Complexity

Overcoming "Big Data" Barriers in Machine Learning Techniques for the Real-Life Applications

Lead Guest Editor: Ireneusz Czarnowski
Guest Editors: Tülay Yildirim, Piotr Jędrzejowicz, and Kuo-Ming Chao

Overcoming "Big Data" Barriers in Machine Learning Techniques for the Real-Life Applications

Complexity

Overcoming "Big Data" Barriers in Machine Learning Techniques for the Real-Life Applications

Lead Guest Editor: Ireneusz Czarnowski
Guest Editors: Tülay Yildirim, Piotr Jędrzejowicz, and Kuo-Ming Chao

Editorial Board

Contents

Hindawi
Complexity
Volume 2018, Article ID 1234390, 3 pages
https://doi.org/10.1155/2018/1234390

Editorial

Overcoming "Big Data" Barriers in Machine Learning Techniques for the Real-Life Applications

Ireneusz Czarnowski,[1] **Piotr Jedrzejowicz**,[1] **Kuo-Ming Chao**,[2] **and Tülay Yildirim**[3]

[1]*Department of Information Systems, Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland*
[2]*Coventry University, Priory Street, Coventry CV1 5FB, UK*
[3]*Department of Electronics and Communication Eng., Yildiz Technical University, Davutpasa Campus, 34220 Esenler/Istanbul, Turkey*

Correspondence should be addressed to Ireneusz Czarnowski; irek@am.gdynia.pl

Received 12 November 2018; Accepted 5 December 2018; Published 30 December 2018

1. Introduction

Data analysis, regardless of whether the data are expected to explain a quantitative (as in regression) or categorical (as in classification) models, often requires overcoming various barriers. They include unbalanced datasets, faulty measurement results, and incomplete data.

A special group of barriers is typical for what nowadays is referred to as "Big Data." The term is used to characterize problems where the available datasets are too large to easily deal with traditional machine learning tools and approaches. It is now generally accepted that dealing with huge and complex data sets poses many processing challenges and opens a range of research and technological problems and calls for new approaches.

Big Data is often characterized by the well-known 5V properties:

(i) Volume: typically a huge amount of data,

(ii) Velocity: a speed at which data are generated including their dynamics and evolution in time,

(iii) Variety: involving multiple, heterogeneous, and complex data representations,

(iv) Veracity: the uncertainty of data and lack of its quality assurance,

(v) Value: a potential business value that big data analysis could offer.

Big Data environment is often a distributed one with a distributed data sources. These sources can be heterogeneous, differing in various respects including storage technologies and representation methods.

Challenges of the Big Data not only involve a need to overcome the 5V properties but also include a need to develop techniques for data capturing, transforming, integrating, and modelling. Yet other important issues are concerned with privacy, security, governance, and ethical aspects of the Big Data analysis.

Current advances in dealing with the Big Data problems, albeit in many cases spectacular, are far from being satisfactory for the real-life applications. This becomes especially true in numerous domains where machine learning tasks are crucial to obtaining knowledge of different processes and properties in areas such as bioinformatics, text mining, or security. Unfortunately, the majority of the current algorithms become ineffective when the problem becomes very large since underlying combinatorial optimization problems are, as a rule, computationally difficult. There exists a variety of methods and tools which are excellent at solving small and medium size machine learning tasks but become unsatisfactory when dealing with the large ones.

Current hot topics in the quest to improve the effectiveness of the machine learning techniques include a search for compact knowledge representation methods and better tools for knowledge discovery and integration. Machine learning may also profit from integrating collective intelligence techniques, applying evolutionary and bioinspired

techniques, and exploring further deep and extreme learning techniques.

2. Contributions Included in the Special Issue

The purpose of this special issue is to publish some of the current research results advancing different techniques for dealing with large and complex problems. The issue consists of fourteen papers covering some novel methods and techniques as well as their applications. The selected of them are extended versions of papers presented at the IEEE INISTA Conference in 2017 year.

The paper of I. Czarnowski and P. Jędrzejowicz proposes an approach to data reduction for learning from Big Data sets by integrating stacking, rotation, and agent population learning techniques. The paper shows that combining the proposed techniques can improve the performance of the classifier learning from large and complex datasets. The approach is based on the classifier ensemble paradigm where stacking ensembles have been produced using the rotation-based techniques, guaranteeing their heterogeneity. To reduce the dimensionality of the data, data reduction in an instance and feature dimensions has been applied.

Dimensionality reduction has been also discussed in the paper of J. Lee and D.-W. Kim. The authors consider a multilabel classification problem. Multilabel classification is a variant of multiclass classification, where multiple labels may be assigned to each instance; i.e., each instance corresponds to multiple class labels. The approach is based on dimensionality reduction by feature selection. The approach is based on analysis of the information content and remedies the computational burden by discarding the labels that are unimportant to feature importance scores.

Semisupervised learning is a class of the machine learning tasks where both labeled and unlabeled data are used to induce a learning model. The paper of E. Protopapadakis et al. deals with the problem of instance selection and training set preprocessing. Several approaches to instance selection based on sampling are discussed and compared. An extensive experimental evaluation of the considered approaches is included in the paper.

In their paper, J. Jedrzejowicz and P. Jedrzejowicz consider an approach to the data stream mining. The problem has been solved using the incremental Gene Expression Programming classifier with metagenes and data reduction. As it has been shown, the proposed concept of metagenes assured increasing the classification accuracy while data reduction allowed controlling computation time. The advantage of the proposed approach is also allowing work with the data stream that results from implementation of simple drift detection mechanisms. The proposed approach offers also scalability through the possibility to adjust computation times to the user needs at the expense of the classification accuracy.

Z. H. Kilimci and S. Akyokus focus on the text classification problem. The authors propose the ensemble learning and deep learning approaches to enhance the text classification performance. The ensemble of base classifiers proposed for solving the considered classification problem includes traditional machine learning algorithms such as naïve Bayes, support vector machine and random forest, and a deep learning based Conventional Network classifier. The different document representations and different ensemble approaches on eight different datasets have been evaluated. Finally, it has been shown that using heterogeneous ensembles together with deep learning methods and word embedding enhances text classification performance.

A. Nowak-Brzezińska deals with the problem of knowledge management and proposes a new approach for the rule management mechanisms in the decision support systems. The approach is based on hierarchically organized rule-base structure. Such a structure is produced based on the clustering approach. Making use of the similarity approach the proposed algorithm tries to discover new facts (new knowledge) from rules and facts already known. The computational experiment involves an analysis of the impact of the proposed methods on the efficiency of a decision support system with hierarchical knowledge representation.

B. Huynh and B. Vo focus on the problem of mining erasable itemsets. Mining erasable itemsets is a class a frequent pattern mining problem. In general, the problem of mining erasable itemsets belongs to the NP-hard class and the existing algorithms for mining erasable itemsets have high computational complexity. Computational experiments results show that the proposed approach ensures quite reasonable and competitive results as compared with earlier approaches.

Y. Guo et al. deal with the problem of complex power system status monitoring and evaluation. In the paper, a special Big Data platform, used as an analytical tool, is presented as discussed. Based on the case study, authors show how to improve the decision-making process in power systems.

The paper written by P. Caşcaval focuses on the problem of modeling and evaluating the reliability of the complex networks. In general, the problem of computing complex network reliability belongs to NP-hard class problems. The paper contributes by proposing a novel approach to network reliability evaluation. The proposed method reduces the computation time for large networks to a great extent, compared with an exact method as well with other known methods.

In his paper, J. P. Paplinski investigates the stability of the Bat Algorithm. The analysis is based on the assumption that the considered algorithm can be treated as a stochastic discrete-time system which allows using the Lyapunov stability theory for analyzing the behavior of the algorithm. The computational experiment proves the correctness of the approach.

The paper of C. A. Martin et al. is dedicated to the problem of classifying comments that tourists publish online. The paper discusses the case study, where Convolutional Neural Networks and Long Short-term Memory Networks are used in the process of decision making with respect to the quality of the service improvements.

A. Wosiak and D. Zakrzewska discuss the real problem of detection and diagnosis of heart diseases. One approach for preparing a model of heart diseases for medical diagnostic is based on clustering. The authors propose a new approach based on combining unsupervised feature selection and

clustering. The proposed approach has been validated using the real-life datasets of cardiovascular cases. The experiment results show the advantage of the approach as compared to other approaches based on feature selection but without the clustering for supporting the statistical inference.

In the paper written by J. Jakubik and H. Kwaśnicka, the music data are analyzed using machine learning methods. SVM is used as the classification tool, but before the adequate data representation has to be prepared using the Recurrent Neural Network. The computational experiment results show that the proposed hybrid machine learning tool is competitive as compared with other approaches.

The paper written by D. Świetlik deals with simulation of natural brain processes, including three typical processes, i.e., learning, memory, and forgetting. The processes are simulated based on the model of the CA1 region of the hippocampus. A possibility of the hardware implementation of the pyramidal cells of the CA1 region of brain hippocampus is also discussed. The problem considered is an example of the problem where different signals influence the brain processes. Their analysis can be useful from the point of view of the medical diagnostics as well from the point of view of extracting knowledge important for preparing and improving artificial models and algorithms applied for brain data analyses.

3. Conclusions

The editors believe that the special issue has been an important and timely initiative. The editors hope that the presented research results will be of value to the scientific community working in the field of Big Data, data science, machine learning, analysis of complex data, data mining, knowledge discovery, and project management. Presented results are also addressed for other researchers who are currently or will be in the future implementing different data analysis tools trying to solve the real-life problems.

We would like to take this opportunity to thank all the authors for their valuable contributions. The submitted papers have been reviewed by at least two referees. We wish to thank all peer reviewers whose invaluable work, suggestions, and detailed feedback have helped to improve the quality of the papers included in the special issue. Special thanks are due to Sergio Gómez and Vincent Labatut, who supported the editors in their work.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this Special Issue.

Ireneusz Czarnowski
Piotr Jedrzejowicz
Kuo-Ming Chao
Tülay Yildirim

Hindawi
Complexity
Volume 2018, Article ID 1297150, 13 pages
https://doi.org/10.1155/2018/1297150

Research Article

Simulations of Learning, Memory, and Forgetting Processes with Model of CA1 Region of the Hippocampus

Dariusz Świetlik

Intrafaculty College of Medical Informatics and Biostatistics, Medical University of Gdańsk, 1 Debinki St., 80-211 Gdańsk, Poland

Correspondence should be addressed to Dariusz Świetlik; dswietlik@gumed.edu.pl

Received 4 April 2018; Revised 30 July 2018; Accepted 28 October 2018; Published 10 December 2018

Guest Editor: Ireneusz Czarnowski

The aim of this paper is to present a computational model of the CA1 region of the hippocampus, whose properties include (a) attenuation of receptors for external stimuli, (b) delay and decay of postsynaptic potentials, (c) modification of internal weights due to propagation of postsynaptic potentials through the dendrite, and (d) modification of weights for the analog memory of each input due to a pattern of long-term synaptic potentiation (LTP) with regard to its decay. The computer simulations showed that CA1 model performs efficient LTP induction and high rate of sub-millisecond coincidence detection. We also discuss a possibility of hardware implementation of pyramidal cells of CA1 region of the hippocampus.

1. Introduction

Hippocampus is a neural structure located in brain in medial temporal lobe, under the cerebral cortex. Being a part of limbic system, hippocampus plays main role in cortical memory [1–4] navigation [5–10] and conditioning [11–13]. In basic hippocampal circuit, a series of narrow zones could be distinguished, part of which are *Cornu Ammonis* (CA) areas filled with pyramidal cells. CA1 and CA3 are proven to be areas of highest significance [14–17].

Most of reviewed models are associated with memory and imply that hippocampus is working as a homogenous network [18]. These models do not assume any differentiation among CA1 and CA3. Among numerous hippocampal models only a few specify a role for CA1; however there are many examples of synaptic integration among pyramidal cells in CA1 area presenting no connection with their basic function.

Scientific research was primarily concentrated on the potential of CA3 areas, mostly the ability of cells to autoassociate [19, 20] or to associate activity in sequences [21, 22]. Treves and Rolls [19] suggested that CA3 requires recording into a stronger code and takes benefits from this process. Otherwise, McClelland and Goddard [20] presented slightly different point of view, in which CA3 are too strong for direct association. As a result, the invertible code might cause confusion among superficial and deep layers of entorhinal cortex.

Following Marr's inspiration [23], Treves and Rolls improved a precise and successful hippocampus memory model, in which CA3 is area associated with recurrent collaterals and memory recall [19, 24–27]. In this model possible functions of CA1 are also mentioned. It is suggested that CA1 are responsible for insurance of effective information transmission and reduction of CA3 excessive activity [28–31].

However, O'Reilly and McClelland [32] presented a slightly different expertise in which CA1 areas are required to solve the problem of associating CA3 activity with the primal entorhinal activity. McClelland and Goddard [20] developed a model, in which CA1 cells contact EC cells and have direct connections to them. Another point of view suggesting connection between CA3 cells and dentate gyrus is given by Lisman [33]. Nevertheless, Lisman and Otmakhova [22] declared that storage of new information in hippocampus requires activation of dopamine receptor which enables temporoammonic input activity. Dopamine has ability to inhibit reaction caused by temporoammonic stimulation and simplify the induction of early LTP in the Schaffer collateral [34] without interfering in their response [35].

