Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 3, 445–459
DOI: 10.34768/amcs-2021-0030

GrNFS: A GRANULAR NEURO–FUZZY SYSTEM FOR REGRESSION IN LARGE VOLUME DATA

KRZYSZTOF SIMINSKI [a]

[a] Department of Algorithmics and Software
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland
email: krzysztof.siminski@polsl.pl

Neuro-fuzzy systems have proved their ability to elaborate intelligible nonlinear models for presented data. However, their bottleneck is the volume of data. They have to read all data in order to produce a model. We apply the granular approach and propose a granular neuro-fuzzy system for large volume data. In our method the data are read by parts and granulated. In the next stage the fuzzy model is produced not on data but on granules. In the paper we introduce a novel type of granules: a fuzzy rule. In our system granules are represented by both regular data items and fuzzy rules. Fuzzy rules are a kind of data summaries. The experiments show that the proposed granular neuro-fuzzy system can produce intelligible models even for large volume datasets. The system outperforms the sampling techniques for large volume datasets.

Keywords: granular computing, neuro-fuzzy system, large volume data, machine learning.

1. Introduction

1.1. Granular computing.

Granular computing is an emerging field of research in data mining. This new paradigm is a shift from computer-centred to human-centred analysis. The phrase "information granulation" was coined by Lotfi Zadeh in 1979 (Zadeh, 1979). It was a very innovative idea (perhaps too innovative) and was not exploited mainly because there were not enough techniques and methods. Later on techniques, algorithms, ideas, concepts were further developed and in 1997 Zadeh published a paper in which he described the idea of "granular computing" (Zadeh, 1997), where he stated three concepts of human cognition: granulation (decomposition of a whole into parts), organization (integration of parts into a whole), and causation (relations of causes and effects). Granular computing (GrC) is an umbrella term for concepts, algorithms, techniques, and methods that mimic the human cognition model (Pedrycz *et al.*, 2015b). Granular computing aims at providing a generic abstraction for methods already known and applied (Salehi *et al.*, 2015). It is a new view on existing methods and simultaneously a starting point for new techniques, algorithms, and concepts. For Zadeh granular computing is a starting point for computing with words (Zadeh, 2002). In recent years granular computing has woken up from long hibernation and has made a huge progress (Yao *et al.*, 2013). Yao (2007) claims granular computing a new field of study.

Yao (2008) also describes a triarchic theory of granular computing with three perspectives, each supporting the other two (Fig. 1):

Philosophical perspective: focused on structured thinking. A complex system can be decomposed into smaller simpler and more fundamental parts. A complex system is viewed as a cooperation of simpler components. The properties of the system are properties of its parts and relations between components (in the case of emergent wholes). This perspective holds both analysis of a whole into parts and synthesis of a whole from parts.

Methodological perspective: focused on structured problem solving. It analyses methods, techniques, and tools aimed at systematic problem solving at multiple levels and with multiple views.

Computational perspective: focused on structured information processing. It tries to address information processing on multiple levels of data

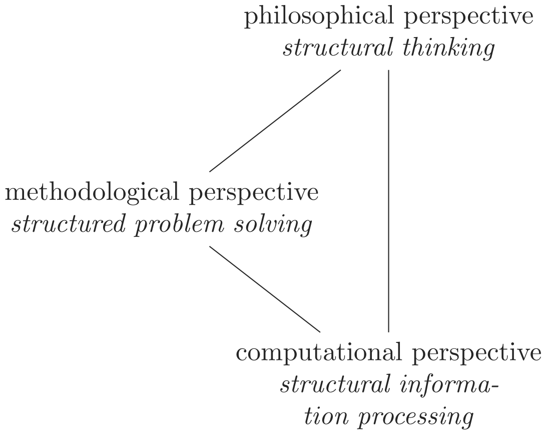
philosophical perspective
structural thinking
methodological perspective
structured problem solving
computational perspective
structural information processing

Fig. 1. Granular computing triangle.

granularity by mimicking information processing in a brain. This perspective enumerates two processes (Fig. 2): granulation (creation of granules) and computing with granules (this is a big challenge for granular computing research).

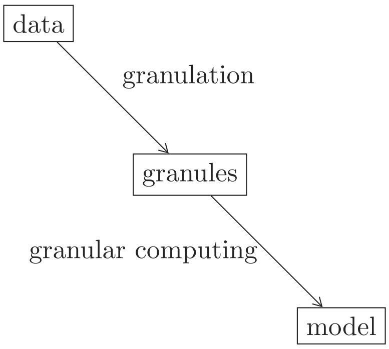

Fig. 2. Computational perspective of the granular approach.

1.2. Granules. A data granule may be defined as a collection of related entities in the sense of similarity, proximity, indiscernibility (Pedrycz, 2013; Yao and Zhong, 2007; Yao, 2008). Data granules originate at the numeric level and are arranged together due to their similarity, functional adjacency, and distinguishabiblity (Shifei *et al.*, 2010). An important difference between granulation and clustering is the semantics of data granules (Bargiela and Pedrycz, 2006). A cluster gathers similar, close objects, whereas granulation requires granules to be tagged with a semantically rich labels. A granule represents a semantic whole and simultaneously, it is in a hierarchical relation with other granules. A granule is a component of a more general granule and simultaneously is a composition of subcomponents—subgranules. Granularity is the ability to represent and operate on different levels of detail in data, information, and knowledge (Keet, 2008). When some granule requires more detailed analysis, it is further decomposed into finer (sub)granules (Yao, 2018). We trace granular entities in the environment and then compose hierarchical structures of granules (Ciucci, 2016).

Granules are commonly represented with intervals, fuzzy sets (Zadeh, 1965), rough sets (Skowron *et al.*, 2016), shadowed sets (Pedrycz, 1998), fuzzy rough sets (Bisi *et al.*, 2017), intuitionistic fuzzy sets (Atanassov, 1986), clusters (Siminski, 2020), etc. The fuzzy set approach enables composition of granules with slightly unequal objects due to the handling of imprecision with fuzzy sets. Recent studies (Pedrycz *et al.*, 2012) address a non-numeric view at membership functions in fuzzy sets—application of interval type-2 fuzzy or general type-2 fuzzy sets. The rough set approach makes use of indiscernibility and knowledge granularity. Indiscernible object builds notions in a very natural way. Shadow sets (Pedrycz, 1998) enable representation of an element whose membership to a set is unknown. Similarly to (interval) type-2 fuzzy sets they also support non-numeric processing. Fuzzy sets with granular membership function are proposed by Pedrycz *et al.* (2012).

Typically granular analysis of data starts with granulation of data, i.e., extraction of granules from data (Fig. 2). The spectrum of granulation techniques is very wide. Commonly used techniques are discretization, quantization, clustering, aggregation, and transformation (Yao, 2020). If the granulation of objects is based on an equivalent class, the elaborated granules are treated as information granules (Pawlak, 1996). If a binary relation is used, a binary granular structure is achieved (Qian *et al.*, 2011; 2010).

Pedrycz *et al.* (2015a) underline the importance of data granulation in data mining. Some data may be available only locally and cannot be shared because of technical or legal constraints, and only some summaries of them may be available. This problem may be solved with the granulation and publishing of granules instead of vulnerable data. Two interesting remarks on data distribution are important (Pedrycz *et al.*, 2015a). Data may have spatial and/or temporal distribution (Yang *et al.*, 2019). The former occurs when data are collected at different locations in various schemes, but granulation may reveal some global abstract structure of data. In the case of insufficient data transfer, granulation may be one of solutions to share data between remote clients. Similarly, data may be distributed over some time horizon and it is impossible to gather all data at a time. Storage and transfer of raw data may be impossible due to their size. This problem may be solved with data granulation.

Table 1. Symbols used in the paper.

