Application of Sequential Neural Networks to Predict the Approximation Degree of Decision-making Systems

Jarosław Szkoła[1] [a] and Piotr Artiemjew[2] [b]

[1] *The Department of Computer Science, University of Rzeszow, 35-310 Rzeszow, Poland*

[2] *Faculty of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn, Poland*

Keywords: Rough Sets, Granulation of Knowledge, Rough Inclusions, Standard Granulation, Estimation of the Approximation Degree, Sequential Neural Networks.

Abstract: The paradigm of granular computing appeared from an idea proposed by L. A Zadeh, who assumed that a key element of data mining techniques is the grouping of objects using similarity measures. He assumed that similar objects could have similar decision classes. This assumption also guides other scientific streams such as reasoning by analogy, nearest neighbour method, and rough set methods. This assumption leads to the implication that grouped data, (granules) can be used to reduce the volume of decision systems while preserving their classification efficiency - internal knowledge. This hypothesis has been verified in practice - including in the works of Polkowski and Artiemjew (2007 - 2015) - where they use rough inclusions proposed by Polkowski and Skowron as an approximation tool - using the approximation scheme proposed by Polkowski. In this work, we present the application of sequential neural networks to estimate the degree of approximation of decision systems (the degree of reduction in the size) based on the degree of indiscernibility of the decision system. We use the standard granulation method as a reference method. Pre-estimation of the degree of approximation is an important problem for the considered techniques, in the context of the possibility of their rapid application. This estimation allows the selection of optimal parameters without the need for a classification process.

1 INTRODUCTION

1.1 A Few Words of Introduction to Rough Set Theory and Granular Computing

As the basic form of collecting knowledge about certain problems, we use information systems - in the sense of collecting intelligence needed to solve problems. The possibility of modelling decision-making processes is provided by indiscernibility relations, see (Pawlak, 1992) We define an information system (Pawlak, 1992) as $InfSys = (Uni, Attr)$, where Uni is the universe of objects, $Attr$ the set of attributes describing the objects. We assume that Uni and $Attr$ are finite and nonempty. Attributes $a \in Atr$ describe the objects of the universe by means of certain values from their domain (V_a) - that is, $a : U \to V_a$. By adding some expert decisions (problem solutions) to the information system, we obtain a decision system that can be defined as a triple $DecSys(Uni, Attr, dec)$, where $dec \notin Attr$. Problems $u, v \in Uni$ are *B–indiscernible* whenever $a(u) = a(v)$ for every $a \in B$, for each attribute set B. This assumption is modelled by indiscernibility relation $IND(B)$.

$$(u, v) \in IND(B) \text{ iff } a(u) = a(v) \text{ for each } a \in B.$$

The $IND(B)$ relation divides the universe of objects into classes

$$[u]_B = \{v \in Uni : (u, v) \in IND(B)\},$$

Defined classes form B-primitive granules, collections over $InfSys$. Connections of primitive granules form elementary granules of knowledge. The descriptor language (Pawlak, 1992) is used to describe information systems, where a descriptor is defined as (a, v) with $a \in Attr, v \in V_a$. Formulas are created from the descriptors using the connectors $\vee, \wedge, \neg, \Rightarrow$. Let us define the semantics of formulas, , where $[[p]]$ denotes the meaning of a formula p:

$$\begin{array}{c} 1.\ [[(a,v)]] = \{u \in U : a(u) = v\} \\ 2.\ [[p \vee q]] = [[p]] \cup [[q]] \\ 3.\ [[p \wedge q]] = [[p]] \cap [[q]] \\ 4.\ [[\neg p]] = U \setminus [[p]]. \end{array} \quad (1)$$

[a] https://orcid.org/0000-0002-6043-3313

[b] https://orcid.org/0000-0001-5508-9856

Szkoła, J. and Artiemjew, P.
Application of Sequential Neural Networks to Predict the Approximation Degree of Decision-making Systems.
DOI: 10.5220/0010992300003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 984-989
ISBN: 978-989-758-547-0; ISSN: 2184-433X

Objects of the universe Uni in attribute terms B are described by a feature vector $inf_B(u) = \{(a = a(u)) : a \in B\}$

1.2 Theoretical Introduction to the Reference Graulation Method

The paradigm of granular computing in the context of approximate reasoning was proposed by Zadeh in (Zadeh, 1979) and has been naturally incorporated into the rough sets paradigm - see (Lin, 2005), (L. Polkowski, 2001), (L.Polkowski, 1999), (Skowron and Stepaniuk, 2001).