Implementation of CA1 presented in model given by Hasselmo and Schnell [36] imputes a crucial role to acetylcholine,

which is presented as the main agent performing suppression [37]. Otherwise, Hasselmo et al. submit implication for CA1 in which the theta rhythm pertains separate phases of storage and recall in CA1 and CA3 [38].

Another model, presented by Levy, concentrates on more general or predicted function of CA1 and might not be compatible with proven activity of hippocampus [39]. A temporoammonic input activity is suggested to take place in CA1 and is associated with activity in CA3 through the Schaffer collaterals. Furthermore, it could be possible that that temporoammonic input blocks Schaffer collateral vividness in order to determine which active CA1 cells can be connected with active CA3 cells. In this model CA1 are viewed as a decoder of CA3 activity, like subiculum and entorhinal cortex, while CA3 recurrent collaterals simplify the preservation of sequences [40]. The prediction of existing dependent on time plasticity in the Schaffer collaterals is examined and supported by Nishiyama et al. [41].

On the contrary, Lorincz and Buzsaki [18] suggest that the mismatch between the current input and events recalled by hippocampus is calculated in the entorhinal cortex. The contribution to CA1 is observed during activity by using delta rule [42]. Those hypothetical learning rules based on mathematics are given by L'orincz [43]; however in the previous version of this theory there is no precise input to CA1 [44].

The hippocampus is an area in human brain, which becomes activated in order to process temporal orders of events. CA1 involve this region in the memory of objects, odors, and, what is more important, their sequences [45]. Another promising conclusion might be a relation of temporal delays in the neural circuitry of the medial entorhinal cortex to temporal adjustment process, which could result in the various volume of spatial grids found in the medial entorhinal cortex [46]. Various types of neuron cells' firing rates are high at different times, within a trial or delay period [47–49].

The role of hippocampus in contextual learning of objects recognition must be also mentioned [50]. And even simply models of hippocampal circuits could prove new explanations in human pathology such as Alzheimer disease or drugs art of work. It is well established that the connections from entorhinal cortex layer 2 to hippocampus play a crucial role in development of Alzheimer pathology [51]. Cannabinoids disrupt memory encoding by functionally isolating hippocampal CA1 from CA3 [52].

In Section 2, we introduce mathematical model of CA1 region of the hippocampus microcircuit and discuss the methods simulation CA1 (Sections 2.1 and 2.2). Additionally, in Sections 2.3–2.5, we describe mathematical presentations: pyramidal, basket, and O-LM cells of CA1 region, while, in Sections 2.6–2.7, we present CA1 network inputs and synaptic properties glutamate and GABA receptors. In Section 3, we present the results of the paper. In Section 4, we fully discuss our results. Section 5 summarizes the conclusions.

2. Materials and Methods

2.1. The Model Description. The CA1 microcircuit is presented of Figure 1. Our simulations of the hippocampus are based on computational models from previous studies [53, 54]. There are four pyramidal cells (P1, P2, P3, and P4), two basket (B1, B2), one O-LM cell (inhibitory interneurons), and 3 independently programmed theta rhythm generators. Such sparse network with strictly topographically related connections is very similar to the CA1 net used by Hasselmo and Cutsuridis [55]. And in our opinion it has more biological plausibility as compared to previous Cutsuridis network with 100 pyramidal cells and nearly to all interconnections [56].

We have used the theta oscillation in our previous studies [53, 54], which were based on faster gamma-frequency oscillations [1, 57–60], spatial information [61–63], in-time locking cell activities [64], and regulation of learning facilities [1, 65–67].

The MS-DBB (Medial Septum-Diagonal Band of Broca) has been classically viewed as the hippocampal theta rhythm generator [57, 64, 68]. However, the role of the MS-DBB in hippocampal theta oscillations must be revised in light of recent discovery that the hippocampus itself can generate a theta-frequency rhythm independent of the MS-DBB [69]. Huh et al. suggest that the MS-DBB is one of several extrinsic rhythm generators that amplify and regulate intrinsic theta generators within the hippocampus [70]. Hence, the hippocampal theta rhythm recorded in vivo may be a product of several interacting intrinsic and extrinsic theta generators working in concert. It remains to be elucidated what role glutamatergic, GABAergic, and most importantly cholinergic MS-DBB neurons [71] play in these interactions; the understanding of these matters will bring new insights into the mechanisms underlying functions such as spatial learning and memory. In our model we have employed the most simple theta generators, which depict the basic Wang [68] suggestions, but we have not considered the proposals of Hajos [72], as the reciprocal Septo-Hippocampal connections are much more complicated as in the skeleton network of Hajos [72]. The T1, T2, and T3 theta generators send the series of 8 bursts every second; it means 8 Hz theta frequency. Each burst consists of 5 spikes at 100 Hz; for T2 and T3 we have a faze delay of 10 and 20 milliseconds for burst activity.

2.2. Art of Work. The mathematical description of equations and parameters that we used in our simulations was based on our previous studies on single neuron model [53] and sparse CA3 network model [54]. All mathematical descriptions of CA1 model neurons are presented in Table 1.

2.3. CA1 Pyramidal Cells. Every CA1 pyramidal cell consists of 16 compartments in which each dendrite has an excitatory or inhibitory synapse. There are glutamate receptors for excitatory inputs: AMPA - E (k, i), NMDA - M (k, i). GABA receptors are for inhibitory inputs: I (k, i) while k is the number of dendrite compartments and i is the number of area register table. There are dendrites constructed within a course of compartments. Each CA1 cell receives somatic synaptic inhibition from CA1 basket cells, proximal excitation from CA1 pyramidal cells, mid-dendritic excitation from Mossy Fibers, and distal apical excitation from the layer 3 entorhinal cortex [73, 74].

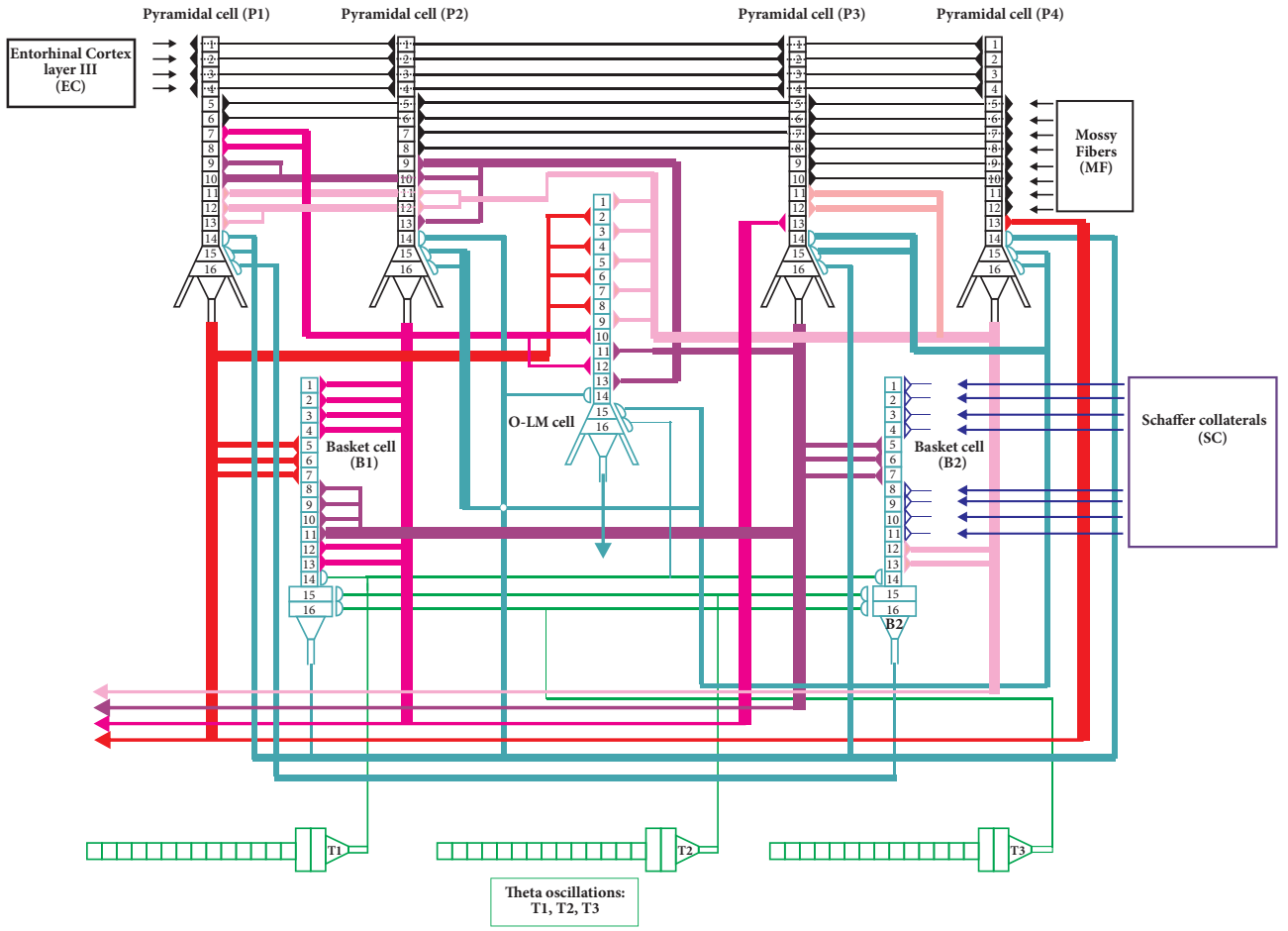


Figure 1: Hippocampal formation CA1 microcircuit showing pyramidal, basket, and OL-M cells. CA1 network: P1, P2, P3, P4: pyramidal cells; B1, B2: basket cells; and OL-M cell. Each CA1 pyramidal cell received distal apical excitation from the layer 3 entorhinal cortex (EC) and mid-dendritic excitation from Mossy Fibers (MF).

2.4. CA1 Basket Cells. Every CA1 basket cell consists of 16 compartments in which each dendrite has an excitatory or inhibitory synapse. There are glutamate receptors for excitatory inputs: AMPA - E (k, i), NMDA - M (k, i). GABA receptors are for inhibitory inputs: I (k, i) while k is the number of dendrite compartments and i is the number of area register table. There are dendrites constructed within a course of compartments. Each basket cell receives somatic synaptic inhibition from the medial septum (theta oscillations) and neighboring basket cells in their soma. Excitatory connections are received to their distal dendrites form layer 3 entorhinal cortex and to medium dendrites from both CA1 pyramidal cells and CA3 Schaffer collaterals.

2.5. CA1 O-LM Cells. Every CA1 O-LM cell consists of 16 compartments in which each dendrite has an excitatory or inhibitory synapse. There are glutamate receptors for excitatory inputs: AMPA - E (k, i), NMDA - M (k, i). GABA receptors are for inhibitory inputs: I (k, i) while k is the number of dendrite compartments and i is the number of area register table. There are dendrites constructed within a course of compartments. Each O-LM cell receives excitatory and inhibitory connections. First ones were received from active CA1 cells, whereas second ones were received from the medial septum (theta oscillations: T1, T2, and T3).

2.6. Model Inputs. According to Witter sources of inputs to CA1 are Mossy Fibers and entorhinal cortex layers III [75] as well as disinhibitory theta input from medial septal area. Every CA1 pyramidal cell input from Mossy Fibers was presented as the firing at an average frequency of 44 Hz and from layer 3 entorhinal cortex of 24 Hz. Each CA1 basket cell input from CA3 Schaffer collaterals was modeled as firing at an average frequency of 50 Hz and from the medial septum at 8 Hz theta rhythm. All initial parameters of microcircuit model of CA1 network are presented in Table 2. Pyramidal cells received somatic synaptic inhibition from CA1 basket cells (B1, B2), proximal excitation from CA1 pyramidal cells (P1, P2, P3, and P4), mid-dendritic excitation from Mossy Fibers (MF), and distal apical excitation from the layer III entorhinal cortex (EC). Basket cell received somatic synaptic inhibition from the medial septum (theta oscillations: T1, T2, and T3). Excitatory connections are received from both CA1 pyramidal cells (P1, P2, P3, and P4) and CA3 Schaffer

Table 1: The most important mathematical issues of the model cells of CA1 region.

