A Framework for Comparative Analysis of Data Mining Algorithms

Duško Mirković*, Ivan Luković**, Nikola Obrenović*, Đurđa Rogić*
*Schneider Electric DMS NS LLC, Novi Sad, Serbia
** Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
{dusko.mirkovic, nikola.obrenovic, djurdja.rogic}@schneider-electric-dms.com, ivan@uns.ac.rs

Abstract—Consumer load profiles play an important role in distribution network analysis. Determining typical consumers can be based on a data mining algorithm called clustering or cluster analysis. Given the multitude of clustering algorithms available today and the disparity of data sets, algorithm selection becomes a non trivial task. One approach to this task is to use multi-criteria decision making algorithms to rank data mining algorithms based on performance and other relevant metrics. In order to move focus from algorithm ranking to selection and tuning, one needs a framework that offers performance data manipulation as well as flexible and customizable ranking algorithms. This paper proposes one such framework that will support research of consumer clusterisation algorithms for power distribution networks.

I. INTRODUCTION

One of the main characteristics of electric energy is that it cannot be directly stored in relevant quantities from the standpoint of distribution systems. Electric energy is cheapest when it is produced in large quantities in facilities such as coal or nuclear power plant. However, these power plants have relatively high response time for changing the output power and starting or stopping may take several days. More responsive power plants are subject to limitations such as geographical location (e.g. hydroelectric power plants require water reservoir such as natural or artificial lake), available capacity (e.g. water level in reservoir), high marginal energy cost, environmental issues etc. Indirect methods for energy storage, such as pumped hydroelectric (PHES) or compressed air (CAES) energy storage, are also subject to certain limitations such as geographical location (e.g. PHES requires two reservoirs at different heights), capital cost, power rating, available capacity and environmental issues [1]. Exact consumer needs at any given time are, in many cases, poorly predictable. Some of the main causes are the multitude of ways electric power is used today as well as number of non-industrial consumers. Home generation units such as photovoltaic cells introduce even more variance. The balance between production and consumption is the reason behind load forecast - a process of estimating future consumer needs.

Typical distribution network consists of thousands of nodes and branches. Having a measurement of relevant physical quantities for each node and branch is not economically justifiable, due to the high price of smart metering devices. However, determining the state of the network is crucial step that is providing necessary input for all other calculations and analyses.

Individual consumer load is a stochastic variable. This makes it very hard to develop a model for each individual consumer. To overcome this uncertainty consumers are grouped based on similar demand and assigned a load profile that represents average consumer in that group. Data mining algorithms that can discover such groups are called clustering or cluster analysis algorithms. Load profiles created this way are used in power distribution system for the processes of state estimation and load forecast.

Nowadays there are many algorithms for data clustering and classification. One of the main issues in any research that is based on data mining algorithms is selecting the algorithm which will give the best results for a given data set. Rice in [2] presented a base for algorithm selection problem in general, based on approximation theory. Wolpert and Macready showed that all algorithms that search for an extremum of a cost function perform exactly the same, according to any performance measure, when averaged over all possible cost functions [3]. Dubes and Jain compared several clustering algorithms from the user's perspective and concluded that rational basis for comparing clustering methods is needed with links to well-understood mathematical and statistical methods [4].

Conclusions of Wolpert and Macready in [3] and Dubes and Jain in [4] imply that data set characteristics are tightly coupled with performance of particular algorithm and play an important role in algorithm selection. However, handling that data and ranking should not take much effort that would be better spent trying new algorithms or tuning existing ones. Therefore, a framework is needed to handle all those tasks and allow researchers to focus on experiment.

This paper proposes a framework that provides benchmarking and a comparative analysis of data mining algorithms with regard to data set characteristics as well as all relevant performance metrics. The framework is to provide flexible data model and an extensible process for performance indices collection. Collected data are used to rank algorithms by one or multiple Multi-Criteria Decision Making (MCDM) algorithms.

Such framework will serve as a workbench for further research in selecting the most appropriate algorithm for consumer clustering in power distribution systems.

Beside the Introduction and Conclusion, this paper consists of three sections. In section 2 we present related work. In Section 3 we analyze main requirements for the framework we present in this paper. The framework architecture is presented in section 4.

