Optimizing Mixed Fuzzy-Rule Formation by Controlled Evolutionary Strategy

Matthias Lermer, Hendrik Kuijs, Christoph Reich
Institute for Cloud Computing and IT Security
Furtwangen University of Applied Science
Furtwangen, Germany
Email: {matthias.lermer, hendrik.kuijs, christoph.reich}@hs-furtwangen.de

***Abstract*—Machine learning algorithms are heavily applied to address many challenges in various fields. This paper specifically takes a look at use cases from the health sector, as well as the industry 4.0 sector. In both cases, the knowledge about the classification process is as important as the classification itself. One current problem is the disregard of expert knowledge provided by adept human beings. In practice, it is possible and also feasible to learn similar knowledge with machine learning algorithms like artificial neural networks (ANNs) or support vector machines (SVMs). However, time and money could be saved if this expert knowledge was used directly. Right now, this is only possible with more transparent algorithms like rule-based systems or decision trees, where knowledge can be incorporated relatively easily. The approach of this paper shows that rules generated by a mixed fuzzy-rule formation algorithm can be optimized by applying a controlled evolutionary strategy while maintaining the interpretability of the decision-making process. The evaluation is performed by executing the evolutionary strategy proposed in this paper on data from two different industries.**

Keywords*–*Evolutionary Strategy; Optimization; Fuzzy Logic; Decision support systems; Industry 4.0.

I. Introduction

Nowadays, there is a trend towards using deep learning algorithms, e.g., Deep Neural Networks (DNN), for almost any kind of Machine Learning problem [1]. One of the earlier disadvantages, the slow computation with those kind of algorithms, has been overcome successfully with the help of graphics cards and their optimized cores [2]. Still, one of the big remaining problems is the interpretability of the results when using black box algorithms like DNNs [3][4]. There are many recent approaches to make those results more transparent, but those are still in their infancy [5][6][7]. Other Machine Learning algorithms are more transparent, e.g. Rule-based systems or Decision trees and can provide a human understandable explanation. In practice however, this transparency often comes with the price of worse prediction results.

The approach depends on the use case or the Machine Learning problem itself. Is it more important to absolutely get the best result possible? Or can a weaker result be tolerated if explanations and knowledge about the results origins can be acquired? In case of the two different scenarios evaluated in this paper, the transparent way to the result is as important as the outcome itself.

The remainder of the paper is organized as follows: Section II provides an overview about related work. Section III describes the genetic adaptation of the Mixed Fuzzy-Rule Formation. In Sections IV and V, the evaluations based on two different Use Cases are conducted. Section VI completes the paper by drawing a conclusion and suggesting future work.

II. Related Work

Elsayed et al. [8] combine fuzzy rules and evolutionary algorithms, albeit in a different way than in our approach. In their solution, two algorithms cooperate by using fuzzy rules with complementary characteristics. This results in a higher success rate when applied on different data sets with different optimization problems. Their approach is especially interesting as it can be used to further optimize the method proposed in this paper.

Schaer et al. have shown that the adjustment of established fuzzy rules and fuzzy set functions can lead to better results [9]. Their work was evaluated within an autonomous car racing competition where they could improve the previous score by 0.5 %. The adjustments and optimizations of the fuzzy components were mainly the product of simulation experiments. In the conclusion, they are mentioning that there are plans to use genetic algorithms for the adjustments which is similar to the evolutionary strategy approach proposed in this paper.

Jariyatantiwait and Yen [10] follow the special approach of Differential Evolution (DE). They apply their modification on the ZDT (Zitzler, Deb and Thiele) and DTLZ (Deb, Thiele, Laumanns and Zitzler) test suits [11], which are used for evaluating the optimization of algorithms and map the optimization directly to fuzzy rules. Those rules adapt certain control parameters during the evolution process. Examples are the degree of greediness and exploration. They successfully show that performance metrics can be combined with human understandable knowledge in the form of fuzzy rules. The work conducted in this paper takes a similar approach, but tries to combine classification tasks themselves with fuzzy rules while control parameters like the degree of exploration are defined by hand.