Name of functions	Functions
Synaptic	$SF(\mathrm{t}) = \begin{cases} 0 & \mathrm{t} = \mathrm{t}_{sd} \\ \frac{A_{MAX}}{\mathrm{t}_r}(\mathrm{t} - \mathrm{t}_{sd}) & \mathrm{t}_{sd} < \mathrm{t} \le \mathrm{t}_r \\ \frac{A_{MAX}}{\mathrm{t}_d}[(\mathrm{t}_d - (\mathrm{t} - (\mathrm{t}_r + \mathrm{t}_{sd})))] & \mathrm{t}_r < \mathrm{t} \le \mathrm{t}_d \end{cases}$
Summarized potential	$S(\mathrm{k}, \mathrm{i}+1) = \mathrm{Re}P + \sum_{m=1}^{NE} (E(\mathrm{m}, \mathrm{i}) - \mathrm{Re}P) \inf(\mathrm{m}, \mathrm{k})$
Memory	$MEM(\mathrm{k}, \mathrm{i}) = 1 + \ln \frac{(C(\mathrm{k}, \mathrm{i}) + 1)}{6\mathrm{clog}}$
Power function	$power = powerA(M(k;i) - ReP)$
Time of memory duration	$C(k, i+1) = C(k, i) + e^{power} - 1$
Summarized postsynaptic potential in neuron	$PSP(i) = ReP + \sum_{\mathrm{m}=1}^{NE} W(k)(E(m, i) - ReP) + \sum_{\mathrm{m}=1}^{NI} (I(m, i) - ReP)$
Threshold function for action potential	$out = \begin{cases} 0 & PSP < threshold \\ 1 & PSP \ge threshold \end{cases}$
Shifting of registers	$\forall_{\mathrm{k}\in N}\forall_{\mathrm{i}\in\langle 0,31\rangle} E(\mathrm{k}, \mathrm{i}, \mathrm{t}+1) = \begin{cases} E(\mathrm{k}, \mathrm{i}+1, \mathrm{t}) & PSP < threshold \\ ADRV(k) & PSP \ge threshold \end{cases}$
	$\forall_{\mathrm{k}\in N}\forall_{\mathrm{i}\in\langle 0,31\rangle} M(\mathrm{k}, \mathrm{i}, \mathrm{t}+1) = \begin{cases} M(\mathrm{k}, \mathrm{i}+1, \mathrm{t}) & PSP < threshold \\ M(k, i+1, t) & PSP \ge threshold \end{cases}$
	$\forall_{\mathrm{k}\in N}\forall_{\mathrm{i}\in\langle 0,31\rangle} I(\mathrm{k}, \mathrm{i}, \mathrm{t}+1) = \begin{cases} I(\mathrm{k}, \mathrm{i}+1, \mathrm{t}) & PSP < threshold \\ ReP & PSP \ge threshold \end{cases}$

Table 2: Initial parameters microcircuit model of CA1 network.

CA1 cells	LTP (Memory)	NE	NI	Threshold	ReP	LSW	CaMT	EPSPd	IPSPd	FQ
Pyramidal cell (P1)	on	13	3	-50	-80	0,2	-68	4,5	-6	10
Pyramidal cell (P2)	on	13	3	-50	-80	0,2	-68	4,5	-6	10
Pyramidal cell (P3)	on	13	3	-50	-80	0,2	-68	4,5	-6	10
Pyramidal cell (P4)	on	13	3	-50	-80	0,2	-68	4,5	-6	10
Basket cell (B1)	off	13	3	-50	-80	1	-68	4	-4,5	10
Basket cell (B2)	off	13	3	-50	-80	1	-68	4	-5,5	10
O-LM cell	off	13	3	-50	-80	0,6	-68	4	-4	10

collaterals (SC). O-LM cells receive excitatory and inhibitory connections. First ones are received from active CA1 cells (P1, P2, P3, and P4), whereas second ones are received from the basket cells (B1, B2). Hippocampal formation of CA1 microcircuit connections is presented in Table 3. As compared with the previous model of CA3 microcircuit, we have inputs from layer III of entorhinal cortex instead of those from layer II (∗). The Mossy Fibers (∗∗) subsequently diminish in range; instead we have intermingled input from pyramidal cells with each other through the Schafer collaterals (Table 3).

2.7. Synaptic Properties. The mathematical description of the glutamate receptors AMPA, NMDA, and the GABA receptor was based on our previous studies [53, 54]. The real value of postsynaptic potential is estimated by using functions from Table 1. During these studies all pyramidal cells had the same LSW parameter (0,2). However during our previous studies on CA3 microcircuit, the pyramidal cells used LSW ranging from 0,6 to 0,7 [53, 54].

3. Results

Two simulations were used, one without LTP induction and one with LTP induction. For LTP induction pyramidal cells (P1, P2, P3, and P4) were strongly excited on inputs 7, 8, and 9 by stimulation at 100 Hz for 400 ms. Such approach was inspired by the well-known phenomenon where during the

Table 3: Connections of hippocampal CA1 microcircuit.

CA1 cells	Inputs															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pyramidal cell (P1)	EC_3*	EC_3*	EC_3*	EC_3*	MF**	MF**	P2	P2	P3	P3	P4	P4	P4	B1	B1	B2
Pyramidal cell (P2)	EC_3*	EC_3*	EC_3*	EC_3*	MF**	MF**	MF**	MF**	P3	P3	P4	P4	P3	B1	B2	B2
Pyramidal cell (P3)	EC_3*	EC_3*	EC_3*	EC_3*	MF**	MF**	MF**	MF**	MF**	MF**	P4	P4	P2	B2	B2	B1
Pyramidal cell (P4)	EC_3*	EC_3*	EC_3*	EC_3*	MF**	MF**	MF**	MF**	MF**	MF**	MF**	MF**	P1	B1	B2	B2
Basket cell (B1)	P2	P2	P2	P2	P1	P1	P1	P3	P3	P3	P3	P2	P2	T1	T2	T3
Basket cell (B2)	SC	SC	SC	SC	P3	P3	P3	SC	SC	SC	SC	P4	P4	T1	T2	T3
O-LM cell	P4	P1	P4	P1	P4	P1	P4	P1	P4	P2	P3	P2	P3	B1	B2	B2

environmental activity the firing rate at particular hippocampal connections increases rapidly [76]. Firing histograms of the 3 cells' groups and theta oscillation including stimulation with and without LTP are given in Figures 2 and 3. Those stimulations refer to Bliss and Lomo research work [77, 78].

The stimulated pyramidal cells have a phase preference compatible with the theta rhythm. O-LM cell are able to intrinsically oscillate at the theta rhythm and become extremely active during theta oscillations. However, there is no visible contribution of O-LM cells to the synchronization of pyramidal cells in the CA1 network. Otherwise, in response to septum inhibitory input, basket cells oscillate.

In both simulations, after the training course, there is a significant increase in the values of LTP and frequency of action-potential generation. The CA1 model represents also clearly the heterosynaptic LTP as emerged from online observations of all numerical values during simulations.

Configuration of CA1 network without LTP inducing protocol was also examined and no output increased frequency was observed. What is more, after the 3rd second of simulation an LTP induction was noticeable because of a narrow coincidence of neighboring excitatory inputs after 2 seconds of simulation. The examination carried out without LTP inducing protocol presents the stabilization of pyramidal cells output frequency at the value of 34.5Hz (SD 0.827). This value stabilizes after 10 seconds of simulation and is similar to average frequency input (35.22) (Figure 4). There might be a conclusion that LTP induction simplifies the additional firing correspondence with less important inputs frequencies, which is clearly explained and visualized on the online simulation (see Supplementary Materials available here).

Process of creating synaptic plasticity is very complex; any change might be triggered in a single and short millisecond action; however the consecutive long-term results last for a long time, even days and years. STDP (the spike time dependent plasticity) is an algorithm for synaptic changes, which could be perceived as an evolution of the old Hebb principle demanding precise timing of pre- and postsynaptic spikes. According to STDP neurons need to wait for the next postsynaptic spike to decide if they prefer to turn on LTP or LTD. Recently, accuracy of such algorithm was being doubted.

In experiment carried out without stimulation the received spikes value after 10 seconds for pyramidal cells was from 229 to 321 (mean 258, SD 43.2); however in research involving LTP inducing protocol spikes value was from 262 to 348. Statistically relevant increase in spikes value in experiment with LTP inducing protocol was not observed ($p=0.3123$) (Figures 2 and 3).

The solution could be the LTP related algorithm. This algorithm functions on a dendrite level, independently of each compartment in compatibility with canonical form of sc. The induced LTP protocol has a specified time of duration.

CA1 pyramidal cells LTP network (called Memory) is dependent on duration of simulation in both cases, connected with LTP induction and without it. In those simulations positive and statistically relevant correlation between time of simulation and average value Memory f of 4 pyramidal cells were received (Figure 5). Score without stimulation was $R=0.97$ ($p=0.0001$) and with LTP stimulation $R=0.89$ ($p=0.0001$). Furthermore, in both cases statistically valid increase of pyramidal cells memory and spikes value was observed ($R=0.98$, $p=0.0001$).

After arrival of action potential at a synapse the subsequent change of synaptic potential remains disposable for further computation. For EPSP this time of duration I pyramidal cells is 15 ms. The amount of interspike intervals combination relies on the number of received action potentials arriving at one synapse at the same time. Making an assumption that the difference of one clock step at one interval is essential for further computing, it could be estimated that there are 6272 combinations from one spike to dense impulse of 8 spikes. If we assume that 1 ms is a significant distinction then we achieve 623 combinations. A total amount of variation for 16 inputs (synapses of pyramidal cells) can be calculated as 627216.

Coworkers of Kasabov presented SPAN which allows recognizing over 200 synapses' spike patterns during 200 ms stimulation [79]. SPAN is a spiking neuron capable of learning connections of arbitrary spike trains in a controlled fashion which enables process of spatiotemporal information encoding in the accurate timing of spikes.

4. Discussion

In biological neurons the precision might be less advanced due to the fact that diverse membranes current need time to raise potential. However, the mathematical models of

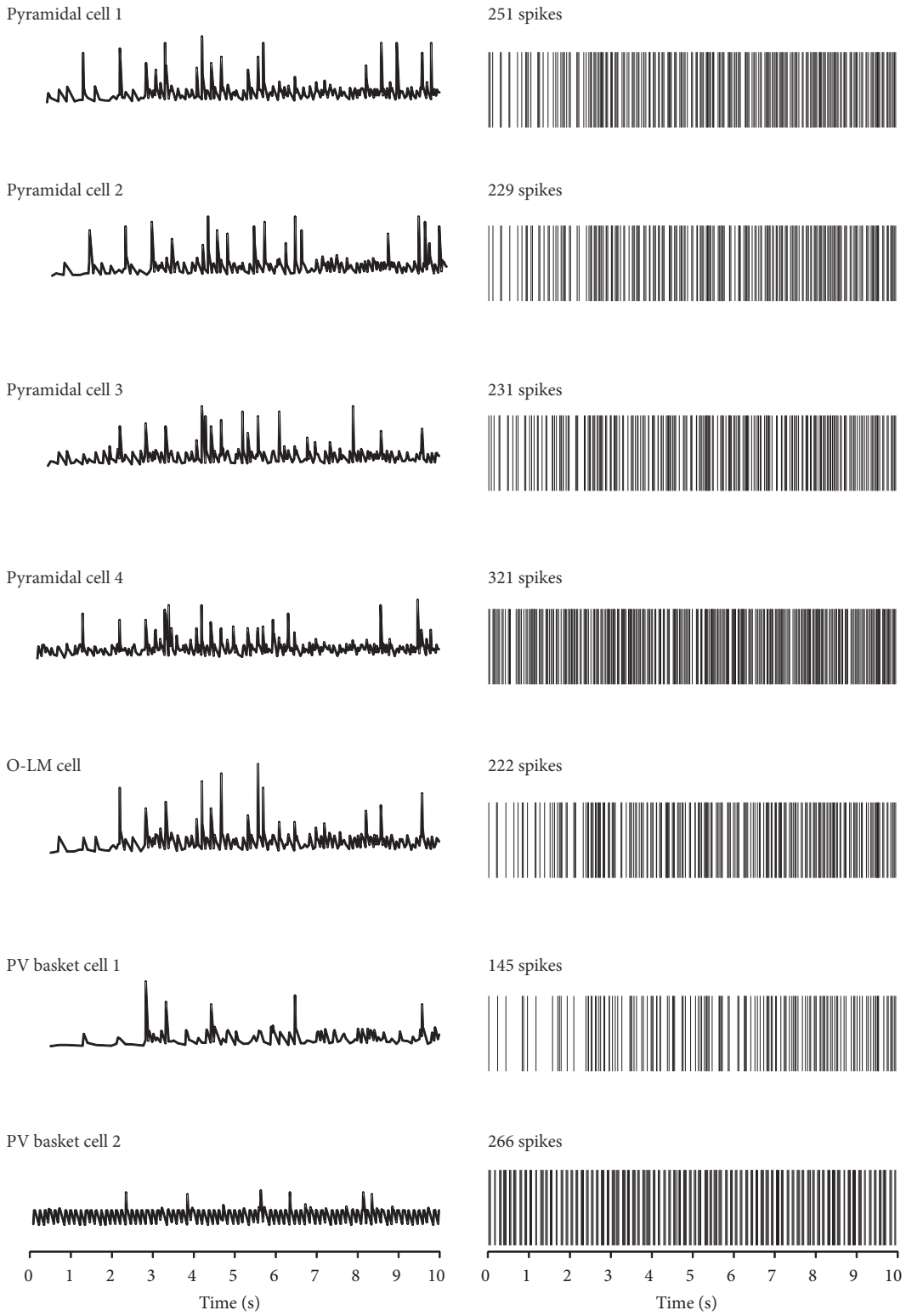


Figure 2: Basic 10-second real-time simulation of pyramidal cells, basket cells, and OL-M cell without LTP inducing protocol. On the left, time course of summarized postsynaptic potential for all cells of CA1 region. On the right, output spikes train. Time (s), forecast time of duration LTP at the end of simulation.