II. RELATED WORK

This section presents related work from three aspects: existing MCDM software, use of MCDM for data mining algorithm ranking (selection) and MCDM algorithms.

International Society on Multiple Criteria Decision Making[1] offers an extensive list of MCDM related software that we considered for our research. However, each of the solutions that we considered had some limitations that drove us to the need to develop a new framework that will support our and possibly many other researches in the field of data mining. Most of the solutions that we considered only supported one MCDM algorithm which can be limiting for research teams that want to try different algorithms to find the one that is most suitable for their actual research. Our framework aims to provide extensible model that allows virtually any MCDM algorithm to be plugged in and tested against collected data.

Another limitation that we encountered is that MCDM solutions require manual entry of all alternatives and their attributes. This may be necessary for project portfolios where none of the attributes can be gathered automatically. Our framework aims to provide means to describe a workflow. For each step of the workflow performance data is collected automatically. Also, each step of the workflow is modular so we can easily explore variations in performance by substituting only one step of the workflow. Solutions that we considered were either web (cloud) based or standalone tools. Our framework aims to provide easy collaboration mechanism such as peer to peer, without the need for central server host.

Microsoft offers a cloud based environment for data mining - Azure Machine Learning[2], which provides means for result evaluation. However, to the best of our knowledge it does not provide any MCDM algorithms that would help in selecting appropriate solution from several non-dominated solutions.

We believe that software that would fulfill most of the requirements for many disparate research projects would be hard to build and would take too much effort. On the other hand, our framework provides basic building blocks that can be used to build custom tailored solutions. It also provides referent implementation that can be either directly used or adapted for particular situation. This means that each research project has flexible starting point that does not require experienced software engineer to customize according to project specifics. Common components allow better knowledge sharing and exchange of experience.

One of the first attempts of using MCDM approach to solve the users' dilemma for selection of data mining algorithm was based on Data Envelopment Analysis (DEA) [5], a method in operations research and economics for measuring productive efficiency of decision making units. This approach is later extended with user profiles that would give more importance to some parameters in order to express user preference [6]. There were also attempts to use multiple algorithms simultaneously [7], [8]. This should provide better stability of the ranking and alleviate single algorithm weaknesses. Our framework is to provide abstract component that will represent both simple and hybrid MCDM algorithms.

Approach that was used in ESPIRIT METAL project was based on similarity to known data set performance [9], [10]. Data set characteristics and performance data were gathered and algorithms ranked using MCDM methods. This data was later used to estimate performance of algorithms for unknown data set. Our framework allows researchers to maintain custom attributes and use the collected data to perform such tasks. Another approach considered multiple human experts from different domains and modeled their preference with fuzzy sets that mapped qualitative expressions to weights that are used in selected MCDM [11], [12]. This approach is supported in our framework by means of custom MCDM modules that can implement any simple or hybrid approach.

The task of algorithm selection lies at intersection of many disciplines so it is surprising how little intersection has been in the relevant developments in different communities [13]. With each community developing its' own vocabulary it was harder to search for relevant papers and build on existing work. Keogh and Kasetty chose 57 of the most relevant papers at the time and re-tested them against 50 diverse data sets [14]. They showed that most of the benchmarks in the field of data mining algorithms were performed on very limited data set without explicit note and this can easily lead to false conclusions about performance. In some cases small variations in implementation of known algorithm gave better performance improvements than the newly proposed algorithm. For this reason it is particularly important to precisely state the characteristics of data sets that are used in benchmark and perform unbiased optimization of the algorithm.

MCDM algorithms are one way to formally define decision making process and thus minimize bias towards certain solutions as well as provide more information for someone that is looking at our conclusions. With this formal definition of our decision process, interested reader can compare our preferences with his and have better idea of how relevant our conclusions are compared to his specific problem. However, collecting of the relevant data and ranking alternatives by one or more MCDM algorithms requires nontrivial effort. This is why we decided to create a framework that will make this process easier and will help in knowledge sharing by providing more details and common nomenclature.

