.4097/kjae.2015.68.6.540

Klein A, Marquardt CA (2006) Fundamentals of accounting losses. Account Rev 81:179–206. https://doi.org/10.2308/accr.2006.81.1.179

Kuhn HW, Tucker AW (1951) Nonlinear programming. In: The Regents of the University of California

Kwon SS, Yin J (2015) A comparison of earnings persistence in high-tech and non-high-tech firms. Rev Quant Finan Acc 44:645–668. https://doi.org/10.1007/s11156-013-0421-5

Lorek KS (2014) A critical assessment of the time-series literature in accounting pertaining to quarterly accounting numbers. Adv Account 30:315–321. https://doi.org/10.1016/j.adiac.2014.09.008

Lorek KS, Willinger GL (2008) Time-series properties and predictive ability of quarterly cash flows. Adv Account 24:65–71. https://doi.org/10.1016/j.adiac.2008.05.010

Lorek KS, Willinger GL (2011) Multi-step-ahead quarterly cash-flow prediction models. Account Horizons 25:71–86. https://doi.org/10.2308/acch.2011.25.1.71

Makridakis S, Andersen A, Carbone R, Fildes R, Hibon M, Lewandowski R, Newton J, Parzen E, Winkler R (1982) The accuracy of extrapolation (time series) methods: results of a forecasting competition. J Forecast 1:111–153. https://doi.org/10.1002/for.3980010202

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. Adaptive computation and machine learning. MIT, Cambridge

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Thissen U, van Brakel R, de Weijer AP, Melssen WJ, Buydens LMC (2003) Using support vector machines for time series prediction. Chemometr Intell Lab Syst 69:35–49. https://doi.org/10.1016/S0169-7439(03)00111-4

United Nations (2008) International Standard industrial classification of all economic activities (ISIC). Rev. 4. Statistical papers. Series M, no. 4, re 4. United Nations, New York

Vapnik V (1995) The nature of statistical learning theory. Springer, New York

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bulletin 1:80. https://doi.org/10.2307/3001968

Zoubi TA, Salama F, Hossain M, Alkafaji YA (2016) The value relevance of components of other comprehensive income when net income is disaggregated. Rev Pac Basin Finan Mark Pol 19:1650027. https://doi.org/10.1142/S0219091516500272