Learnt Topology Gating Artificial Neural Networks

Petr Kadlec and Bogdan Gabrys

Abstract— **This work combines several established regression and meta-learning techniques to give a holistic regression model and presents the proposed Learnt Topology Gating Artificial Neural Networks (LTGANN) model in the context of a general architecture previously published by the authors. The applied regression techniques are Artificial Neural Networks, which are on one hand used as local experts for the regression modelling and on the other hand as gating networks. The role of the gating networks is to estimate the prediction error of the local experts dependent on the input data samples. This is achieved by relating the input data space to the performance of the local experts, and thus building a performance map, for each of the local experts. The estimation of the prediction error is then used for the weighting of the local experts predictions. Another advantage of our approach is that the particular neural networks are unconstrained in terms of the number of hidden units. It is only necessary to define the range within which the number of hidden units has to be generated. The model links the topology to the performance, which has been achieved by the network with the given complexity, using a probabilistic approach. As the model was developed in the context of process industry data, it is evaluated using two industrial data sets. The evaluation has shown a clear advantage when using a model combination and meta-learning approach as well as demonstrating the higher performance of LTGANN when compared to a standard combination method.**

I. Introduction

The original idea of Artificial Neural Networks (ANN) was to mimic the operation of biological neurons, as the basic information processing units in the biological nervous system. Probably the most common in terms of the reported number of applications is the Multi-Layer Perceptron (MLP) (e.g. [1]). MLPs are universal function approximators, which means that provided enough training data and given a complex enough structure, they can be trained to approximate any possible function. Since the introduction of the *back-propagation* learning algorithm to ANNs [2] and due to their generalisation power and ability to solve non-linear problems, MLPs have been applied to many practical classification and regression problems. The drawback of MLPs and of the back-propagation algorithm is that during the learning phase they can get stuck in local minima, which results in sub-optimal performance on the test data. Another problem is the difficulty with the estimation of correct topology of the networks since the generalisation power of MLPs depends to a high extent on the complexity of the networks and thus an appropriate choice of the topology is critical. There is also an issue with the interpretability of the learnt knowledge as it is distributed in the weights between particular neurons which can not be easily interpreted in terms of a human understandable representation.

Petr Kadlec and Bogdan Gabrys are with the Computational Intelligence Research Group, Bournemouth University, Bournemouth, BH12 5BB, United Kingdom (email: {pkadlec, bgabrys}@bournemouth.ac.uk).

The problems listed above are not unique for ANNs. In fact they are quite common to many computational learning approaches. In the meta-learning terminology models such as this learning input-output mappings given training data sets often using a fixed model structure are called *base-learners*. In contrast to the base-learners, the task of the meta-learning approach is to extract high-level knowledge from the base-learner and to use this knowledge to improve them. One could therefore describe meta-learning as *learning to learn*. This task can be approached from different directions. Probably the most direct one is to link the performance of the base-learners to meta-features and thus to identify their areas of expertise. This corresponds to the regions of the meta-feature space for which a particular algorithm or class of algorithms perform well. The simplest examples of such meta-features are statistics of the data, like mean value, variance, kurtosis, etc. [3], [4], [5]. Another way of using meta-learning for model building is by combining the predictions of several base learners to a global prediction. This approach is known under many different names including ensembles methods [6], multiple classifier systems [7], model stacking [8] [9], etc. The aim of combining is to train a meta-learner, whose input space is formed by the predictions of a set of particular base-learners. The target feature of the meta-learner is equivalent to the target feature of the base-learners. There are several ways to build the combined predictions. In general, one can distinguish between trainable and fixed combiners. Typical examples of fixed combiners are the building mean, or more outlier resistant median, values of the predictions. In contrast to the fixed combiners, the trainable combiners are much more powerful with the most common example including weighted linear combinations with trainable weights. Good reviews and discussions of various combination methods can be found in [9], [10], [11], [12].

In the context of this work a particularly relevant approach to combination is discussed in [13] [14], where a gating network is used to decide which of the models from a set of available base-learner ANNs, or local experts in the terminology of the cited work, is responsible for the prediction of the given input sample. The predictions of the particular local experts are weighted using weights, which are predicted by the gating networks. In [13] Jordan and Jacobs proposed a special algorithm for the training of the gating networks, which learns and memorizes the experts responsible for a significant improvement of the performance of the global model. The work described in this paper motivated by the Gating Artificial Neural Networks proposes a number of

978-1-4244-1821-3/08/$25.00

important modifications further discussed in Section III.