Symbol	Meaning
$\mathbb{L}$	set of rules, rule base
l	rule, $l \in \mathbb{L}$
L	number of rules, $L = \|\|\mathbb{L}\|\|$
β	number of data items in a part
γ	number of data generated from a set of granules
$\mathbb{G}$	set of granules
G	number of granules in $\mathbb{G}$
$\mathbb{D}$	data set
c_i	cardinality of the i-th granule
e_i	error of the i-th granule
q_i	quality of the i-th granule
n_i	number of items generate from the i-th granule
y_i	expected value for the i-th data item
$\hat{y}_i$	elaborated value for the i-th data item

1.3. Neuro-fuzzy systems.

Neuro-fuzzy systems have proved their effectiveness in both regression and classification tasks. They can generalise presented data and elaborate an intelligible model. Their great advantage is the form of the elaborated model: it is a set of fuzzy rules that can be easily understood by humans. Neuro-fuzzy systems have found wide applications in time series modelling, regression and classification tasks (Siminski, 2021), automatic control, etc. In our system any neuro-fuzzy system for regression can be used. We use the Takago–Sugeno–Kang neuro-fuzzy system described in Section 2.3.

1.4. Objective of the paper.

Nowadays the volume of data is still growing and they have to be handled. Application of neuro-fuzzy systems may be difficult for large data, because neuro-fuzzy systems require access to all the data (in the paper we avoid the "big data" term, because we focus only on the size of data and "big data" is a semantically richer term). A simple approach suggests the sampling of data and thus reducing the volume of data. However, this approach loses data and uses only a small part of data to build a model. This is why we would like to apply the granular approach: instead of sampling data we granularise them. Granules may be treated as specific summaries of data with clear semantics, whereas sampled data are just subsets of original full data.

Our objective is a reliable neuro-fuzzy system that is able to handle data that are not possible to fit into machine memory. The proposed system can both create a fuzzy model and granulate data. It performs the latter, thus reducing its volume until the final model is elaborated. In our approach we use a novel representation of them: we represent granules with fuzzy rules (Fig. 3).

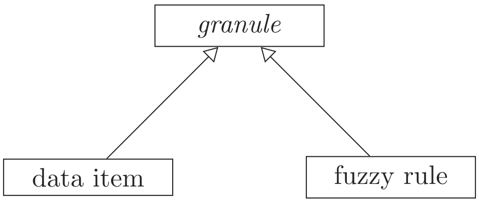

Fig. 3. Hierarchy of granules.

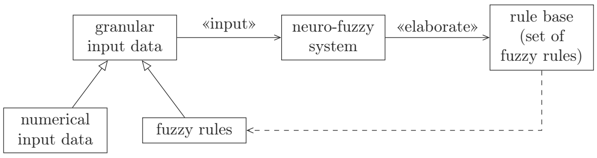

Fig. 4. Scheme of the train procedure for a granular neuro-fuzzy system.

In the paper we use the following convention for symbols: the blackboard bold characters ($\mathbb{A}$) are used to denote sets, bolds ($\mathbf{a}$) represent matrices and vectors, upper case italics (A) stand for the cardinality of sets, lower case italics (a) for scalars and set elements. A detailed list of symbols is gathered in Table 1.

2. Granular neuro-fuzzy system for large volume data

The objective of our research is a neuro-fuzzy system for large volume data. We use granules to represent summaries of data in elaboration of a fuzzy rule base for a whole large dataset. The main idea of our approach is to process data by parts. The system elaborates a set of fuzzy rules as a model of processed data. In our approach fuzzy rules represent granules and a set of granules is an input data set for the system. We describe our system in this section in detail.

2.1. Representation of granules.

Often in papers the term "granular neuro-fuzzy system" is used for neuro-fuzzy systems with identification of the fuzzy rule base with the clustering approach. This step (clustering) is often labelled "granulation." In our approach we would like to go deeper. In Introduction we list several techniques used to represent granules. Now we add one more: fuzzy rules. These are composed of premises and consequences. Premises of rules split the input domain into regions.

A more detailed description of fuzzy rules used in experiments is presented in Sections 2.3.1 and 2.3.2.

Each region represents a part of data due to the intelligibility of the fuzzy rule base. This is why we can use fuzzy rules as granules—they have clear semantics.

The structure of rules is described in Section 2.3. Premises of rules are composed of fuzzy sets. We follow the definition of fuzzy rules by Ahmed and Isa (2017), who define fuzzy granules as "sufficiently interpretable fuzzy sets."

In our approach granules are represented both by data items (by a data item we understand a vector of numbers, a "classical" numeric data representation) and fuzzy rules (Fig. 3). In the training mode our neuro-fuzzy system takes numerical data items (vectors of numbers) or fuzzy rules as input and elaborates a set of fuzzy rules—a fuzzy rule base (Fig. 4). The produced fuzzy rules are granules, thus may be an input for our neuro-fuzzy system (dashed arrow in Fig. 4). Training a neuro-fuzzy system with data granules is described in detail in Section 2.2.

The principle of justifiable granularity (Pedrycz and Homenda, 2013) is an important issue in granulation. It focuses on the semantics of granules. If a granule covers a lot of data items, it loses its meaning. On the other hand, if a granule represents data very precisely, it may cover only a few items. A granule has to fulfil two contradictory criteria: (i) data coverage and (ii) specificity. The former requires a sufficient experimental evidence behind a granule, the latter—clear semantics of a granule. These two criteria are contradictory, because highly specific granules have usually poor data coverage, and granules with wide data coverage usually lack specificity and semantics. A balance between these two criteria is the principle of justifiable granularity (Wang *et al.*, 2019). In neuro-fuzzy systems, the intelligibility of rules is their crucial property (Cpalka *et al.*, 2014; Alcalá *et al.*, 2006; Alonso and Magdalena, 2011; Botta *et al.*, 2009; Evsukoff *et al.*, 2009; Gacto *et al.*, 2011; Herrera *et al.*, 2005; Yen *et al.*, 1998; Siminski, 2015; 2014; 2017). Rules that have poor semantics (either too specialised or too general) break the principle of justifiable granularity. Application of neuro-fuzzy systems makes the risk of unjustifiable granularity lower, because in neuro-fuzzy systems the fuzzy rule base has to be interpretable and have clear semantics of rules, which is in full concordance with the principle of justifiable granularity.

2.2. Creating a fuzzy rule base.

To handle data that are too large to fit a machine memory, we read data by parts (blocks) and build a fuzzy rule base part by part (Alg. 1). Each block of data holds β data items. Because numerical data items are granules (Fig. 3), they are treated as such—they build a zeroth grade set $\mathbb{G}_0$ of granules (line 9). When a block of data is read, fuzzy rules are elaborated (line 12). The resulting rules are treated as granules and added to a first grade set $\mathbb{G}_1$ of granules (line 14). If the set $\mathbb{G}_1$ is too large (line 16), the granules are used as input to produce a new set of rules (line 17). The procedure iterates until all blocks of data are read and transformed into granules. If a number of elaborated rules is larger than required, one more granulation is run (line 21). Finally, the set of rules (granules) is returned. The number L of rules (granules) in a fuzzy rule base is a parameter of the procedure.

2.2.1. Forming of fuzzy rules from data granules.

Lines 12 and 20 in Algorithm 1 seem very similar (in both lines the procedure create_rules_from_granules is called), but actually these two calls are completely different. The pseudocode for this procedure is presented in Algorithm 2. It is trivial, but its behaviour depends on how granules are represented. Implementation of degranulation depends on the data that constitute a granule.

