Robust Fuzzy Modeling and Symbolic Regression for Establishing Accurate and Interpretable Prediction Models in Supervising Tribological Systems

Edwin Lughofer[2], Gabriel Kronberger[1], Michael Kommenda[1], Susanne Saminger-Platz[2],
Andreas Promberger[3], Falk Nickel[3], Stephan Winkler[1] and Michael Affenzeller[1]

[1] *Heuristic and Evolutionary Algorithms Laboratory, School of Informatics, Communications and Media, Softwarepark 11, 4232 Hagenberg, Austria*

[2] *Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, 4040 Linz, Austria*

[3] *Miba Frictec, Peter-Mitterbauer-Str., 4661 Roitham, Austria*

Keywords: Tribological Systems, Robust Fuzzy Modeling, Generalized Takagi-Sugeno Fuzzy Systems, Symbolic Regression, Multi-objective Accuracy/Complexity Tradeoff, Enhanced Regularized Learning.

Abstract: In this contribution, we discuss data-based methods for building regression models for predicting important characteristics of tribological systems (such as the friction coefficient), with the overall goal of improving and partially automatizing the design and dimensioning of tribological systems. In particular, we focus on two methods for synthesis of interpretable and potentially non-linear regression models: (i) robust fuzzy modeling and (ii) enhanced symbolic regression using genetic programming, both embedding new methodological extensions. The robust fuzzy modeling technique employs *generalized Takagi-Sugeno fuzzy systems*. Its learning engine is based on the *Gen-Smart-EFS approach*, which in this paper is (i) adopted to the batch learning case and (ii) equipped with a new enhanced *regularized learning scheme* for the rule consequent parameters. Our enhanced symbolic regression method addresses (i) direct gradient-based optimization of numeric constants (in a kind of memetic approach) and (ii) multi-objectivity by adding complexity as a second optimization criterion to avoid over-fitting and to increase transparency of the resulting models. The comparison of the new extensions with state-of-the-art non-linear modeling techniques based on nine different learning problems (including targets wear, friction coefficients, temperatures and NVH) shows indeed similar errors on separate validation data, but while (i) achieving much less complex models and (ii) allowing some insights into model structures and components, such that they could be confirmed as very reliable by the experts working with the concrete tribological system.

1 INTRODUCTION

Friction models have been studied for more than hundred years (Berger, 2002) and are central for understanding and accurately describing tribological systems which occur in almost all mechanical systems. The main difficulty in modeling friction is that it is a very complicated phenomenon which depends on a large variety of parameters including mechanical properties (e.g. surface roughness and hardness, lubrication), load (e.g. pressure, energy and sliding speed), and environmental conditions (e.g. humidity, temperature) (Berger, 2002). Additionally, friction is a dynamic phenomenon as abrasive effects and material deterioration as well as temperature changes strongly influence friction characteristics (De Wit et al., 1995).

Mathematical models of friction systems derived solely from mechanical principles (De Wit et al., 1995) are strongly simplified and therefore limited. The forces acting in tribological systems on a micro- and nano-level depend on the surface characteristics of the friction materials as well as on the lubrication and are difficult to describe mathematically (Sellgren et al., 2003) (Berger, 2002). Many of these factors are hard to capture in a mathematical model based solely on physical principles (Aleksendric and Carlone, 2015). In particular, we study friction systems as applied in clutches employed in power-trains (Senatore et al., 2011). Relevant influence factors include

Lughofer, E., Kronberger, G., Kommenda, M., Saminger-Platz, S., Promberger, A., Nickel, F., Winkler, S. and Affenzeller, M.
Robust Fuzzy Modeling and Symbolic Regression for Establishing Accurate and Interpretable Prediction Models in Supervising Tribological Systems.
DOI: 10.5220/0006068400510063
In *Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016) - Volume 2: FCTA*, pages 51-63
ISBN: 978-989-758-201-1

the composition and mechanical properties of the employed friction material, the oil, or the geometry of groovings on friction material.

The overall objective is the improvement of the design process of these tribological systems by using accurate models for the most important characteristics. These models should be included in an expert system for the virtual design and dimensioning of friction systems to support engineers (Aleksendric and Carlone, 2015), thus they should be not too complex and interpretable.

1.1 State-of-the-Art

Friction models can be roughly categorized into analytical models derived from physical principles and purely empirical models (Berger, 2002). Analytical models such as the one given by (De Wit et al., 1995) are usually rather sophisticated and describe friction forces based on surface characteristics of the friction materials. Drawbacks of these models are the high complexity as well as the limited applicability in real scenarios, because important effects such as non-linear dynamics or abrasive effects are not captured by the models. Models based on finite elements simulation, such as the model described in (Sellgren et al., 2003), are computationally expensive and also have the drawback that relevant effects in practical applications are not described accurately.

Empirical models have predictive capabilities but do not provide a detailed, physically correct and general description of friction. Depending on the simulation objectives these empirical models can be rather simple, such as the non-linear numerical model presented in (Loh et al., 2000) which estimates the coefficient of friction based solely on load parameters. An example for a more complex numerical model that also include effects of surface characteristics is given in (Xiao et al., 2007). In both of these cases, the model structure has been manually (and not automatically) defined based on experience and intuition about the main effects and the parameters of the model have been optimized to fit the model to measurements (which requires high development effort).

Recently, especially artificial neural networks have been applied successfully in multiple occasions for data-based modeling of tribological systems (Aleksendric and Carlone, 2015). For example, neural networks have been used to predict wear of brake friction materials where the complete formulation of the friction material, important manufacturing conditions, as well as load parameters, sliding velocity and the temperature have been used as input for the network (Aleksendri, 2010). Hosenfeldt and colleagues describe that they trained an artificial neural network model that “can predict the tribological behaviour of camshaft and bucket tapped systems” (Hosenfeldt et al., 2014), and state that they achieved “a deviation of 8%” which “is a very good result, especially when considering that the measurement error with reference to friction is 5%” (Hosenfeldt et al., 2014). On the other hand, ANNs deliver models which typically appear as black boxes with high structural complexity. This makes them unattractive and not usable within an interpretable expert system.

1.2 Our Approach

We design prediction models for a variety of *nine* important system variables during the design and dimensioning of tribological systems: four different measurements of the coefficient of friction, two different measurements of wear, two temperature measurements, vibration and harshness (NVH). We concentrated on two architectures for data-driven regression models, Takagi-Sugeno-type (TS) fuzzy systems (Takagi and Sugeno, 1985), as recently introduced in generalized form (Lemos et al., 2011), and symbolic regression models containing functional terms achieved through genetic progamming (GP) (Affenzeller et al., 2009), which both allow interpretation and inspection of the resulting models, e.g., for gaining better insights into the system behavior. We propose two major enhancements in our learning engines to cope robustly with binary input variables and high noise levels.