The next section gives a brief overview of the general modelling architecture. Section III describes the LTGANN meta-learning model as an instance of the architecture, which is in Section IV evaluated on two real-life industrial data sets. Finally, this work is summarised in Section V.

II. Regression Model Architecture

While more details can be found in [15], this section provides a summary of the meta architecture previously published by the authors. A broad overview of the architecture is shown in Fig. 1. A significant part of the proposed architecture are the two pools, firstly the *Pre-processing Methods Pool* (PPMP), which is further split into actual Pre-Processing (PP) methods (e.g. filtering, normalisation), Feature Selection (FS) methods (e.g. correlation-based feature selection) and Instance Selection (IS) methods (e.g. receptive fields filtering). The second pool, *Computational Learning Methods Pool* (CLMP), consists of various computational learning methods (e.g. linear regression, multi-layer perceptron models, etc.). The two pools provide the methods to the *Path/Pool Management* (PPM) module. Within this module the methods are instantiated and linked to form *transformation paths*. A transformation path may be for example built from the following elements: feature standardisation, correlation-based feature selection and a multi-layer perceptron method. The particular transformation paths within this module are managed within Path/Pool Management Control. From here the paths can be created, adapted and eliminated. The decisions are made on the basis of information coming from high level decision making parts of the architecture which are described later in this section.

Another key aspect of the architecture is the *Path Combination* (PC) module. This module provides the possibility to make use of model combination and selection techniques which is beneficial for the performance of the final model [12]. The combinations are performed at the transformation path level which provides additional flexibility. One can do the combination while including different methods from PPMP (e.g. a combination of several paths consisting of MLP with different approaches to feature selection as a pre-processing step). Another advantage is that it is possible to combine different methods from the CLMP, in this way it is possible to do combinations across different types of computational learning methods (e.g. a combination of MLPs and RBF together with linear regression models). The path combination module together with the instance selection methods from PPMP provide also the possibility to combine different local paths (local learning models) to a global path. The Path Combination Control plays a similar role to the control unit in the PPMP but at the combination level.

The architecture provides also the possibility of using meta-learning approaches [10][16]. There are two modules in the architecture for this purpose. The first one, *Meta-Feature Management*, having information about the data together with the performance of the particular paths builds the meta-features. This module may e.g. extract the information about the performance of the different paths in the different parts of the input data space and pass this information further to the *Meta-Level Learning* module which can, using the provided information, control the Path/Pool Management and Path Combination modules.

The *Instance Selection Management* module is responsible for the filtering of the instances and thus providing the possibility for building of local models, i.e. local experts, [14][17][18][19]. The local approach to the model building is, apart from the pool and path concepts and meta-learning techniques, one of the key aspects of the proposed architecture.

The next section describes an instance of this general architecture using Artificial Neural Networks.

III. Learnt Topology GANN

The Learnt Topology Gating Artificial Neural Network (LTGANN) approach presented in this Section is based on [13] [14] but in contrast to the cited work the Gating ANNs (GANNs) use the standard back-propagation algorithm for the training of the gating networks. The next difference is that there is one gating network trained for each of the local experts. In this way it can be guaranteed that the GANN becomes an *expert* for the performance prediction of the assigned local expert.

The gating networks are trained to estimate the prediction error of the assigned local expert. The aim of the GANN is therefore to learn the performance of the experts dependent on the input samples. This is achieved by training the GANN using the local expert's prediction error on a validation data set as the target value, the training set for the *ith* GANN has thus the following form: $T^i_{train} = \{X^i_{val}, e^i_{val}\}$, where X^i_{val} is the validation input samples of the *ith* local expert and e^i_{val} represents the prediction error of the same local expert on the validation data.

After their training, the gating networks are able to estimate the prediction errors of the local experts. This estimation is then used to weight the predictions of the local experts and to obtain the final prediction in the following way:

$$y^f_p = \sum_{i=1}^{N} w^i y^i_p = \sum_{i=1}^{N} \frac{1}{1+e^i_p} y^i_p, \tag{1}$$

where N is the number of available experts, y^f_p the final output of the model, y^i_p the prediction of the *ith* local expert and w_i the weight of the local expert i based on the local expert's predicted error e^i_p.