If a granule is represented by a single data item, the procedure simply returns the data item (Algorithm 3). If

Algorithm 1. Procedure create_rulebase_by_parts.

```
Require: $L$ {number of rules to elaborate}, $\beta$ {number of data items to read in one block}
 1: $\mathbb{G}_1 := \emptyset$
 2: while more data exist do
 3:     data := read a block of $\beta$ data
 4:     $\mathbb{G}_0 := \emptyset$
 5:     for all datum $d$ in data do
 6:         {"classical" numeric data items:}
 7:         $g$ := create a granule from each datum $d$;
 8:         {add granule to the set of granules:}
 9:         $\mathbb{G}_0 := \mathbb{G}_0 \cup g$
10:     end for
11:     {a block of data read and transformed into unit granules}
12:     $\mathbb{L} = \{l_1, \ldots, l_L\}$ := create_rules_from_granules ($\mathbb{G}_0$){Fig. 2}
13:     for all rule $l$ in $\mathbb{L}$ do
14:         $\mathbb{G}_1 := \mathbb{G}_1 \cup l$ {add each rule of the set of granules}
15:     end for
16:     if $\|\mathbb{G}_1\| > \beta$ then
17:         $\mathbb{G}_1$ := create_rules_from_granules ($\mathbb{G}_1$)
18:     end if
19: end while
20: if $\|\mathbb{G}_1\| > L$ then
21:     $\mathbb{G}_1$ := create_rules_from_granules ($\mathbb{G}_1$){Fig. 2}
22: end if
23: return $\mathbb{G}_1$
```

Algorithm 2. Procedure create_rules_from_granules.

```
Require: $\mathbb{G}$ {set of granules}
if granules represented by data items then
    $\mathbb{D}$ := degranulate_data_items ($\mathbb{G}$){Alg. 3}
end if
if granules represented by data items then
    $\mathbb{D}$ := degranulate_fuzzy_rules ($\mathbb{G}$){Alg. 4}
end if
$\mathbb{L}$ := train_neuro_fuzzy_system ($\mathbb{D}$)
```

a granule is represented by a fuzzy rule, the procedure returns a set of data items generated by the fuzzy rule (Algorithm 4).

The granule is a kind of data summary. Now we "open" a granule to get the data. It is not a container for data items. A granule holds information extracted from data in the form of premises and consequences. A granule is able to reconstruct data in the degranulation process (Reyes-Galaviz and Pedrycz, 2015; Hu *et al.*, 2017). The reconstructed data are not exactly the same data the granule was created with.

First the rule cardinalities are calculated (line 1 in Algorithm 4). By the cardinality c of a rule we mean the number of data items that are matched (recognized, covered) by this rule. Because rules are fuzzy, their cardinalities are not integers. Then we evaluate the root mean square errors RMSE e (Eqn. (23)) for each rule (line 2 in Algorithm 4).

Finally, in line 5 we assess the quality of rules with the formula

$$q_i = \frac{c_i}{\sum_{k=1}^{G} c_k} \left(1 - \frac{e_i}{\sum_{k=1}^{G} e_k} \right), \tag{1}$$

where c_i stands for the cardinality of the i-th rule, and e_i for its error and G for the number of granules.

The principle of justified granularity (Pedrycz and Gomide, 2007) highlights two features of granules: coverage and precision. The first factor in (1) is a normalised cardinality that represents the coverage of a granule. The second factor stands for the precision of a granule (the complement of a normalised error of a granule).

Algorithm 3. Procedure degranulate_data_items.

```
Require: $\mathbb{G} = \{g_1, \dots, g_G\}$ {set of granules represented by data items (vectors of numbers)}
$\mathbb{D} := \emptyset$
for all granule $g$ in $\mathbb{G}$ do
    $\mathbb{D} := \mathbb{D} \cup g$ {granule is a data item}
end for
return $\mathbb{D}$
```

Algorithm 4. Procedure degranulate_fuzzy_rules.

```
Require: $\mathbb{G} = \{g_1, \dots, g_G\}$ {set of granules represented by fuzzy rules}
 1: $\{c_1, \dots, c_G\}$ := elaborate_cardinalities_of_rules($\mathbb{G}$)
 2: $\{e_1, \dots, e_G\}$ := elaborate_errors_of_rules($\mathbb{G}$);
 3: $\mathbb{D} := \emptyset$ {set of elaborated data items}
 4: for all granule $g_i$ in $\mathbb{G}$ do
 5:     $q_i$ := evaluate quality of granule $g_i$ with (1);
 6:     $n_i$ := evaluate number of data items to generate from granule $g_i$ whose quality is $q_i$ with (2)
 7:     $\mathbb{d}$ := generate $n_i$ data items from granule $g_i$
 8:     $\mathbb{D} := \mathbb{D} \cup \mathbb{d}$
 9: end for
10: return $\mathbb{D}$
```

We calculate the numbers of data items to generate from each rule:

$$n_i = \left[\gamma \frac{q_j}{\sum_{j=1}^{G} q_j} \right], \tag{2}$$

where γ is the number of generated data items from the set of fuzzy rules (it a parameter of the system) and $[\cdot]$ is the rounding operator defined as

$$[x] = \begin{cases} \lfloor x \rfloor, & x - \lfloor x \rfloor < 0.5, \\ \lfloor x \rfloor + 1, & x - \lfloor x \rfloor \geq 0.5. \end{cases} \tag{3}$$

Data items are generated with normal distribution, because premises of rules are represented with Gaussian fuzzy sets (cf. Section 2.3.1). Rules with higher qualities have higher probability of data item generation. Eventually the data set with generated data items is returned from the procedure.

In our approach granules are specific summaries of data. They do not hold data literally, data are not zipped or packed, but they are represented by granules and representatives of data can be extracted (produced) from granules.

2.3. Takagi–Sugeno–Kang neuro-fuzzy system.

In our approach any neuro-fuzzy system may be used. We use the Takagi–Sugeno–Kang (TSK) neuro-fuzzy system (Takagi and Sugeno, 1985; Sugeno and Kang, 1988).

The structure of the TSK system is presented in Fig. 5. TSK is a multiple-input single-output (MISO) system. Its rule base $\mathbb{L}$ contains fuzzy rules l in the IF-THEN form:

$$l : \text{IF } \mathbf{x} \text{ is } \mathfrak{a} \text{ THEN } y \text{ is } \mathfrak{b}, \tag{4}$$

where $\mathbf{x} = [x_1, x_2, \dots, x_D]^{\mathrm{T}}$ is a vector of attribute values (descriptors) of a data item and y is a decision attribute for this data item. All attributes are real numbers.

2.3.1. Premises.

Below we describe the architecture of the system only for one rule in order to keep the notation simpler.

The linguistic variable $\mathfrak{a}$ in the rule premise is represented as a fuzzy set $\mathbb{A}$ in a D-dimensional space. In each dimension d, the set $\mathbb{A}_d$ is described with a Gaussian fuzzy set:

$$u_{\mathbb{A}_d}(x_d) = \exp\left(- \frac{(x_d - v_d)^2}{2 s_d^2} \right), \tag{5}$$

where v_d is the core location for the d-th attribute and s_d is the fuzziness of this attribute. The Gaussian membership function is differentiable in its whole domain, which enables application of the gradient descent optimisation procedure. The memberships of all attributes (descriptors) are aggregated in order to elaborate the membership $u_{\mathbb{A}}$ of a data item to the premise of the rule. A T-norm $\star$ is used as the aggregation operator:

$$u_{\mathbb{A}} = u_{\mathbb{A}_1} \star u_{\mathbb{A}_2} \star \ldots \star u_{\mathbb{A}_D}. \qquad (6)$$

The T-norm is implemented as a product (thus Eqn. (6) becomes

$$u_{\mathbb{A}} = \prod_{d=1}^{D} u_{\mathbb{A}_d}. \qquad (7)$$

From now on the rule index l will be used again. Thus $u_{\mathbb{A}}$ becomes $u_{l\mathbb{A}}$.

To avoid misunderstandings, please keep in mind the meanings of the symbols:

$u_{\mathbb{A}_d}$ stands for the membership of the d-th attribute to the fuzzy set $\mathbb{A}_d$ in the premise of a certain rule (the index of which we omit here) as in the formulae (5)–(7),

$u_{l\mathbb{A}}$ stands for the membership of the whole data item to the premise of the l-th rule.

Combining (5) and (7), we get the activation (firing strength) F of the premise of the l-th rule for data item $\mathbf{x}$:

$$F_l(\mathbf{x}) = u_{l\mathbb{A}}(\mathbf{x}) = \prod_{d=1}^{D} \exp\left[-\frac{(x_d - v_{ld})^2}{2s_{ld}^2}\right], \qquad (8)$$

which is a real number for any $\mathbf{x}$: $F(\mathbf{x}) \in (0,1]$.