Our framework does not aim to provide exhaustive set of MCDM algorithms, but rather enable flexible interface that will allow virtually any algorithm to be plugged in. In order to derive a common algorithm interface, we studied several of the most widely known MCDM algorithms. Algorithms that were studied are: Weighted Sum Model (WSM) and similar but less used Weighted Product Model (WPM), Preference Ranking Organization Method for Enrichment of Evaluations (PROMETHEE) [15], Višekriterijumska Optimizacija i Kompromisno Rešenje (VIKOR) [16]–[18], Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [19], [20], Logic Scoring of Preference (LSP) [21]–[24], Data Envelopment Analysis (DEA) [25]–[27], Adjusted Ratio of Ratios (ARR), and Analytic Hierarchy Process (AHP) [28].

[1] http://www.mcdmsociety.org/ available in December 2014.
[2] http://azure.microsoft.com/en-us/services/machine-learning/ available in December 2014.

III. REQUIREMENTS

This section presents main requirements that will drive the framework architecture and design. We first describe typical process for clustering consumers in power distribution system. We then abstract the steps where possible so that the requirements we identify may be attributed to general data mining task.

Typical process of clustering consumers in power distribution system consists of 5 major steps: input data cleansing, populating consumer model, dimensionality reduction, clustering and cluster evaluation. We outline these steps in Fig. 1. More details about each of the steps is given in the following paragraphs.

The first step is cleansing of the input data. Input data for consumer clustering is collection of measurements of active and reactive power at regular intervals (usually 15 minutes) for one or several years. It is possible to have missing data or peaks. Peaks are unusually large values that are caused by a network disturbance or a measuring equipment malfunction. Both missing values and peaks can have negative effect on clustering algorithm. There are several strategies to resolve such situations, such as interpolation, but they are out of scope of this paper.

Almost any data mining task based on real world data will require such step. Type of irregularities may differ but conceptually this step remains the same: processing step that requires certain amount of resources and changes the quality of the input data which can have non-trivial effect on further processing steps. Resources that it uses as well as data quality change may be measured and used to select the most appropriate algorithm for data cleansing for particular workflow.

The second step is populating the consumer model, which is explained in details in the remaining of the section. Two consumers behave similarly not only if they draw similar power from the network at one moment in time, but rather on the entire interval (e.g. one or multiple years). However, comparing all measurements of two consumers, for the given period of one or multiple years, would produce complex model with too many details that would make it hard to identify groups of similar consumers, due to phenomenon called *dimensionality curse* [31]. Vladimir Pestov in [29] discusses one aspect of this phenomenon: a point in high-dimensional space can have many "close" neighbors. This is much wider subject and there are many more relevant papers that deal with it, but it is out of scope of this paper.

In order to partially avoid the dimensionality curse, first level of abstraction is introduced, a consumer model. A consumer models consists of a set of curves where each curve represents consumer's consumption during 24 hours, for a particular season (e.g. Spring, Summer, Autumn and Winter) and a particular day type (e.g. working day, weekend or holiday). Each curve from the consumer model is calculated as an average from consumer's daily curves which correspond to the given season and the given day type. By using the consumer model to represent a consumer, we raise the level of abstraction and move away from the detailed measurements. Also, the aggregation of detailed measurements reduces the influence of missing or invalid data, that missed to be cleaned in the previous step.

In the previous step we reduced consumer representation to approximately few thousand dimensions

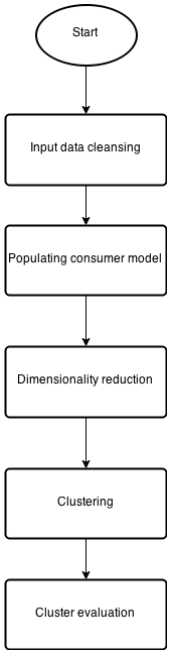

Figure 1 Typical process of clustering consumers in power distribution system

(192 measurements per day, 3-5 day types, 4 seasons) which is still an order of magnitude less than direct representation (192 measurements per day, 365 days per year). However, this is still enough to manifest the aforementioned dimensionality curse. This is why another step is taken to reduce a number of model dimensions.