In case of (generalized) TS fuzzy systems, we integrate a convex combination of Lasso and ridge regression for optimizing the linear consequent parameters. This is leaned on the concept of *elastic net regularization* (Hastie et al., 2010) (Zou and Hastie, 2005), but adopted to the specific (locally weighted) optimization problem for generalized Takagi-Sugeno fuzzy systems for the first time. Such a more sophisticated learning concept is expected to enhanced the robustness of the solutions. For learning the rule structure and the rules’ antecedent parts, we employ the *Gen-Smart-EFS* technique, originally developed for streaming data in (Lughofer et al., 2015), which we adjust for the batch off-line case by specific two-stage iterative optimization procedure (see Section 2.2.2). Joining rule antecedent and enhanced regularized consequent learning, the new method is termed as *Robust-GenFIS* (short for *Robust Generalized Fuzzy Inference Systems*) and described in detail in Section 2.2.2.

In case of symbolic regression models, the genetic programming (GP) approach in (Affenzeller et al.,

2009) is extended by gradient-based optimization of numeric constants (Kommenda et al., 2013) and by multi-objective optimization of complexity and estimation accuracy (Kommenda et al., 2016). These extensions are firstly combined to a joint enhanced optimization and applied to a real-world scenario in this paper. In traditional GP-based symbolic regression numeric constants are optimized side-by-side with the model structure relying solely on evolutionary operators of crossover, mutation and selection to find optimal constants. Incorporating a gradient-based optimizer for constants increases the efficiency of GP-based symbolic regression and can thus improve final solution quality. A drawback of GP-based symbolic regression in practical applications is the tendency to produce very large ("bloated") solutions (Langdon et al., 1999). Selection in combination with crossover operations has the effect that ineffective parts accumulate in solution candidates and therefore the structural complexity of solution candidates increases. We thus employ multi-objective optimization to produce a set of Pareto-optimal solutions in the context of GP, where one dimension corresponds to the model complexity and the other to the model error — see Section 2.3.

In this article, we compare the newly extended fuzzy and symbolic regression modeling approaches with other state-of-the-art (SoA) techniques such as support vector regression (Smola and Vapnik, 1997), random forests (Breiman, 2001) or gradient boosted trees (Friedman, 2001), as well as with related and widely used fuzzy system extraction algorithms such as *LoLiMoT* (Nelles, 2001), *genfis2* (modified version of it) (Chiu, 1994) and *FLEXFIS* (Lughofer, 2008), in terms of predictive accuracy as well as model complexity.

The major finding is that linear models are insufficient for modeling the main characteristics of tribological systems while the newly proposed robust fuzzy modeling and enhanced symbolic regression approaches produce significantly more accurate models (Section 4.1), which satisfactory for the usage in virtual design of tribological systems. The predictive quality on separate test data indeed turned out to be similar (similar error ranges) to the predictive quality achieved by related SoA methods. However, the models produced with our methods have much lower complexity (see Section 4.2) than those achieved by related non-linear techniques. They also can be naturally represented in form of linguistically readable fuzzy rules and in form of symbolic, physically interpretable terms; both can be casted into a tree-like structure, achieving synergy on structural level, which induces the possibility of a direct and transparent model comparison for operators and experts (Section 4.3).

Table 1: Variables in the preprocessed data set.

Number	Type	Identification
22	continuous features	$x_1 \ldots x_{22}$
3	binary features representing the friction mat. type	$b_1 \ldots b_3$
2	binary features representing the test procedure	Source_1, Source_2
25	binary features representing the friction material	$\mathrm{Mat}_1 \ldots \mathrm{Mat}_{25}$
10	binary features representing the grooving type	$\mathrm{Gro}_1 \ldots \mathrm{Gro}_{10}$
11	binary features representing the oil	$\mathrm{Oil}_1 \ldots \mathrm{Oil}_{11}$
4	continuous measurements of coeff. of friction (target)	$\mathrm{Cf}_1 \ldots \mathrm{Cf}_4$
2	continuous measurements of wear (target)	Wear_1, Wear_2
1	ordinal rating of noise, vibration, and harshness (target)	NVH
2	continuous measurements of temperature (target)	Temp_1, Temp_2

2 METHODOLOGY

2.1 Data Acquisition and Preparation

The data used for modeling have been acquired through extensive testing of many different friction plates on tribological test benches. Each of the friction plates has been tested using one of two different test programs. The first test program is designed to test and measure coefficient of friction and wear under different loads, the second test program is designed to test noise and vibration characteristics. In total, data from almost 1300 individual test runs are considered. The combined data set contains measurements from test procedures for many different combinations of friction materials and oils, whose major influence factors are friction material, the oil, design parameters of the friction plate as well as temperature.

In a preprocessing phase we cleaned the data set by removing obvious measurement errors and incomplete test runs so that the resulting data set contains only valid measurements for the most commonly used combinations of friction material and oil. To prepare for the modeling phase we created additional binary indicator features for each friction material, grooving, and oil. After preprocessing the dataset contains variables shown in Table 1. Depending on the target variable the number of rows in the data sets is between 300 and 4000 rows.

2.2 Robust Fuzzy Modeling

2.2.1 Data-driven Fuzzy Modeling Architecture

The advantage of data-driven fuzzy systems among other types of soft computing based model architectures and machine learning techniques is their joined characteristics of (i) universal approximation property (Castro and Delgado, 1996), being able to model any implicitly contained non-linear relationship with an arbitrary degree of accuracy and (ii) linguistic interpretability (Lughofer, 2013), allowing some insights into the system dependencies and variable interrelations. While the antecedent parts of the rules embed the linguistically interpretable description of feature interrelations, there are various possibilities how to design the consequent parts, inducing different types of fuzzy systems (Lughofer, 2011).

For the robust modeling intentions in this paper, we will employ the (recently introduced) *generalized* version of TS fuzzy systems (Lemos et al., 2011), which induced more compact rule bases with similar or even less model errors than conventional TS fuzzy systems during past studies (Lughofer et al., 2015) because of its ability to model piecewise local correlations between variables in a more compact and accurate way. In the generalized case, the rules are defined by:

$$\begin{aligned} &\text{IF } \vec{x} \text{ IS (about) } \mu_i \\ &\text{THEN } l_i(\vec{x}) = w_{i0} + w_{i1}x_1 + w_{i2}x_2 + \ldots + w_{ip}x_p \end{aligned} \quad (1)$$

where l_i the hyper-plane defining the consequent of the ith rule, μ_i denotes a high-dimensional kernel function, which, in accordance to the basis function networks spirit, is given by the multivariate Gaussian distribution:

$$\mu_i(\vec{x}) = \exp(-\frac{1}{2}(\vec{x}-\vec{c}_i)^T \Sigma_i^{-1}(\vec{x}-\vec{c}_i)) \quad (2)$$

with $\vec{c}_i$ the center and Σ_i^{-1} the inverse covariance matrix of the ith rule, allowing any possible rotation and spread of the rule.

The output of a (generalized) TS system consisting of C rules is a weighted linear combination of the outputs produced by the individual rules (through the l_i's), thus:

$$\hat{f}(\vec{x}) = \hat{y} = \sum_{i=1}^{C} \Psi_i(\vec{x}) \cdot l_i(\vec{x}) \quad \Psi_i(\vec{x}) = \frac{\mu_i(\vec{x})}{\sum_{j=1}^{C}\mu_j(\vec{x})} \quad , \quad (3)$$

with $\mu_i(\vec{x})$ the rule firing degree obtained through (2).