2.3.2. Consequences. The term $\mathfrak{b}$ in (4) describing the l-th rule's consequence is represented by a singleton fuzzy set. The localisation y_l of the singleton is determined by a linear combination of input attribute values:

$$y_l = \mathbf{p}_l^{\mathrm{T}} \cdot \left[1, \mathbf{x}^{\mathrm{T}}\right]^{\mathrm{T}} = [p_{l0}, p_{l1}, \ldots, p_{lD}] \cdot \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{bmatrix}. \qquad (9)$$

The above formula can also be written as

$$y_l = \sum_{d=1}^{D} p_{ld} x_d + p_{l0} = \sum_{d=0}^{D} p_{ld} x_d, \qquad (10)$$

where $x_0 = 1$.

The height of the singleton is the firing strength $F_l(\mathbf{x})$ (Eqn. (8)). Singletons of all rules are aggregated a into final answer of the TSK system with the formula

$$y_0 = \frac{\sum_{l=1}^{L} F_l(\mathbf{x})\, y_l(\mathbf{x})}{\sum_{l=1}^{L} F_l(\mathbf{x})}. \qquad (11)$$

2.3.3. Identification of system parameters. The initial system parameters are identified with two different methods. The premises of rules are produced with the fuzzy C-means (FCM) clustering algorithm (Dunn, 1973). The consequences are calculated precisely with the least squares method for regression.

2.3.4. Tuning system parameters. The identified parameters of the TSK system are tuned with a gradient optimisation technique (for premises) and the least squares method (for consequences).

3. Experiments

In experiments we used a TSK neuro-fuzzy system as a part of granular neuro-fuzzy system and as a standalone reference system. For the TSK neuro-fuzzy system we use the implementation available from the public library—NFL (Siminski, 2019).

3.1. Datasets. In the experiments we use large volume data. This is why we use artificial data sets that are defined with mathematical formulae. This enables preparation of datasets of large volume we need in our experiments.

3.1.1. Two dimensional surface. The input domain of the system has two attributes, x_1 and x_2. The former is partitioned with two triangular fuzzy sets, the latter—with two semitriangular fuzzy sets. The descriptors in premises of rules are joined with a product T-norm. The consequences follow the Mamdami–Assilan paradigm and are composed of four triangular fuzzy sets, and produce output value y for a pair $\langle x_1, x_2 \rangle$. A data item in this data set follows the scheme $\langle x_1, x_2, y + N(-0.5, 0.5) \rangle$, where $N(-0.5, 0.5)$ is a random value with normal distribution $N(m = -0.5, \sigma = 0.5)$.

3.1.2. Mackey–Glass series. This data series represents concentration of leukocytes in blood modelled with the Mackey–Glass equation (Mackey and Glass, 1977):

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{ax(t-\tau)}{1 + (x(t-\tau))^{10}} - bx(t), \qquad (12)$$

where x is the concentration of leukocytes, $a = 0.2$, $b = 0.1$ and $\tau = 17$ are constants. The equation was solved for the initial condition $x(0) = 0.1$ with the Runge–Kutta method with step $k = 0.1$ (Leski, 2008). First, 500 items were removed, and from the other each tenth item is taken. The data series was the base for creation of tuples with the template

$$[x(t), x(y-6), x(t-12), x(t-18), x(k+6)]. \qquad (13)$$

To each value in a tuple, a random noise value $N(0, 0.05)$ is added.

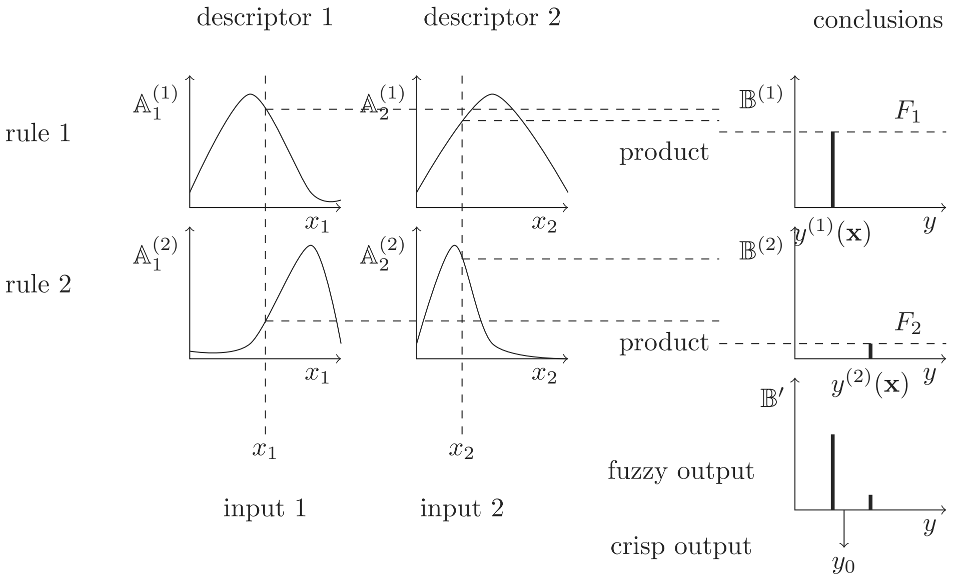

Fig. 5. Structure of the TSK fuzzy system with two rules and two input descriptors. The membership values of the 1st and the 2nd input are aggregated with the product T-norm. The final crisp output is calculated with Eqn. (11).

3.1.3. Lorenz system.

The Lorenz system describes a model of atmospheric convection (Lorenz, 1963):

$$x' = a(y - x), \tag{14}$$

$$y' = x(c - z) - y, \tag{15}$$

$$z' = xy - bz, \tag{16}$$

where parameters a, b, and c represent physical values in the model. For values $a = 10$, $b = 8/3$, and $c = 28$ the Lorenz system has chaotic behaviour. The first variable x is used to produce data tuples with the pattern

$$[x(t), x(t + 1), x(t + 5), x(t + 7), x(t + 13)]\,. \tag{17}$$

To each value in a tuple, a random noise value $N(0, 0.05)$ is added.

3.1.4. Rössler attractor.

This chaotic attractor was first proposed theoretically (Rössler, 1976), but later found applications in analysis of chemical reactions. The Rössler system is defined with three equations:

$$x' = -y - z, \tag{18}$$

$$y' = x + ay, \tag{19}$$

$$z' = b + z(x - c), \tag{20}$$

with the original parameters $a = 0.2$, $b = 0.2$, and $c = 5.7$. The first variable x is used to produce data tuples with the pattern

$$[x(t), x(t + 1), x(t + 5), x(t + 7), x(t + 13)]\,. \tag{21}$$

To each value in a tuple, a random noise value $N(0, 0.05)$ is added.

3.1.5. Noisy plant identification dataset.

This data set is used for comparison with other researchers. This is why the data set is not a huge one, because the goal of other researchers was different from ours. The data are created with the equation

$$y(t + 1) = \frac{y(t)}{1 + y^2(t)} + u^3(t), \tag{22}$$

where $u(t) = \sin(2\pi t/100)$, $t = 1, 2, \ldots, 200$, and $y(1) = 0$. Data tuples are inputs $y(t)$ and $u(t)$, output $y(t + 1)$. The noisy data tuples are used as the training data set. The original, clean data tuples are used as the test set. The number of iterations for this data is 500. The experiments were repeated 20 times. The scheme of the experiment for this data set differs, because we mimic the experiments by Juang and Chen (2013).

3.2. Error measures.

We use two error measures commonly applied for the regression task. The root mean square error (RMSE) and the mean absolute error (MAE) are defined respectively as

$$E_{\mathrm{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}, \tag{23}$$

$$E_{\mathrm{MAE}} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|\,, \tag{24}$$

where y_i denotes the i-th expected value and $\hat{y}_i$ the predicted value, N stands for the number of data items.

3.3. Experiments. Experiment 1 aims at computation of optimal parameters (β and γ) of the algorithm (Section 3.3.1). This experiment has two parts: 1a and 1b. The objective of Experiment 2 is to verify whether our system can handle large volume datasets. Experiment 3 compares the proposed granular approach and the sampling of large volume datasets.

3.3.1. Experiment 1. This experiment analyses the influence of parameters β and γ on the quality of set of granules. Parameter β denotes the size of data read in one block and γ the number of data items generated from the set of granules (fuzzy rules). We would like to find some recommendation for values of these two parameters.