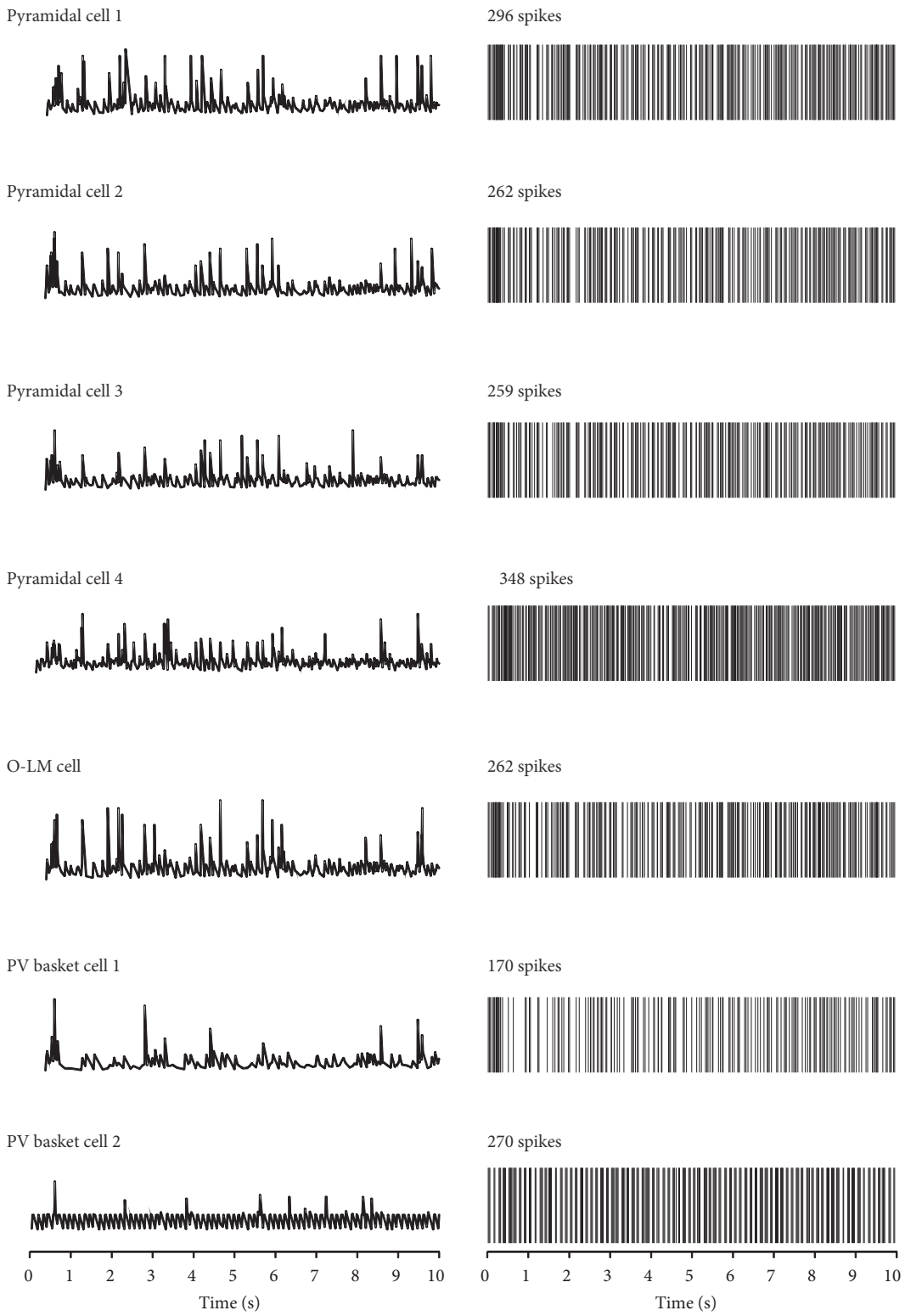


Figure 3: Basic 10-second real-time simulation of pyramidal cells, basket cells, and OL-M cell with LTP inducing protocol. On the left, time course of summarized postsynaptic potential for all cells of CA1 region. On the right, output spikes train. Time (s), forecast time of duration LTP at the end of simulation.

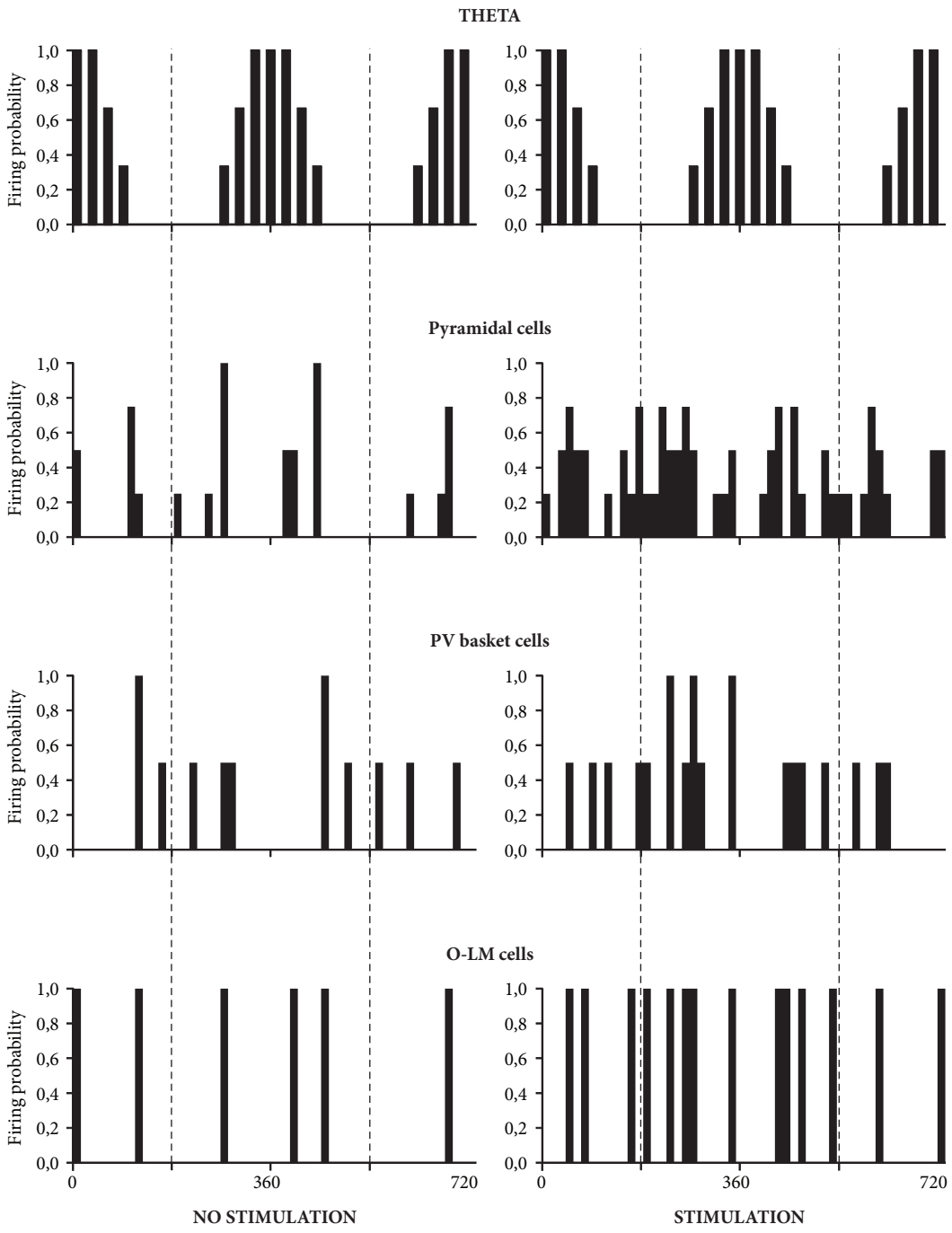


Figure 4: Temporal relationship between theta oscillations, pyramidal cells, basket cells, and OL-M cell. On the left, temporal relationship without LTP inducing protocol. On the right, temporal relationship with LTP inducing protocol. Histograms show the firing probability rates of all cells CA1 region.

dendrites present the capacity to perform sub-millisecond compatibility detection.

The presence of chaos process inside brain network is acknowledged and there is no possibility for precise prediction of multiple inputs data and finding any analytical solution seems to be rather impossible. The Izhikevivh model, similarly to many other models, consists of accessible differential equations with only a few parameters which appears to be easy to define [80]. Every constructed model of biological feasibility needs to be introduced with an efficient algorithm. Although there is a precise set of internal parameters for mature living neuron cells, some differences might appear in various brain areas. We have already determined the number of excitatory and inhibitory inputs with their exact location on dendrites. Then, we have estimated the parameters of postsynaptic potential, amount of time needed for threshold

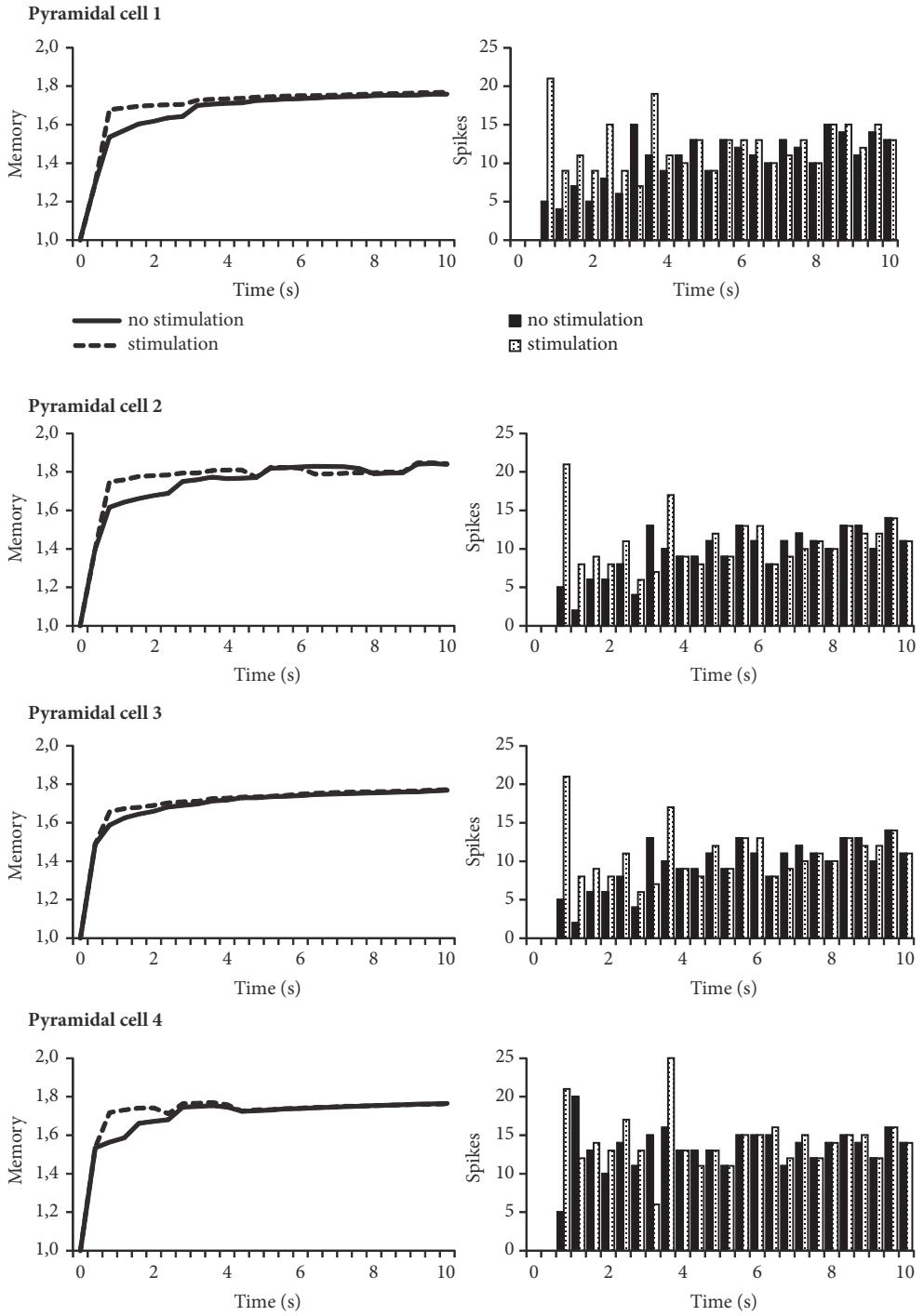


Figure 5: Comparison of memory during the simulation without and with LTP inducing protocol. On the left, relationship between memory function without and with LTP inducing protocol. On the right, histograms reflect number of firing spikes during 10s simulation without and with LTP inducing protocol.

and refraction, and finally two fundamental physiological values: resting potential and synaptic reversal potential. Thanks to those parameters, we have ability to present for model any algorithm weight change (learning) for the inputs (synapses). In order to inaugurate the simulation a signal for all excitatory impulses must be determined and in the Izhikevivh equation it is named as "I."

We performed more than a thousand simulations, using not only single pyramidal sells, but also a small network of ten neurons connected like in CA1 hippocampal area. Any

changes appearing in initial values or input patterns were the reason for further alterations of the interspike intervals (ISI) time series on the output. The real time of course simulations was even 10 or 20 seconds. This is the fundamental proof that the CA1 model has chaotic, dynamic characteristics.

There are some notions for the stochastic resonance phenomenon, which were firstly observed in 1950 by Bernard Langenbeck [81]. Nowadays such diagnostic methods are commonly used; however the physical white noise signal extends the capacity of inner ear receptors to react.