2.2.2 Our Robust Learning Engine for Generalized TS Fuzzy Models

The learning engine is based on the generalized smart evolving fuzzy systems approach, shortly termed *Gen-Smart-EFS* (Lughofer et al., 2015), which has been designed for (fast) streaming data, thus allowing only a single-pass over incoming data samples for expanding or shrinking the structure and recursively updating the parameters. We thus adopt it here to the batch, off-line case by performing multiple passes over the entire (training) data set to optimize the positioning, shape and direction of the rules.

Rule Structure Elicitation and Initialization of Non-linear Parameters. The first phase performs a single-pass over the whole data set in order to elicit the optimal number of rules for the current problem (data set) at hand. It follows the same procedure as used in the *Gen-Smart-EFS* approach for the antecedent parts by compactly applying the following steps (with $C = 0$ initially) (see (Lughofer et al., 2015) for detailed formulation and motivation of algorithmic parts):

1. Load a new sample $\vec{x}$; if it is the first one, Goto Step 5 (there, ignoring the if-part);
2. Elicit the winning rule, i.e. the rule closest to the current sample, which is then denoted as $\vec{c}_{win}$; for the distance calculation, standard Mahalanobis distance is used (as on the right hand side in (4) below).
3. Check whether the following criterion is met (the *rule evolution criterion*):

$$\begin{aligned} min_{i=1,\ldots,C}\sqrt{(\vec{x}-\vec{c})^T\Sigma^{-1}(\vec{x}-\vec{c})} &> r_i \\ r_i = \text{fac}p^{1/\sqrt{2}}\frac{1.0}{(1-1/(k_i+1))^m} & \end{aligned} \quad (4)$$

   with p the dimensionality of the input feature space and fac an a priori defined parameter, steering the tradeoff between stability (update of an old cluster) and plasticity (evolution of a new cluster). This is the only sensitive parameter and is varied during the model evaluation phase, see Section 3.1.

4. If (4) is not met, the centre of the winning rule is updated by

$$\vec{c}_{win}(N+1) = \vec{c}_{win}(N) + \eta_{win}(\vec{x} - \vec{c}_{win}(N)) \quad (5)$$

   and its inverse covariance matrix by (the index *win* neglected due to transparency reasons):

$$\begin{aligned} \Sigma^{-1}(k+1) &= \frac{\Sigma^{-1}(k)}{1-\alpha} \\ &- \frac{\alpha}{1-\alpha}\frac{(\Sigma^{-1}(k)(\vec{x}-\vec{c}))(\Sigma^{-1}(k)(\vec{x}-\vec{c}))^T}{1+\alpha((\vec{x}-\vec{c})^T\Sigma^{-1}(k)(\vec{x}-\vec{c}))} \end{aligned} \quad (6)$$

   with N the number of samples seen so far and $\alpha = \frac{1}{k_{win}+1}$ with k_{win} the number of samples seen so far for which c_{win} has been the winning rule (cluster). The former stems from the idea in vector quantification (Gray, 1984) by minimizing the expected squared quantization error. The latter is a recursive exact update, which is analytically derived with the usage of the Neumann series (Lughofer and Sayed-Mouchaweh, 2015).

5. If (4) is met, a new rule is evolved as covering a new region in the feature space (i.e. having sufficient *novelty content*) by setting its center $\vec{c}_{C+1}$ to the coordinates of $\vec{x}$ and initialize its inverse covariance matrix Σ_{win}^{-1} by setting it to a diagonal matrix with entries 1 divided by a small fraction of the range; $C = C+1$.

Optimizing Non-linear Parameters (New Extension #1). After the initial phase is finished, a fine-tuning phase is conducted which optimizes the centers and inverse covariance matrices of the rules as initially formed in one single pass. This is achieved over multiple iterations of the whole data set by successively moving the centers and inverse covariance matrices according to (5) and (6), with

$$\eta_{win} = \frac{1}{\text{iterations}+1} \qquad \alpha = \frac{1}{\text{iterations}+1} \tag{7}$$

This is in accordance with the Robbins-Monroe conditions and thus assures convergence to the optimal solution in terms of the quantization error.

The iterations are performed as long as the stopping criterion is not fulfilled. We have chosen the degree of change between two cluster partitions from two consecutive cycles $t-1$ and t; so, if the following condition is met

$$\sum_{i=1}^{C} \|\vec{c}_i(t) - \vec{c}_i(t-1)\| < \varepsilon, \tag{8}$$

with ε set to a small positive number, the optimization iterations are stopped.

Robust Learning of Linear Consequent Parameters (New Extension #2). Once the rule structure and antecedent parts are obtained, the aim is to estimate the consequent parameters in a way to best match predicted model outputs with observed target values. Therefore, the least squares error criterion is the most conventional choice in data-driven regression modeling as appropriate optimization problem. We use its weighted version in order to induce a *local learning* scheme, i.e. to estimate the parameters per rule separately and independently due to robustness and accuracy reasons (Lughofer, 2011) (Chapter 2).

The (locally) weighted least squares optimization problem for consequent parameters in all C rules is formalized as:

$$J_i = \sum_{k=1}^{N} \Psi_i(\vec{x}(k)) e_i^2(k) \longrightarrow \min_{\vec{w}_i} \qquad i = 1,...,C \tag{9}$$

where $e_i(k) = y(k) - \hat{y}_i(k)$ represents the error of the local linear model in the kth sample. It can be solved in closed analytical form:

$$\hat{\vec{w}}_i = (R^T Q_i R)^{-1} R^T Q_i \vec{y} \tag{10}$$

with R the regression matrix, additionally containing a column of ones for the intercept, and Q_i the diagonal weighting matrix $\in R^{NxN}$, containing the membership degrees of all N training data samples to the ith rule.

Obviously, the solution in (10) may become unstable once the matrix $R^T Q_i R$ is of low rank or even singular. The likelihood for such an occurrence increase with significant noise levels and especially when binary variables (containing only either 0s and 1s) are involved — both is the case in the application of tribological systems, especially the usage of a particular oil, grooving or friction material type results in binary inputs (compare with Figure 1). Therefore, it is indispensable to perform a regularization of the optimization problem in order to assure robust solutions. The most convenient option is to integrate Tichonov-type regularization, which leads to the form of classical ridge regression, see (Hastie et al., 2009) (Chapter 2), which has been also used before in fuzzy systems training.

Here, we go one step further and employ a generalization of ridge regression, termed as *elastic net* (Hastie et al., 2010), and adopt it for learning of consequent parameters in fuzzy systems. It incorporates a convex combination of Lasso and ridge regularization term, thus its optimization problem in the context of fuzzy systems consequent training is defined as:

$$J_i = \sum_{k=1}^{N} \Psi_i(\vec{x}(k)) e_i^2(k) + \lambda \sum_{j=1}^{p} (\alpha w_{ij}^2 + (1-\alpha)|w_{ij}|) \longrightarrow \min_{\vec{w}_i} \quad i = 1,...,C \tag{11}$$

with λ the regularization parameter and α a parameter in $[0,1]$, steering the degree of influence of the 'Lasso term' $\sum_{j=1}^{p} |w_{ij}|$ versus the 'ridge term' $\sum_{j=1}^{p} w_{ij}^2$

Obviously, the solution representation of the consequent vector $\vec{w}_i$ of (11) in a closed analytical form is given by the (convex) combination of the representations obtained by Lasso and ridge regression, thus:

$$\hat{\vec{w}}_i = (R^T Q_i R + \lambda_1 I)^{-1} (R^T Q_i \vec{y} - \frac{\lambda_2}{2} \text{sign}(\vec{w}_i)), \tag{12}$$

with $\lambda_1 = \lambda * \alpha$ and $\lambda_2 = \lambda * (1-\alpha)$ which results in a least squares problem with 2^{p+1} inequality constraints, as there are 2^{p+1} possible sign patterns $\in \{-1,1\}$ for the entries in the consequent parameter vector $\vec{w}_i$. This can be efficiently solved through a quadratic programming approach, termed as LARS-EN, see (Zou and Hastie, 2005).