Alcalá-Fdez et al. [12] show that their modification of a evolutionary fuzzy-rule based system leads to an improved performance within monotonic classification problems. In contrast to this paper, the authors used genetic algorithms and concentrated on adjusting crossover mechanisms, including customised incest prevention and restarting processes while the mutation mechanism was kept relatively simple by hardcoding the mutation rate.

The works from various other authors in this section show that evolutionary strategies within classification problems hold a high value, given the good results and the preserved interpretability by humans. This can be observed for many more use cases, e.g., financial market [13] [14], medicine [15], computer science [16], etc. and reinforce the choice to take a deeper look at the two uses cases of this paper. However, a direct comparison to other works with different use cases is very difficult to accomplish and would go beyond the scope of this paper. Furthermore, modifying different parts of the evolution process is still a heavily pursued research topic, regardless of the use case.

III. Preliminary

This work relies on the fuzzy rules generated by the mixed fuzzy-rule formation proposed by Berthold [17]. The decision to use this specific kind of fuzzy logic as a basis for genetic adaptation was based on its ability to cope with high-dimensional data sets while delivering good classification performance. Additionally, the created rules can be interpreted by humans and further expanded on using knowledge engineering. Table I shows a quick and shortened example of a rule.

TABLE I. EXAMPLE RULE

age	*operation year*	*axillary nodes*	*survival*
(24, 48, 60, 64)	(03, 06, 07, 07)	(19, 21, 35, 46)	true

The attributes age, operation year and axillary nodes are described textually in this example. Those 4 values per attribute are to be seen in the usual fuzzy partition context. Looking at the age value, this means that people ranging from 48 - 60 years are members of the complete (survival = true) set, while the rest of the people in ranges from 24-48 and 60-64 years are only partial members of the (survival = true) set. The same applies for the other attributes and the rule is only valid when logically combining all the attributes. The following configuration is used during the fuzzy rule generation with the mixed fuzzy-rule formation algorithm:

- Shrink rules after each commit to reduce conflicting rules
- Use the class with maximum coverage for training
- Min/Max fuzzy norm for the rule activation computation
- Volume Border Based shrink function after the complete rule set has been established to further reduce conflicting rules

Even with shrinking mechanisms in place, many rules are created by the algorithm. It depends on the size of the dataset and its attributes. More data usually results in more created rules when using mixed fuzzy-rule formation. In order to further minimize possible conflicts, only the two rules that represent a class with the highest weight are chosen for the Evolutionary Strategy (ES). This has mainly two reasons.

Firstly, the application of many rules to a big data set can become time consuming, which is still a practical problem. Fernandez et. al show in [18] that solutions for this challenge are still in their infancy. Practical solutions proposed by Rio et al. [19] furthermore show that, depending on the use case, there is always a speed-accuracy trade-off. The second reason is the easier comprehensibility by human experts. To have less, but more robust rules additionally aids in the process of battling over fitting. From a research point of view, this is especially interesting as one of the major problems when implementing predictive maintenance in the context of Industry 4.0 is the ability to generalize the created or extracted knowledge to subsets of machine types, like Computer numerical control (CNC) grinding machines.

A. *Workflow*

The following list provides a quick overview of the workflow depicted in Figure 1.

1) Split the data into Training/Validation (70%) and Test sets (30%)
2) Create a complete rule-set using the mixed fuzzy-rule formation algorithm
3) Filter the two rules with the highest weight per prediction class
4) Apply the rules to the Test Set
5) Adapt the filtered rules by mutation
6) Compare the results based on F-measure
7) Keep on mutating
8) Stop the iteration after a defined Terminal Condition for Mutation (TC M) has been met. This can take on the form of an interval, result or event
9) Save the adapted rules and results
10) Split the data into training/validation (70%) and test sets (30%) again, but in a different way than before
11) Adapt the mutated rules from the previous iteration
12) Apply the adapted rules to Test Set and save the result
13) Compare the results
14) Stop the iteration after a defined interval, result or event