The experiment is divided in two parts. Experiment 1a is run for smaller datasets and a grid of parameters β and γ. This may be treated as a preliminary experiment to tune parameter values for the other experiments. In Experiment 1b we use a results of Experiment 1a and run it for larger datasets.

Experiment 1a. The first experiment was run for the number of items in one block $\beta \in \{100, 200, 500, 1000, 2000, 5000, 10000\}$ and the number of data items generated from the set of granules $\gamma \in \{100, 200, 500, 1000, 2000, 5000, 10000\}$. We present here the results for the 'Rössler' and 'Lorenz' data sets. For each pair of values (β, γ) the experiment was repeated 10 times. In Tables 2 and 3 we present averages of the RMSE, MAE, and execution time for the 'Rössler' and 'Lorenz' data sets.

The parameters γ and β influence both execution time and the attendant errors. With an increase in the number β of data items read in one block, both errors and time decrease. The number β of data items read in one block should be as large as possible to keep errors and execution time similar to the values of the reference system.

For the number γ of items generated from a set of rules the behaviour of the system differs. With an increase in γ the errors decrease (cf. Fig. 6), but execution time increases (cf Fig. 7). The execution time is constant for $\gamma \leq \beta$. If $\gamma > \beta$ the execution time increases linearly with γ (cf. Tables 2 and 3). To keep the errors low, we should use higher values of γ. To keep the execution time short, we should use lower γ values. Some reasonable trade-off should be found for these two contradictory conclusions. We decided to fix $\gamma = \beta$. This is the largest value of γ that keeps the execution time short.

Experiment 1b. This experiment is designed to analyse the effectiveness of the system for larger data sets. In the experiment we use the results from the first experiment and set $\beta = \gamma$. The results presented in Table 4 reveal that β should be as large as possible. The last row in Table 4 presents results produced by the reference Takagi–Sugeno–Kang neuro-fuzzy system. The penultimate row shows the results produced by our granular neuro-fuzzy system when all input data are read in one block.

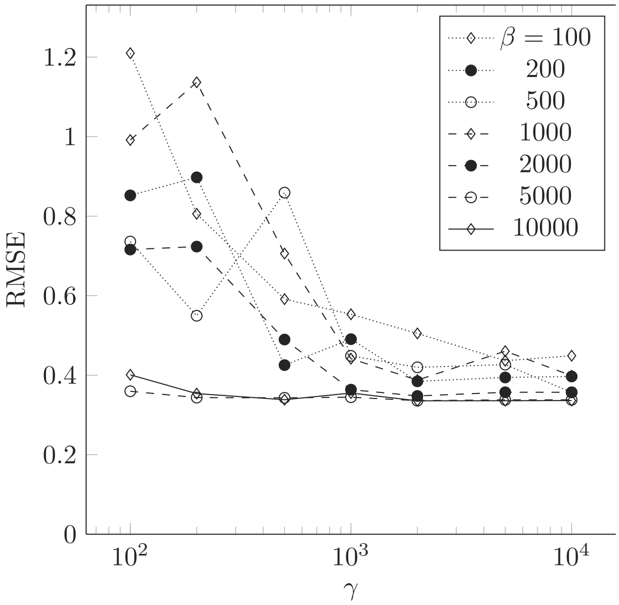

Fig. 6. Root mean square errors for the 'Lorenz' data set.

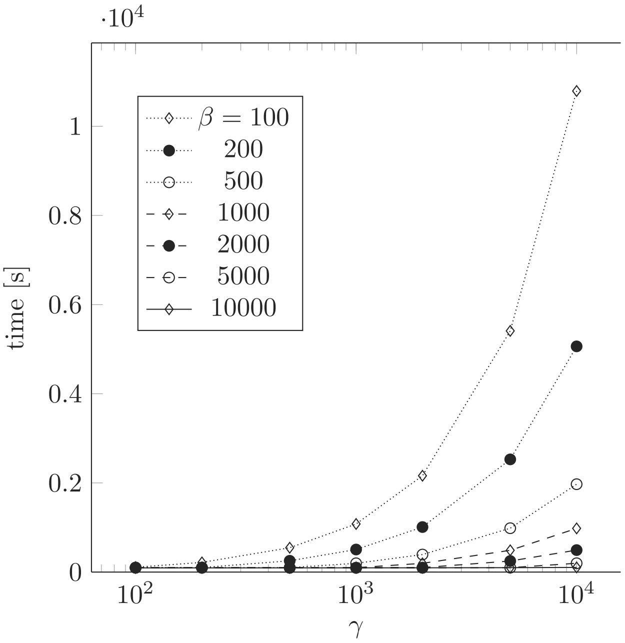

Fig. 7. Execution times for the 'Lorenz' data set.

The obtained results show that very low values of β and γ imply higher errors and longer execution times. For higher numbers β of items read in one block, the errors are

Table 2. Results produced for the ‘Rössler’ data set in the first experiment.

Number β of		Number γ of items generated from the set of granules						
items read		100	200	500	1000	2000	5000	10000
100	RMSE	0.2341	0.2404	0.2266	0.2297	0.2276	0.2315	0.2309
	MAE	0.1897	0.1996	0.1875	0.1906	0.1887	0.1929	0.1919
	time [s]	129	257	647	1283	2567	6435	12768
200	RMSE	0.4008	0.4305	0.4430	0.4247	0.4197	0.4221	0.4210
	MAE	0.3257	0.3469	0.3440	0.3382	0.3351	0.3368	0.3368
	time [s]	127	127	323	634	1268	3174	6372
500	RMSE	0.1330	0.1287	0.1300	0.1337	0.1314	0.1322	0.1291
	MAE	0.1060	0.1030	0.1041	0.1071	0.1051	0.1057	0.1033
	time [s]	133	132	132	264	521	1281	2510
1000	RMSE	0.1033	0.1032	0.1033	0.1032	0.1032	0.1034	0.1031
	MAE	0.0823	0.0823	0.0823	0.0823	0.0823	0.0824	0.0822
	time [s]	119	119	119	119	240	600	1198
2000	RMSE	0.1034	0.1032	0.1032	0.1031	0.1031	0.1031	0.1031
	MAE	0.0824	0.0822	0.0823	0.0822	0.0822	0.0822	0.0822
	time [s]	99	99	100	100	101	254	504
5000	RMSE	0.1035	0.1039	0.1031	0.1031	0.1031	0.1031	0.1031
	MAE	0.0825	0.0829	0.0822	0.0822	0.0822	0.0822	0.0822
	time [s]	94	94	94	95	96	98	197
10000	RMSE	0.1033	0.1037	0.1031	0.1031	0.1031	0.1031	0.1031
	MAE	0.0823	0.0827	0.0822	0.0822	0.0822	0.0822	0.0822
	time [s]	92	92	92	93	93	96	101
TSK	RMSE				0.1031			
	MAE				0.0821			
	time [s]				95			

almost the same as in the reference system (the differences are less than 0.1).

3.3.2. Experiment 2.

The second experiment is designed to show the ability of the system to handle large volume data, which is our objective. In this case we use data sets with 1 000 000, 5 000 000, and 10 000 000 data items. The data are so large that it is impossible to use a standalone reference TSK system, and only results for our granular neuro-fuzzy system can be presented in Table 5. The TSK system reads all data at once to identify and tune parameters of a fuzzy model (a set of fuzzy rules). If a data set is too large, it cannot be stored in computer memory. In this case we cannot compare the results of our system with the reference one. The granular system has linear time complexity in the number of data items.

Figure 9 presents the premises of produced rules for the ‘Lorenz’ data set with the granular TSK system. The premises of rules split the input domain in a regular way, and each region can be labelled with semantically rich tags (e.g., ‘very low’, ‘low’, ‘medium’, ‘high’, ‘very high’).

3.3.3. Experiment 3.

In this experiment we would like to test if the granulation technique can produce better results (lower error rates) than the simple sampling of large volume data. The granular TSK system is trained exactly in the same way as in Experiment 2. A TSK system is trained not with the whole train data set with 10 000 000 data items, but with a sampled data set that holds 100 000 data vectors taken at random from the whole data set. The granular approach can elaborate results with lower errors (Table 6). However, it takes 100 times more time to build the model, because the volume of data is 100 times larger.