The third step is dimensionality reduction, which applies one of many algorithms to transform input data into another problem space with fewer dimensions. One of such algorithms is Principal Component Analysis (PCA) [30]. This algorithm is used to represent cross-correlation matrix of the consumer type model by several orthogonal vectors - principal components. We can either choose fixed number of principal components or variable number of principal components depending on their cumulative influence (e.g. top n principal components that will amount for 90% of the variation in data set). The second and the third step perform major compression of source data by creating more and more abstract models. Just for quick comparison we can presume that we had measurements at least for one year at 15 minutes intervals which amounts to approximately 70.000 measurements per customer. After the third step we represent the same data by not more than 60 data points per customer. It is obvious that these models should be selected carefully in order to provide good input data for further steps. The same as for the first step we can measure the quality of derived model as well as resources that were used in processing. This will later influence our decision on what algorithm to use and how to choose parameters, if any, for the selected algorithm.

The fourth step is clustering of the data represented in model from previous step. Each consumer is one data point represented in n-dimensional data space where n is the number of attributes in input model (e.g. if we used PCA in previous step then n is the number of principal components). There are many clustering algorithms available nowadays, but all of them have one thing in

common. They must have a metric defined on a problem space. Examples of such metrics are Manhattan distance, Euclidean distance, Chebyshev distance etc. [31]. Each of these algorithms uses different amount of resources depending on both the quantity of input data as well as metric that is chosen. They also produce results of different quality.

The last, fifth step is determining quality of the results when there is no ideal clustering to compare to. There are several widely used internal and external evaluation measures such as Davies-Bouldin index [32] and Rand measure [33]. Usually more than one is used to get a better comparison between results for different algorithms.

At this point we have some measurements of how much resources we have used and what is the quality of the results (both intermediate and final). The next question is if it is the most appropriate solution for this environment. To address this issue we would have to try some other approach and compare the results. As Keogh and Kasetty showed in [14], experiment result is strongly tied to data set characteristics as well as algorithm implementation on the specific platform. Resources that were used, input data quality and result quality will be analyzed and compared to alternatives to achieve a solution closer to global optimum. This is where MCDM algorithms can help with formally defined criteria and procedure for selection of one out of possibly many non dominated solutions.

The framework is to provide the components that wrap the steps described above as well as the input and output data sets. These components enable access to data that is used in decision making step, such as elapsed time, memory used, data quality etc. They also provide signaling and control flow operations that will enable creating the experiment workflow such as the one described above. This includes, but is not limited to, start step, step completed, cancel step, step error, etc. Actual step implementation is not included in wrapper component. In other words, wrapper component should be able to wrap around existing implementation of certain step (e.g. data cleansing). This is also valid for data set wrapper components. This enables use of the framework with many different databases, such as Hadoop, Vertica, SQL Server, Oracle, and many different languages, such as Java, C#, R, etc.

The framework is to provide abstract model of the entire consumer clustering process in such way that will enable implementation of at least following MCDM algorithms: WSM, WPM, PROMETHEE, VIKOR, TOPSIS, LSP, DEA, ARR, ELECTRE, AHP. The framework should also allow implementation of a custom MCDM algorithm that can be based on the same model of the consumer clustering process.

It is not unusual that more than one research team will work on one task such as selecting consumer clustering algorithm. Even if they belong to single organization such as corporation or several different organizations such as universities, they will possibly work on different platforms, such as Windows or Linux. The framework is to provide interface for collaboration for teams that use different platforms. In order to avoid one centralized location that would require special maintenance and administration, the framework is to provide distributed operation. In other words, there is no one central server that all the clients will connect to but rather every client can connect to any other client and form a network for collaboration. The framework is to define means for discovery of the first peer and all other peers already connected to it.

Collaboration communication may contain sensitive research data. The framework is to enable encrypted communication over public channels to protects sensitive research data from eavesdropping or content change. This should be modular, allowing research team to use encryption scheme that they consider appropriate for required data confidentiality. This includes no encryption scheme for situations where no confidential data is exchanged over public networks.

The framework is to define modular and loosely coupled architecture that will allow any component to be customized in a plug-in manner. In other words, research team should be able to provide custom implementation of any module and be able to use it without the need to recompile the entire workbench.

