

# Data Mining of Agricultural Yield Data: A Comparison of Regression Models

Georg Ruß

Otto-von-Guericke-Universität Magdeburg

**Abstract.** Nowadays, *precision agriculture* refers to the application of state-of-the-art GPS technology in connection with small-scale, sensor-based treatment of the crop. This introduces large amounts of data which are collected and stored for later usage. Making appropriate use of these data often leads to considerable gains in efficiency and therefore economic advantages. However, the amount of data poses a data mining problem – which should be solved using data mining techniques. One of the tasks that remains to be solved is *yield prediction* based on available data. From a data mining perspective, this can be formulated and treated as a multi-dimensional regression task. This paper deals with appropriate regression techniques and evaluates four different techniques on selected agriculture data. A recommendation for a certain technique is provided.

**Keywords:** Precision Agriculture, Data Mining, Regression, Modeling

## 1 Introduction

In the past decades, information technology (IT) has become more and more part of our everyday lives. With IT, improvements in efficiency can be made in almost any part of industry and services. Nowadays, this is especially true for agriculture. Due to the modernization and better affordability of state-of-the-art GPS technology, a farmer nowadays harvests not only crops but also growing amounts of data. These data are precise and small-scale – which is essentially why the combination of GPS, agriculture and data has been termed *precision agriculture*.

However, collecting large amounts of data often is both a blessing and a curse. There is a lot of data available containing information about a certain asset – here: soil and yield properties – which should be used to the farmer's advantage. This is a common problem for which the term *data mining* has been coined. Data mining techniques aim at finding those patterns or information in the data that are both valuable and interesting to the farmer.

A common specific problem that occurs is yield prediction. As early into the growing season as possible, a farmer is interested in knowing how much yield he is about to expect. In the past, this yield prediction has usually relied on farmers' long-term experience for specific fields, crops and climate conditions. However, this knowledge might also be available, but hidden, in the small-scale, precise data which can nowadays be collected in-season using a multitude of sensors. These sensors essentially aim to measure a field's heterogeneity.

Therefore, the problem of yield prediction encountered here is one of data mining and, specifically, multi-dimensional regression. This article should serve as an overview on the capabilities of different regression techniques used on agricultural yield data. Furthermore, this article can be seen as a continuation of [26]: in the previous article artificial neural networks have been evaluated and established as a well-suited reference model, which further models would have to compete against. The current work compares this particular neural network model with suitable further techniques (such as regression trees or support vector machines) to find the best prediction model. To accomplish this, the model output on site-year data from different years and sites is compared. Results on the parameterization of the different models are presented.

### 1.1 Research Target

The overall research target is to find those indicators of a field's heterogeneity which are suited best to be used for a yield prediction task. The sub-task here is one of multi-dimensional regression – predicting yield from past and in-season attributes. Furthermore, from the agricultural perspective, it is interesting to see how much the factor “fertilization” influences the yield in the current site-year. For this purpose, modeling techniques can be used, but have to be evaluated first. Therefore, this work aims at finding suitable data models that achieve a high accuracy and a high generality in terms of yield prediction capabilities. For this purpose, different types of regression techniques will be evaluated on different data sets.

Since models usually are strongly parameterized, an additional question is whether the model parameters can be carried over from one field to other fields which are comparable in (data set) size. This issue will also be addressed in this work. This is especially useful when new data have to be evaluated using one of the presented models.

### 1.2 Article structure

Section 2 lays out the data sets that this work builds upon. The attributes and their properties will be presented shortly. Section 3 presents four selected regression techniques from the data mining area which will be used for yield prediction. The free model parameters will be described. Section 4 shows the results from the modeling steps and provides answers to the aforementioned research questions. At the end of this article, future work is pointed out and implementation details are provided.

## 2 Data Description

The data available in this work have been obtained in the years 2003–2006 on three fields near Köthen, north of Halle, Germany<sup>1</sup>. All information available for these 65-, 72- and 32-hectare fields<sup>2</sup> was interpolated using kriging [30] to a grid with 10 by 10 meters grid cell sizes. Each grid cell represents a record with all available information. During the growing season of 2006, the latter field was subdivided into different

<sup>1</sup> GPS: Latitude N 51 40.430, Longitude E 11 58.110

<sup>2</sup> called *F04*, *F330* and *F131*, respectively

strips, where various fertilization strategies were carried out. For an example of various managing strategies, see e.g. [27], which also shows the economic potential of PA technologies quite clearly. The field grew winter wheat, where nitrogen fertilizer was distributed over three application times during the growing season.