3.3.4. Experiment 4.

This experiment aims at a comparison of the proposed neuro-fuzzy systems with other research result. The parameters are $\beta = \gamma = 25$, the number of rules $L = 6$, the number of tuning iterations 500. The experiments were repeated 20 times. The results are presented in Table 7. The proposed system yields quite good results results, although not the best ones.

Table 3. Results elaborated for the ‘Lorenz’ data set in the first experiment.

Number β of items read		Number γ of items generated from the set of granules						
		100	200	500	1000	2000	5000	10000
100	RMSE	1.2099	0.8059	0.5913	0.5532	0.5050	0.4364	0.4490
	MAE	0.9493	0.5792	0.4536	0.4222	0.3747	0.3281	0.3392
	time [s]	108	217	546	1079	2161	5410	10791
200	RMSE	0.8522	0.8976	0.4254	0.4906	0.3846	0.3943	0.3968
	MAE	0.6328	0.6892	0.3325	0.3689	0.3032	0.3095	0.3095
	time [s]	102	102	253	506	1010	2528	5063
500	RMSE	0.7361	0.5495	0.8591	0.4486	0.4198	0.4266	0.3569
	MAE	0.4902	0.4122	0.5494	0.3414	0.3240	0.3209	0.2795
	time [s]	98	99	99	197	393	984	1971
1000	RMSE	0.9914	1.1367	0.7059	0.4414	0.3876	0.4605	0.3988
	MAE	0.5747	0.6067	0.4684	0.3356	0.2947	0.3322	0.3007
	time [s]	97	97	97	98	196	488	980
2000	RMSE	0.7160	0.7234	0.4897	0.3636	0.3478	0.3571	0.3570
	MAE	0.4492	0.4576	0.3574	0.2752	0.2665	0.2743	0.2749
	time [s]	99	98	99	99	100	248	493
5000	RMSE	0.3596	0.3437	0.3431	0.3446	0.3360	0.3379	0.3380
	MAE	0.2698	0.2636	0.2637	0.2650	0.2578	0.2597	0.2595
	time [s]	96	96	95	94	96	99	198
10000	RMSE	0.4010	0.3537	0.3383	0.3551	0.3356	0.3358	0.3363
	MAE	0.3110	0.2732	0.2590	0.2746	0.2570	0.2574	0.2577
	time [s]	94	94	95	95	96	99	104
TSK	RMSE				0.3386			
	MAE				0.2591			
	time [s]				93			

4. Conclusions and future work

Neuro-fuzzy systems have proven their ability to build intelligible nonlinear models for presented data. However, their bottleneck is the volume of data. They have to read all data in order to produce a model. We addressed this problem with granular computing.

In the paper we applied the granular approach and proposed granular neuro-fuzzy system for large volume data. In our approach the data are read by parts and granulated. In the next stage the fuzzy model is produced not on data, but on granules.

We introduced a new form of granule—a fuzzy rule. In our system the granules are represented by numeric vectors and by fuzzy rules. Fuzzy rules are specific summaries of data. The experiments showed that the proposed granular neuro-fuzzy system can produce intelligible models even for large volume data sets. The quality of the elaborated fuzzy models is very close to that of the models produced by classical neuro-fuzzy sets (for small data sets where application of classical neuro-fuzzy systems and comparison are possible).

The experiments reveal that the granular approach can produce more precise models than preprocessing data and reducing the volume of data by sampling.

In future research we would like to focus on the precision of the degranulation process. Degranulation commonly introduces some errors. We would like to reduce them and simultaneously decrease the number of degranulated data items, which is supposed to shorten the time needed to run the system. We would also like to propose prototype-based fuzzy rules with a new form of granularised premises.

Acknowledgment

This work was co-financed by a Silesian University of Technology grant for the maintenance and development of research potential.

References

Ahmed, M.M. and Isa, N.A.M. (2017). Knowledge base to fuzzy information granule: A review from the

Table 4. Experiment 1b: root mean square errors, mean absolute errors, and execution time elaborated with the granular TSK and reference TSK systems in the function of the number of items read in one block (β) and the number of items generated from the set of granules (γ); for all granular TSK systems, $\beta = \gamma$. The results are presented in the form of average ± standard deviation.

		Data sets			
$\beta = \gamma$		'Surface'	'Mackey–Glass'	'Lorenz'	'Rössler'
100	RMSE	0.3107 ± 0.0029	0.0749 ± 0.0007	0.9614 ± 0.1400	0.2302 ± 0.0189
	MAE	0.2628 ± 0.0018	0.0596 ± 0.0006	0.7170 ± 0.0997	0.1865 ± 0.0145
	time [s]	44.00 ± 0.00	98.70 ± 2.28	102.30 ± 0.90	120.80 ± 0.40
200	RMSE	0.3079 ± 0.0024	0.0731 ± 0.0003	0.5820 ± 0.1153	0.4152 ± 0.0168
	MAE	0.2610 ± 0.0015	0.0582 ± 0.0003	0.4393 ± 0.0687	0.3299 ± 0.0143
	time [s]	42.70 ± 0.45	95.70 ± 2.10	97.10 ± 1.04	120.70 ± 0.45
500	RMSE	0.3057 ± 0.0011	0.0728 ± 0.0001	0.6124 ± 0.2033	0.1319 ± 0.0028
	MAE	0.2596 ± 0.0007	0.0579 ± 0.0001	0.4198 ± 0.1035	0.1055 ± 0.0025
	time [s]	42.00 ± 0.00	94.40 ± 2.05	93.50 ± 0.50	124.00 ± 0.00
1000	RMSE	0.3042 ± 0.0007	0.0727 ± 0.0001	0.6389 ± 0.1578	0.1032 ± 0.0000
	MAE	0.2586 ± 0.0005	0.0579 ± 0.0001	0.4255 ± 0.0710	0.823 ± 0.0000
	time [s]	42.00 ± 0.00	93.40 ± 1.85	93.10 ± 0.30	119.10 ± 0.30
2000	RMSE	0.3041 ± 0.0004	0.0727 ± 0.0001	0.4108 ± 0.0521	0.1031 ± 0.0000
	MAE	0.2586 ± 0.0002	0.0579 ± 0.0000	0.3022 ± 0.0261	0.0822 ± 0.0000
	time [s]	42.00 ± 0.00	94.50 ± 1.96	93.10 ± 0.30	100.30 ± 0.45
5000	RMSE	0.3042 ± 0.0005	0.0727 ± 0.0000	0.3375 ± 0.0016	0.1031 ± 0.0000
	MAE	0.2586 ± 0.0003	0.0579 ± 0.0000	0.2592 ± 0.0014	0.0822 ± 0.0000
	time [s]	43.00 ± 0.00	97.50 ± 1.91	96.00 ± 0.00	98.00 ± 0.00
10000	RMSE	0.3042 ± 0.0002	0.0727 ± 0.0000	0.3363 ± 0.0008	0.1031 ± 0.0000
	MAE	0.2586 ± 0.0002	0.0579 ± 0.0000	0.2576 ± 0.0007	0.0822 ± 0.0000
	time [s]	45.00 ± 0.00	102.60 ± 2.10	100.30 ± 0.45	101.00 ± 0.00
20000	RMSE	0.3046 ± 0.0004	0.0728 ± 0.0001	0.3360 ± 0.0005	0.1031 ± 0.0000
	MAE	0.2589 ± 0.0003	0.0579 ± 0.0000	0.2572 ± 0.0004	0.0822 ± 0.0000
	time [s]	49.10 ± 0.30	111.70 ± 2.23	110.00 ± 0.00	110.70 ± 0.90
50000	RMSE	0.3048 ± 0.0008	0.0727 ± 0.0001	0.3358 ± 0.0003	0.1031 ± 0.0000
	MAE	0.2590 ± 0.0005	0.0579 ± 0.0001	0.2570 ± 0.0002	0.0822 ± 0.0000
	time [s]	62.00 ± 0.00	140.00 ± 2.52	137.90 ± 0.30	139.20 ± 1.99
100000	RMSE	0.3045 ± 0.0003	0.0728 ± 0.0004	0.3387 ± 0.0000	0.1031 ± 0.0000
	MAE	0.2588 ± 0.0002	0.0579 ± 0.0004	0.2591 ± 0.0000	0.0822 ± 0.0000
	time [s]	41.40 ± 0.48	93.30 ± 1.61	92.00 ± 0.00	92.00 ± 0.00

interpretability-accuracy perspective, *Applied Soft Computing* **54**: 121–140.