One direction in the development of granulation methods in the context of rough sets has been the use of rough inclusions, formally derived from the paradigm of rough mereology see (L. Polkowski, 2001), (L.Polkowski, 1999), (Polkowski, 2005), (Polkowski, 2006).

In this particular work we will consider one of the approximate inclusions derived from Łukasiewicz t–norm. Considering a given system $(Uni, Attr)$, the rough inclusion $\mu_L(u,v,r)$ is defined by means of the formula,

$$\mu_L(u,v,r) \Leftrightarrow \frac{|IND(u,v)|}{|Attr|} \geq r$$

$$IND(u,v) = \{a \in Attr : a(u) = a(v)\}$$

The formula $\mu_L(u,v,r)$ defines a collection of objects indiscernible from the central object at some fixed degree r (radius of granulation). The granulation idea that we use in this work as a base method was proposed by Polkowski in (Polkowski, 2005). It consists in forming for the $InfSys$ a collection of granules $Gr = mathcalG(Uni)$, with which the whole universe of objects Uni is covered, by forming a collection $Cov(U)$ according to a fixed strategy Str. For each granule $g \in Cov(Uni)$, and each attribute $a \in A$, a value $\overline{a}(g) = \mathcal{S}\{a(u) : u \in g\}$ is determined. The system is reduced in size as determined by the granulation radius r used and the covering strategy. We use majority voting as a reference strategy when forming objects from granules see (Duda et al., 2000).

The hypothesis that information systems can be approximated by rough inclusions according to the described scheme - (Polkowski, 2005) - has been verified in many works by Polkowski and Artiemjew, including (Polkowski, 2015).

1.3 Goal of This Work

In this work, we use Polkowski's approximation method (standard granulation) as a reference method to check the possibility of estimating the degree of approximation (percentage of reduction of decision systems). For the estimation, we used the internal degree of r-indiscernibility computed as the number of combinations without repetition of system objects r-indiscernible from each other - similar at least in degree r. To achieve our goal, we used sequential neural networks; we used the naive forecasting method as a reference method. Our aim was not to find the best existing technique, but to test the initial possibility of modelling the problem under study.

The rest of the paper is organised as follows. Sect. 2 contains the methodology. Sect. 3 contains the results of experiments. Section 4 contain the conclusions and future works.

Let us move on to present a toy example of creating a granular reflection of a decision system using standard granulation (see Fig. 1).

2 RESEARCH METHODOLOGY

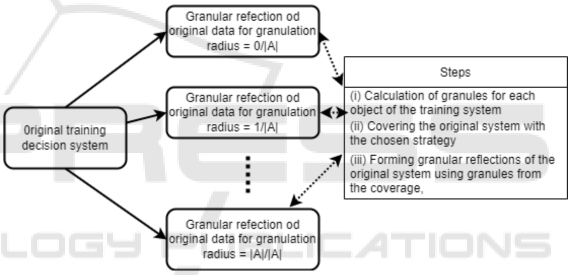

Granular refection od original data for granulation radius = 0/|A|
Granular refection od original data for granulation radius = 1/|A|
Granular refection od original data for granulation radius = |A|/|A|
0riginal training decision system
Steps
(i) Calculation of granules for each object of the training system
(ii) Covering the original system with the chosen strategy
(iii) Forming granular reflections of the original system using granules from the coverage,

Figure 1: In the figure we have a brief demonstration of the standard granulation process.

2.1 Standard Granulation - Toy Example

Assuming that $g_{r_{gran}}(u_i) = \{u_j \in U_{trn} : \frac{|IND(u_i,u_j)|}{|A|} \geq r_{gran}\}$

$IND(u_i,u_j) = \{a \in A; a(u_i) = a(u_j)\}$

U_{trn} is the universe of training objects,

and $|X|$ is the cardinality of set

The sample standard granules with a 0.25 radius, derived from decision systems from Tab. 1 look as follows,

$$g_{0.25}(u_1) = \{u_1,\}$$

$$g_{0.25}(u_2) = \{u_2,u_4,\}$$

$$g_{0.25}(u_3) = \{u_3,u_6,u_8,u_{10},\}$$

$$g_{0.25}(u_4) = \{u_2, u_4, \}$$
$$g_{0.25}(u_5) = \{u_5, \}$$
$$g_{0.25}(u_6) = \{u_3, u_6, u_8, u_{10}, \}$$
$$g_{0.25}(u_7) = \{u_7, \}$$
$$g_{0.25}(u_8) = \{u_3, u_6, u_8, u_{10}, \}$$
$$g_{0.25}(u_9) = \{u_9, \}$$
$$g_{0.25}(u_{10}) = \{u_3, u_6, u_8, u_{10}, \}$$