Overall, for each field there are seven input attributes – accompanied by the respective current year’s yield (2004 or 2006) as the target attribute. Those attributes will be described in the following. In total, for the F04 field there are 5241 records, for F131 there are 2278 records, for F330 there are 4578 records, thereof none with missing values and none with outliers. In addition, a subset for F131 was available: in this subset, a special fertilization strategy was carried out which used a neural network for prediction and optimization – this data set is called F131net and has 1144 records of the following attributes.

### **2.1 Nitrogen Fertilizer – N1, N2, N3**

The amount of fertilizer applied to each subfield can be easily measured. It is applied at three points in time into the vegetation period, which is the standard strategy for most of Northwest Europe [20].

### **2.2 Vegetation – REIP32, REIP49**

The *red edge inflection point* (REIP) is a second derivative value calculated along the red edge region of the spectrum, which is situated from 680 to 750nm. Dedicated REIP sensors are used in-season to measure the plants’ reflection in this spectral band. Since the plants’ chlorophyll content is assumed to highly correlate with the nitrogen availability (see, e.g. [18]), the REIP value allows for deducing the plants’ state of nutrition and thus, the previous crop growth. For further information on certain types of sensors and a more detailed introduction, see [13] or [32]. Plants that have less chlorophyll will show a lower REIP value as the red edge moves toward the blue part of the spectrum. On the other hand, plants with more chlorophyll will have higher REIP values as the red edge moves toward the higher wavelengths. For the range of REIP values encountered in the available data, see Tables 1(b) and 1(c). The numbers in the REIP32 and REIP49 names refer to the growing stage of winter wheat, as defined in [16].

### **2.3 Electric Conductivity – EM38**

A non-invasive method to discover and map a field’s heterogeneity is to measure the soil’s conductivity. Commercial sensors such as the EM-38<sup>3</sup> are designed for agricultural use and can measure small-scale conductivity to a depth of about 1.5 metres. There is no possibility of interpreting these sensor data directly in terms of its meaningfulness as yield-influencing factor. But in connection with other site-specific data, as explained in the rest of this section, there could be coherences. For a more detailed analysis of this particular sensor, see, e.g. [4]. For the range of EM values encountered in the available data, see Tables 1(a) to 1(c).

<sup>3</sup> trademark of Geonics Ltd, Ontario, Canada

## 2.4 YIELD

Here, yield is measured in metric tons per hectare ( $\frac{t}{ha}$ ). For the yield ranges for the respective years and sites, see Tables 1(b) and 1(c). It should be noted that for the F131 and F330 data sets the yield was reduced significantly due to bad weather conditions (lack of rain) during the growing season 2006.

## 2.5 Data Overview

In this work, data sets from three different fields are being evaluated. A brief summary of the available data attributes for both data sets is given in Tables 1(a) to 1(c). On each field, different fertilization strategies have been used. One of those strategies is based on a technique that uses a multi-layer perceptron (MLP) for prediction and optimization. This technique has been presented and evaluated in, e.g., [25,26] or [32]. For each field, one data set will contain all records, thus containing all the different fertilization strategies. In addition, a subset of F131 has been chosen to serve as a fourth data set to be evaluated.

Table 1: Overview of the F04, F131 and F330 data sets. The additional data set *F131net*, which is a subset of *F131*, is not shown as its statistical properties are very similar to those of *F131*.

<b>F04</b>	<i>min</i>	<i>max</i>	<i>mean</i>	<i>std</i>
YIELD03	1.19	12.38	6.27	1.48
EM38	17.97	86.45	33.82	5.27
N1	0	100	57.7	13.5
N2	0	100	39.9	16.4
N3	0	100	38.5	15.3
REIP32	721.1	727.2	725.7	0.64
REIP49	722.4	729.6	728.1	0.65
YIELD04	6.42	11.37	9.14	0.73

(a) Data overview, F04

<b>F131</b>	<i>min</i>	<i>max</i>	<i>mean</i>	<i>std</i>
YIELD05	1.69	10.68	5.69	0.93
EM38	51.58	84.08	62.21	8.60
N1	47.70	70	64.32	6.02
N2	14.80	100	51.71	15.67
N3	0	70	39.65	13.73
REIP32	719.6	724.4	722.6	0.69
REIP49	722.3	727.9	725.8	0.95
YIELD06	1.54	8.83	5.21	0.88