2.3 Enhanced Symbolic Regression

Symbolic regression is a nonparametric regression and function discovery method (Koza, 1992), where the generated prediction model is a mathematical expression. An advantage of symbolic regression is that despite the generated models are able to express nonlinear relationships, they can be interpreted and inspected by domain experts (Kronberger, 2011). Furthermore, because the model is a mathematical expression, it can be transformed and manipulated, easily incorporated into expert systems (Affenzeller et al., 2014) — see Figure 1 for an example.

$$f(x) = x_1 x_2 + 2.5\, x_3^2 + \frac{x_4 + 5.0}{0.8\, x_5 - x_1}$$

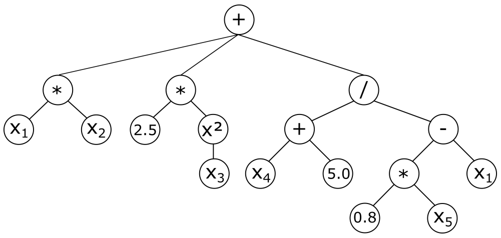

Figure 1: Symbolic regression model represented as mathematical expression and the equivalent symbolic expression tree.

Tree-based genetic programming (GP) (Koza, 1992; Poli et al., 2008), an evolutionary metaheuristic optimization method, is commonly used to solve symbolic regression problems. GP has been originally developed to evolve solution candidates that solve a given task without explicitly programming them. In the case of symbolic regression, the computer programs are mathematical expressions in the form of a symbolic expression trees predicting the dependent variable.

2.3.1 Constants Optimization in Symbolic Regression

An advantage of symbolic regression compared to other regression methods is that neither the model structure nor its parameters nor the used variables are predetermined. As a consequence, solving a symbolic regression problem can be divided into three separate but correlated subproblems:

- Selecting the appropriate variables
- Detecting the best suited model structure for the selected variables
- Determining the numerical constants of a model structure

Typical symbolic regression systems tend to solve those three subproblems at the same time by combining variables, numerical constant and the functions forming the model structure in the symbolic expression tree. This has the disadvantage that although the appropriate model structure and variables are identified, the model might have a high prediction error due to wrong numerical constants. For example, when the function $y = 5\,x^2 - x + 2$ should be identified, a possible candidate solution $f(x) = -2\,x^2 + 2.4\,x - 3$ using the appropriate variables and model structure would have a high prediction error due to the wrong numerical constants ($[-2, 2.4, -3]$ instead of $[5, -1, 2]$).

One of the first attempts to diminish the effects of wrong numerical constants has been the introduction of linear scaling terms in symbolic regression (Keijzer, 2003). For every generated model scaled predictions in the form of $a * f(x) + b$ instead of the raw predictions $f(x)$ are used for comparison with the dependent variable. Therefore, it is not necessary anymore to identify the correct scale and offset for the prediction models, as these can be calculated by a simple linear transformation. Other methods for determining numerical constants in symbolic regression range from using heuristic methods such as evolution strategies (Alonso et al., 2009) or differential evolution (Mukherjee and Eppstein, 2012) to gradient-based optimization (Topchy and Punch, 2001).

We use a gradient-based optimization method, similar to (Topchy and Punch, 2001), and combine it with linear scaling for tuning numerical constants of symbolic regression models (Kommenda et al., 2013). The Levenberg-Marquardt algorithm is used for tuning of numerical constants β by minimizing the least squares functional: $Q(\beta) = \sum_{i=0}^{n} (y - f(x_i, \beta))^2$. Starting from an initial guess for β, which are the current numerical values of the expression tree, the constants are iteratively adapted using the current gradient information w.r.t β that can be efficiently calculated by automatic differentiation (Rall, 1981). The Levenberg-Marquardt algorithm only finds a local optimum. In combination with evolutionary search, it leads to a kind of memetic approach, where global and local optimization is intervened to find better solutions more quickly. The final results of symbolic regression with constants optimization are more accurate compared to symbolic regression without constants optimization (Kommenda et al., 2013). The rational behind it is that although fewer models are generated and evaluated, the search direction for the algorithm becomes clearer, because it can focus on building the best suited model structure with appropriate variables.

2.3.2 Multi-objective Optimization of Accuracy and Complexity

Symbolic regression is in general performed as a single-objective optimization, where only the prediction accuracy of the models is considered as an op-

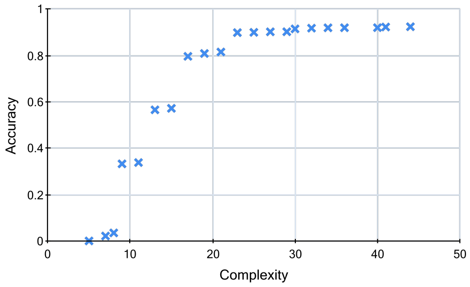

Figure 2: Pareto-front evolved by NSGA-II showing the trade-off between accuracy and complexity for symbolic regression models.

timization objective. This may weaken the compactness and transparency, therefore we switched from single-objective to multi-objective optimization, where next to the accuracy the complexity is optimized as well. The result is a whole Pareto-front of models that shows the trade-off between accuracy and complexity (Smits and Kotanchek, 2005). An exemplary Pareto-front is depicted in Figure 2, where the accuracy is measured as the correlation coefficient between the dependent variable y and the model's prediction $f(x)$ and the complexity is measures as the symbolic expression tree length.

The non-dominated sorting genetic algorithm (NSGA-II) (Deb et al., 2002) is used as multi-objective optimization algorithm to generate the Pareto-front of symbolic regression models. Similar to standard genetic programming for single-objective symbolic regression the models are encoded as expression trees and initialization, recombination and mutation work in the same way. Before the prediction accuracy of a model is evaluated its numeric constants are improved (see Section 2.3.1) and afterwards its complexity is calculated. The benefits of this method are that the generated models are in general simpler, thus easier to interpret, and the occurrence of bloat (Poli, 2003), an increase in tree length without accompanying accuracy improvement, is less likely.

3 EXPERIMENTAL SETUP

3.1 Test Protocol and Evaluation Strategy

Cross-validation (CV) was used for validation of the learning methods over all their tunable parameters (see subsequent section), whereas a deterministic assignment of measurements to folds has been used. We therefore ordered tests by time and assigned them to ten folds using a round robin principle. This was necessary in this case as simply shuffling the data would have delivered a too optimistic CV performance.