The algorithm stops the current iteration and starts a new one as soon as a defined terminal condition (TC M) is satisfied. The fitness function used for comparing the results is defined as the classification F-measure, which is selected in order to consider precision, as well as recall. At first, the TC M is called when the algorithm does not improve the F-measure after a certain number of iterations, which could be rather limiting considering that mutations are based on randomness. Another factor was that the mutated rules should never mutate so much that they completely change their meaning. So, a rather low number of maximum 42 mutations per mutation iteration is allowed. When applied to the use cases described in Section V, the final terminal condition is defined as a maximum of 15 mutation iteration rounds. This definition is based on the decision to optimize the existing rules and not to create new ones. In test runs, it was evaluated that with a very high number of mutation rounds, the underlying rule could not be identified any more. To mitigate over fitting, the following two procedures are implemented. Firstly, the first training set in the first mutation round includes data which is held back and used only for validation purposes. Secondly, after each mutation iteration the dataset is split again in a different way based on the pseudo random generator provided by the python random library.

IV. Evolutionary Strategy

The adaptation process concentrates on mutating the generated rules in order to optimize those rules. This procedure pursues a slightly different approach compared to classical genetic algorithms (GAs). Evolutionary Strategy usually does not include a crossover mechanism for the population adaptation.

There are mainly two reasons to concentrate on ES. Firstly, when including crossover mechanisms, the fuzzy rules often drastically change and do not represent their original meaning any more. This stands in opposition to the focus in this paper, which is to optimize existing rules which are built upon