Undoubtedly, certain emotions such as curiosity or fear strengthen the capability to learn and memorize new models and patterns. In this process pyramidal neurons located in the cerebral cortex or hippocampus get supplementary inputs from the excited emotion areas such as amygdaloid body, which might be perceived as an indirect supervised learning algorithm. The elementary mechanism for long-term potentiation induction requires presence of NMDA channels and removal of magnesium ions blockade to enable calcium ions influx through them [82, 83]. This is accomplished by depolarization of postsynaptic region; however instant depolarization is dependent on history of input patterns. This information may lead to a conclusion that any accessory input of all possible characteristics could potentially enlarge learning capability and should be considered as an unlike equivalent of the stochastic resonance phenomenon.

5. Conclusions

The most influential conclusion of all our studies seems to be the prospect of extracting pure information processing algorithm from biological backgrounds as channels and membranes. The fundamental concept of our pyramidal neuron model was derived precisely from the theory of transistors with floating gates capacitor coupling, computer language, and usual models of all biological details measured in hitherto models of neurons. We did not use any of Hodgkin-Huxley, integrate and fire, or spike timing dependent plasticity formalisms. The received outcome is a mathematical circuit which matches major apparent features of living nervous cells. Moreover, we are able to repeat the Bliss and Lomo trial for induction of long-term synaptic potentiation in the rabbit hippocampus carried out in 1973, which is presented in Figures 2 and 3 from our previous work [54].

The circuit model within shift registers working as memory buffer for any synapse is believed to have a great potential to future development of spatiotemporal computing. Such an accessible mathematical model can become a starting point for constructing biologically inspired processors which could be slightly implemented in hardware like Neuron-MOS Transistor of Shibata or Ohmi, which nowadays are arousing great interest [84, 85].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that they have no conflicts of interest.

Supplementary Materials

CA1 network simulation: the video screen captures file hippocampal formation CA1 microcircuit from the simulation. On the top four pyramidal cells and on the bottom two basket cells and OL-M cell. 10-second real-time simulation of hippocampal cells with LTP inducing protocol. CA1 linear chart simulation: 10-second real-time simulation of pyramidal cells, basket cells, and OL-M cell with LTP inducing protocol showing linear chart. On the top four pyramidal cells and on the bottom two basket cells and OL-M cell (in the middle). Linear chart shows spikes and firing hippocampal cells formation of CA1 microcircuit. *(Supplementary Materials)*

References

[1] G. Buzsaki, "Two-stage model of memory trace formation: a role for 'noisy' brain states," *Neuroscience*, vol. 31, no. 3, pp. 551–570, 1989.

[2] P. Alvarez and L. R. Squire, "Memory consolidation and the medial temporal lobe: A simple network model," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 91, no. 15, pp. 7041–7045, 1994.

[3] J. M. J. Murre, "TraceLink: A model of amnesia and consolidation of memory," *Hippocampus*, vol. 6, no. 6, pp. 675–684, 1996.

[4] S. Káli and P. Dayan, "Off-line replay maintains declarative memories in a model of hippocampal-neocortical interactions," *Nature Neuroscience*, vol. 7, no. 3, pp. 286–294, 2004.

[5] R. U. Muller and M. Stead, "Hippocampal place cells connected by Hebbian synapses can solve spatial problems," *Hippocampus*, vol. 6, no. 6, pp. 709–719, 1996.

[6] A. D. Redish and D. S. Touretzky, "The role of the hippocampus in solving the Morris water maze," *Neural Computation*, vol. 10, no. 1, pp. 73–111, 1998.

[7] D. J. Foster, R. G. M. Morris, and P. Dayan, "A model of hippocampally dependent navigation, using the temporal difference learning rule," *Hippocampus*, vol. 10, no. 1, pp. 1–16, 2000.

[8] A. Arleo and W. Gerstner, "Spatial cognition and neuromimetic navigation: a model of hippocampal place cell activity," *Biological Cybernetics*, vol. 83, no. 3, pp. 287–299, 2000.

[9] P. Gaussier, A. Revel, J. P. Banquet, and V. Babeau, "From view cells and place cells to cognitive map learning: Processing stages of the hippocampal system," *Biological Cybernetics*, vol. 86, no. 1, pp. 15–28, 2002.

[10] R. A. Koene, A. Gorchetchnikov, R. C. Cannon, and M. E. Hasselmo, "Modeling goal-directed spatial navigation in the rat based on physiological data from the hippocampal formation," *Neural Networks*, vol. 16, no. 5-6, pp. 577–584, 2003.

[11] M. A. Gluck and C. E. Myers, "Hippocampal mediation of stimulus representation: A computational theory," *Hippocampus*, vol. 3, no. 4, pp. 491–516, 1993.

[12] C. V. Buhusi and N. A. Schmajuk, "Attention, configuration, and hippocampal function," *Hippocampus*, vol. 6, no. 6, pp. 621–642, 1996.

[13] P. Rodriguez and W. B. Levy, "A model of hippocampal activity in trace conditioning: Where's the trace?" *Behavioral Neuroscience*, vol. 115, no. 6, pp. 1224–1238, 2001.

[14] R. Granger, J. Whitson, J. Larson, and G. Lynch, "Non-Hebbian properties of long-term potentiation enable high-capacity encoding of temporal sequences," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 91, no. 21, pp. 10104–10108, 1994.

[15] B. P. Graham, "Pattern recognition in a compartmental model of a CA1 pyramidal neuron," *Network: Computation in Neural Systems*, vol. 12, no. 4, pp. 473–492, 2001.

[16] M. Migliore, "On the integration of subthreshold inputs from Perforant Path and Schaffer Collaterals in hippocampal CA1 pyramidal neurons," *Journal of Computational Neuroscience*, vol. 14, no. 2, pp. 185–192, 2003.

[17] P. Poirazi, T. Brannon, and B. W. Mel, "Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell," *Neuron*, vol. 37, no. 6, pp. 977–987, 2003.

[18] A. Lörincz and G. Buzsáki, "Two-phase computational model training long-term memories in the entorhinal-hippocampal region," *Annals of the New York Academy of Sciences*, vol. 911, pp. 83–111, 2000.

[19] A. Treves and E. T. Rolls, "Computational analysis of the role of the hippocampus in memory," *Hippocampus*, vol. 4, no. 3, pp. 374–391, 1994.

[20] J. L. McClelland and N. H. Goddard, "Considerations arising from a complementary learning systems perspective on hippocampus and neocortex," *Hippocampus*, vol. 6, no. 6, pp. 654–665, 1996.

[21] W. B. Levy, N. L. Desmond, and D. X. Zhang, "Perforant path activation modulates the induction of long-term potentiation of the schaffer collateral-hippocampal CA1 response: Theoretical and experimental analyses," *Learning & Memory*, vol. 4, no. 6, pp. 510–518, 1998.

[22] J. E. Lisman and N. A. Otmakhova, "Storage, recall, and novelty detection of sequences by the hippocampus: Elaborating on the SOCRATIC model to account for normal and aberrant effects of dopamine," *Hippocampus*, vol. 11, no. 5, pp. 551–568, 2001.

[23] D. Marr, "Simple memory: a theory for archicortex.," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 262, no. 841, pp. 23–81, 1971.

[24] E. Rolls, *Parallel Distributed Processing, chapter Parallel distributed processing in the brain: implications of the functional architecture of neuronal networks in the hippocampus*, Oxford University Press, 1989.

[25] A. Treves and E. T. Rolls, "What determines the capacity of autoassociative memories in the brain?" *Network: Computation in Neural Systems*, vol. 2, no. 4, pp. 371–397, 1991.

[26] A. Treves and E. T. Rolls, "Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network," *Hippocampus*, vol. 2, no. 2, pp. 189–199, 1992.

[27] A. Treves, "Quantitative estimate of the information relayed by the Schaffer collaterals," *Journal of Computational Neuroscience*, vol. 2, no. 3, pp. 259–272, 1995.

[28] B. L. McNaughton and R. G. M. Morris, "Hippocampal synaptic enhancement and information storage within a distributed memory system," *Trends in Neurosciences*, vol. 10, no. 10, pp. 408–415, 1987.

[29] K. Nakazawa, M. C. Quirk, R. A. Chitwood et al., "Requirement for hippocampal CA3 NMDA receptors in associative memory recall," *Science*, vol. 297, no. 5579, pp. 211–218, 2002.

[30] K. Nakazawa, L. D. Sun, M. C. Quirk, L. Rondi-Reig, M. A. Wilson, and S. Tonegawa, "Hippocampal CA3 NMDA receptors are crucial for memory acquisition of one-time experience," *Neuron*, vol. 38, no. 2, pp. 305–315, 2003.

[31] J. M. Lassalle, T. Bataille, and H. Halley, "Reversible inactivation of the hippocampal mossy fiber synapses in mice impairs spatial learning, but neither consolidation nor memory retrieval, in the Morris navigation task," *Neurobiology of Learning and Memory*, vol. 73, no. 3, pp. 243–257, 2000.

[32] R. C. O'Reilly and J. L. McClelland, "Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off," *Hippocampus*, vol. 4, no. 6, pp. 661–682, 1994.

[33] J. E. Lisman, "Relating hippocampal circuitry to function: recall of memory sequences by reciprocal dentate-CA3 interactions," *Neuron*, vol. 22, no. 2, pp. 233–242, 1999.

[34] N. A. Otmakhova and J. E. Lisman, "D1/D5 dopamine receptor activation increases the magnitude of early long-term potentiation at CA1 hippocampal synapses," *The Journal of Neuroscience*, vol. 16, no. 23, pp. 7478–7486, 1996.

[35] N. A. Otmakhova and J. E. Lisman, "Dopamine, serotonin, and noradrenaline strongly inhibit the direct perforant path-CA1 synaptic input, but have little effect on the Schaffer collateral input," *Annals of the New York Academy of Sciences*, vol. 911, pp. 462–464, 2000.

[36] M. E. Hasselmo and E. Schnell, "Laminar selectivity of the cholinergic suppression of synaptic transmission in rat hippocampal region CA1: Computational modeling and brain slice physiology," *The Journal of Neuroscience*, vol. 14, no. 6, pp. 3898–3914, 1994.

[37] M. E. Hasselmo and J. M. Bower, "Cholinergic suppression specific to intrinsic not afferent fiber synapses in rat piriform (olfactory) cortex," *Journal of Neurophysiology*, vol. 67, no. 5, pp. 1222–1229, 1992.

[38] M. E. Hasselmo, C. Bodelón, and B. P. Wyble, "A proposed function for hippocampal theta rhythm: separate phases of encoding and retrieval enhance reversal of prior learning," *Neural Computation*, vol. 14, no. 4, pp. 793–817, 2002.

[39] W. B. Levy, "A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks," *Hippocampus*, vol. 6, no. 6, pp. 579–590, 1996.

[40] W. B. Levy, "A Computational Approach to Hippocampal Function," in *Computational Models of Learning in Simple Neural Systems*, vol. 23 of *Psychology of Learning and Motivation*, pp. 243–305, Elsevier, 1989.

[41] M. Nishiyama, K. Hong, K. Mikoshiba, M.-M. Poo, and K. Kato, "Calcium stores regulate the polarity and input specificity of synaptic modification," *Nature*, vol. 408, no. 6812, pp. 584–588, 2000.

[42] B. Widrow and M. Hoff, "Adaptive switching circuits," *IRE WESCON Convention Record*, vol. 4, pp. 96–104, 1960.

[43] A. Lorincz, "Forming independent components via temporal locking of reconstruction architectures: A functional model of the hippocampus," *Biological Cybernetics*, vol. 79, no. 3, pp. 263–275, 1998.

[44] R. P. Kesner and E. T. Rolls, "A computational theory of hippocampal function, and tests of the theory: new developments," *Neuroscience & Biobehavioral Reviews*, vol. 48, no. 1, pp. 92–147, 2015.

[45] H. Lehn, H.-A. Steffenach, N. M. Van Strien, D. J. Veltman, M. P. Witter, and A. K. Håberg, "A specific role of the human hippocampus in recall of temporal sequences," *The Journal of Neuroscience*, vol. 29, no. 11, pp. 3475–3484, 2009.

[46] E. Kropff and A. Treves, "The emergence of grid cells: Intelligent design or just adaptation?" *Hippocampus*, vol. 18, no. 12, pp. 1256–1269, 2008.

[47] B. J. Kraus, M. P. Brandon, R. J. Robinson, M. A. Connerney, M. E. Hasselmo, and H. Eichenbaum, "Grid cells are time cells," *Society for Neuroscience - Abstracts*, pp. 769-719, 2013.

[48] B. Kraus, R. Robinson, J. White, H. Eichenbaum, and M. Hasselmo, "Hippocampal "Time Cells": Time versus Path Integration," *Neuron*, vol. 78, no. 6, pp. 1090–1101, 2013.

[49] E. T. Rolls, "A computational theory of episodic memory formation in the hippocampus," *Behavioural Brain Research*, vol. 215, no. 2, pp. 180–196, 2010.

[50] Patrick Greene, Mike Howard, Rajan Bhattacharyya, and Jean-Marc Fellous, "Hippocampal Anatomy Supports the Use of Context in Object Recognition: A Computational Model," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 294878, 19 pages, 2013.

[51] H. Braak and E. Braak, "Neuropathological stageing of Alzheimer-related changes," *Acta Neuropathologica*, vol. 82, no. 4, pp. 239–259, 1991.

[52] R. A. Sandler, D. Fetterhoff, R. E. Hampson, S. A. Deadwyler, and V. Z. Marmarelis, "Cannabinoids disrupt memory encoding by functionally isolating hippocampal CA1 from CA3," *PLoS Computational Biology*, vol. 13, no. 7, 2017.