(b) Data overview, F131

<b>F330</b>	<i>min</i>	<i>max</i>	<i>mean</i>	<i>std</i>
YIELD05	4.64	14.12	10.62	0.97
EM38	25.08	49.48	33.69	2.94
N1	24.0	70	59.48	14.42
N2	3.0	100	56.38	13.35
N3	0.3	91.6	50.05	12.12
REIP32	719.2	724.4	721.5	1.03
REIP49	723.0	728.5	726.9	0.82
YIELD06	1.84	8.27	5.90	0.54

(c) Data overview, F330

## 2.6 Fertilization Strategies

There were three different strategies that have been used to guide the nitrogen fertilization of the fields. The three strategies are described in the following. A subset of the F131 data set was chosen where the strategy was “N”. This is used as the smallest data set in this work that models will be built upon.

- F** – uniform distribution of fertilizer according to long-term experience of the farmer
- N** – fertilizer distribution was guided by an economic optimization with a multi-layer perceptron model; the model was trained using the above data with the current year’s yield as target variable that is to be predicted (see, e.g., [26]).
- S** – based on a special nitrogen sensor – the sensor’s measurements are used to determine the amount of nitrogen fertilizer that is to be applied.

## 3 Advanced Regression Techniques

As mentioned in the preceding section, the task of yield prediction is essentially a task of multi-dimensional regression. Therefore, this section will serve as an overview about different regression techniques that are applicable to the yield data sets. It is aimed to evaluate these techniques on the data sets presented in the preceding section.

The regression task can be formalized as follows: the training set

$$T = \{\{x_1, \dots, x_n\}, y_i\}_{i=1}^N \quad (1)$$

is considered for the training process, where  $x_i, i = 1, \dots, n$  are continuous input values and  $y_i, i = 1 \dots, N$  are continuous output values. Given this training set, the task of the regression techniques is to approximate the underlying function sufficiently well. The quality of the approximated function can be measured by error values, some of which are specified in section 3.6.

### 3.1 Introduction to regression techniques

Since one particular technique, namely MLPs, has been used successfully in previous work [24, 26], it is used as a reference model here. Three additional modeling techniques will be presented that are suitable for the task of yield prediction.

In the past, numerous regression techniques have been used successfully on data from agriculture. Neural networks have shown to be quite effective in modeling yield of different crops ([7, 28]). In [31] and [32], artificial neural networks, namely multi-layer perceptrons (MLPs) have been trained to predict wheat yield from fertilizer and additional sensor input. The basic framework for MLPs will be given in section 3.2.

Radial basis function (RBF) networks are similar to multi-layer perceptrons in that they can also be used to model non-linear relationships between input data. Nevertheless, there has been almost no research into RBF networks when applying them to agriculture data. Some of the theoretical properties and differences between MLPs and RBFs will be pointed out in section 3.2.

Regression trees have seen some usage in agriculture [6, 12, 14]. Essentially, they are a special case of decision trees where the outcome (in the tree leaves) is a continuous function instead of a discrete classification. Further details can be found in section 3.3.

A fourth technique that has, to the best of the author's knowledge, not been used on similar yield data, but for similar regression tasks, is a derivative of support vector machines (SVMs). Similar to decision trees, if the target attribute is discrete, SVMs would solve a classification task, whereas in the case of a continuous attribute, a regression task would be solved. Hence, support vector regression (SVR) will be explained in section 3.4.

The aforementioned techniques have not been compared to each other when used with different data sets in the agriculture context. This section presents the necessary background for each of the techniques before they will be evaluated in section 4.

### 3.2 Neural Networks

In previous work multi-layer perceptrons (MLPs), a type of neural networks, have been used for a modeling task [24, 26] similar to the one laid out in the preceding section. The MLP model has been established as a reference model against which further regression techniques would have to compete. Hence, the MLP will be explained shortly in the following section. Furthermore, a different type of neural network, a radial basis function (RBF) network, will be presented since it is well-suited to the regression task.

**MLP** According to [19], "neural networks provide a general, practical method for learning [...] vector-valued functions from examples." In the previous work multi-layer perceptrons (MLPs) with backpropagation learning have been used to learn from agricultural data and predict yield. Generally, MLPs can be seen as a practical vehicle for performing a non-linear input-output mapping [10]. The results from [24, 26] lead us to assume that the extension to more than one hidden layer only marginally increases the generalization performance of MLPs, but rather drastically increases the computation time for the backpropagation algorithm. Hence, here it is assumed that one hidden layer is sufficient to approximate the underlying function sufficiently well. For a more detailed and formal description of MLP neural networks, it is referred to [9] or [10].