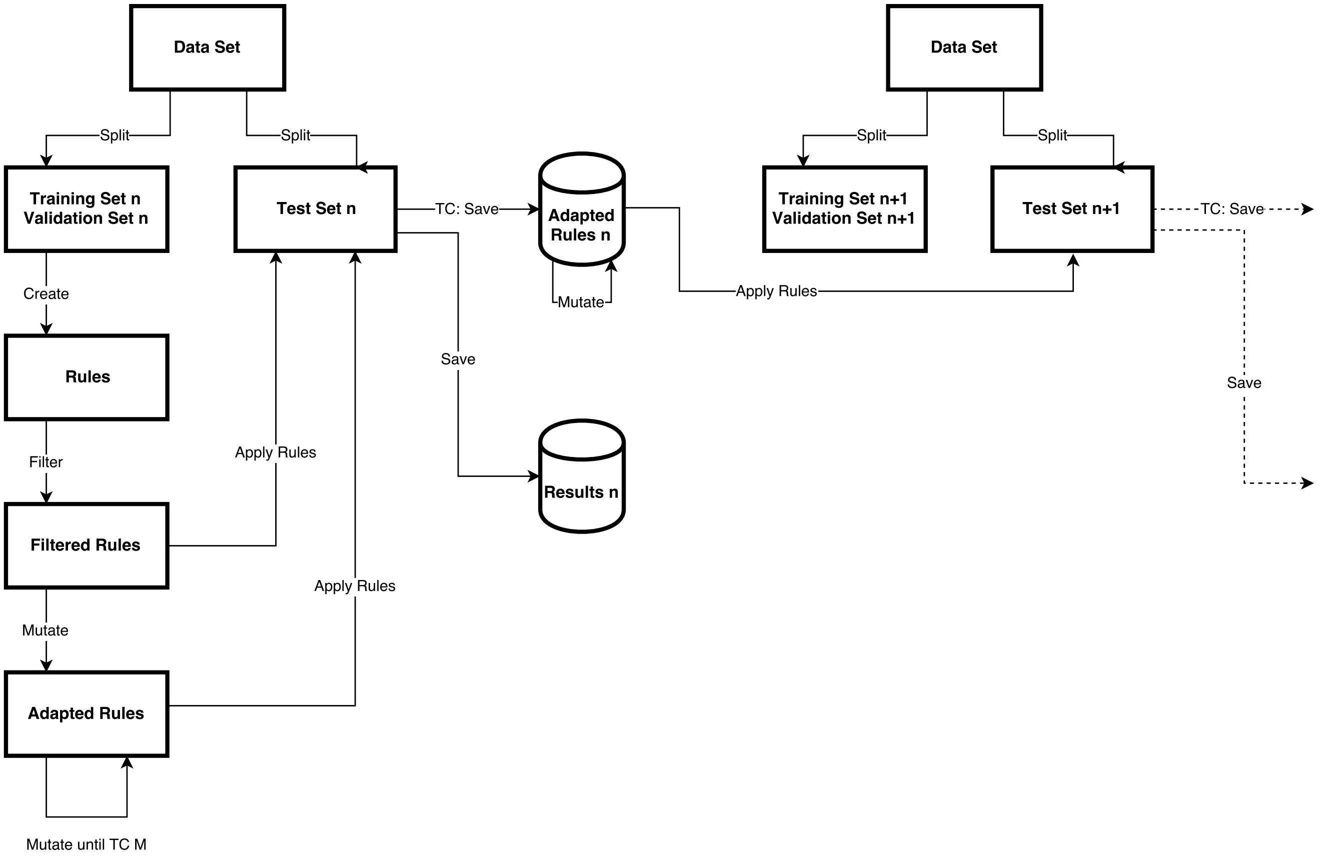

Figure 1. Workflow

specific knowledge. Secondly, when only using mutation, the algorithm can, if needed, dynamically be controlled to a high degree. The first approach was to start with a lower mutation rate (meaning values within the rules were allowed to change up to +-5%) and let the rate steadily increase. The second approach was to start with a higher mutation rate of 40 % and let it steadily decline. In practice however, the best results were achieved using a random mutation rate, confined to a change within -40 % to + 40 %. Surprisingly, this is true for both data sets described in Section V.

V. Evaluation with Use Cases

The following use cases will show the practical applicability of the proposed genetic adaptation. The decision to evaluate with the help of two different data sets is made in order to get a brief look at the generalization potential of the algorithm.

A. Evaluation 1: Health Sector

The ’Haberman’s Survival Data’ [20] provides information about the survival status of breast cancer patients who underwent surgery. This multivariate dataset contains 306 examples and was gathered by the Billings Hospital in Chicago. The data was provided by the Machine Learning Repository of the University of California, Irvine [21] with the following description of the attributes:

- Age of patient at time of operation (numerical)
- Patient’s year of operation (numerical)
- Number of positive axillary nodes detected (numerical)
- Survival Status (binary class attribute)
    - ○ 1 = the patient survived 5 years or longer
    - ○ 2 = the patient died within 5 years

The mutation is performed on the fuzzy rules attributes age, operation year and axillary nodes, shown in Table II. Like previously mentioned, the maximum change of one attribute per mutation lies between -40 % and +40 %. As the data in this use case only consisted of integers, the mutation also delivers only integers. A control mechanism detects if violations of the hard and soft boundaries of the fuzzy rules were the result of a mutation and rolls the fuzzy rule back to the state one step before the violating mutation has occurred. Table II shows one rule created by mixed fuzzy-rule formation and two mutations. Mutation 1 is violating the fuzzy rule in the last value of the attribute age while mutation 2 represents a valid mutation.

TABLE II. MUTATIONS OF AN EXAMPLE RULE

mutation	*age*	*operation year*	*axillary nodes*	*survival*
-	(24, 48, 60, 64)	(03, 06, 07, 07)	(19, 21, 35, 46)	true
1	(24, 48, 60, 59)	(03, 06, 07, 07)	(19, 21, 35, 46)	true
2	(24, 48, 60, 62)	(03, 06, 07, 07)	(19, 21, 35, 46)	true

Figure 2 gives a graphical overview of one part of a rule. The soft limits can clearly be seen in green colour at the operation years 60 and 67 while the hard limits are represented by years 59 and 69.