The parameter setting achieving the minimal CV error over the whole parameter grid (see below) was applied to train a final model on the whole training data set and to test it on an independent test set — this was repeated for each of the nine targets (listed in Table 1).

We use the relative mean of absolute errors $\frac{1}{Range(y)}\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$ to express model accurracy on the separate test data sets. The separate test sets are drawn from the complete data sets by using the latest (timely recorded) 30% of all values.

3.2 Parameters for Fuzzy Methods

For the fuzzy modeling variants, the following parametrization grids has been used within a 10-fold CV procedure:

- For all methods: iteration over $q=\{1,\ldots,p\}$ input features, either in form of original variables or in form of principal components (see below): in the qth iteration, the first q inputs are used for fuzzy modeling from the ranked list.
- *Rob-GenFIS* (this paper): iteration over $fac=\{0.5,\ldots,3.0\}$ in step of 0.35 in order be more or less conservative in rule evolution criterion; iteration over $\alpha=\{0,\ldots,1\}$ in step of 0.1 to balance lasso versus ridge regression in consequent learning.
- *FLEXFIS* (Lughofer, 2008): iteration over $vigi=\{0.1,\ldots,0.9\}$ in order to be more or less conservative in rule evolution criterion.
- *Genfis2 loc* (Chiu, 1994) (Lughofer, 2011): iteration over $radius=\{0.1,\ldots,0.9\}$ (denoting the range of influence of rules) in steps of 0.1 in order be more or less conservative in rule evolution criterion.
- *LoLiMoT* (Nelles, 2001): iteration over $maxNumLL=\{1,\ldots,15\}$ in steps of 1, i.e. over the maximal number of local linear models allowed as outcome of the splitting operation.

The dimensionality reduction of the (very) high-dimensional input space (up to 60 inputs for the various targets) turned out to be indispensable for preventing severe curse of dimensionality effects and over-fitting. Therefore, we conducted two variants, one based on partial least squares (PLS) (Haenlein and Kaplan, 2004), which transforms the input space into a principal component space by successively looking for directions in the data to best explain the variance in the target, the other performs a ranking

Table 2: Parameter settings for symbolic regression.

Settings	Single-objective Symb. Regr.	Multi-objective Symb. Regr.
Algorithm	OSGA	NSGA-II
Objective function	$\max R^2$	$\max R^2$, min complexity
Constants optim.	10 iterations	5 iterations
Maximum tree length	50 nodes	50 nodes
Allowed functions	$+, -, *, /$	$+, -, *, /, e^x, \log(x)$
Allowed terminals	constant, constant $*$ variable	constant, constant $*$ variable
Population size	1000	1000
Tree Initialization	Probabilistic Tree Creator 2	Probabilistic Tree Creator 2
Parent selection	Gender specific selection	Crowded tournament selection
Crossover probability	100%	90%
Crossover operation	Subtree swapping	Subtree swapping
Mutation probability	25%	25%
Mutation operations	Change node, Shake node+tree, Remove and replace branch	Change node, Shake tree, Remove and replace branch
Termination	Generations > 50 Selection pressure > 100	Generations > 200

of the original features based on a modified variant of forward selection — as also successfully used in (Cernuda et al., 2011) in combination with fuzzy systems training.

3.3 Parameters for Symbolic Regression

We have tested two variants of symbolic regression, single-objective symbolic regression using an offspring selection genetic algorithm (OSGA) and multi-objective symbolic regression solved by NSGA-II. The algorithm parameters of both variants are described in Table 2 and have been chosen according to prior experience with the algorithms. The reason that parameters have been manually chosen instead of determining them by using grid-search in combination with cross validation is that due to the stochastic nature of symbolic regression getting reliable estimates of the effects of parameters is hardly possible. The whole training partition has been used for learning the symbolic regression models and 50 repetitions of each variant have been performed. The most accurate models on the training partition have been selected and manually simplified and pruned (Affenzeller et al., 2014).

3.4 Parameters for Standard Methods

In order to achieve a fair comparison with related SoA methods in linear and non-linear regression modeling, such as *partial least squares* (Haenlein and Kaplan, 2004), *elastic net* (Hastie et al., 2010), *random forests* (Breiman, 2001), *Gaussian process regression* (Rasmussen and Williams, 2006), *support vector machines* (Smola and Vapnik, 1997), *Gradient boosted trees* (Friedman, 2001) and *Gradient boosted trees (standard)* (Friedman, 2001), we varied the most sensitive learning parameters in each of these via a fine grid and performed the same cross-validation procedure using exactly the same folds as in case of our methods.

4 RESULTS AND DISCUSSION

4.1 Model Accuracy

Table 3 shows the results achieved for all modeling methods The values are the relative mean of absolute errors (MAE in percent) on separate validation data. The best methods (with lowest perceptual MAE) for each target over the linear SoA ones, over the non-linear SoA ones as well as over our enhanced modeling techniques are highlighted in bold font. From this, it can be quite easily recognized that the linear methods are outperformed by the non-linear methods (state-of-the-art and our approaches), which has been verified in statistical preference analysis tests using Mann-Whitney test and a (default) significance level of $\alpha = 0.05$ (Mann and Whitney, 1947). Furthermore, the non-linear SoA methods seem to outperform our proposed modeling techniques for most of the cases — except for Temp$_2$ where there is a tie between enhanced symbolic regression employing NSGA-II and gradient boosting, however for targets Wear$_1$, CF$_3$, CF$_4$, and Temp$_1$ there is no statistical evidence for the out-performance. It is also remarkable that in four cases enhanced symbolic regression can outperform all fuzzy modeling variants, but in other four cases the new method *Rob-GenFIS* produces the lowest MAEs, whereas in two additional targets it also produces lower MAEs than all other fuzzy modeling variants.

From first glance, the results seem not to be satisfactory for our new proposed methods compared to related state-of-the-art approaches, however an important issue in our application is to achieve some sort of interpretability in the model outcomes. Thus, we performed a deeper investigation of the results, based on:

1. The *compactness* of the models measured by their internal structural complexity (see Section 4.2), which respects the number of inputs, the number of components and the number of parameters.
2. The *expressiveness power in interpretability* of model structures (Section 4.3).

4.2 Compactness of Models

We examined the compactness of all models achieved by the various non-linear modeling variants listed in Table 3. It is represented by the internal structural complexity of the final model and is elicited by:

- TS fuzzy models: $C * (p * 3 + 1)$ with C the number of rules and p the number of inputs; this is because each rule represents a structural component,

Table 3: Summary of results for all target variables and all methods that have been tested. The cell values are relative mean of absolute errors (MAE in percent) on separate validation data; the best methods (with lowest MAE) for each target over the linear SoA ones, over the non-linear SoA ones as well as over our enhanced modeling techniques are highlighted in bold font.