Once the number of hidden layers of the network is fixed, there remain a few parameters to be determined. Similar to the remaining modeling techniques, those are usually determined experimentally. This also means that often a large parameter space has to be searched. For standard MLP networks, the size of the hidden layer, the learning rate, the activation function and the minimum gradient are the most important parameters that have to be set. In this case, the matlab implementation for the MLP network was used: `newff`.<sup>4</sup>

**RBF** While the MLP networks in the preceding section had a variable number of layers, the number of layers in an RBF network is fixed. There are three layers of neurons that

---

<sup>4</sup> For details on matlab implementations and scripts see the link contained in the *acknowledgements* section.

constitute an RBF network and perform different roles. While the input layer is the same as in an MLP network, the only hidden layer applies a nonlinear transformation from the input space to the hidden space. The output layer is again linear. The idea behind this approach is that a regression (or classification) problem is much more likely to be solvable in a high-dimensional space than in a low-dimensional space [5]. The main difference to MLPs is in the hidden layer. The activation function of each hidden unit in the RBF network computes the Euclidean norm, i.e. the distance, between the input vector and the center of that unit. In MLP networks, the activation function computes the inner product of the input vector and the synaptic weight vector of that unit.

One of the simpler learning algorithms that can be employed for RBF networks is described in the following.

1. The network is simulated: for all training examples, the output of the network is compared to the actual target value of the respective example.
2. The input vector with the greatest error is determined.
3. An RBF neuron is added to the hidden layer with weights equal to that vector.
4. The connection weights from the hidden layer to the output layer are adapted to minimize the error.

According to the above algorithm, the RBF network training algorithm has at least the following parameters: a) an error goal that must be met, b) a radius (or spread) of the radial basis function and c) a maximum number of neurons that should be added before stopping. These parameters are usually determined experimentally, although some strategies for computing them are presented in [10]. Since the current approach aims to compare four basic techniques for non-linear regression, it was chosen to employ the above training algorithm without further tweaking. It has been implemented in matlab's `newrb` function and the parameters have been determined experimentally.

### 3.3 Regression Tree

Learning decision trees is a paradigm of *inductive learning*: a model is built from data or observations according to some criteria. The model aims to learn a general rule from the observed instances. Decision trees can therefore accomplish two different tasks, depending on whether the target attribute is discrete or continuous. In the first case, a classification tree would result, whereas in the second case a regression tree would be constructed. Since the focus is on solving a regression task, the regression tree will be explained shortly in the following.

Regression trees approximate learning instances by sorting them down the tree from the root to some leaf node, which provides the value of the target attribute. Each node in the tree represents a split of some attribute of the instance and each branch descending from that node corresponds to one part left or right of the split. The value of the target attribute for an instance is determined by starting at the root node of the tree and testing the attribute specified by this node. This determines whether to proceed left or right of the split. Then the algorithm moves down the tree and repeats the procedure with the respective subtree. In principle, there could be more than one split in a tree node, which would result in more than two subtrees per node. However, in this application

scenario, regression trees with more than two subtrees per split node are not taken into consideration.

Regression as well as decision trees are usually constructed in a top-down, greedy search approach through the space of possible trees [19]. The basic algorithms for constructing such trees are CART [2], ID3 [22] and its successor C4.5 [23]. The idea here is to ask the question “which attribute should be tested at the top of the tree?” To answer this question, each attribute is evaluated to determine how well it is suited to split the data. The best attribute is selected and used as the test node. This procedure is repeated for the subtrees. An attribute selection criterion that is employed by ID3 and C4.5 is the entropy and, resulting from it, the information gain. Entropy is a measure from information theory that describes the variety in a collection of data points: the higher the entropy, the higher the variety. An attribute split aims to lower the entropy of the two resulting split data sets. This reduction in entropy is called the information gain. For further information it is referred to [19].

However, if the addition of nodes is continued without a specific stopping criterion, the depth of the tree continues to grow until each tree leaf covers one instance of the training data set. This is certainly a perfect tree for the training data but is likely to be too specific – the problem of overlearning occurs. For new, unseen data, such a specific tree will probably have a high prediction error. Therefore, regression trees are usually pruned to a specific depth which is a trade-off between high accuracy and high generality. This can easily be achieved by setting a lower bound for the number of instances covered by a single node below which no split should occur. For this work the standard matlab implementation of `classregtree` was used.