Random coverage of training systems is as follows, $Cover(U_{trn}) = \{g_{0.25}(u_1), g_{0.25}(u_4), g_{0.25}(u_5), g_{0.25}(u_7), g_{0.25}(u_8), g_{0.25}(u_9), \}$

To summarise, the example described is the granulation of system from Tab. 1, as an assisting granularity tool array from Tab. 2 is used. The granular reflection of the original system is in the Tab. 3.

Table 1: Exemplary decision system: diabetes, 9 attributes, 10 objects.

Day	*a*1	*a*2	*a*3	*a*4	*a*5	*a*6	*a*7	*a*8	*class*
u_1	6	148	72	35	0	33.6	0.627	50	1
u_2	1	85	66	29	0	26.6	0.351	31	0
u_3	8	183	64	0	0	23.3	0.672	32	1
u_4	1	89	66	23	94	28.1	0.167	21	0
u_5	0	137	40	35	168	43.1	2.288	33	1
u_6	5	116	74	0	0	25.6	0.201	30	0
u_7	3	78	50	32	88	31.0	0.248	26	1
u_8	10	115	0	0	0	35.3	0.134	29	0
u_9	2	197	70	45	543	30.5	0.158	53	1
u_{10}	8	125	96	0	0	0.0	0.232	54	1

Table 2: Triangular indiscernibility matrix for standard granulation ($i < j$), derived from Tab. 1 $c_{ij} = 1,\ if\ \frac{|IND(u_i,u_j)|}{|A|} \geq 0.25\ and\ d(u_i) = d(u_j),\ 0,\ otherwise.$

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
u_1	1	0	0	0	0	0	0	0	0	0
u_2	x	1	0	1	0	0	0	0	0	0
u_3	x	x	1	0	0	1	0	1	0	1
u_4	x	x	x	1	0	0	0	0	0	0
u_5	x	x	x	x	1	0	0	0	0	0
u_6	x	x	x	x	x	1	0	1	0	1
u_7	x	x	x	x	x	x	1	0	0	0
u_8	x	x	x	x	x	x	x	1	0	1
u_9	x	x	x	x	x	x	x	x	1	0
u_{10}	x	x	x	x	x	x	x	x	x	1

Let us now turn to the introduction of the architecture and description of the sequential neural network used to achieve the goal of the work.

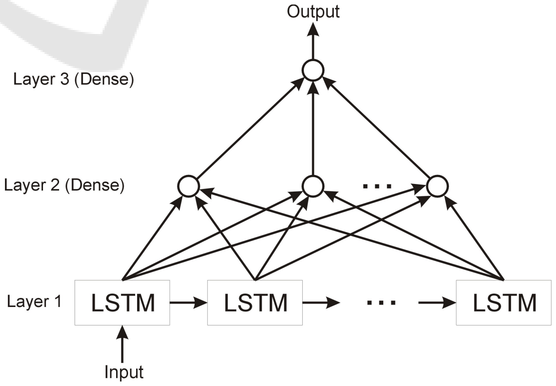
Table 3: Standard granular reflection of the exemplary training system from Tab. 1, in radius 0.25, 9 attributes, 6 objects; MV is Majority Voting procedure (the most frequent descriptors create a granular reflection).