IV. THE FRAMEWORK ARCHITECTURE

In this section we propose an architecture of the framework. Main components are outlined in Fig. 2 and their characteristics presented from two broad aspects with regard to requirements stated in the previous section.

The first aspect is a support for the individual work of a researcher. This includes process modeling, execution and analysis of the results. The second aspect is a support for collaboration of researchers. This includes verification of results from other researchers as well as using their results to improve decisions about current research. For example, one researcher performs experiments with one algorithm and another researcher performs experiments on some other algorithm but the same or very similar input data set. They can share and compare experiment results in order to determine which algorithm is more appropriate.

Three major components provide researcher with functionality for design, execution and analysis of experiments. Process designer component provides modeling of the process as well as individual process steps. Each process step can be either nested process or primitive component. Primitive components are those that cannot be further decomposed, such as data set or algorithm. Primitive components have basic attributes that are measured and recorded such as number of data rows, average computation time, or used memory. Process has aggregated attributes that are calculated by aggregating process steps attributes. This enables two researchers to work on different layers of abstraction but still be able to compare the results. Such approach will effectively create attribute hierarchy that will define aggregation rules. For example: if we consider process that executes three sequential steps then we can simply sum the individual step elapsed time to get aggregated elapsed time; however, if the process executes three parallel steps then aggregated elapsed time will be equal to the maximum elapsed time of the three steps.

Each process step defines the implementation plug-in that will connect the process model with the actual executing process during the execution phase. Basic operations include, but are not limited to the start step, stop step, pause step, report progress (on demand or via callback method), set parameter, get parameter and get attribute. Get attribute is the only mandatory operation.

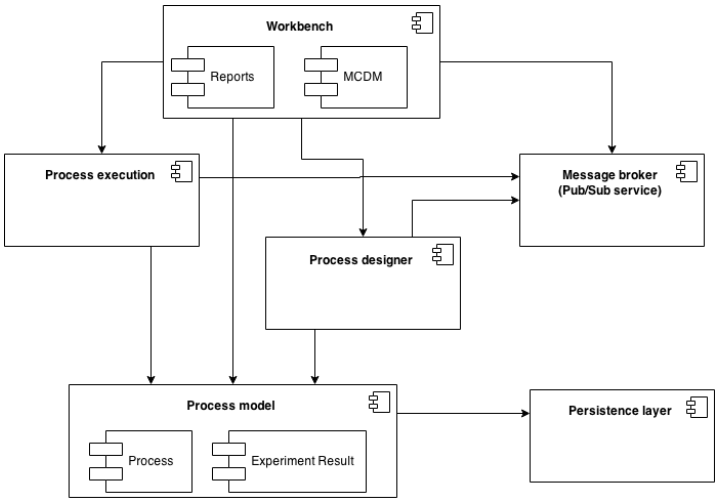

Figure 2 Proposed framework architecture

All others may or may not be supported by different implementations. For example: if we have offline trace processing plug-in it may only able to extract step attributes from execution trace but it will not be able to start or stop execution as it is already completed.

A result of the design phase is a data mining process model with all relevant attributes defined at each step. This model may be persisted and shared with other researchers. Model persistence is handled by a common component that provides basic operations such as store and retrieve, as well as model versioning. Model persistence includes execution results persistence. Each execution is uniquely identified and associated with process model and execution context. Execution context describes environment parameters such as platform, number of processors, available physical memory, as well as a user that executed the process.

The process execution component loads a process model from the persistent storage. Depending on the model it then offers supported operations to the user such as start processing or trace execution status. The common persistence component is used to record execution results as a part of the process model. Execution results contain values for step attributes defined in the process model. It is possible that not all attributes can be recorded for certain environments. In such case, researcher will be able to either create projection that does not contain those attributes with missing values or specify default values to use in the decision making.

Workbench is the third major component. It serves as a central point of individual work and binds all other components together. It is used to initiate design or execution sessions as well as provide modules for MCDM and reporting based on process model and experiment results. Workbench fills the process model based on messages it receives from other peers via message broker component that is described in more detail in the following paragraphs. The reports module provides many different views of the process model that help in analysis. The MCDM module provides plug-in port for implementation of various MCDM algorithms. A result of the MCDM algorithm is ranking or preference list of selected processes based on the experiment results. We rank processes characterized by algorithms used in each step so that we account for synergy between certain algorithms. The MCDM algorithms can also be used to rank algorithms used in a particular step of the process given that other steps are not changed in the selected experiments. These results are attached to process model together with context information and published to all subscribed peers.