### 3.4 Support Vector Regression

Support Vector Machines (SVMs) are a supervised learning method discovered by [1]. However, the task here is regression, so the focus is on support vector regression (SVR) in the following. A more in-depth discussion can be found in [8]. Given the training set, the goal of SVR is to approximate a linear function  $f(x) = \langle w, x \rangle + b$  with  $w \in \mathbb{R}^N$  and  $b \in \mathbb{R}$ . This function minimizes an empirical risk function defined as

$$R_{emp} = \frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(\hat{y} - f(x)), \quad (2)$$

where  $L_{\varepsilon}(\hat{y} - f(x)) = \max(|\xi| - \varepsilon, 0)$ .  $|\xi|$  is the so-called slack variable, which has mainly been introduced to deal with otherwise infeasible constraints of the optimization problem, as has been mentioned in [29]. By using this variable, errors are basically ignored as long as they are smaller than a properly selected  $\varepsilon$ . The function here is called  $\varepsilon$ -insensitive loss function. Other kinds of functions can be used, some of which are presented in chapter 5 of [8].

To estimate  $f(x)$ , a quadratic problem must be solved, of which the dual form, according to [17] is as follows:

$$\max_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \quad (3)$$



with the constraint that  $\sum_{j=1}^N (\alpha_j - \alpha_j^*) = 0$ ,  $\alpha_j, \alpha_j^* \in [0, C]$ . The regularization parameter  $C > 0$  determines the tradeoff between the flatness of  $f(x)$  and the allowed number of points with deviations larger than  $\varepsilon$ . As mentioned in [8], the value of  $\varepsilon$  is inversely proportional to the number of support vectors. An adequate setting of  $C$  and  $\varepsilon$  is necessary for a suitable solution to the regression problem.

Furthermore,  $K(x_i, x_j)$  is known as a kernel function which allows to project the original data into a higher-dimensional feature space where it is much more likely to be linearly separable. Some of the most popular kernels are radial basis functions (equation 4) and a polynomial kernel (equation 5):

$$K(x, x_i) = e^{-\frac{\|x-x_i\|^2}{2\sigma^2}} \quad (4)$$

$$K(x, x_i) = (\langle x, x_i \rangle + 1)^p \quad (5)$$

The parameters  $\sigma$  and  $p$  have to be determined appropriately for the SVM to generalize well. This is usually done experimentally. Once the solution for the above optimization problem in equation 3 is obtained, the support vectors can be used to construct the regression function:

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (6)$$

In the current experiments, the SVMtorch implementation from [3] has been used, which also points out further details of the SVR process.

### 3.5 Performance Measurement

The performance of the models will be determined using the root mean squared error (RMSE) and the mean absolute error (MAE). For the RMSE, first the difference between an actual target value  $y_a$  and the model output value  $y$  is computed. This difference is squared and averaged over all training examples before the root of the mean value is taken, see equation 7. The MAE is computed similarly, see equation 8.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_{a,i})^2} \quad (7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_{a,i}| \quad (8)$$

### 3.6 Model Parameter Estimation

Each of the aforementioned four different models will be evaluated on the same data sets. One of the research goals is to establish whether a model that has been used on one data set can be used on a different data set without changing its parameters. This would mean that comparable fields could use the same prediction model. Hence, the

*F04* data set is used to determine the model parameters experimentally. Afterwards, the models are re-trained on the remaining data sets using the settings determined for *F04*. The parameter settings are given in section 4.

For training the models, a cross-validation approach is taken. As mentioned in e.g. [11], the data will be split randomly into a training set and a test set. The model is trained using the training data and after each training iteration, the error on the test data is computed. During training, this error usually declines towards a minimum. Beyond this minimum, the error rises – overlearning (or overfitting) occurs: the model fits the training data perfectly but does not generalize well. Hence, the model training is stopped when the error on the test set starts rising. A size ratio of 9:1 for training and test set is used. The data sets are partitioned randomly 20 times, the models are trained and the error values are collected.

## 4 Results

The models are run with the following parameter settings, which were determined experimentally on *F04* and carried over to the remaining data sets.

**MLP** For the multi-layer perceptron model, a relatively small number of 10 hidden neurons is used and the network is trained until a minimum gradient of 0.001 is reached, using a learning rate of 0.25 and the *tangens hyperbolicus* sigmoid activation function.