Day	*a*1	*a*2	*a*3	*a*4	*a*5	*a*6	*a*7	*a*8	*class*
$MV(g_{0.25}(u_1))$	6	148	72	35	0	33.6	0.627	50	1
$MV(g_{0.25}(u_4))$	1	85	66	29	0	26.6	0.351	31	0
$MV(g_{0.25}(u_5))$	0	137	40	35	168	43.1	2.288	33	1
$MV(g_{0.25}(u_7))$	3	78	50	32	88	31.0	0.248	26	1
$MV(g_{0.25}(u_8))$	8	183	64	0	0	23.3	0.672	32	1
$MV(g_{0.25}(u_9))$	2	197	70	45	543	30.5	0.158	53	1

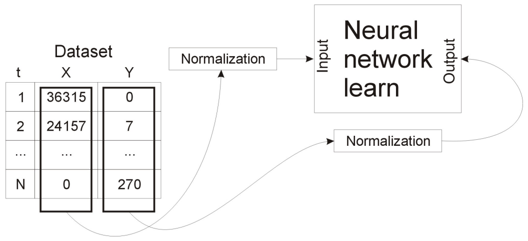
2.2 Forecasting using a Sequential Neural Network

We used the MSE (Mean Square Error) parameter to estimate the quality of the presented forecasting method. Which, when comparing these parameters on specific data, is an objective solution.

The three-layer sequential neural network architecture (Szkoła et al., 2011) (Szkola et al., 2011) (Li et al., 2020) was proposed for predicting the output data, the first layer contains LSTM cells, in subsequent layers full connections between the preceding layer and the current one. In the first layer, was used from 50 to 100 neurons for different input datasets, in the second layer 25 and in the last layer - one neuron. This configuration works well in most cases of the input datasets. To obtain a sufficiently high quality of prediction, number of neurons in the network should be carfully selected. After many attempts, we can recommend using 2 - 5 times more neurons in the first layer than the input time samples of the data from the individual time steps. With this range of input neurons, we could achieve good balans beetwen genralization and overfiting.

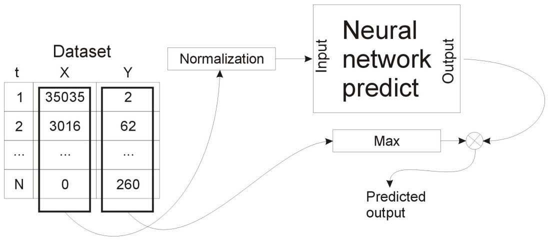


Before the learning process begins, the input data must be properly processed to the form in which learning can take place (Szkoła et al., 2011). The input data supplied to the network should be in the range in which the training algorithms based on gradient methods do not cause rapid saturation. In practice, the input data is converted to the range [0 - 1] or [-1, 1]. The presented algorithm uses the method of normalizing the input and output data separately, to the range of values [0 - 1]. Attempts were also made to transform the input data by means of logarithmic normalization, but it did not improve the classification / prediction of the data, therefore it was decided that the classical normalizing algorithm would be used.



The learning process involves the sequential feeding of individual samples to the input of the network, based on a method known as online learning (batch size = 1). The network is learned through 1000 epochs, if less epochs was used, accuracy was lower value. As the loss function is used mean_squared_error, the optimizer used is the Adaptive Moment Estimation algorithm.

A simple pre-processing and post-processing operation was used to match the network response to the expected output values (Szkoła and Pancerz, 2019). For the input data we used normalization algorithm. For the compute output value, in the first step, the maximum value that is possible on the output for the training data is calculated. Then this value is multiplied by the network response to the given excitation, as a result of which we obtain the predicted response in the range of values consistent with the input data. Due to the range of values that the network can return, it is not possible to obtain the target value directly from the network, therefore a simple operation of multiplying the network response by an appropriate factor is required.



The number of neurons in the first layer has a great influence on the correct operation of the network. The number of neurons in this layer should be greater than the number of data records. The tests show that good results are obtained if the number of the neurons in the first layer is at least two or more times greater than the number of input records. Changes in the number of neurons in the next layer have a smaller impact on the functioning of the network.

$$MSE = \sum_{i=1}^{number\ of\ data\ points} (observed_i - predicted_i)^2$$

Let us move on to discuss the experimental verification of the objective defined in Sect. 1.3.

3 EXPERIMENTAL SESSION

3.1 Data Preparation

For the experiments, we selected a few distinctly different datasets from the UCI repository (Dua and Graff, 2017).

(i) Australian Credit (dims.: 15, items: 690);

(ii) Car (dims.: 7, items: 1728);

(iii) Congressional house votes (dims.: 17, items: 435);

(iv) Fertility (dims.: 10, items: 100);

(v) German credit (dims.: 20, items: 1000);

(vi) Heart Disease (dims.: 14, items: 270);

(vii) Pima Indians Diabetes (dims.: 9, items: 768).