Another advantage of the presented approach is that the number of local experts can be changed dynamically, i.e. the number of experts can be increased (or decreased) without the need to change the other local experts and GANNs. We exploit this feature and gradually increase the number of local experts. While increasing the number of local experts, the optimal network topologies for both, the local experts and the gating networks, are being learnt.

Restricting ourselves to networks with one hidden layer, the topology (i.e. the number of hidden units) of the LE

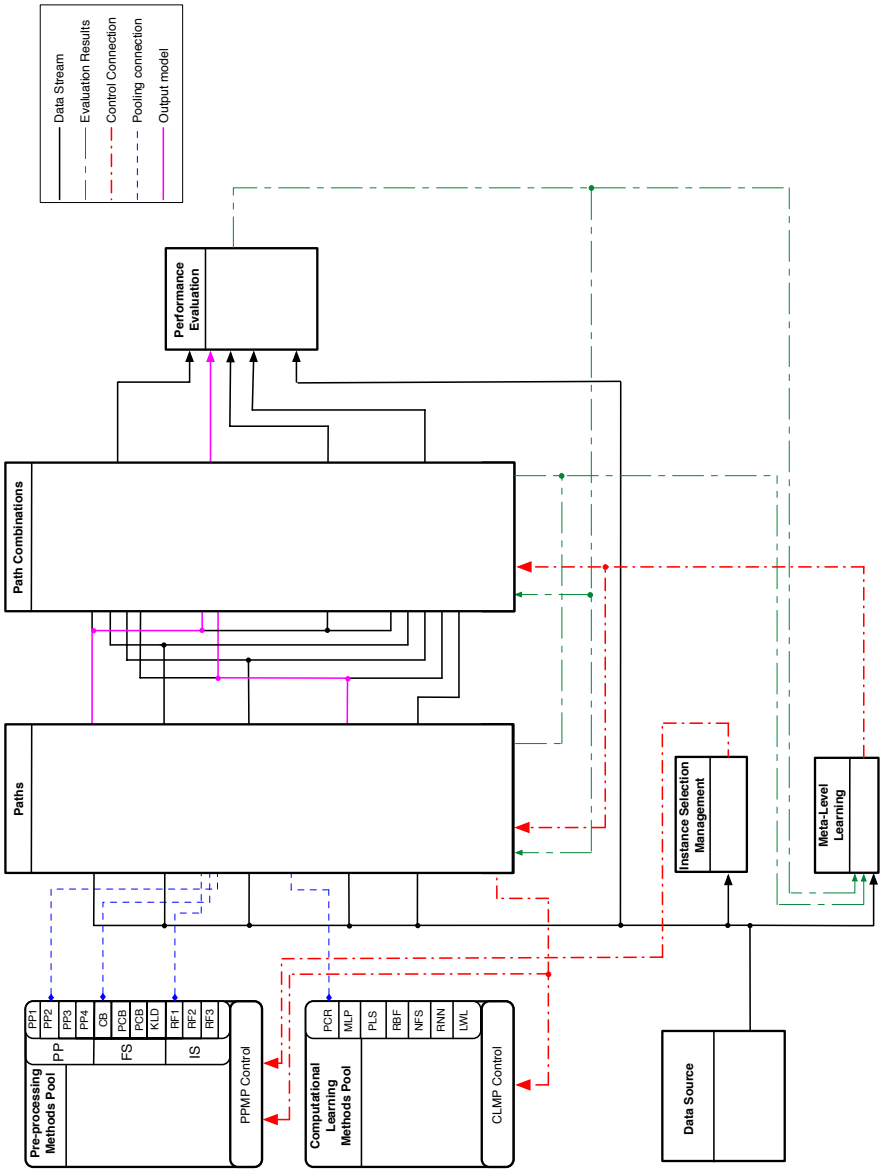

Fig. 1. The general architecture model

and GANN is initially determined randomly by drawing the numbers of hidden units from a equal distribution $\mathcal{U}(H_{LE})$ and $\mathcal{U}(H_{GANN})$, where H_x is a pre-defined range of possible hidden number units for the local experts and gating networks respectively. After evaluating the performance of the LEs and GANNs with the number of hidden units $h_{LE} \in H_{LE}$ and $h_{GANN} \in H_{GANN}$, the relative performances q_{LE} and q_{GANN} are used to modify the originally equal distribution towards the conditional distributions for both topologies $P(H_{LE}|q_{LE})$ and $P(H_{GANN}|q_{GANN})$:

$$P(H) \xrightarrow{init.} \mathcal{U}(H) \xrightarrow{learning} P(H|q). \quad (2)$$