**RBF** For the radial basis function network, a radius of 1 is used for the radial basis neurons in the hidden layer. The algorithm, which incrementally adds neurons until the error goal of 0.001 is met, uses a maximum number of 70 neurons, which results in a relatively long training time.

**RegTree** For the regression tree, the default settings of `classregtree` perform optimal; the full tree is pruned automatically and the minimum number of training examples below which no split of a tree node should be done is 10.

**SVR** For the support vector regression model, the radial basis function kernel yields the best results, using the parameter  $C = 60$  (tradeoff between training error and margin), and the standard deviation  $\sigma = 4.0$ . The slack variable  $\xi = 0.2$  is also determined as yielding the best results on *F04*.

### 4.1 Detailed Results

Considering the results in Table 2, support vector regression obviously performs best on all but one of the data sets, regarding both error measures. Furthermore, SVR also is the model taking the least amount of computation time (not shown in table). Hence, the slight difference between the RMSE of SVR and RBF on the *F330* data set may be considered insignificant in practice when computational cost is also taken into account when deciding for a model.

Regarding the understandability of the generated models, it would certainly be desirable to have the regression tree as the best model since simple decision rules can

Error Measure / Model	F04	F131	F131net	F330
MAE MLP:	0.3706	0.2468	0.2300	0.3576
RMSE MLP:	0.4784	0.3278	0.3073	0.5020
MAE RBF:	0.3838	0.2466	0.2404	0.3356
RMSE RBF:	0.5031	0.3318	0.3205	<b>0.4657</b>
MAE REGTREE:	0.4380	0.2823	0.2530	0.4151
RMSE REGTREE:	0.5724	0.3886	0.3530	0.6014
MAE SVR:	<b>0.3446</b>	<b>0.2237</b>	<b>0.2082</b>	<b>0.3260</b>
RMSE SVR:	<b>0.4508</b>	<b>0.3009</b>	<b>0.2743</b>	0.4746

Table 2: Results of running different models on different data sets. The best predictive model for each data set is marked in **bold** font.

easily be generated from the tree. However, the regression tree performs worst in all of the cases. On the other hand, when comparing the hitherto reference model MLP with the current best model SVR, the understandability of both models is equally limited. Further research into understanding these models has been and should be undertaken.

Figure 1 shows eight graphs which depict the MAE and RMSE for the different data sets vs. different models, when run on a total of 20 random cross-validation splits. Except for the regression tree, which often produced much higher errors than the other models, the three remaining models usually agree in their error values – with slight, but constant differences in favor of the SVR.

## 4.2 Conclusion

The results clearly show that support vector regression can serve as a better reference model for yield prediction. It is computationally less demanding, at least as understandable as the hitherto multi-layer perceptron and, most importantly, produces better yield predictions.

The results also show that model parameters which have been established on one data set can be carried over to different (but similar with respect to the attributes) data sets.

## 4.3 Future Work

One aspect that should be considered in future work is the understandability of the model. While regression trees would be the easiest to understand, they bear the burden of providing worse results compared to SVR. There has already been quite a lot of work towards understanding the inner workings of an MLP, but it remains one of the more intransparent models. The same holds for SVR, but there has been some work using the support vectors for visualization, such as [33, 15].

Rather than selecting one of the four presented models, there might be a certain combination of models that performs better than a single one. Some prerequisites would have to be fulfilled – such as the error for a certain data record would have to be alternatively low in one model and high in another. A similar idea is presented in [21].