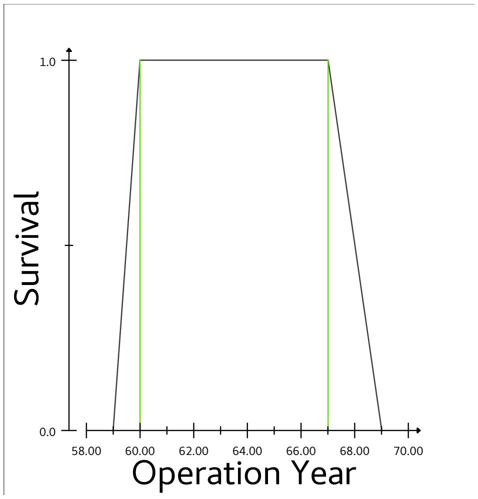

Figure 2. Part of a fuzzy rule

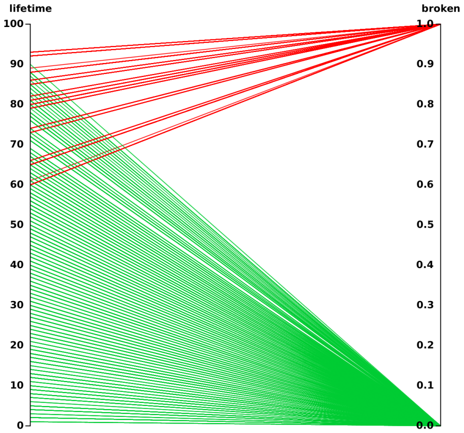

Figure 3. Correlation of lifetime and machine health

TABLE III. SCORING STATISTICS: HEALTH SECTOR

Iteration	Mutations	Rate	F-measure
0	-	-	0.696
1	30	24 %	0.695
2	35	2 %	0.691
3	7	10 %	0.684
4	42	4 %	0.682
5	29	30 %	0.681
6	14	13 %	0.684
7	31	29 %	0.685
8	13	9 %	0.690
9	5	21 %	0.693
10	10	37 %	0.699
11	31	29 %	0.697
12	13	9 %	0.686
13	28	21 %	0.689
14	10	7%	0.693
15 (TC) = 10			0.699

Table III shows the individual iterations of the adaptation process. The number of mutations is listed next to the mutation rate, which represents the average change of the values during that iteration. It can be seen that in iterations 10 and 11 the algorithm leads to a better result. Iteration 15 triggers the terminal condition and iteration 10 is selected, as it improves the F-measure by 0.3 %. Although the improvement may seem small, the new knowledge in form of the mutated fuzzy rules could be evaluated and used by human experts.

B. *Evaluation 2: Industry 4.0*

Industry 4.0 and predictive maintenance is a hot topic in research and business right now. Companies want to precisely predict the date and time a machine needs maintenance in order to produce more efficiently [22]. Often, many sensors are added to machines or along the production line. The data gathered by these sensors is then used to build the predictive models. The dataset used in this evaluation has been provided by Ludovic in [23] and consists of 1001 records. The interesting fact about this dataset is that it also provides additional information like the responsible maintenance team for a certain machine. The following attributes are provided in the data set:

- lifetime of the machine in weeks (numeric)
- pressure (numeric)
- moisture (numeric)
- temperature (numeric)
- provider of the machine (string)
- responsible maintenance team (string)
- status
  - 1 = machine is broken
  - 0 = machine is still working

Another interesting part about this data set is that the information provided is relatively easy to obtain for different kind of machines. Corresponding cost-effective sensors for temperature or moisture measurement can usually be additionally installed, regardless of the age of the machine. In contrast to the first use case, the values of the sensor attributes consist of floating-point numbers. The mutation was performed on the numerical attributes lifetime, pressure, temperature and moisture. It turns out that mutating the string type attributes like provider and maintenance team had a too strong impact on the original fuzzy rule. This makes sense as a slight change in those data types can completely turn a fuzzy rule on its head. Figure 3 shows the correlation of lifetime and health of a machine. Looking at this graph, it makes sense to use a rule-based system to model this correlation and use it for classifications and predictions. However, Figure 4 shows that it is not as easy when looking for correlations of the temperature and the health of a machine as there seems to be a rather equal distribution. Thankfully, fuzzy rule-based systems can cover the correlations between attributes thanks to the soft- and hard boundaries as shown in Table II and, at the same time, retain transparency.