At each new step (i.e. adding new local expert), the up-to-date distributions are used to generate the topologies of the new networks. This mechanism provides the means to deal with one of the disadvantages of ANNs, namely the manual estimation of the optimal network topology as the proposed algorithm learns the well-performing network topologies automatically. One needs only to define the range from which the number of hidden units has to be drawn.

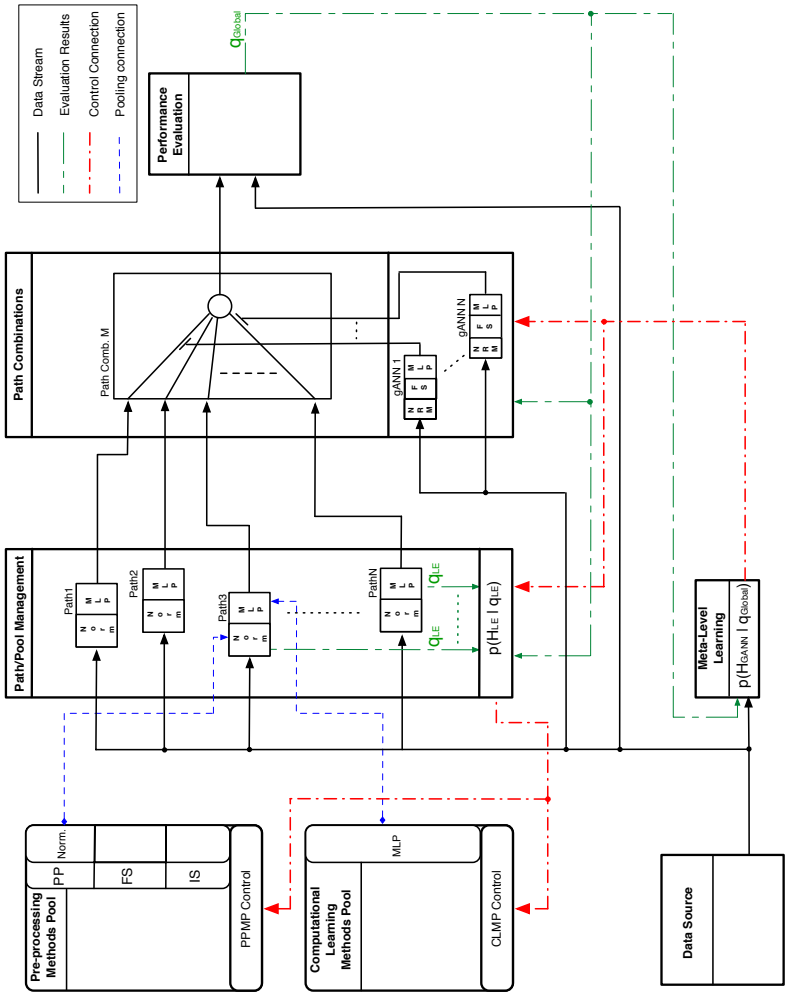

Fig. 2. The Learnt Topology Gating Artificial Neural Network

The proposed approach can easily be presented in the context of the architecture from Fig. 1 and be described as an instance of this architecture. In this simple case, there are only few techniques necessary within the pre-processing and computational learning methods pools, namely the normalisation, and feature selection methods in the pre-processing pool and MLP in the computational learning pool. The local experts, which correspond to the transformation paths, are managed in the Path/Pool Management module. New local experts added to this module are built having the topology generated in accordance to the probability distribution $P(H_{LE}|q)$, which is being managed in the control part of the paths module. In this work we have not used any pruning mechanism, using which the local experts could be removed. Going further, the weighted sum of a particular set of local experts responses is built in the Path Combination module. There is only one combination of all available predictions present. The weights of this combination are set in the control part of the module. In this case we use the GANNs, which predict the weights dependent on the input data, present there. The topology of the gating networks is controlled from the meta-level learning part of the architecture, where the probability distribution of the number of hidden units is controlled. This control is based on the evaluation of the performance provided by the Performance Evaluation module. The LTGANN instance of the general architecture is shown in Fig. 2.