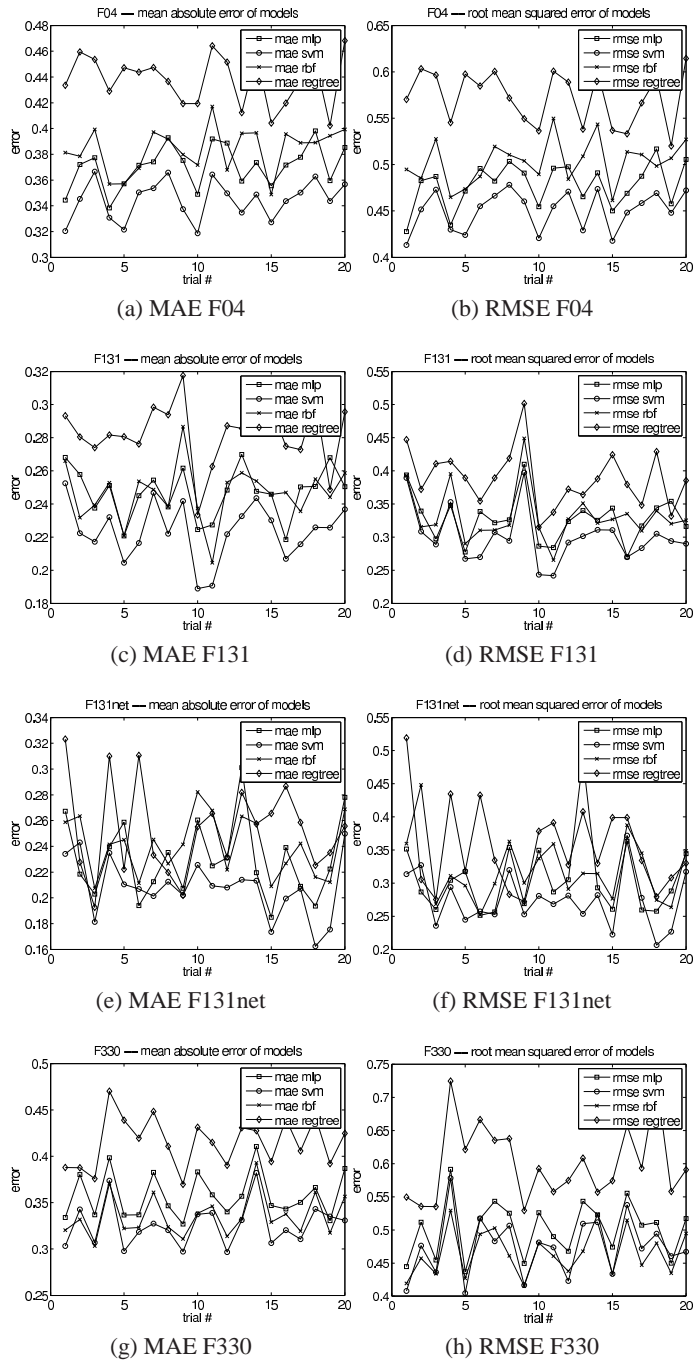


Fig. 1: Errors of different data sets vs. different models

The most time-consuming part of this article is the manual determination of parameters for one or more models since a large parameter space has to be considered. Even though it has been established that the model parameters which have been determined on one data set can be carried over to different data sets, heuristics for model parameters might reduce the size of the parameter space which has to be searched.

#### 4.4 Acknowledgements

Experiments have been conducted using Matlab 2008a and the corresponding Neural Network Toolbox. The field trial data came from the experimental farm Görzig of Martin-Luther-University Halle-Wittenberg, Germany. The trial data have kindly been provided by Martin Schneider and Prof. Dr. Peter Wagner<sup>5</sup> of the aforementioned institution. Supplementary material, such as Matlab scripts and plots can be found at the author's research site <http://research.georgruss.de/?cat=20>.

#### References

1. Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.
2. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
3. Ronan Collobert, Samy Bengio, and C. Williamson. Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1:143–160, 2001.
4. D. L. Corwin and S. M. Lesch. Application of soil electrical conductivity to precision agriculture: Theory, principles, and guidelines. *Agron J*, 95(3):455–471, May 2003.
5. T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. In *IEEE Transactions on Electronic Computers*, volume EC-14, pages 326–334, 1965.
6. S. F. Crone, S. Lessmann, and S. Pietsch. Forecasting with computational intelligence - an evaluation of support vector regression and artificial neural networks for time series prediction. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 3159–3166, 2006.
7. S. Drummond, A. Joshi, and K. A. Sudduth. Application of neural networks: precision farming. In *International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence*, volume 1, pages 211–215, 1998.
8. S.R. Gunn. Support vector machines for classification and regression. Technical Report, School of Electronics and Computer Science, University of Southampton, Southampton, U.K., 1998.
9. Martin T. Hagan. *Neural Network Design (Electrical Engineering)*. Thomson Learning, December 1995.
10. Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, July 1998.
11. Robert Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, September 1990.