Algorithm	CF_1	CF_2	CF_3	CF_4	NVH	$Temp_1$	$Temp_2$	$Wear_1$	$Wear_2$
Linear SoA Methods									
Constant	17.22	11.99	9.99	11.57	31.18	4.82	16.51	8.46	5.27
Linear Regression	**5.70**	7.00	8.28	**6.79**	23.29	4.78	**12.56**	5.38	**3.68**
PLS	6.06	**6.53**	**8.02**	7.15	**23.10**	**4.61**	12.60	5.37	4.48
Elastic Net	5.85	6.83	8.46	7.82	23.90	4.80	13.40	5.24	4.32
Elastic Net + PLS	6.11	6.79	8.66	7.48	24.10	4.80	12.70	**5.15**	4.11
Non-Linear SoA Methods									
SVM-RBF	4.97	**5.04**	**6.90**	5.89	16.65	**3.78**	**11.37**	5.18	3.52
Random Forest	**4.19**	5.05	7.22	5.90	17.83	4.20	11.70	**4.73**	3.32
GPR SEard	4.79	5.18	7.43	6.21	18.62	4.01	12.13	4.78	3.37
GBT	4.56	5.05	7.15	5.83	18.04	4.24	11.81	4.81	**3.18**
GBT standard	4.44	5.05	7.03	**5.64**	**15.92**	3.82	11.61	4.89	3.44
Enhanced Symb. Regr									
Symbolic Regr. OSGA	5.34	6.95	8.05	6.72	20.56	4.12	12.08	5.64	3.28
Symbolic Regr. NSGA-II	5.34	6.64	**7.75**	**6.11**	20.74	4.42	**11.75**	5.44	**3.18**
Fuzzy Modeling Variants									
FLEXFIS + FS	5.91	6.73	8.45	7.62	21.30	4.00	12.00	5.30	3.34
FLEXFIS + PLS	10.30	6.16	8.43	20.30	21.00	4.48	11.90		3.48
Rob-GenFIS + FS	5.79	5.56	9.31	7.54	21.90	4.19	14.40	5.92	3.93
Rob-GenFIS + PLS	**5.01**	**5.53**	7.94	7.08	**19.40**	**3.94**	11.90	5.60	3.36
Genfis2 loc + FS	5.52	5.76	8.73	8.79	24.60	4.19		**5.13**	3.32
Genfis2 loc + PLS	5.39	5.80	7.96	6.71	21.20	4.13	13.70	5.30	3.54
LoLiMoT +FS	5.85	5.85	8.46	6.34	21.30	4.39	12.10	5.19	3.20
LoLiMoT + PLS	5.61	5.88	7.69	6.93	19.60	4.61	12.60	5.40	3.41

whereas each of the p antecedent parts is composed of two parameters (the input variable and the assigned fuzzy set with its associated linguistic term) and each consequent part is composed by a hyper-plane with $p+1$ parameters (p inputs and one intercept). This complexity definition shows up in a structural representation of TS fuzzy models which is also in full accordance with the representation obtained by symbolic regression, as will be further analyzed in Section 4.3.

- Symbolic regression: the number of operators, variables and constants in the tree (cf. with Figure 3).
- Support vector machines: $C * p$ with C the number of support vectors and the p the number of inputs; this is because each support vector can be interpreted as one structural component (localizer) similarly to one rule, whereas each support vector has only one parameter, namely in form of one numeric coordinate value.
- Random forests: $\sum_{m=1}^{M} C_m * (2 * d_m)$ with M the number of trees in the forest, C_m the number of leaf nodes (=rules) and d_m the depth of the mth tree (i.e. the number of conditions from the root to the leaves); in this way, it is also in accordance to the TS fuzzy model and symbolic regression tree complexity representation.
- Gradient boosted trees (GBT): the calculation is done in the same way as for random forest.
- Gaussian process regression (GPR): same calculation as for support vector machines. The only difference is that each row is considered as one structural component (localizer).

In order to combine accuracy and complexity within one value to represent a compact index for indicating a feasible tradeoff between accuracy and complexity, we also investigated an accuracy/complexity (A/C) index which is defined by:

$$AC_{ind} = \frac{100 - 2 * MAE}{log(C)} \tag{13}$$

We applied this measure for the purpose of comparing the compactness (and thus transparency) of the fuzzy rule bases obtained by the fuzzy modeling approaches against each other. In this sense, C denotes the number of rules plus the number of inputs and MAE the mean absolute error achieved in the separate test data set. This measure punishes the accuracy of more complex models, thus the higher its value, the better the method becomes in terms of representing a good accuracy/complexity tradeoff.

Table 4 shows the internal structural complexity across all methods, and also the A/C index values for the fuzzy modeling and symbolic regression approaches after the slashes in each cell.

The interpretation of this table is obvious: the non-linear state-of-the-art regression modeling techniques produce models with extremely huge complexity (at least a few thousands of parameters in all cases), which makes them inapplicable for our purposes to represent transparent and readable models to the experts. Compared to non-linear SoA methods, the structural complexity of the symbolic regression models is much smaller and includes around 50 up to maximally 67 nodes for all target variables. The fuzzy models produced by our new robust method Rob-GenFIS (and also by FLEXFIS) are slightly larger

Table 4: Summary of results for all target variables and all non-linear methods that have been tested. The cell values are the values obtained from the internal structural complexity values as defined in the text, the models with lowest complexity for each target are highlighted in bold font; for the fuzzy modeling and the enhanced symbolic regression methods, after the slashes the accuracy/complexity index as defined by (13) is reported.

Algorithm	CF_1	CF_2	CF_3	CF_4	NVH	$Temp_1$	$Temp_2$	$Wear_1$	$Wear_2$
Non-Linear SoA Methods									
SVM-RBF	20265	120776	14473	10004	79827	13735	14350	33001	25600
RF	164010	506010	69860	78435	487475	53780	78055	153160	53210
GPR SEard	21385	129890	18040	17876	108035	17999	17876	39040	40896
GBT	267000	267000	267000	267000	267000	267000	267000	267000	267000
GBT standard	721814	745000	708128	703553	739721	709422	706279	744640	744748
Enhanced Symbolic Regr									
Symbolic Regr. OSGA	26 / 27.41	48 / 22.24	41 / 22.59	46 / 22.61	43 / 15.65	38 / 25.23	46 / 19.81	42 / 23.74	28 / 28.04
Symbolic Regr. NSGA-II	43 / 23.75	67 / 20.62	36 / 23.58	56 / 21.81	44 / 15.46	47 / 23.68	48 / 19.76	50 / 22.78	49 / 24.06
Fuzzy Modeling Variants									
FLEXFIS	96 / 17.40	128 / 18.07	104 / 17.90	144 / 11.95	208 / 10.87	182 / 17.49	130 / 15.65		50 / 23.78
Rob-GenFIS	104 / 19.37	**130 / 18.27**	**21 / 27.63**	144 / 17.27	**70 / 14.41**	182 / 17.70	130 / 15.65	**38 / 24.41**	**48 / 24.10**
Genfis2 loc	285 / 15.78	560 / 13.97	160 / 16.57	**64 / 20.82**	130 / 11.83	1444 / 12.61	**95 / 15.94**	185 / 17.13	95 / 20.40
LoLiMoT	**57 / 21.96**	780 / 13.25	301 / 14.83	125 / 17.84	364 / 10.31	**10 / 39.43**	152 / 14.89	150 / 17.80	171 / 18.12

and in the range of around 100 up to maximal 208 for all targets. The difference between fuzzy models and symbolic regression are small, such that, together with the error results in the table above, we can conclude that both perform almost equally.