IV. Experimental Evaluation

The LTGANN was applied to two industrial data sets. The results of the experiments are presented in this section.

A. Drier Data Set

The target values of this data set are laboratory measurements of the residual humidity of the process product.

The data set has 19 input features, most of them being temperatures, pressures and moistures measured within the process plant. The output feature is, as already described, the humidity of the process product. The data set consists of 1219 data samples covering almost seven months of the operation of the process. It consists of raw unprocessed data as it were recorded by the process information and measurement system.

The experiments were carried out using two-fold cross-validation. A justification for using two-fold CV is that the training data for the gating networks use the prediction error of the local experts on the validation data and thus using two folds balances the number of the sizes of the training data set for the local experts and the gating networks.

The interval of hidden units numbers is $[1, 10]$ for both, the local experts and the gating networks. These values were found during preliminary experiments. The following two figures (Fig. 3 and 4) show the probability distribution of the hidden unit number after 200 training steps, i.e after training 200 local experts and GANNs. One can observe that in

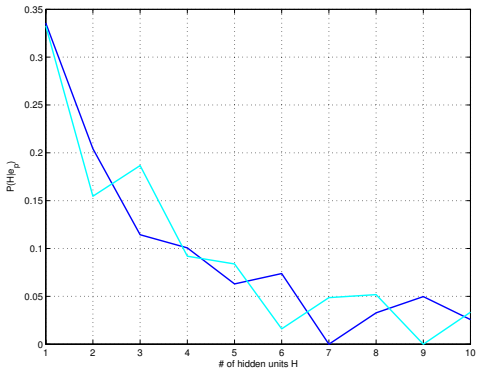

Fig. 3. Probability distribution $P(H_{LE}|q_{LE})$ of the local experts hidden units number after 200 learning steps for the two CV folds.

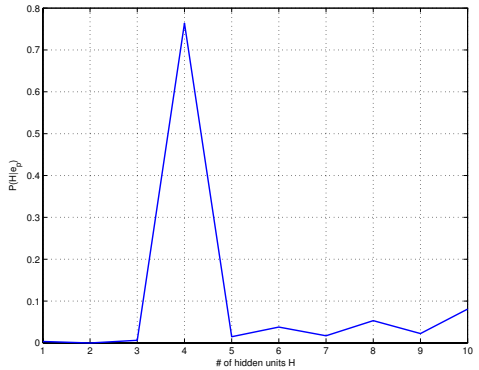

Fig. 4. Probability distribution $P(H_{GANN}|q_{GANN})$ of the GANN hidden units number after 200 learning steps.

the case of the local experts, there is a preference for rather simple topologies which in general seem to achieve better performance. In the case of the GANN, networks with four hidden units achieve the best performance.

To be able to asses the performance of the LTGANN approach, it is on one hand compared to the performance of the particular local experts (referred to as 'Local Experts' in the following figures) and on the other hand to the base-line mean combination approach of the local experts, where the combination is carried out using the mean value of the predictions (referred to as 'Mean Comb.').

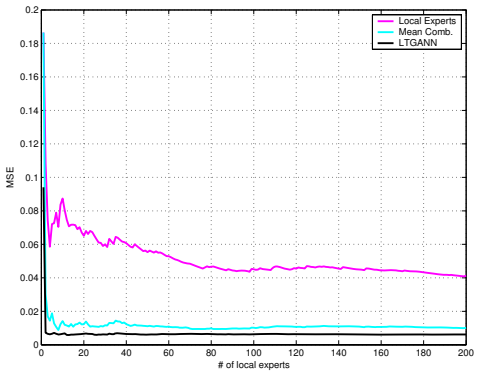

Fig. 5. MSE performance of the LTGANN compared to the mean combination approach and to the averaged performance of the local experts.

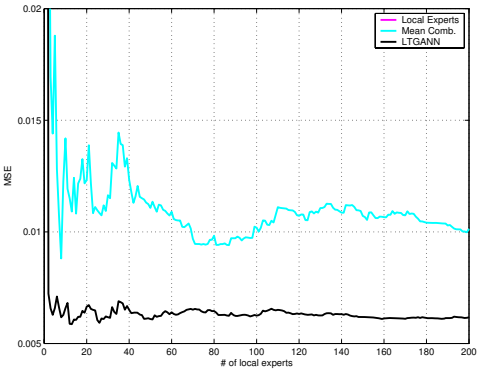

Fig. 6. Detailed view of the MSE performance of the LTGANN and of the mean combination approach.