The second aspect is a support for collaboration of researchers. The aforementioned workbench component uses the message broker component to provide pub/sub (publish/subscribe) functionality. Researchers create groups or teams on peer-to-peer principle. Each member of the group is able to invite new members. Each group member defines their level of interest that will dictate data that is exchanged. This prevents exchange of excessive data that could divert focus or increase pressure on communication channels.

Communication is based on the pub/sub principle. Each researcher subscribes to topic that are relevant for their research. Subscription requests are broadcasted to all peers. All peers also act as brokers, determining which publications are dispatched to which subscribers. Each peer dispatches only publications published on that peer. At any time subscriber can request special publication - integrity update. Integrity update publication transmits current state for the requested topic only to the requesting peer. It is used to initialize state of the subscriber after the subscription to certain topic as well as to reinitiate state after suspected communication failure. Integrity update request contains current state of the subscriber that the publisher can use to detect differences and send only data that is missing on the subscriber. Message broker component supports several protocols and communication channels in order to provide seamless collaboration even in the situations of complex network topology structures.

V. CONCLUSION

We started our research in the direction of finding the most appropriate consumer clustering algorithm. This search lead us to more elaborate problem of algorithm selection and MCDM problems. As Smith-Miles reported in [13] there was little intersection between relevant developments in different communities. This was mostly due to a different terminology in different problem domains and different communities.

We studied algorithm selection problems and MCDM use in data mining problems and identified the need for a

support in a form of a framework that will enable easier collection of characteristic performance data, as well as decision making by some of the widely used MCDM methods. Such framework is to support easier collaboration by common nomenclature and decision making process description. This allows researchers to formally state what are the important aspects of both data and algorithms in their problem domain and more efficiently communicate this information to fellow researchers. With our framework, we strive to give more confidence in published results and allow both result confirmation and further research based on those results.

We described our consumer clustering process and derived basic requirements for such framework that will help us in our further research. This requirements serve as the base on which main elements of the framework are defined. Focus of this paper is on the requirements. More detailed description of the framework is subject of the future work.

Our future work is directed to detailed design of the framework and reference implementation of the tool based on that framework. Further, we continue our research of the most appropriate algorithm for consumer clustering. It will serve as a proof of concept of our framework. The framework supports future development of many other data mining processes that will be based on large amount of data that is generated and collected by advanced distribution management systems, e.g. theft detection, outage prediction, predictive maintenance and many more.

ACKNOWLEDGMENT

The research presented in this paper was partially supported by Ministry of Education, Science and Technological Development of Republic of Serbia, Grant III-44010.

REFERENCES

[1] H. L. Ferreira, R. Garde, G. Fulli, W. Kling, and J. P. Lopes, "Characterisation of electrical energy storage technologies," *Energy*, vol. 53, pp. 288–298, May 2013.

[2] J. Rice, "The algorithm selection problem," *Adv. Comput.*, vol. 15, 1975.

[3] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Search," Santa Fe Institute, Feb. 1995.

[4] R. Dubes and A. K. Jain, "Clustering techniques: The user's dilemma," *Pattern Recognit.*, vol. 8, no. 4, pp. 247–260, Oct. 1976.

[5] G. Nakhaeizadeh and A. Schnabl, "Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms.," *KDD*, 1997.

[6] G. Nakhaeizadeh and A. Schnabl, "Towards the Personalization of Algorithms Evaluation in Data Mining.," *KDD*, pp. 289–293, 1998.

[7] Y. Peng, G. Kou, G. Wang, and Y. Shi, "FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms," *Omega*, vol. 39, no. 6, pp. 677–689, Dec. 2011.

[8] G. Kou, Y. Peng, and G. Wang, "Evaluation of clustering algorithms for financial risk analysis using MCDM methods," *Inf. Sci. (Ny).*, vol. 275, pp. 1–12, Aug. 2014.