(viii) Soybean large (dims.: 36, items: 683);

(ix) SPECTF (dims.: 45, items: 267);

(x) SPECT (dims.: 23, items: 267);

For simplicity in the standard granulation process, all attributes are treated as symbolic.

3.2 Results Description

For the Australian Credit Dataset. Considering the granulation (approximation) radii as: $\{\frac{0}{|A|}, \frac{1}{|A|}, ..., \frac{|A|}{|A|}\}$, the size of the granulated (approximated) systems is as follows:1 1 2 3 4 9 24 67 141 355 551 663 682 684 690. A naive prediction solution applying the previous step to the evaluation gives an estimate: 1 2 3 5 7 13 33 91 208 496 906 1214 1345 1366 1374. Finally our sequential neural network which is based on indiscernibility degrees 237705 237360 233062 215300 178050 125809 73344 33300 10990 2432 351 42 18 9 0 gives an estimate: 2.954362 -1.041474 0.421173 0.848999 1.51634 5.706955 19.51853 62.379723 138.676437 353.199066 550.998535 660.025024 684.502258 687.227844 686.925232. Calculating the MSE, for the naive solution is 1827233, and for our approach done with a sequential neural network is around 117. The exact data on which the neural network worked can be seen in Tab 4.

Table 4: Data for **Australian Credit Diabetes**: In the table we have summary information about the values used by the neural network (input), the expected values, and the values predicted by the network.

input	expected	predicted
237705.000000	0.000000	2.954362
237360.000000	0.000000	-1.041474
233062.000000	1.001451	0.421173
215300.000000	2.002903	0.848999
178050.000000	3.004354	1.516340
125809.000000	8.011611	5.706955
73344.000000	23.033382	19.518530
33300.000000	66.095791	62.379723
10990.000000	140.203193	138.676437
2432.000000	354.513788	353.199066
351.000000	550.798258	550.998535
42.000000	662.960813	660.025024
18.000000	681.988389	684.502258
9.000000	683.991292	687.227844
0.000000	690.000000	686.925232

For the Pima Indians Diabetes Dataset. The size of the granulated systems is as follows:1 23 122 393 633 757 768 768 768. A naive prediction solution: 1 24 145 515 1026 1390 1525 1536 1536. Finally our sequential neural network gives an estimate: 7.4e-05 22.028669 121.157761 392.511017 632.823975 756.985718 768 767.999939 768. Calculating the MSE, for the naive solution is 2323249, and for our approach done with a sequential neural network is around 3. The exact data on which the neural network worked can be seen in Tab 5.

Table 5: Data for **Pima Indians Diabetes**: In the table we have summary information about the values used by the neural network (input), the expected values, and the values predicted by the network.

input	expected	predicted
294528.000000	0.000000	0.000074
115994.000000	22.028683	22.028669
38060.000000	121.157757	121.157761
5271.000000	392.511082	392.511017
375.000000	632.823990	632.823975
13.000000	756.985658	756.985718
0.000000	768.000000	768.000000
0.000000	768.000000	767.999939
0.000000	768.000000	768.000000

For the Heart Disease Dataset. The size of the granulated systems is as follows:1 1 3 3 8 12 25 63 135 214 261 270 270 270. A naive prediction solution: 1 2 4 6 11 20 37 88 198 349 475 531 540 540. Finally our sequential neural network gives an estimate: 2.384996 1.747474 5.694103 4.539211 7.185006 16.565092 35.658192 74.358444 142.456848 216.848175 255.135269 260.954498 261.508392 263.523743 Calculating the MSE, for the naive solution is 282764, and for our approach done with a sequential neural network is around 570. The exact data on which the neural network worked can be seen in Tab 6.

Table 6: Data for **Heart Disease**: In the table we have summary information about the values used by the neural network (input), the expected values, and the values predicted by the network.

input	expected	predicted
36315.000000	0.000000	2.384996
36160.000000	0.000000	1.747474
35035.000000	2.007435	5.694103
31203.000000	2.007435	4.539211
24157.000000	7.026022	7.185006
15514.000000	11.040892	16.565092
7882.000000	24.089219	35.658192
3016.000000	62.230483	74.358444
780.000000	134.498141	142.456848
137.000000	213.791822	216.848175
10.000000	260.966543	255.135269
0.000000	270.000000	260.954498
0.000000	270.000000	261.508392
0.000000	270.000000	263.523743

For the rest of the examined data, we only present the calculated MSE values, which can be seen in the summary - see Tab. 7.