In terms of the fuzzy modeling variants, when comparing their achieved A/C index values, our newly proposed method *Rob-GenFIS* is able to outperform the other methods for five targets, whereas for CF_1 and $Temp_1$ *LoLiMoT* turns now out to be the best option, although for these two targets *Rob-GenFIS* produces the lowest MAEs. On the other hand, *Rob-GenFIS* can significantly outperform its 'predecessor' *FLEXFIS* for 8 targets. This underlines that the extension to generalized rules and to an improved, more robust training of consequents (with the usage of elastic net regularization) pays-off

4.3 Model Interpretation on Structural Level — Gaining Insights

In case of the well-performing non-linear state-of-the-art methods, no such insight on structural model level is possible at all — all models generated by them are rather appearing as black boxes.

In case of fuzzy models, the interpretable insight can be achieved through the representation of fuzzy rules in IF-THEN form (compare with (1)). This yields some sort of linguistically readable (partial, local) dependencies between the input variables and the targets, which can then be associated with predominant system conditions in certain different operation ranges/modes. This is possible as a fuzzy model internally provides a granulated viewpoint on the whole system behavior by partitioning all the inputs into several parts with the usage of *fuzzy sets*, usually associated with linguistic terms such as LOW, MEDIUM and HIGH, which are then combined to form the antecedent parts of the rules. In case when the fuzzy models are learned from data, the fuzzy sets and rule antecedent parts can be obtained by the projection concept (Lughofer et al., 2015) and typically by also applying some post-operating procedure for removing redundant, significantly overlapping sets/rules (Lughofer, 2013).

We obtained the following five transparent fuzzy rules with TS-type consequent hyper-planes after removing three inputs whose partitioning resulted in one global fuzzy set (which can be associated with a don't care part in all rules) and after projecting the generalized rules to the axes (according to the projection concept developed in (Lughofer et al., 2015)) to form standard linguistic fuzzy partitions, Finally, it ends up with only three antecedent parts each and a maximum of two fuzzy sets per input (ON/OFF resp. LOW/HIGH):

Rule 1: IF $Source_1$ is ON and x_{22} is HIGH and Mat_{24} is OFF THEN $y = 6.169 * Source_1 + 0.00205 * x_{22} - 1.113 * Mat_{24} - 20.54$

Rule 2: IF $Source_1$ is OFF and x_{22} is LOW and Mat_{24} is OFF THEN $y = 3.778 * Source_1 + 0.00013 * x_{22} - 1.973 * Mat_{24} + 2.201$

Rule 3: IF $Source_1$ is OFF and x_{22} is HIGH and Mat_{24} is OFF THEN $y = 5.705 * Source_1 + 0.00013 * x_{22} - 2.335 * Mat_{24} + 1.811$

Rule 4: IF $Source_1$ is ON and x_{22} is LOW and Mat_{24} is OFF THEN $y = 4.176 * Source_1 + 0.00034 * x_{22} - 0.03488 * Mat_{24} + 1.784$

Rule 5: IF $Source_1$ is OFF and x_{22} is LOW and Mat_{24} is ON THEN $y = 6.136 * Source_1 + 0.0002029 * x_{22} - 2.399 * Mat_{24} + 2.069$

where the variable y stands for the target '$Wear_2$'. The consequents in this form may be interpretable by experts if she/he is able to associate a physical meaning of the weighted linear regression formula. Alternatively, they can be even transferred to linguistic terms as realized in the equivalent tree-structured representation (Figure 3 left).

In order to obtain a direct valid comparison with symbolic regression, we transformed this conventional fuzzy rule base representation into a tree structure in accordance with the conventional tree structure representation of symbolic regression by still keeping

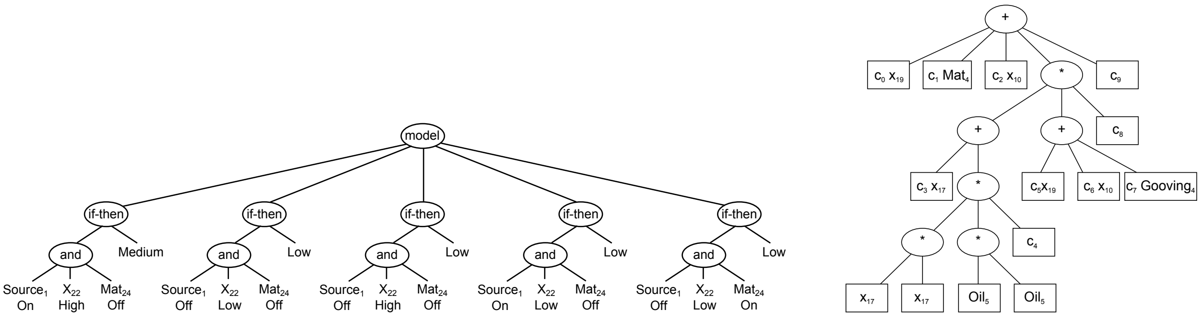

Figure 3: Tree-based model structures obtained through fuzzy modeling (left) and symbolic regression (right); all nodes and assignments of parameters therein are counted to the internal structural complexity.

the correct semantic meaning of the rule base. For the example above, the corresponding tree is shown in Figure 3 (left), whereas in (right) the model tree for the same target (*Wear*$_2$) spanned by symbolic regression is shown.

In conventional notation the symbolic regression model shown in Figure 3 (b) is:

$$\begin{aligned}\mathrm{Wear}_2 = {} & c_0 \cdot x_{19} + c_1 \cdot \mathrm{Mat}_4 + c_2 \cdot x_{10} \\ & + c_8 \cdot \left(c_3 \cdot x_{17} + x_{17}^2 \cdot \mathrm{Oil}_5^2 \cdot c_4\right) \\ & \cdot \left(c_5 \cdot x_{19} + c_6 \cdot x_{10} + c_7 \cdot \mathrm{Grooving}_4\right) + c_9\end{aligned}$$

Comparing the two different models shown in Figure 3 it can be observed that the fuzzy model has more nodes than the symbolic regression model. However, when determining structural complexity we have not counted the internal structural nodes 'IF-THEN' and 'AND' because they are fixed and can not be changed by the learning algorithm. In comparison, genetic programming is free to combine the allowed operators, functions and their operands in any semantically valid way. Therefore, we have also counted internal nodes for symbolic regression models. The symbolic regression model can be implemented rather easily in any programming language as it only relies on standard operators. Implementing the fuzzy model is a little bit more difficult as it is also necessary to implement the fuzzy inference algorithm which is necessary to interpret the rules correctly.

5 CONCLUSIONS

We have for the first time demonstrated that fuzzy modeling and symbolic regression can be applied for empirical modeling of wet tribological systems and have shown that they can be used successfully for modeling key metrics of tribological systems. The errors achieved for nine important targets are significantly lower than those achieved by linear methods and similar to those achieved by related non-linear modeling regression methods. However, 1.) the structural complexity is much lower (around 100 versus a few 1000s components) and 2.) the related methods do not offer any interpretable meaning as appearing as complete black boxes. Such a high complexity bears the risk of significant over-fitting. Both of our methods, fuzzy modeling and symbolic regression can be represented in tree-based structures, based on which interpretation is pretty easy (we provided a concrete example for target "wear"). Upon such higher level interpretation possibilities offer by our methods, the experts and operators working with the system reach a higher confidence in the models.