Figures 5 and 6 present the Mean Squared Error (MSE) of the LTGANN compared to the other two base-line approaches as a function of the number of involved local experts. The three MSE curves for calculated using the

following equations:

Local Experts:

$$MSE_{LE} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_p^i(x_{test}(j)) - y(j)\right]^2$$

Mean Comb.:

$$MSE_{MC} = \frac{1}{M}\sum_{j=1}^{M}\left[\left(\frac{1}{N}\sum_{i=1}^{N}y_p^i(x_{test}(j))\right) - y(j)\right]^2$$

LTGANN:

$$MSE_{LTGANN} = \frac{1}{M}\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{M}w^i y_p^i(x_{test}(j))\right) - y(j)\right]^2,$$

where y are the correct target values, x_{test} the input sample from the test set, y_p^i the prediction of the particular local expert i, w^i the weights predicted by the GANN, N the number of local experts which is 200 for the experiments presented in this work and M the number of test samples.

One can observe a convergence of the MSE curve with increasing number of involved local experts. After a certain number of combined local experts the performance remains stable. One can observe similar behaviour also for the mean combination approach ('Mean Comb.'), but in this case the convergence value is higher compared to the one of LTGANN and it takes more learning steps, i.e. there are more local experts needed, till the model approaches the convergence value (see Fig. 6). Fig. 6 also shows that the LTGANN model performance stability is higher than that of the mean combination. Another effect which can be observed from Fig. 5, more precisely from the 'Local Experts' curve which is showing the averaged performance of the local experts, is the effect of the learning of the optimal topology which is demonstrated by the decrease of the curve with increasing number of involved local experts. The probability distribution of the number of hidden units is updated and thus improves with each added local expert.

Figures 7 and 8 show the boxplot statistical representation of the MSE curves presented in Fig. 5. The leftmost box shows the MSE statistics of the local experts without doing any combination. One can observe several model properties from the boxplot representation. For example the high variance of the single local expert results. This has its origin in the fact that due to problems with local minima artificial neural networks are prone to give sub-optimal performance on the test data. Unless one explores the whole parameter space of the weights, there is no guarantee of finding the global minimum of the training error but even if the global minimum is found it can, because of overfitting of the model, happen that the performance on the test data remains still sub-optimal. Especially from Fig. 8 one can see the superior performance of the model combinations, the median values of their MSE curves are far bellow that of the non-combined models. This figure also shows that there are particular local experts which perform better than LTGANN but as it was already mentioned it is virtually impossible to find these models during the training. Fig. 8 also confirms that the LTGANN achieves significantly better performance than the mean combination technique. The size of the LTGANN box is smaller than the other boxes. This demonstrates the fact that once the curve nears the convergence value it remains stable. Finally to be able to judge the performance of the

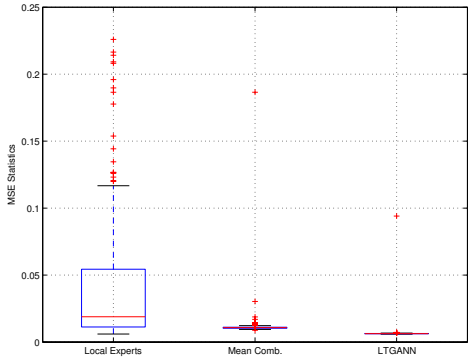

Fig. 7. Boxplots of the MSE curves.

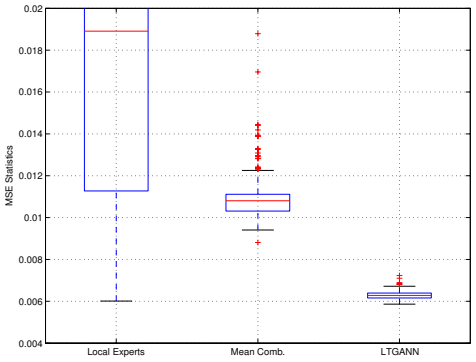

Fig. 8. Boxplots of the MSE curves, details of the combination approaches.