Table IV shows that in this use case, the algorithm improves in iterations 2, 3, 4 and 9 compared to the original fuzzy rule. The F-measure is improved by 0.4 %. This time, the best results are obtained after relatively few iterations.

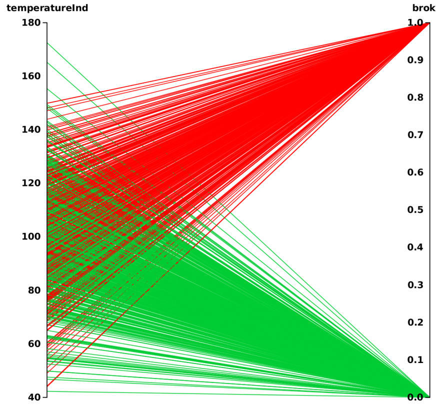

Figure 4. Correlation of temperature and machine health

TABLE IV. SCORING STATISTICS: INDUSTRY 4.0

Iteration	Mutations	Rate	F-measure
0	-	-	0.650
1	3	13 %	0.650
2	23	20 %	0.654
3	30	14 %	0.651
4	1	2 %	0.651
5	13	21 %	0.645
6	16	19 %	0.647
7	25	1 %	0.647
8	40	28 %	0.650
9	31	13 %	0.651
10	27	9 %	0.649
11	9	5 %	0.645
12	22	39 %	0.646
13	8	17 %	0.642
14	20	7 %	0.640
15 (TC) = 2			0.654

VI. Conclusion and Future Work

The evaluation within two different use cases with different data sets shows that fuzzy rules, generated by mixed fuzzy-rule formation, can be optimized by using the proposed method. The evolutionary strategy is primary based on mutation in order to keep the changes assessable. The implemented control mechanism while mutating ensures that the fuzzy partitioning within the fuzzy rule were not violated. Currently, this optimization is only possible with numerical type attributes within the fuzzy rules.

In the future, it should be evaluated how to deal with string and binary values during the mutation process as one minor change already results in big changes to the fuzzy rule itself. Future work includes the application to bigger data sets with more attributes and records to see if the algorithm can scale accordingly. At the same time, the implications of a bigger data set on the trade of between accuracy and coverage of the data set have to be evaluated. The impact on the computation time has to be analysed as well for those cases. Additionally, it must be evaluated if the algorithm can work for regression problems, too. In general, the algorithm must be tested with more use cases to be able to make comprehensible assumptions about generalisation potential and possibilities. The algorithm will additionally be evaluated with machine data gathered from grinding machines and lathes. This will be a similar use case to the second use case described in this paper. But it should be kept in mind that even simple changes, like slightly different positioned sensors could already complicate the ability to generalize well, even when using the exactly same types of sensors. Furthermore, both use cases should be evaluated in a detailed comparison with other machine learning algorithms, e.g., artificial neural networks.

VII. Acknowledgement

This work has received funding from INTERREG Upper Rhine (European Regional Development Fund) and the Ministries for Research of Baden-Wuerttemberg, Rheinland-Pfalz and from the Region Grand Est in the framework of the Science Offensive Upper Rhine.

References

[1] Statista, "Deep learning market size in U.S. by segment 2014-2025 — Statistic," Available: https://www.statista.com/statistics/779696/united-states-deep-learning-market-size, [retrieved: 03, 2018].

[2] NVIDIA, "Deep Learning & Artificial Intelligence Solutions from NVIDIA," Available: https://www.nvidia.com/en-us/deep-learning-ai, [retrieved: 03, 2018].

[3] S. Shirataki and S. Yamaguchi, "A study on interpretability of decision of machine learning," in 2017 IEEE International Conference on Big Data (Big Data), Dec 2017, pp. 4830–4831.