---

<sup>5</sup> {martin.schneider,peter.wagner}@landw.uni-halle.de

12. Chengquan Huang, Limin Yang, Bruce Wylie, and Collin Homer. A strategy for estimating tree canopy density using landsat 7 etm+ and high resolution images over large areas. In *Proceedings of the Third International Conference on Geospatial Information in Agriculture and Forestry*, 2001.
13. J. Liu, J. R. Miller, D. Haboudane, and E. Pattey. Exploring the relationship between red edge parameters and crop variables for precision agriculture. In *2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages 1276–1279, 2004.
14. David B. Lobell, J. Ivan Ortiz-Monasterio, Gregory P. Asner, Rosamond L. Naylor, and Walter P. Falcon. Combining field surveys, remote sensing, and regression trees to understand yield variations in an irrigated wheat landscape. *Agronomy Journal*, 97:241–249, 2005.
15. Tomasz Maszczyk and Włodzisław Duch. Support vector machines for visualization and dimensionality reduction. *Artificial Neural Networks - ICANN 2008*, pages 346–356, 2008.
16. U. Meier. *Entwicklungsstadien mono- und dikotyler Pflanzen*. Biologische Bundesanstalt für Land- und Forstwirtschaft, Braunschweig, Germany, 2001.
17. Iván Mejía-Guevara and Ángel Kuri-Morales. Evolutionary feature and parameter selection in support vector regression. In *Lecture Notes in Computer Science*, volume 4827, pages 399–408. Springer, Berlin, Heidelberg, 2007.
18. E. M. Middleton, P. K. E. Campbell, J. E. McMurtrey, L. A. Corp, L. M. Butcher, and E. W. Chappelle. “Red edge” optical properties of corn leaves from different nitrogen regimes. In *2002 IEEE International Geoscience and Remote Sensing Symposium*, volume 4, pages 2208–2210, 2002.
19. Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, March 1997.
20. Jacques J. Neeteson. *Nitrogen Management for Intensively Grown Arable Crops and Field Vegetables*, chapter 7, pages 295–326. CRC Press, Haren, The Netherlands, 1995.
21. Mark Orr, John Hallam, Alan Murray, Seishi Ninomiya, Mari Oide, and Tom Leonard. Combining regression trees and radial basis function networks. *International Journal of Neural Systems*, 10, 1999.
22. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.
23. Ross J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.
24. Georg Ruß, Rudolf Kruse, Martin Schneider, and Peter Wagner. Estimation of neural network parameters for wheat yield prediction. In Max Brammer, editor, *Artificial Intelligence in Theory and Practice II*, volume 276 of *IFIP International Federation for Information Processing*, pages 109–118. Springer, July 2008.
25. Georg Ruß, Rudolf Kruse, Martin Schneider, and Peter Wagner. Optimizing wheat yield prediction using different topologies of neural networks. In José Luis Verdegay, Manuel Ojeda-Aciego, and Luis Magdalena, editors, *Proceedings of IPMU-08*, pages 576–582. University of Málaga, June 2008.
26. Georg Ruß, Rudolf Kruse, Peter Wagner, and Martin Schneider. Data mining with neural networks for wheat yield prediction. In Petra Perner, editor, *Advances in Data Mining (Proc. ICDM 2008)*, pages 47–56, Berlin, Heidelberg, July 2008. Springer Verlag.
27. M. Schneider and P. Wagner. Prerequisites for the adoption of new technologies - the example of precision agriculture. In *Agricultural Engineering for a Better World*, Düsseldorf, 2006. VDI Verlag GmbH.
28. C. Z. Serele, Q. H. J. Gwyn, J. B. Boisvert, E. Pattey, N. McLaughlin, and G. Daoust. Corn yield prediction with artificial neural network trained using airborne remote sensing and topographic data. In *2000 IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 384–386, 2000.
29. Alex J. Smola and Bernhard Sch Olkopf. A tutorial on support vector regression. Technical report, Statistics and Computing, 1998.

30. Michael L. Stein. *Interpolation of Spatial Data : Some Theory for Kriging (Springer Series in Statistics)*. Springer, June 1999.
31. P. Wagner and M. Schneider. Economic benefits of neural network-generated site-specific decision rules for nitrogen fertilization. In J. V. Stafford, editor, *Proceedings of the 6th European Conference on Precision Agriculture*, pages 775–782, 2007.
32. Georg Weigert. *Data Mining und Wissensentdeckung im Precision Farming - Entwicklung von ökonomisch optimierten Entscheidungsregeln zur kleinräumigen Stickstoff-Ausbringung*. PhD thesis, TU München, 2006.
33. Sitao Wu and T. W. S. Chow. Support vector visualization and clustering using self-organizing map and vector one-class classification. In *Proceedings of the International Joint Conference on Neural Networks*, pages 803–808 vol.1, 2003.