approach presented in this paper, Fig. 9 shows the correct target values and the prediction of the LTGANN model. Another effect, which can also be observed in Fig. 9, is the deterioration of the model performance with increasing time. The final model performs better for the first half of the test samples. For the second half, the model performance starts to drop and the model is no more able to predict the data as accurately as for the first half. This shows a clear need for retuning or adaptation of the model. The adaptation possibilities of the general architecture, presented in Section II, are discussed in [15]. As it was shown here, the adaptation of the model is vital for maintaining its performance and therefore, further, more complex instances of the architecture will focus on the implementation of efficient adaptation

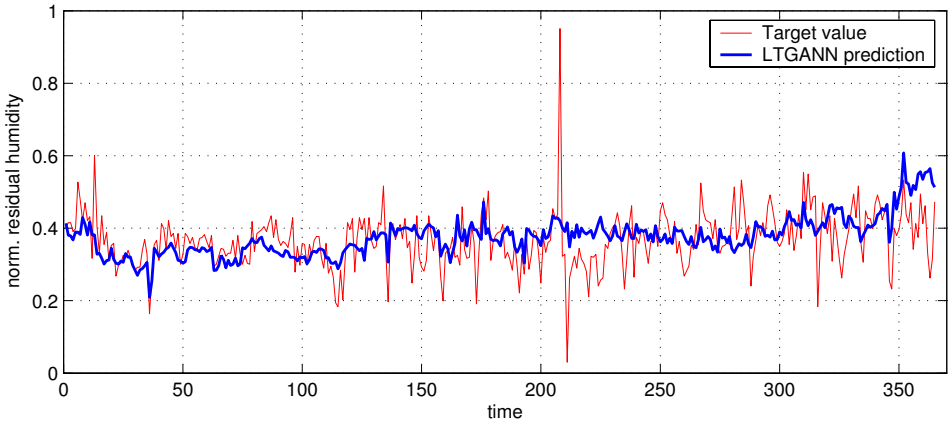

Fig. 9. The target values and the prediction of the LTGANN.

mechanisms which are out of the scope of this paper.

B. Debutanizer Data Set

This data set is publicly available[1], and described in [20]. The data was recorded in a debutanizer column which is a part of the desulfuring and naphtha splitter plant. The data set consists of seven manually pre-selected input features, consisting mainly of temperature, pressure and reflux measurements at different positions within the column. The target value is the concentration of butane at the output of the column.

For this experiment the same methodology as for the previous one was applied. In this case, the target feature is very hard to model which is demonstrated by the weak performance of the local expert ANNs. Again, using the model combination approaches the performance of the regression model can be significantly improved, as can be seen in Figures 10 and 12. For this data set the LTGANN method again achieves significantly better performance if compared to the base-line mean combination method (see Figures 11, 13) and similar conclusions to the previous experiment can be drawn. An interesting fact in the case of this experiment is that the MSE performance of the combination methods can be better values than the MSE performance of the best local expert ANN.

V. Conclusions

This paper describes a meta-learning regression model called Learnt Topology Gating Artificial Neural Networks (LTGANN). LTGANN is based on the Gating Artificial Neural Networks (GANN) method which is a well established approach to model combination. In this work GANN was modified to allow the training of the gating networks using the standard back-propagation algorithm. This has the advantage that the model can be easily enlarged by adding new local experts without the need to make any changes to the already existing networks. The gating networks are trained to link the performance of the local experts to the position of a sample in the input space. This provides a performance map which can be used for estimation of the performance of the particular local expert given the input

[1]Data set available at: www.springer.com/1-84628-479-1

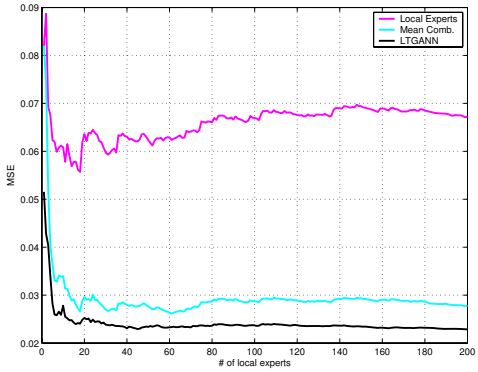

Fig. 10. MSE performance of the LTGANN compared to the mean combination approach and to the averaged performance of the local experts.