Alcalá, R., Alcalá-Fdez, J., Casillas, J., Cordón, O. and Herrera, F. (2006). Hybrid learning models to get the interpretability-accuracy trade-off in fuzzy modeling, *Soft Computing* **10**(9): 717–734.

Alonso, J.M. and Magdalena, L. (2011). Special issue on interpretable fuzzy systems, *Information Sciences* **181**(20): 4331–4339.

Atanassov, K.T. (1986). Intuitionistic fuzzy sets, *Fuzzy Sets and Systems* **20**(1): 87–96.

Bargiela, A. and Pedrycz, W. (2006). The roots of granular computing, *2006 IEEE International Conference on Granular Computing, Atlanta, USA*, pp. 806–809.

Bisi, C., Chiaselotti, G., Ciucci, D., Gentile, T. and Infusino, F.G. (2017). Micro and macro models of granular computing induced by the indiscernibility relation, *Information Sciences* **388–389**: 247–273.

Botta, A., Lazzerini, B., Marcelloni, F. and Stefanescu, D.C. (2009). Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index, *Soft Computing* **13**(5): 437–449.

Table 5. Experiment 2: root mean square errors, mean absolute errors, and execution time elaborated by the granular TSK and reference TSK systems in the function of the number of items read in one block (β) and the number of items generated from the set of granules (γ); for all granular TSK systems, $\beta = \gamma$. The results are presented in the form of average ± standard deviation. For larger data sets the results for TSK are missing, because the data are too large for the system to process them.

		Data sets							
Number		‘Surface’		‘Mackey–Glass’		‘Lorenz’		‘Rössler’	
of items		GrTSK	TSK	GrTSK	TSK	GrTSK	TSK	GrTSK	TSK
100 000	RMSE	0.3007	0.3004	0.0720	0.0720	0.3369	0.3369	0.1031	0.1031
	MAE	0.2565	0.2563	0.0572	0.0572	0.2579	0.2579	0.0822	0.0822
	time [s]	41	41	94	91	92	91	93	93
1 000 000	RMSE	0.3014	0.2911	0.0719	0.0704	0.3356	0.3283	0.1034	0.1034
	MAE	0.2568	0.2509	0.0571	0.0558	0.2564	0.2513	0.0824	0.0824
	time [s]	473	421	1054	945	1021	922	1018	944
5 000 000	RMSE	0.3015	–	0.0718	–	0.3356	–	0.1030	–
	MAE	0.2569	–	0.0570	–	0.2574	–	0.0821	–
	time [s]	2156	–	4786	–	4709	–	4724	–
10 000 000	RMSE	0.3017	–	0.0722	–	0.3356	–	0.1032	–
	MAE	0.2571	–	0.0574	–	0.2568	–	0.0823	–
	time [s]	4220	–	9406	–	9512	–	9350	–

Table 6. Experiment 3: averaged of root mean square errors, mean absolute errors, and execution time elaborated by the granular TSK and reference TSK systems for the training dataset with 10 000 000 items. The granular system processes the whole data set. The reference TSK works on a 100 000-item random sample of data.

Dataset	Method	RMSE	MAE	time [s]
‘Surface’	GrTSK	0.3017	0.2571	4220
	sampling + TSK	0.7114	0.5862	42
‘Mackey–Glass’	GrTSK	0.0722	0.0574	9406
	sampling + TSK	0.0816	0.0669	91
‘Lorenz’	GrTSK	0.3356	0.2568	9512
	sampling + TSK	0.3391	0.2613	91
‘Rössler’	GrTSK	0.1032	0.0823	9350
	sampling + TSK	0.1235	0.1025	91

Table 7. Experiment 4: comparison of root mean square error for the ‘noisy’ data set. The RMSE column holds averages ± standard deviations; when only one value is present, it is the average.

Method	RMSE	source
SONFIN	0.169	Juang and Lin, 1998
T2FLS	0.167	Mendel, 2004
T2SONFS	0.138	Juang and Tsao, 2008
DIT2NFS-AC	0.124	Juang and Chen, 2013
DIT2NFS-IP	0.124	Juang and Chen, 2013
DIT2NFS-S1	0.123	Juang and Chen, 2013
GrTSK	0.116664 ± 0.003857	This paper
IT2NFSIB	0.107919 ± 0.002393	Siminski, 2017

Ciucci, D. (2016). Orthopairs and granular computing, *Granular Computing* **1**: 159–170.

Cpałka, K., Łapa, K., Przybył, A. and Zalasiński, M. (2014). A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects, *Neurocomputing* **135**: 203–217.

Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters, *Journal Cybernetics* **3**(3): 32–57.

Evsukoff, A.G., Galichet, S., de Lima, B.S. and Ebecken, N.F. (2009). Design of interpretable fuzzy rule-based classifiers using spectral analysis with structure and parameters optimization, *Fuzzy Sets and Systems* **160**(7): 857–881.

Gacto, M.J., Alcalá, R. and Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, *Information Sciences* **181**(20): 4340–4360.

Herrera, L.J., Pomares, H., Rojas, I., Valenzuela, O. and Prieto, A. (2005). TASE, a Taylor series-based fuzzy system model that combines interpretability and accuracy, *Fuzzy Sets and Systems* **153**(3): 403–427.

Hu, X., Pedrycz, W., Wu, G. and Wang, X. (2017). Data reconstruction with information granules: An augmented method of fuzzy clustering, *Applied Soft Computing* **55**: 523–532.

Juang, C.-F. and Chen, C.-Y. (2013). Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability, *IEEE Transactions on Cybernetics* **43**(6): 1781–1795.

Juang, C.-F. and Lin, C.-T. (1998). An online self-constructing neural fuzzy inference network and its applications, *IEEE Transactions on Fuzzy Systems* **6**(1): 12–32.

Juang, C.F. and Tsao, Y.W. (2008). A type-2 self-organizing neural fuzzy system and its FPGA implementation, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**(6): 1537–1548.

Keet, C.M. (2008). *A Formal Theory of Granularity*, PhD thesis, Free University of Bozen-Bolzano, Bolzano.

Leski, J. (2008). *Neuro-Fuzzy Systems*, Scientific and Engineering Publishers WNT, Warsaw, (in Polish).

Lorenz, E.N. (1963). Deterministic nonperiodic flow, *Journal of the Atmospheric Sciences* **20**(2): 130–141.

Mackey, M.C. and Glass, L. (1977). Oscillation and chaos in physiological control systems, *Science* **197**(4300): 287–289.

Mendel, J.M. (2004). Computing derivatives in interval type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems* **12**(1): 84–98.

Pawlak, Z. (1996). Rough sets, rough relations and rough functions, *Fundamenta Informaticae* **27**(2): 103–108.

Pedrycz, A., Hirota, K., Pedrycz, W. and Dong, F. (2012). Granular representation and granular computing with fuzzy sets, *Fuzzy Sets and Systems* **203**: 17–32.

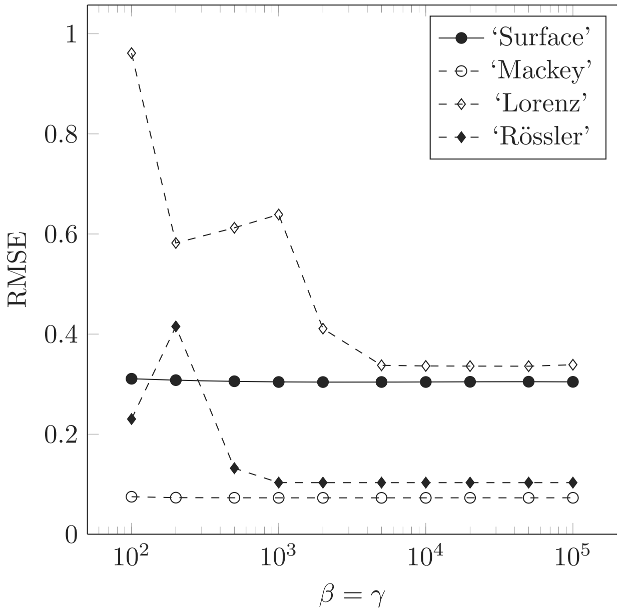

Fig. 8. Experiment 1b: root mean square error for various values $\beta = \gamma$.