[53] D. Świetlik, J. Białowąs, A. Kusiak, and D. Cichońska, "Memory and forgetting processes with the firing neuron model," *Folia Morphologica*, vol. 77, no. 2, pp. 221–233, 2018.

[54] D. Świetlik, J. Białowąs, A. Kusiak, and D. Cichońska, "A computational simulation of long-term synaptic potentiation inducing protocol processes with model of CA3 hippocampal microcircuit," *Folia Morphologica*, vol. 77, no. 2, pp. 210–220, 2018.

[55] V. Cutsuridis and M. Hasselmo, "GABAergic contributions to gating, timing, and phase precession of hippocampal neuronal activity during theta oscillations," *Hippocampus*, vol. 22, no. 7, pp. 1597–1621, 2012.

[56] V. Cutsuridis, S. Cobb, and B. P. Graham, "Encoding and retrieval in a model of the hippocampal CA1 microcircuit," *Hippocampus*, vol. 20, no. 3, pp. 423–446, 2010.

[57] M. Stewart and S. E. Fox, "Do septal neurons pace the hippocampal theta rhythm?" *Trends in Neurosciences*, vol. 13, no. 5, pp. 163–169, 1990.

[58] G. Buzsáki, L. W. Leung, and C. H. Vanderwolf, "Cellular bases of hippocampal EEG in the behaving rat.," *Brain Research*, vol. 287, no. 2, pp. 139–171, 1983.

[59] A. Bragin, G. Jando, Z. Nadasdy, J. Hetke, K. Wise, and G. Buzsaki, "Gamma (40-100 Hz) oscillation in the hippocampus of the behaving rat," *The Journal of Neuroscience*, vol. 15, no. 1, pp. 47–60, 1995.

[60] A. Pietak and M. Levin, "Bioelectrical control of positional information in development and regeneration: A review of conceptual and computational advances," *Progress in Biophysics and Molecular Biology*, vol. 137, pp. 52–68, 2018.

[61] J. O'Keefe and M. L. Recce, "Phase relationship between hippocampal place units and the EEG theta rhythm," *Hippocampus*, vol. 3, no. 3, pp. 317–330, 1993.

[62] W. E. Skaggs, B. L. McNaughton, M. A. Wilson, and C. A. Barnes, "Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences," *Hippocampus*, vol. 6, no. 2, pp. 149–172, 1996.

[63] Y. Omura, M. M. Carvalho, K. Inokuchi, and T. Fukai, "A log-normal recurrent network model for burst generation during hippocampal sharp waves," *The Journal of Neuroscience*, vol. 35, no. 43, pp. 14585–14601, 2015.

[64] G. Buzsáki, "Theta oscillations in the hippocampus," *Neuron*, vol. 33, no. 3, pp. 325–340, 2002.

[65] P. T. Huerta and J. E. Lisman, "Heightened synaptic plasticity of hippocampal CA1 neurons during a Cholinergically induced rhythmic state," *Nature*, vol. 364, no. 6439, pp. 723–725, 1993.

[66] R. P. Kesner and E. T. Rolls, "A computational theory of hippocampal function, and tests of the theory: new developments," *Neuroscience & Biobehavioral Reviews*, vol. 48, no. 1, pp. 92–147, 2006.

[67] K. R. Hedrick and K. Zhang, "Megamap: Flexible representation of a large space embedded with nonspatial information by a hippocampal attractor network," *Journal of Neurophysiology*, vol. 116, no. 2, pp. 868–891, 2016.

[68] X.-J. Wang, "Pacemaker neurons for the theta rhythm and their synchronization in the septohippocampal reciprocal loop," *Journal of Neurophysiology*, vol. 87, no. 2, pp. 889–900, 2002.

[69] R. Goutagny, J. Jackson, and S. Williams, "Self-generated theta oscillations in the hippocampus," *Nature Neuroscience*, vol. 12, no. 12, pp. 1491–1493, 2009.

[70] C. Y. L. Huh, R. Goutagny, and S. Williams, "Glutamatergic neurons of the mouse medial septum and diagonal band of broca synaptically drive hippocampal pyramidal cells: Relevance for hippocampal theta rhythm," *The Journal of Neuroscience*, vol. 30, no. 47, pp. 15951–15961, 2010.

[71] J. Bialowas and M. Frotscher, "Choline acetyltransferase-immunoreactive neurons and terminals in the rat septal complex: A combined light and electron microscopic study," *Journal of Comparative Neurology*, vol. 259, no. 2, pp. 298–307, 1987.

[72] M. Hajós, W. E. Hoffmann, G. Orbán, T. Kiss, and P. Érdi, "Modulation of septo-hippocampal θ activity by GABAA receptors: An experimental and computational approach," *Neuroscience*, vol. 126, no. 3, pp. 599–610, 2004.

[73] D. Amaral and P. Lavenex, "Hippocampal neuroanatomy," in *The hippocampus book*, P. Andersen, R. Morris, D. Amaral, T. Bliss, and J. O'Keefe, Eds., pp. 37–114, University Press, Oxford, UK, 2007.

[74] P. Andersen, R. Morris, D. Amaral, T. Bliss, and J. OKeefe, *The hippocampus book*, University Press, Oxford, UK, 2007.

[75] M. Witter, "Connectivity of the hippocampus," in *Hippocampal microcircuits: A computational modeler's resource book*, V. Cutsuridis, Ed., pp. 27–67, Springer, NY, USA, 2010.

[76] J. H. L. P. Sadowski, M. W. Jones, and J. R. Mellor, "Ripples make waves: Binding structured activity and plasticity in hippocampal networks," *Neural Plasticity*, vol. 2011, Article ID 960389, 2011.

[77] T. V. P. Bliss and G. L. Collingridge, "A synaptic model of memory: long-term potentiation in the hippocampus," *Nature*, vol. 361, no. 6407, pp. 31–39, 1993.

[78] T. V. P. Bliss, G. L. Collingridge, and R. G. M. Morris, "Synaptic plasticity in health and disease: Introduction and overview," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1633, Article ID 20130129, 2014.

[79] N. Kasabov, "To spike or not to spike: a probabilistic spiking neuron model," *Neural Networks*, vol. 23, no. 1, pp. 16–19, 2010.

[80] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks and Learning Systems*, vol. 15, no. 5, pp. 1063–1070, 2004.

[81] B. Langenbeck, "Geräuschaudiometrische Diagnostik. Die Absolutauswertung," *Archiv für Ohren- Nasen- und Kehlkopfheilkunde*, vol. 158, no. 2-6, pp. 458–471, 1950.

[82] M. Borjkhani, F. Bahrami, and M. Janahmadi, "Computational modeling of opioid-induced synaptic plasticity in hippocampus," *PLoS ONE*, vol. 13, no. 3, 2018.

[83] L. Y. Prince, T. J. Bacon, C. M. Tigaret, and J. R. Mellor, "Neuromodulation of the feedforward dentate gyrus-CA3 microcircuit," *Frontiers in Synaptic Neuroscience*, vol. 8, p. 32, 2016.

[84] T. Shibata and T. Ohmi, "An intelligent MOS transistor featuring gate-level weighted sum and threshold operations," in *Proceedings of the International Electron Devices Meeting, IEDM 1991*, pp. 919–922, USA, December 1991.

[85] T. Shibata and T. Ohmi, "Neuron MOS Binary-Logic Integrated Circuits—Part I: Design Fundamentals and Soft-Hardware-Logic Circuit Implementation," *IEEE Transactions on Electron Devices*, vol. 40, no. 3, pp. 570–576, 1993.

Hindawi
Complexity
Volume 2018, Article ID 2065491, 14 pages
https://doi.org/10.1155/2018/2065491

Research Article

Enhancing the Efficiency of a Decision Support System through the Clustering of Complex Rule-Based Knowledge Bases and Modification of the Inference Algorithm

Agnieszka Nowak-Brzezińska

Institute of Computer Science, Faculty of Computer Science and Material Science, Silesian University, ul.Będzińska 39, 41-200 Sosnowiec, Poland

Correspondence should be addressed to Agnieszka Nowak-Brzezińska; agnieszka.nowak@us.edu.pl

Received 19 April 2018; Revised 9 September 2018; Accepted 30 September 2018; Published 6 December 2018

Guest Editor: Ireneusz Czarnowski

Decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. Effective exploration of new knowledge in every domain of human life requires new algorithms of knowledge organization and a thorough search of the created data structures. In this work, the author introduces an optimization of both the knowledge base structure and the inference algorithm. Hence, a new, hierarchically organized knowledge base structure is proposed as it draws on the cluster analysis method and a new forward-chaining inference algorithm which searches only the so-called representatives of rule clusters. Making use of the similarity approach, the algorithm tries to discover new facts (new knowledge) from rules and facts already known. The author defines and analyses four various representative generation methods for rule clusters. Experimental results contain the analysis of the impact of the proposed methods on the efficiency of a decision support system with such knowledge representation. In order to do this, four representative generation methods and various types of clustering parameters (similarity measure, clustering methods, etc.) were examined. As can be seen, the proposed modification of both the structure of knowledge base and the inference algorithm has yielded satisfactory results.

1. Introduction

Big Data is no longer just about processing a huge number of bytes, but doing things with data that you could not do previously. It is not just tabular data you can easily stick into a spreadsheet or a database [1]. Where computer scientists were once limited to mere gigabytes or terabytes of information, they are now studying petabytes and even exabytes of information. At the same time, the tools to sift all that data are getting better as computer scientists refine and improve the algorithms they use to extract meaning from the deluge of data [2]. There is no doubt that big data are now rapidly expanding in all science and engineering domains. While the potential of these massive data is undoubtedly significant, fully making sense of them requires new ways of thinking and novel learning techniques to address the various challenges. Most traditional machine learning techniques are not inherently efficient or scalable enough to handle the data with the characteristics of large volume, different types, high speed, uncertainty and incompleteness, and low value density. In response, machine learning needs to reinvent itself for big data processing [3]. Current hot topics in the quest to improve effectiveness of the machine learning techniques include search for compact knowledge representation methods and better tools for knowledge discovery and integration.

The main subject of the author's scientific work lies at the boundary of artificial intelligence, methods of representation and exploration of domain knowledge, statistical methods of data analysis, and machine learning methods. Recent work focuses on managing complex knowledge bases with rule representation and the development of new inference algorithms in such data sets.

In order to extract useful domain knowledge from the studied area, a lot of data should be collected beforehand. Much also depends on how the rules are induced. For example, effective rule induction algorithms can generate a compressed set of several dozen or several hundred rules for a data set consisting of several thousand objects. That is why when talking about domain knowledge bases, files with several thousand rules are often considered to be too large [4]. The author's experience of working on such amount of data is presented in [5]. In this research, the author has focused on discovering the optimal methods for big data storage, managing management, and exploration. In order to do this, the preliminary experiments, using medium-sized knowledge bases with various types and sizes of data, were carried out. The goal is to specify the most important parameters that facilitate a quick and effective discovery of new knowledge in knowledge bases.

In inference processes based on the rule-based knowledge bases, we explore new domain knowledge by activating the rules (components of a rule-based system with form: IF premises THEN conclusion) with true premises—the ones which may have been covered by the facts given a priori. The process of activating a given rule results in dealing with its conclusion as a new fact. The more rules and initial facts in a given knowledge base, the more rules that can be activated. Of course, the recent solutions in the area of decision support systems require that they additionally perform the task in the shortest time and with the least human involvement. Let us take an example of the medical system, in which we aim to make a decision as fast as possible, based on the knowledge (facts) about a particular patient. The system searches a knowledge base with rules in order to find all the rules relevant to the given set of facts. In case of a big data set, with many rules, such a process can be too time-consuming. The classic approach is then inefficient, as it has to search every rule in a given knowledge base, which in case of big dataset takes too much time. Thus, new solutions need to be discovered and developed. Such solutions should result in the effectiveness not worse than it is in the case of the classic approach, doing it as quickly and as efficiently as possible. It requires a deep analysis of the knowledge stored in the knowledge bases and exploration of the information about a given domain, for example, in the form of so-called meta-knowledge (knowledge about knowledge). In the literature, there is a lot of research devoted to the subject of meta-knowledge and meta-rules [6–8].

It is widely known that the best way to learn a new field is to use generalization skills. Generalization is the process of discovering general features, important features, and the features common for a given class of objects. Following this path, the generalization of the information saved in the rules allows us to gain knowledge about those rules. By attributing similar rules to one group and through the generalization of such groups, we obtain knowledge about many rules without having to review each rule separately.

The notion proposed in this paper is built around the idea of the similarity analysis between the rules and then their subsequent clustering. Among numerous clustering algorithms, the agglomerative hierarchical clustering (AHC) algorithm was chosen (the author previously analysed many other algorithms as well [9, 10]). Its most important feature (and advantage) is the fact that it clusters (agglomerates) the most similar rules and forms a group from them. Regarding the rules in the knowledge base, we must take into account that from a certain moment of clustering, the rules cease to be similar in any respect and there is no reason to cluster them any longer. Thus, the classic clustering AHC algorithm requires a modification. Furthermore, to effectively (efficiently and quickly) find the right group of rules to activate, it is necessary to describe them optimally. The author has recently devoted much attention to the proposal and analysis of methods for representing groups of rules, using the generalization approach [11]. This paper is aimed at verifying the effectiveness of inference, i.e., the ability to activate rules by reviewing only a selected part of the entire knowledge base, most relevant to the given facts. An inference process can be considered successfully finished where only a small part of the entire knowledge base is searched and we are able to successfully find and activate a given rule (or rules).