[4] W. Knight, "The dark secret at the heart of ai," 2017, Available: https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/, [retrieved: 03, 2018].

[5] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," Digital Signal Processing, vol. 73, 2018, pp. 1 – 15.

[6] C. Wu, M. J. F. Gales, A. Ragni, P. Karanasou, and K. C. Sim, "Improving interpretability and regularization in deep learning," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 2, Feb 2018, pp. 256–265.

[7] D. Wang, C. Quek, A. H. Tan, C. Miao, G. S. Ng, and Y. Zhou, "Leveraging the trade-off between accuracy and interpretability in a hybrid intelligent system," in 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Dec 2017, pp. 55–60.

[8] S. Elsayed, R. Sarker, and C. A. C. Coello, "Fuzzy rule-based design of evolutionary algorithm for optimization," IEEE Transactions on Cybernetics, vol. PP, no. 99, 2017, pp. 1–14.

[9] M. Schaer, M. Brennenstuhl, and R. Dornberger, "Improving the fuzzy logic controller of a car racing competition with adjusted fuzzy sets," in 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), Sept 2016, pp. 118–124.

[10] C. Jariyatantiwait and G. G. Yen, "Fuzzy multiobjective differential evolution using performance metrics feedback," in 2014 IEEE Congress on Evolutionary Computation (CEC), July 2014, pp. 1959–1966.

[11] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization. London: Springer London, 2005, pp. 105–145.

[12] J. Alcalá-Fdez, R. Alcalá, S. González, Y. Nojima, and S. García, "Evolutionary fuzzy rule-based methods for monotonic classification," IEEE Transactions on Fuzzy Systems, vol. 25, no. 6, Dec 2017, pp. 1376–1390.

[13] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, "Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification," IEEE Transactions on Fuzzy Systems, vol. 25, no. 2, April 2017, pp. 249–264.

[14] J. A. Sanz, D. Bernardo, F. Herrera, H. Bustince, and H. Hagras, "A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data," IEEE Transactions on Fuzzy Systems, vol. 23, no. 4, Aug 2015, pp. 973–990.

[15] N. L. Tsakiridis, J. B. Theocharis, and G. C. Zalidis, “A fuzzy rule-based system utilizing differential evolution with an application in vis-nir soil spectroscopy,” in 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), July 2017, pp. 1–7.

[16] T. T. Huan, P. D. Huynh, C. V. Kien, and H. P. H. Anh, “Implementation of hybrid adaptive fuzzy sliding model control and evolutionary neural observer for biped robot systems,” in 2017 International Conference on System Science and Engineering (ICSSE), July 2017, pp. 77–82.

[17] M. R. Berthold, “Mixed fuzzy rule formation,” International Journal of Approximate Reasoning, vol. 32, no. 2, 2003, pp. 67 – 84, soft Computing in Information Mining.

[18] A. Fernandez, C. J. Carmona, M. J. Jesus, and F. Herrera, “A view on fuzzy systems for big data: Progress and opportunities,” in International Journal of Computational Intelligence Systems, vol. 9, March 2016, pp. 69–80.

[19] S. del Río, V. Lòpez, J. M. Benítez, and F. Herrera, “A mapreduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules,” International Journal of Computational Intelligence Systems, vol. 8, no. 3, 2015, pp. 422–437.

[20] S. J. Haberman, “Generalized residuals for log-linear models,” in Proceedings of the 9th International Biometrics Conference, Boston, 1976, pp. 104–122.

[21] M. Lichman, “UCI machine learning repository,” 2013, Available: http://archive.ics.uci.edu/ml, [retrieved: 03, 2018].

[22] Roland Berger, “Predictive Maintenance / Servicing tomorrow and where we are really at today,” VDMA, Tech. Rep., 2017, Available: https://www.rolandberger.com/de/Publications/pub_predictive_maintenance_2017.html, [retrieved: 03, 2018].

[23] B. Ludovic, “Predictive maintenance,” 2018, Available: https://www.kaggle.com/ludobenistant/predictive-maintenance-1/data, [retrieved: 03, 2018].