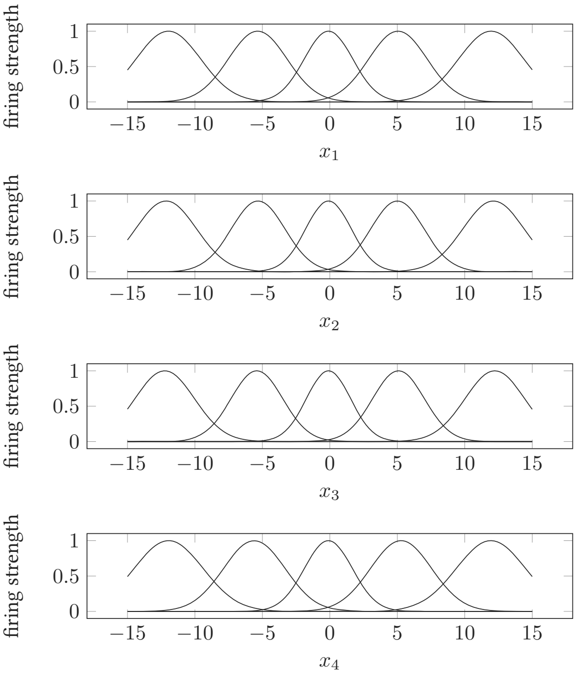

Fig. 9. Premises of five rules for attributes x_1, x_2, x_3, and x_4 produced for the 'Lorenz' data set with the granular TSK system. The splits of the input domain are similar but not identical.

Pedrycz, W. (1998). Shadowed sets: Representing and processing fuzzy sets, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **28**(1): 103–109.

Pedrycz, W. (2013). *Granular Computing: Analysis and Design of Intelligent Systems*, CRC Press, Boca Raton.

Pedrycz, W. and Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*, John Wiley, Hoboken.

Pedrycz, W., Hmouz, R.A., Balamash, A.S. and Morfeq, A. (2015a). Hierarchical granular clustering: An emergence of information granules of higher type and higher order, *IEEE Transactions on Fuzzy Systems* **23**(6): 2270–2283.

Pedrycz, W. and Homenda, W. (2013). Building the fundamentals of granular computing: A principle of justifiable granularity, *Applied Soft Computing* **13**(10): 4209–4218.

Pedrycz, W., Succi, G., Sillitti, A. and Iljazi, J. (2015b). Data description: A general framework of information granules, *Knowledge-Based Systems* **80**: 98–108.

Qian, Y.H., Liang, J.Y., Yao, Y.Y. and Dang, C.Y. (2010). MGRS: A multi-granulation rough set, *Information Sciences* **180**(6): 949–970.

Qian, Y., Liang, J., Wu, W.-Z. and Dang, C. (2011). Information granularity in fuzzy binary GrC model, *IEEE Transactions on Fuzzy Systems* **19**(2): 253–264.

Reyes-Galaviz, O.F. and Pedrycz, W. (2015). Granular fuzzy models: Analysis, design, and evaluation, *International Journal of Approximate Reasoning* **64**: 1–19.

Rössler, O.E. (1976). An equation for continuous chaos, *Physics Letters A* **57**(5): 397–398.

Salehi, S., Selamat, A. and Fujita, H. (2015). Systematic mapping study on granular computing, *Knowledge-Based Systems* **80**: 78–97.

Shifei, D., Li, X., Hong, Z. and Liwen, Z. (2010). Research and progress of cluster algorithms based on granular computing, *International Journal of Digital Content Technology and Its Applications* **4**(5): 96–104.

Siminski, K. (2014). Neuro-fuzzy system with weighted attributes, *Soft Computing* **18**(2): 285–297.

Siminski, K. (2015). Rough subspace neuro-fuzzy system, *Fuzzy Sets and Systems* **269**: 30–46.

Siminski, K. (2017). Interval type-2 neuro-fuzzy system with implication-based inference mechanism, *Expert Systems with Applications* **79C**: 140–152.

Siminski, K. (2019). NFL—free library for fuzzy and neuro-fuzzy systems, *in* S. Kozielski *et al.* (Eds), *Beyond Databases, Architectures and Structures. Paving the Road to Smart Data Processing and Analysis*, Springer International Publishing, Cham, pp. 139–150.

Siminski, K. (2020). GrFCM—Granular clustering of granular data, *in* A. Gruca *et al.* (Eds), *Man–Machine Interactions 6*, Springer, Cham, pp. 111–121.

Siminski, K. (2021). An outlier-robust neuro-fuzzy system for classification and regression, *International Journal of Applied Mathematics and Computer Science* **31**(2): 303–319, DOI: 10.34768/amcs-2021-0021.

Skowron, A., Jankowski, A. and Dutta, S. (2016). Interactive granular computing, *Granular Computing* **1**: 95–113.

Sugeno, M. and Kang, G.T. (1988). Structure identification of fuzzy model, *Fuzzy Sets ans Systems* **28**(1): 15–33.

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

Wang, D., Pedrycz, W. and Li, Z. (2019). Granular data aggregation: An adaptive principle of the justifiable granularity approach, *IEEE Transactions on Cybernetics* **49**(2): 1–10.

Yang, X., Li, T., Liu, D. and Fujita, H. (2019). A temporal-spatial composite sequential approach of three-way granular computing, *Information Sciences* **486**: 171–189.

Yao, J.T., Vasilakos, A.V. and Pedrycz, W. (2013). Granular computing: Perspectives and challenges, *IEEE Transactions on Cybernetics* **43**(6): 1977–1989.

Yao, Y. (2007). The art of granular computing, *in* M. Kryszkiewicz *et al.* (Eds), *Rough Sets and Intelligent Systems Paradigms,* Springer, Berlin/Heidelberg, pp. 101–112.

Yao, Y. (2008). Granular computing: Past, present and future, *2008 IEEE International Conference on Granular Computing, GrC 2008, Hangzhou, China*, pp. 80–85.

Yao, Y. (2018). Three-way decision and granular computing, *International Journal of Approximate Reasoning* **103**: 107–123.

Yao, Y. (2020). Three-way granular computing, rough sets, and formal concept analysis, *International Journal of Approximate Reasoning* **116**: 106–125.

Yao, Y. and Zhong, N. (2007). Granular computing, *in* B.W. Wah (Ed.), *Wiley Encyclopedia of Computer Science and Engineering,* Wiley, Hoboken.

Yen, J., Wang, L. and Gillespie, C. W. (1998). Improving the interpretability of TSK fuzzy models by combining global learning and local learning, *IEEE Transactions on Fuzzy Systems* **6**(4): 530–537.

Zadeh, L.A. (1965). Fuzzy sets, *Information and Control* **8**: 338–353.

Zadeh, L.A. (1979). Fuzzy sets and information granularity, *in* N. Gupta *et al.* (Eds), *Advances in Fuzzy Set Theory and Applications*, North-Holland Publishing, Amsterdam, pp. 3–18.

Zadeh, L.A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems* **90**(2): 111–127.

Zadeh, L.A. (2002). From computing with numbers to computing with words—From manipulation of measurements to manipulation of perceptions, *International Journal of Applied Mathematics and Computer Science* **12**(3): 307–324.

Krzysztof Siminski received his DSc from the Faculty of Automatic Control, Electronics and Computer Science of the Silesian University of Technology, Gliwice, Poland, where he currently works. His main scientific interests focus on data analysis, data mining, machine learning, computational intelligence, granular computing, and formal languages.

Received: 25 March 2021
Revised: 28 May 2021
Accepted: 28 June 2021