It turns out that some clustering parameters have a significant impact on the structure of groups of rules (a tendency to create small or large clusters, to identify atypical rules and separate them from groups). Moreover, certain methods of representation of rule clusters (representative generation methods) are characterized by a tendency to create overly general representatives (or sometimes empty) or overly detailed representatives that have ceased to reflect the content of the whole group. Having knowledge about which clustering parameters and which representative generation methods ensure the best efficiency, we will be able to strive to achieve optimal results.

The structure of the paper is as follows. Section 2 introduces the rule-based knowledge bases and inference processes in decision support systems. Managing of rules in knowledge bases is the main subject of Section 3. The proposed approach with a description of the clustering algorithm and inference algorithm for a hierarchical structure of a knowledge base with rule clusters is presented in Section 4. The results of experiments with their interpretation are included in Section 5. The summary is presented in Section 6.

2. Knowledge-Based Systems

The knowledge-based system (KBS) is a system that uses artificial intelligence to solve problems. It focuses on using knowledge-based techniques to support human decision making, learning, and action. Such systems are capable of cooperating with human users and are fit for purpose. We may even say that they are better than humans are, as they are enriched with the virtues of efficiency and effectiveness. They are able to diagnose diseases, repair electrical networks, control industrial workplaces, create geological maps, etc. Representation of knowledge is difficult because an expert knowledge can be imprecise and/or uncertain. In general, the knowledge is represented as a large set of simple rules. Conclusions are generally obtained through the inference process. The expert systems have been pioneers in the field of knowledge-based systems. They replace one or more

experts for problem solving. In many situations, they may be more useful than traditional computer-based information systems. There are many circumstances when they become particularly useful: when an expert is not available, when expertise is to be stored for future use or when expertise is to be cloned or multiplied, when intelligent assistance and/or training are required for decision-making or problem-solving, or when more than one expert's knowledge has to be stored on one platform. All these situations make them very useful nowadays, and thus, it is very important to improve their performance and usability. The improvement may concern both the structure of the knowledge base and the inference algorithms.

2.1. Rule-Based Knowledge Bases. Among various methods of knowledge representation, rules are the most popular form.

Rule-based knowledge representation uses the Horn clause form: "if premise then conclusion." This is one of the most natural ways for domain experts to explain and present their knowledge. Activation of the rules during the inference process results in adding their conclusions as new facts (new knowledge). Let us assume that the knowledge base KB is a set of N rules: $\mathrm{KB} = \{r_1, r_2, \ldots, r_N\}$. Every rule $r \in \mathrm{KB}$ has a form $r = \mathrm{cond}_1(r) \wedge \mathrm{cond}_2(r) \wedge \cdots \wedge \mathrm{cond}_m(r) \longrightarrow \mathrm{concl}(r)$, where $\mathrm{cond}_1(r) \wedge \cdots \wedge \mathrm{cond}_m(r)$ is the conjunction of the rule's conditions (premises) and $\mathrm{concl}(r)$ is the conclusion of the rule r.

Rules may be generated automatically using one of many possible algorithms based on the machine learning techniques. The knowledge base can be composed of different types of rules: classification rules, association rules, regression rules, or the so-called survival ones [12]. In addition, the rule set can be obtained by transforming the decision tree [13]. They also can be given by experts, but such process is a very difficult task. Usually, the value of experts' knowledge is rated so highly that experts are reluctant to share it. Therefore, to carry out the right number of experiments, it was decided to use the knowledge base with rules generated automatically from data shared within the UCI machine learning repository [14]. An efficient algorithm for generating rules automatically from data is the LEM algorithm [15]. It is based on the rough set theory [16–18] and induces a set of certain rules from the lower approximation (lower approximation is a description of the domain objects that are known with certainty to belong to the subset of interest), and, respectively, a set of possible rules from the upper approximation (upper approximation is a description of the objects that possibly belong to the subset of interest). This algorithm follows a classical greedy scheme which produces a local covering of each decision concept. It covers all examples from the given approximation using a minimal set of rules.

The procedure for preparing knowledge bases for this work was as follows. Each selected set of data from the repository was rewritten as a decision table, which was then subject to the process of rule induction (LEM2 algorithm) using the RSES tool [19].

As an example, let us take a heart disease dataset [20], which originally contains 303 instances, described by 14 nominal and numerical attributes (age: in years, sex: (1 = male; 0 = female), cp: chest pain type with values (1): typical angina, (2): atypical angina, (3): nonanginal pain, and (4): asymptomatic and others). The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

The piece of the original dataset is as follows:

```
63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0,
2.3,3.0,0.0,6.0,0
67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0,
1.5,2.0,3.0,3.0,2
67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0,
2.6,2.0,2.0,7.0,1
37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0,
3.5,3.0,0.0,3.0,0
41.0,0.0,2.0,130.0,204.0,0.0,2.0,172.0,0.0,
1.4,1.0,0.0,3.0,0
56.0,1.0,2.0,120.0,236.0,0.0,0.0,178.0,0.0,
0.8,1.0,0.0,3.0,0
```

A knowledge base with 99 rules has been achieved. The source file is as shown in Sourcecode 1.

The rule `(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(sex=0)&(pain_type=3.0)=>(disease=1[23]) 23` should be read as: *if* (blood sugar = 0) and (thal = 3.0) and (sex = 0) and (pain_type = 3.0) *then* (disease = 1) which is covered by 23 of the 303 instances in the original dataset (8% of 303 instances cover this rule).

When the size of the input data (which rules are to be generated from) increases, the number of generated rules does too. Let us look at the *diabetes* data set [14]. It contains the data for 768 objects described with 8 continuous attributes. Processing the data with LEM2 and RSES with an implementation of the LEM2 algorithm, 490 rules have been created. For the *nursery* dataset, which originally contains 12,960 instances described with 9 conditional attributes, 867 rules have been generated. Such numbers make it difficult or even impossible to be analysed by a person. It is also important to note that the generated rules might have a varying number of premises. It can be said that the fewer premises a rule has, the easier it is to determine if it is true (it requires less number of conditions to cover). On the other hand, making a decision dependent on the highest possible number of conditions may suggest that if all the conditions have been met, the decision must be correct.

When looking globally at a knowledge base with rules, it turns out that it may contain a large number of short rules (with one premise or a few) but also some rules described with a large number of premises with only a few premises that differentiate them. This, in turn, brings about various problems at the rule analysis stage in the inference process. When there is a set of many long rules (described with several premises) which differ from one another by a single premise, it can extend the inference process which then attempts to check all the rules which are deemed fit to be activated. Another possible outcome might be that in a given knowledge base there is an uneven distribution of rules connected with given premises. This may result in a large group of rules dedicated to one area only and one or very few rules describing other areas of the domain (the particular part of the

```
RULE_SET heart_disease
ATTRIBUTES 14
agenumeric 1
sex numeric 1
.....
diseasesymbolic
DECISION_VALUES 2
2
1
RULES 99
(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(sex=0)&(pain_type=3.0)=>(disease=1[23]) 23
(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(no._of_vessels=0)&(sex=0)&(electrocardiograph=0.0)
=>(disease=1[22]) 22
....
...
(blood_sugar=0)&(sex=1)&(electrocardiograph=2.0)&(angina=0.0)&(pain_type=1.0)&(age=42)
=>(disease=1[1]) 1
(blood_sugar=0)&(sex=1)&(electrocardiograph=2.0)&(no._of_vessels=0)&(thal=7.0)&(angina=1.0)&(age=53)
=>(disease=1[1]) 1
```

Sourcecode 1

domain has not been sufficiently explored). Finding rare rules might become a nontrivial task. When taking into consideration the matter of big sets of often dispersed rules, it turns out that for the effectiveness of the inference processes, decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. In other words, they are methods and tools which help to review the rules effectively and quickly find those to be activated. One of the available solutions is rule clustering. In the subject literature, this issue has been extensively described and most of the time it focuses on cluster analysis [21]. Assuming every rule cluster as a group of similar rules, it is possible to create its representative as a set of all the features that describe the group in the best possible way. Let us imagine there is a knowledge base with a large number of rules which are subject to clustering. As a result, there will be a structure of groups of rules which are similar to one another. The extent of cohesiveness of a knowledge base will translate into the number and size of the resulting clusters of rules. There are several possible scenarios: a small number of clusters which contain a large number of rules in each of them or a large number of clusters which contain a few rules in each of the clusters. Of course, the scenarios described above are at the extreme ends of the scale. However, the generated structure of clusters may be well-balanced where each cluster contains a comparable number of rules and the number of rules is close to the size of each cluster (e.g., if there are 100 rules which are divided into 10 clusters with 10 rules in each).

Subsequently, the effectiveness of the knowledge extraction from rule clusters depends on the rule cluster quality and the efficiency of inference algorithms. For rule clusters, we create representatives and they are then searched in the process of inference. Due to the fact that the quality of representatives and the optimization of inference processes are so important, better solutions are still being sought.

To make the rule activation process possible, apart from the gathered knowledge, an inference mechanism is necessary. The following subsection presents the definition of inference and a short description of the existing inference algorithms and discusses the parameters and the inference control strategies.

2.2. Inference Algorithm. An inference engine is a software program that refers to the existing knowledge, manipulates the knowledge in line with needs, and makes decisions about actions to be taken. It generally utilizes pattern matching and search techniques for conclusions. Through these procedures, the inference engine examines existing facts and rules and adds new facts when possible. There are two common methods of deriving new facts from rules and known facts. These are data-driven (forward chaining) and goal-driven (backward chaining) inference algorithms. The most popular one, with respect to the usability in real-life applications, is the data-driven algorithm based on the *modus ponens* rule—a common inference strategy. It is simple and easy to understand [22]. The framework can be given as follows: the rule states that when A is known to be true and a rule states "if A, then B," it is valid to conclude that B is true.

The data-driven algorithm starts with some facts and applies rules to find all possible conclusions. It is applicable when the goal of inference is undefined. The inference with a given goal is provided until this goal is considered as a new fact. The case in which there are more than one possible rule to activate, in a given iteration of the inference algorithm, is called in the literature *a conflict set*, and the method which deals with the issue is called the conflict set resolution strategy [23]. It should be emphasized, especially in case of a big dataset, that such situation occurs very often. There are many possible strategies proposed in the literature, but the most popular ones are to use the *FIFO* (First In First Out) or *LIFO* (Last In First

Out) techniques familiar in programming languages. When there are many rules and facts involved in an expert system, classic inference algorithms become ineffective. Inference times become unacceptable, and the number of newly generated facts exceeds the limit of the new knowledge that can be properly absorbed.

In such cases, it is necessary to find new inference algorithms which ensure effective management of the analysis process for rules to be activated. One may also consider changing the structure of the knowledge base with the rules to organize them in a specific and well-described structure so that later its search would be effective.

In this paper, the author continues her research on modification of a knowledge base structure with rules into a hierarchical one where the quality of representatives of the created rule clusters is as important as the quality of these clusters.

Therefore, the author proposes the following method of optimization. At the first stage, the knowledge base structure is modified. In the classic approach where the knowledge base is a set of rules written without any specific order, it is necessary to search the entire set of rules. The author proposes to cluster the rules with similar premises into the rule clusters. Among various methods, the agglomerative hierarchical clustering algorithm is used in this research (the author has also studied the use of other algorithms [10]). Its classic approach assumes merging, in every iteration, the two most similar rules or groups of rules into one group. The proposed modification of this approach is based on finding the optimal moment to cut the created hierarchical structure of rules. It should be finished when there is not enough similarity between the rules or groups of rules which remained to be clustered. Details of the proposed approach are presented in the following section.

3. Rule Clustering

Too many rules in the knowledge base can negatively affect the effectiveness of management of rules. One of the ways of managing the rules is to cluster them into groups and to describe the groups by their representatives. Each cluster is described using a so-called group representative (Profile). The notion of cluster analysis indicates that objects in the analysed dimension are split into clusters which collect the objects most similar to one another and the resulting clusters are as different as possible [21]. The optimal structure of rule clusters assumes a maximum internal similarity and a minimal external similarity between groups of rules. It guarantees an optimum internal cohesion and external separateness of clusters. In the next subsection, the author briefly introduces other clustering methods.

3.1. A Short Characteristic of Clustering Algorithms. Within the scope of cluster analysis algorithms, it is possible to select either partitional (sometimes called k-optimizing algorithms, as exemplified by k-means) or hierarchical algorithms (which provide additional knowledge about the order of clustering the most similar objects together, e.g., the agglomerative hierarchical clustering algorithm (AHC)). Both partitional and hierarchical algorithms utilize the distance or similarity measurement in the process of finding similar objects. Moreover, there are algorithms based on the intracluster density (DBSCAN [24] and OPTICS [25]) and, most recently, spectral analysis algorithms (SMS (spectral mean shift) [26]).

Assuming that clustering is an automated process performed on a random set of rules with an unknown structure, the best solution which helps to avoid other possible problems is to use a hierarchical algorithm. The above-mentioned problems are, among others, an inability to determine an optimum number of clusters (necessary for partitional algorithms), the need to separate rare objects (rules) from the created clusters, and a motivation to gain additional knowledge on the sequence of rule clustering so that for each rule, another most similar rule or cluster can be found. In the density-based algorithms, similarly to partitional algorithms, additional clustering parameters like a minimum proximity threshold or the number of objects in a cluster need to be defined. The agglomerative hierarchical clustering algorithm (AHC) is free of such limitations [9, 10]. This algorithm has many modifications which vary from the original with respect to a changing stop condition of the clustering process.