ACKNOWLEDGEMENTS

The work described in this paper was done within the COMET Project Heuristic Optimization in Production and Logistics (HOPL), #843532 funded by the Austrian Research Promotion Agency (FFG) and within the project “Smart Factory Lab” funded within the EU programme IWB 2020 by the provincial government of Upper Austria.

REFERENCES

Affenzeller, M., Winkler, S., Kronberger, G., Kommenda, M., Burlacu, B., and Wagner, S. (2014). Gaining deeper insights in symbolic regression. In Riolo, R., Moore, J. H., and Kotanchek, M., editors, *Genetic Programming Theory and Practice XI*, Genetic and Evolutionary Computation. Springer.

Affenzeller, M., Winkler, S., Wagner, S., and Beham, A. (2009). *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman & Hall, Boca Raton, Florida.

Aleksendri, D. (2010). Neural network prediction of brake friction materials wear. *Wear*, 268(12):117 – 125.

Aleksendric, D. and Carlone, P. (2015). *Soft Computing in the Design and Manufacturing of Composite Mate-*

rials: Applications to Brake Friction and Thermoset Matrix Composites. Woodhead Publishing.

Alonso, C. L., Montana, J. L., and Borges, C. E. (2009). Evolution strategies for constants optimization in genetic programming. In *21st International Conference on Tools with Artificial Intelligence, ICTAI '09*, pages 703–707.

Berger, E. (2002). Friction modeling for dynamic system simulation. *Applied Mechanics Reviews*, 55(6):535–577.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Castro, J. and Delgado, M. (1996). Fuzzy systems with defuzzification are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, 26(1):149–152.

Cernuda, C., Lughofer, E., Maerzinger, W., and Kasberger, J. (2011). NIR-based quantification of process parameters in polyetheracrylat (PEA) production using flexible non-linear fuzzy systems. *Chemometrics and Intelligent Laboratory Systems*, 109(1):22–33.

Chiu, S. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3):267–278.

De Wit, C. C., Olsson, H., Astrom, K. J., and Lischinsky, P. (1995). A new model for control of systems with friction. *Automatic Control, IEEE Transactions on*, 40(3):419–425.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Gray, R. (1984). Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29.

Haenlein, M. and Kaplan, A. (2004). A beginner's guide to partial least squares (PLS) analysis. *Understanding Statistics*, 3(4):283–297.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction - Second Edition*. Springer, New York Berlin Heidelberg.

Hastie, T., Tibshirani, R., and Friedman, J. (2010). Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1).

Hosenfeldt, T., Schulz, E., Glerl, J., and Steinmetz, S. (2014). Friction tailored to your requirements. In KG, S. T. G. . C., editor, *Solving the Powertrain Puzzle - 10th Schaeffler Symposium*, chapter 23, pages 331 – 344. Springer Vieweg.

Keijzer, M. (2003). Improving symbolic regression with interval arithmetic and linear scaling. In Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R., and Costa, E., editors, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 70–82, Essex. Springer-Verlag.

Kommenda, M., Kronberger, G., Affenzeller, M., Winkler, S. M., and Burlacu, B. (2016). Evolving simple symbolic regression models by multi-objective genetic programming. In *Genetic Programming Theory and Practice XIII (not yet published)*. Springer.

Kommenda, M., Kronberger, G., Winkler, S., Affenzeller, M., and Wagner, S. (2013). Effects of constant optimization by nonlinear least squares minimization in symbolic regression. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pages 1121–1128. ACM.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Kronberger, G. (2011). *Symbolic Regression for Knowledge Discovery – Bloat, Overfitting, and Variable Interaction Networks*. Number 64 in Johannes Kepler University, Linz, Reihe C. Trauner Verlag+Buchservice GmbH.

Langdon, W., Soule, T., Poli, R., and Foster, J. (1999). The evolution of size and shape. In *Advances in Genetic Programming*, volume 3, chapter 8, pages 163–190. MIT Press.

Lemos, A., Caminhas, W., and Gomide, F. (2011). Multivariable gaussian evolving fuzzy modeling system. *IEEE Transactions on Fuzzy Systems*, 19(1):91–104.

Loh, W.-Y., Basch, R. H., Li, D., and Sanders, P. (2000). Dynamic modeling of brake friction coefficients. Technical report, SAE Technical Paper.

Lughofer, E. (2008). FLEXFIS: A robust incremental learning approach for evolving TS fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410.

Lughofer, E. (2011). *Evolving Fuzzy Systems — Methodologies, Advanced Concepts and Applications*. Springer, Berlin Heidelberg.

Lughofer, E. (2013). On-line assurance of interpretability criteria in evolving fuzzy systems — achievements, new concepts and open issues. *Information Sciences*, 251:22–46.

Lughofer, E., Cernuda, C., Kindermann, S., and Pratama, M. (2015). Generalized smart evolving fuzzy systems. *Evolving Systems*, 6(4):269–292.

Lughofer, E. and Sayed-Mouchaweh, M. (2015). Autonomous data stream clustering implementing incremental split-and-merge techniques — towards a plug-and-play approach. *Information Sciences*, 204:54–79.

Mann, H. and Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of mathematical Statistics*, 18:50–60.

Mukherjee, S. and Eppstein, M. J. (2012). Differential evolution of constants in genetic programming improves efficacy and bloat. In Rodriguez, K. and Blum, C., editors, *GECCO 2012 Late breaking abstracts workshop*, pages 625–626, Philadelphia, Pennsylvania, USA. ACM.

Nelles, O. (2001). *Nonlinear System Identification*. Springer, Berlin.

Poli, R. (2003). A simple but theoretically-motivated method to control bloat in genetic programming. In *Genetic programming*, pages 204–217. Springer.

Poli, R., Langdon, W. B., and McPhee, N. F. (2008). *A field guide to genetic programming*. Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk.

Rall, L. B. (1981). *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Springer, Berlin.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*. The MIT Press.

Sellgren, U., Bjrklund, S., and Andersson, S. (2003). A finite element-based model of normal contact between rough surfaces. *Wear*, 254(11):1180 – 1188. Papers presented at the 10th Nordic Conference on Tribology, {NORDTRIB} 2002.

Senatore, A., D'Agostino, V., Giuda, R. D., and Petrone, V. (2011). Experimental investigation and neural network prediction of brakes and clutch material frictional behaviour considering the sliding acceleration influence. *Tribology International*, 44(10):1199 – 1207.

Smits, G. F. and Kotanchek, M. (2005). Pareto-front exploitation in symbolic regression. In O'Reilly, U. M., Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and Practice II*, volume 8 of *Genetic Programming*, pages 283–299. Springer Verlag, New York.

Smola, A. and Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1):116–132.

Topchy, A. and Punch, W. F. (2001). Faster genetic programming based on local gradient search of numeric leaf values. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 155–162, San Francisco, California, USA. Morgan Kaufmann.

Xiao, L., Bjrklund, S., and Rosn, B. (2007). The influence of surface roughness and the contact pressure distribution on friction in rolling/sliding contacts. *Tribology International*, 40(4):694 – 698. {NORDTRIB} 2004.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, pages 301–320.