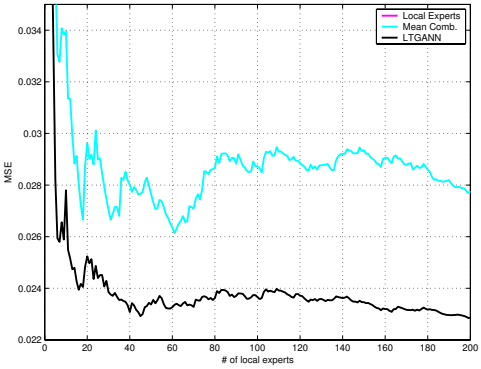

Fig. 11. Detailed view of the MSE performance of the LTGANN and of the mean combination approach.

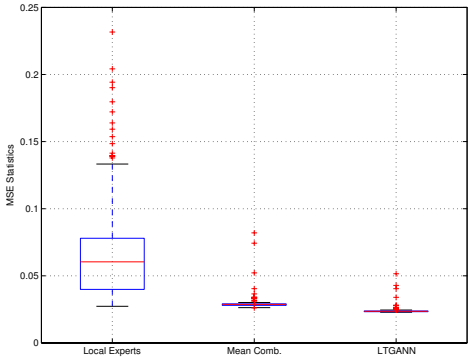

Fig. 12. Boxplots of the MSE curves.

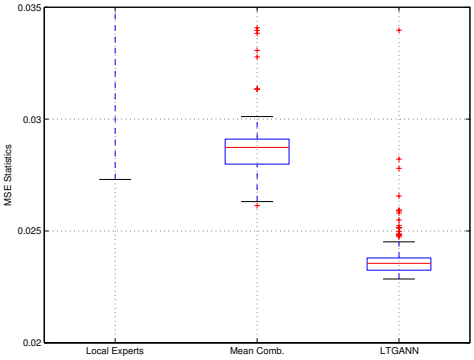

Fig. 13. Boxplots of the MSE curves, details of the combination approaches.

sample. Another key point of this work is the provided flexibility for the ANN topology selection. There is no need to define the exact number of hidden units either for the local experts or for the gating networks. The model learns well-performing topologies and gives preference to these when generating new local experts and gating networks. The LTGANN model is presented as an instance of a more general architecture for the building of regression models. As the architecture is very general this instance is only the first step towards a more complex and holistic model which will involve more sophisticated approaches to data modelling. The discussed model architecture has been developed with the focus on application within the process industry environment, which provides the possibility to deal with application oriented issues common to a large number of industrial applications. Applying LTGANN to two industrial problems has shown a significant performance gain using this method when compared to the performance of a baseline model combination method.

REFERENCES

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1995.

[2] Y. Chauvin and D. E. Rumelhart, *Back Propagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates, 1995.

[3] D. W. Aha, "Generalizing from case studies: A case study," in *Proceedings of the Ninth International Conference on Machine Learning*, 1992, pp. 1–10.

[4] P. Brazdil, "Data transformation and model selection by experimentation and meta-learning," *Workshop NotesUpgrading Learning to the Meta-Level: Model Selection and Data Transformation, number CSR-98-02 in Technical Report*, p. 1117.

[5] C. Giraud-Carrier, "Beyond predictive accuracy: what?" *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, pp. 78–85, 1998.

[6] T. G. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[7] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 1, pp. 66–75, 1994.

[8] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[9] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-IEEE, 2004.

[10] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[11] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information Systems*, vol. 7, no. 1, pp. 1–10, 2000.

[12] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[13] M. I. Jordan and A. G. Barto, "Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks," *COGNITIVE SCIENCE*, vol. 15, pp. 219–250, 1991.

[14] R. Jacobs, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[15] P. Kadlec and B. Gabrys, "Application of computational intelligence techniques to process industry problems," in *Knowledge Processing and Reasoning for Information Society*, N. Nguyen, G. Kolaczek, and B. Gabrys, Eds. Wroclaw: EXIT Warsaw, 2008.

[16] C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine Learning*, vol. 54, no. 3, pp. 187–193, 2004.

[17] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11, no. 1, pp. 11–73, 1997.

[18] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 10, no. 8, pp. 2047–2084, 1998.

[19] E. Frank, M. Hall, and B. Pfahringer, "Locally weighted naive bayes," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 249–256.

[20] L. Fortuna, *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer, 2007.