[9] J. K. Helmut Berrer, Iain Paterson, "Evaluation of Machine-Learning Algorithm Ranking Advisors," 2000.

[10] P. B. Brazdil, C. Soares, and J. P. da Costa, "Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results," *Mach. Learn.*, vol. 50, no. 3, pp. 251–277, Mar. 2003.

[11] A. Sanayei, S. Farid Mousavi, and A. Yazdankhah, "Group decision making process for supplier selection with VIKOR under fuzzy environment," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 24–30, Jan. 2010.

[12] M. Noor-E-Alam, T. F. Lipi, M. Ahsan Akhtar Hasin, and A. M. M. S. Ullah, "Algorithms for fuzzy multi expert multi criteria decision making (ME-MCDM)," *Knowledge-Based Syst.*, vol. 24, no. 3, pp. 367–377, Apr. 2011.

[13] K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Comput. Surv.*, vol. 41, no. 1, pp. 1–25, Dec. 2008.

[14] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Min. Knowl. Discov.*, vol. 7, no. 4, pp. 349–371, Oct. 2003.

[15] J. P. Brans, P. Vincke, and B. Mareschal, "How to select and how to rank projects: The Promethee method," *Eur. J. Oper. Res.*, vol. 24, no. 2, pp. 228–238, Feb. 1986.

[16] S. Opricovic and G.-H. Tzeng, "Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS," *Eur. J. Oper. Res.*, vol. 156, no. 2, pp. 445–455, Jul. 2004.

[17] G.-H. Tzeng, C.-W. Lin, and S. Opricovic, "Multi-criteria analysis of alternative-fuel buses for public transportation," *Energy Policy*, vol. 33, no. 11, pp. 1373–1383, Jul. 2005.

[18] S. Opricovic and G.-H. Tzeng, "Multicriteria Planning of Post-Earthquake Sustainable Reconstruction," *Comput. Civ. Infrastruct. Eng.*, vol. 17, no. 3, pp. 211–220, May 2002.

[19] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, "A new approach for multiple objective decision making," *Comput. Oper. Res.*, vol. 20, no. 8, pp. 889–899, Oct. 1993.

[20] Y. Lai, T. Liu, and C. Hwang, "Topsis for MODM," *Eur. J. Oper. Res.*, vol. 76, pp. 486–500, 1994.

[21] J. J. Dujmović and H. Nagashima, "LSP method and its use for evaluation of Java IDEs," *Int. J. Approx. Reason.*, vol. 41, no. 1, pp. 3–22, Jan. 2006.

[22] J. J. Dujmović and H. Bai, "Evaluation and comparison of search engines using the LSP method," *Comput. Sci. Inf. Syst.*, vol. 3, no. 2, pp. 31–56, 2006.

[23] J. J. Dujmović and H. L. Larsen, "Generalized conjunction/disjunction," *Int. J. Approx. Reason.*, vol. 46, no. 3, pp. 423–446, Dec. 2007.

[24] J. J. Dujmović, G. De Tré, and N. Weghe, "LSP suitability maps," *Soft Comput.*, vol. 14, no. 5, pp. 421–434, Jun. 2009.

[25] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *Eur. J. Oper. Res.*, vol. 2, no. 6, pp. 429–444, Nov. 1978.

[26] P. Andersen and N. C. Petersen, "A procedure for ranking efficient units in data envelopment analysis," *Manage. Sci.*, vol. 39, no. 10, pp. 1261–1264, Oct. 1993.

[27] A. Charnes, W. W. Cooper, B. Golany, L. Seiford, and J. Stutz, "Foundations of data envelopment analysis for Pareto-Koopmans efficient empirical production functions," *J. Econom.*, vol. 30, no. 1–2, pp. 91–107, Oct. 1985.

[28] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *Eur. J. Oper. Res.*, vol. 48, no. 1, pp. 9–26, Sep. 1990.

[29] V. Pestov, "On the geometry of similarity search: Dimensionality curse and concentration of measure," *Inf. Process. Lett.*, vol. 73, no. 1–2, pp. 47–51, Jan. 2000.

[30] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987.

[31] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Addison Wesley, 2006, p. 769.

[32] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[33] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.