3.3 Summary of the Results

The results in Tab. 7 show the great advantage of forecasting with a sequential neural network over the naive method. Our method allows us to estimate the degree of approximation (expected reduction of the original decision systems) in a reasonably accurate way. The goal we set in section 1.3 has been achieved. It is worth noting that the estimation presented is a preview of the possibility of estimating the approximation degrees and there is a slight overfitting process on the examined data. The only way to compensate for this process was to select an appropriate neural network structure. Certainly, the possibility of estimating successive degrees of approximation is higher than the efficiency of the naive method.

Table 7: Summary results of the efficiency of predicting the size of granular systems vs the efficiency of a naive soluation. A smaller MSE value means better prediction precision.

dataset	MSE naive	MSE of neural network
Australian credit	1827233	117
Pima Indians Diabetes	2323249	3
Heart disease	282764	570
Car	126860	133
Fertility	13879	11
German credit	4732593	2239
Hepatitis	108585	789
Congressional house votes	42254	780
Soybean large	477873	6786
SPECT	78780	
SPECTF	2596795	116

4 CONCLUSIONS

In this ongoing work, we verified that it is possible to predict the degree of approximation of decision systems based on their internal degree of indiscernibility. To achieve this goal, we used sequential neural networks, whose efficiency proved statistically superior to the naive prediction method. In the initial estimation model used, we are aware of a slight overfitting process due to the characteristics of the data sequences used. Despite promising initial results, much is left to be done to evaluate the final performance and determine the application of this new method. The discovery of the ability to estimate approximation degrees described in the paper opens up several new research threads. First, we intend to investigate whether the estimated degrees allow us to estimate the behaviour of decision systems in a double granulation process. That is, to verify whether the optimal approximation parameters can be directly estimated from the degrees of indiscernibility of the decision systems. Another horizon of potential research is to try estimating the course of the approximation on previously unseen data based on other data with a similar degree of indiscernibility.

REFERENCES

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition.

L. Polkowski, A. S. (2001). Rough mereological calculi of granules: A rough set approach to computation. *Computational Intelligence. An International Journal*, pages 472–492.

Li, G., Xu, J., Shen, W., Wang, W., Liu, Z., and Ding, G. (2020). Lstm-based frequency hopping sequence prediction. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 472–477.

Lin, T. (2005). Granular computing: examples, intuitions and modeling. In *2005 IEEE International Conference on Granular Computing*, volume 1, pages 40–44.

L.Polkowski, A. (1999). Towards an adaptive calculus of granules. *In: L.A.Zadeh, J.Kacprzyk (Eds.), Computing with Words in Information/Intelligent Systems 1., Physica Verlag*, pages 201–228.

Pawlak, Z. (1992). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, USA.

Polkowski, L. (2005). Formal granular calculi based on rough inclusions. pages 57–62. In Proceedings of the 2005 IEEE Conference on Granular Computing.

Polkowski, L. (2006). A model of granular computing with applications. granules from rough inclusions in information systems. pages 9 – 16.

Polkowski, L.; Artiemjew, P. (2015). *Granular Computing in Decision Approximation - An Application of Rough Mereology*. Springer: Cham, Switzerland.

Skowron, A. and Stepaniuk, J. (2001). Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.*, 16:57–85.

Szkoła, J. and Pancerz, K. (2019). Pattern recognition in sequences using multistate sequence autoencoding neural networks. In *2019 International Conference on Information and Digital Technologies (IDT)*, pages 443–448.

Szkoła, J., Pancerz, K., and Warchoł, J. (2011). A bezier curve approximation of the speech signal in the classification process of laryngopathies. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 141–146.

Szkoła, J., Pancerz, K., and Warchoł, J. (2011). Recurrent neural networks in computer-based clinical decision support for laryngopathies: An experimental study. *Intell. Neuroscience*, 2011.

Szkola, J., Pancerz, K., Warchol, J., Babiloni, F., Fred, A., Filipe, J., and Gamboa, H. (2011). Improving learning ability of recurrent neural networks-experiments on speech signals of patients with laryngopathies. In *BIOSIGNALS*, pages 360–364.

Zadeh, L. (1979). Fuzzy sets and information granularity. Technical Report UCB/ERL M79/45, EECS Department, University of California, Berkeley.