3.2. Agglomerative Hierarchical Clustering Algorithm. The author proposes the clustering of rules with similar premises which produces a hierarchical structure (dendrogram). In the classic form of the agglomerative hierarchical clustering algorithm (AHC), the clustering process of individual rules should be continued until a single cluster of rules is obtained with a reservation that at each step a cluster is created by joining pairs of the most similar rules or clusters of rules. Accordingly, for the N number of rules in a knowledge base, the number of the algorithm's iterations is equal to $N-1$. It is easy to notice that for numerous knowledge bases the inference's duration time might be a problem. This is an unacceptable feature for big knowledge bases, and modifications which reduce the number of iterations are welcome.

3.3. Clustering Parameters. There are various clustering parameters that help to achieve optimal clustering results. In this research, the author has analysed such parameters as similarity measures, the number of clusters to create, and clustering methods.

3.3.1. Similarity Measures. Clustering of similar objects requires that similarities (or distances) between the object be defined. In the literature, there is a lot of research devoted to the analysis of available measures of similarity and dissimilarity of objects [27, 28]. These measures (in this paper) have been used to determine the similarities of rules between one another as well as the similarities of rules and clusters of rules in relation to the cluster representatives. The same measures can be subsequently used to measure the similarity of representatives for clusters of rules and facts in the inference process. To provide the universality of the solution, both the single rules and

clusters use the conjunction of pairs which consist of an attribute and its value. The values of attributes may be symbolic and continuous.

Generally, a similarity value for a pair of rules r_i and r_j which belong to a set of rules R is calculated in the following way:

$$\mathrm{sim}(r_i, r_j) = \sum_{f=1}^{m} w_f * \mathrm{sim}_f(r_{if}, r_{jf}), \quad (1)$$

where sim_f is a similarity value between two rules r_i and r_j in relation to the $f-$th attribute and the value w_f is the weight of the attribute a_f (usually determined as $w_f = 1/d$ for $f = 1, \ldots, d$, where d is the number of attributes). Alternatively, weights 0 and 1 can be used for attributes (where 0 for the $f-$th attribute's weight means that the attribute does not appear in the rule while 1 means that a given attribute constitutes the rule's premise part). The similarity value can be obtained by using one of a various possible similarity measures. The author dealt with the influence of measures of similarity on the clustering quality in [29, 30]. In [29], nine various measures were described and analysed: SMC (simple matching coefficient) and its modification wSMC (weighted simple matching coefficient), Gower's measure (widely known in the literature), two measures used for information search in large text files (OF and IOF) and four measures based on the probability of occurrence for a given feature in the description of a rule or a group of rules (Goodall's measures) [27, 28]. In this research, the author uses the same set of similarity measures (in the experimental stage, each of these methods was used). The measures have been widely described by the author in [29, 30]; therefore, the issue is not discussed again in this work.

For example, the similarity value sim_f based on the wSMC equals 1 if rules r_i and r_j contain the same value for the f attribute. Otherwise it equals 0. Hence, only if rules r_i and r_j contain the same values for the every attribute in their premises and weight w_f is determined as $w_f = 1/d$ for $f = 1, \ldots, d$ and d is the number of attributes, then the similarity value $\mathrm{sim}(r_i, r_j)$ equals 1. If the rules differ at least for one attribute, the value is less than 1. Value 0 for $\mathrm{sim}(r_i, r_j)$ (in case of wSMC similarity measure) means that there was not even one attribute for which rules r_i and r_j would have the same value.

Some of the analysed measures determine the similarity of rules using the frequency $f(r_{if})$ of occurrence of a certain pair of attributes and its values in the entire set of rules ($f(r_{if})$ denotes the number of times a premise r_{jf} appears in rules), while others are based on probabilities $p_f(r_{if})$ ($p_f(r_{if})$ denotes the sample probability of the case when a premise r_{if} appears in rules: $p_f(r_{if}) = f(p_f(r_{if}))/N$).

3.3.2. Number of Clusters. To determine an optimum similarity threshold might be impossible if the algorithm needs to be made independent from the type of data. It must be remembered that when similar rules are to be clustered, the threshold has to be set up at a reasonably high level or the clustering within a knowledge base can be initiated for rules which are practically dissimilar to one another and it might be impossible to reach a high level of similarity. In [9, 10], the author has presented an approach based on the termination of clustering when the intercluster similarity is no greater than the intracluster similarity. Unfortunately, the computations required for this approach are too burdening as far as the clustering algorithm is concerned. Another solution is the termination of clustering at a certain level as an attempt to force upon the number of clusters. Then, the AHC algorithm joins the rules and their clusters as long as the assumed number of clusters is reached. The above-described solution is presented in this paper.

In the literature, there are multiple papers which deal with the issue of an optimum selection of the number of clusters in the clustering algorithms [31, 32]. The most prevalent approach to be found in these papers underlines the necessity to perform numerous iterations for a gradually changing number of clusters and then choosing an optimum solution. Theoretically, it means that the number of possible partitions for a knowledge base with N rules equals N because, having 5 rules to cluster, we may place every rule in 1 or 2, 3, 4 and even into 5 clusters. Of course, the first and last solutions do not make sense (we would achieve one big cluster with an entire set of rules or 5 singular rule clusters). For this reason, the starting parameter value pertaining to the number of groups is 2 and increases by 1 in every partition until the number of clusters is smaller than the number of rules. If numerous knowledge bases are concerned, such an approach would not be time-effective.

The author has attempted to propose heuristics which help to determine an optimum number of clusters. The number of clusters K to be created is calculated with respect to the equations $K_1 = \lceil \sqrt{N} + i * \%N \rceil$ and $K_2 = \lceil \sqrt{N} - i * \%N \rceil$. K_1 and K_2 are the numbers of clusters to create, and N denotes the number of rules. It is easy to see that the modification consists in the clustering for a gradually changing (one step at a time, iteratively relative to the variable i, for $i = 1, 2, \ldots$) parameter K. Such a solution makes it possible to find the optimal number of clusters to create and does not require checking all possible scenarios but only some of them. For example, in case of a heart disease dataset with 99 rules, all the possible rule partitions, based on the proposed heuristics, are as follows: $K = 1, \ldots, 20$. Hence, instead of generating 99 different rule partitions, only 20 are created and analysed.

3.3.3. Clustering Methods. In this paper, the author has used four most popular methods as found in the literature. The first of them, the single-link method (SL), measures the distance between clusters R_p and R_q as a minimum distance between a random pair of rules r_i and r_j where $r_i \in R_p$ and $r_j \in R_q$. The second one is called the complete-link method (CL) and defines the distance between the cluster R_p and R_q as the longest distance between any two objects in two clusters.

Algorithm 1: hierarchical clustering for rules

Data: $KB = \{r_1 \dots r_N\}$ – rules from knowledge base; K - number of clusters to create;
Result: $PR = \{R_1, R_2, \dots, R_K\}$ - the structure of a K number of clusters of rules; $Profiles(PR) = \{Profile(R_1), Profile(R_2), \dots, Profile(R_K)\}$ - a set of representatives for these clusters;
begin
  /* create a clusters structure $PR := \{R_1, R_2, \dots, R_N\}$ in which each cluster $R_i = \{r_i\}$ is a single cluster, $i = 1, 2, \dots, N$; */
  $M := N$;
  while $M > K$ **do**
    /* create similarity matrix $S_{M \times M}$ for all clusters of rules $R_j, R_l \in PR$ in which every cell $s[j, l]$ contains similarity value for a pair of clusters R_j and R_l: */
    $s[j, l] := sim(Profile(R_j), Profile(R_l))$;
    /* find a cell with the maximum value of similarity */
    $(j, l) = \arg\max_{1 \leq j, l \leq M} \{s[j, l]\}$;
    /* create a new cluster R_q (and its representatative $Profile(R_q)$) which contains clusters R_j and R_l: $R_q := R_j \cup R_l$. Remove clusters R_j and R_l from the PR and add R_q to PR: */
    $PR := PR \cup R_q \setminus \{R_j, R_l\}$;
    $M := M - 1$;
  end
  /* return PR and $Profiles(PR)$; */
end

Algorithm 2: data-driven inference algorithm

Data: $PR = \{R_1, R_2, \dots, R_K\}$ - K clusters of rules;
$F = \{f_1, f_2, \dots, f_f\}$- set of facts; $goal$ - goal of the inference;
Result: $F_{new} = \{f_1, f_2, \dots, f_p\}$ - set of new facts explored from PR
begin
  boolean $stop = false$, $result = false$, $F_{new} = \{\emptyset\}$, $iterationCounter = 0$;
  if $goal != null$ **then**
    foreach *fact* $f_h \in F$ **do**
      if $f_h == goal$ **then**
        $stop = true$;$result = true$;
      end
    end
  end
  while $stop == false$ *and* $iterationCounter < 2$ **do**
    /* Find clusters of rules -the most relevant to the set F */
    foreach *cluster* $R_i \in PR$ **do**
      calculate $sim(Profile(R_i), F)$;
    end
    $Facts_{counter} = 0$; $R_{relevant} = R_i : sim(R_i, F)$ is maximal;
    if $R_{relevant} != null$ **then**
      $iterationCounter$ ++;
      activate($R_{relevant}$);
      $F_{new} := F_{new} \cup concl(R_{relevant})$;
      foreach *fact* $f_g \in F_{new}$ **do**
        if $goal != null$ **then**
          if $f_g == goal$ **then**
            $stop = true$;$result = true$;
          end
        end
      end
    end
    else
      $stop = true$;$iterationCounter = 0$;
    end
  end
  return $result$;
end

Figure 1: The pseudocodes of the hierarchical clustering algorithm for rules and the data-driven algorithm for rule clusters.

There are two more methods known in the literature—the average link method and the centroid link method. The former, marked as AL in this paper, measures the distance between the luster R_p and R_q as an average distance of all pairs of objects located within the examined clusters. The latter, marked in this paper as CoL, always calculates the distance between the clusters R_p and R_q as a distance between their centroids. A centroid is a pseudo-object whose attribute values are mean values of all objects in the cluster.

4. Proposed Approach

Having obtained groups which consist of similar rules, in fact only a small part of the knowledge base is searched. The previous object-by-object analysis, where the searched objects need to match the knowledge in the most possible way, can be reduced to matching the input data to each cluster's representative and selecting the best matching representative.

4.1. Hierarchical Structure of a Knowledge Base. As the resulting structure is one or more binary trees with M number of nodes, it is easier to reduce the computing complexity of the inference algorithm from the linear to the $\log_2 M$ complexity as the former emerges from the necessity of review of all rules in the knowledge base in order to find a set of activable rules. The knowledge base's structure with rule clusters shall be defined as a sorted pair $(\mathrm{PR}, \mathrm{Profiles}(\mathrm{PR}))$ where $\mathrm{PR} = \{R_1, R_2, \dots, R_K\}$ represents the structure of a K number of clusters and $\mathrm{Profiles}(\mathrm{PR}) = \{\mathrm{Profile}(R_1), \mathrm{Profile}(R_2), \dots, \mathrm{Profile}(R_K)\}$ constitute a set of representatives for these clusters (for $K \ll N$). The following two conditions must be met: $\bigcup_{j=1,2,\dots,K} R_j = \mathrm{KB}$ and $R_l \cap R_j = \varnothing$ for $j \neq l$ and $j, l = 1, 2, \dots, K$. A *hierarchical knowledge base* contains a structure of clusters of rules together with their representatives. As a result of the application of the AHC algorithm with a set criterion of stopping the agglomeration, we get a number of clusters (equal to K) containing other rule clusters or single rules. This structure is then searched in the inference process.

4.2. Agglomerative Hierarchical Clustering: A Proposed Approach. The pseudocodes of the hierarchical clustering algorithm for rules and data-driven inference algorithm for rule clusters are presented in Figure 1. Iteratively, until a given number of clusters (K) is not achieved, at every step of the clustering process, we create a similarity matrix for all rule clusters. Each cell contains a similarity value for a pair of rule clusters R_l and R_j. Then, we have to choose a matrix cell with the biggest similarity. At the end of each iteration, we create a new cluster R_q which contains the merged clusters R_l and R_j and we remove the clusters from the structure PR and add the new cluster R_q to it. The cluster analysis in effect produces fairly homogeneous groups of rules together with their representatives.

4.3. Knowledge Extraction in Rule Clusters. The decision-making process consists of extraction of new knowledge based on both the rules in a knowledge base and the facts. Since the rules have been merged into groups, the inference process must apply to the rule clusters. The idea proposed by the author is based on the method widely known in the literature within the domain of retrieval information systems and searching within hierarchical structures. Rule clustering with the AHC algorithm creates a hierarchical structure in the form of a dendrogram. A similar structure was obtained in the SMART system [33] where textual documents were subject to clustering. The clusters therein were defined as such sets of documents where each item is similar to all the remaining parts of the set. The obtained hierarchy of documents was then searched through analysis of the similarity between the groups' representatives and the given query. At each level of the hierarchy, the most similar group was chosen. The process ended when the most relevant group (document) was found [34]. The objective of the procedure is to maximise the search efficiency by matching a request with only a small subset of the stored documents, at the same time minimizing the loss of the relevant documents retrieved in the search. It is necessary to remember that cluster representatives are analysed; thus, the efficiency of searching within documents depends on the quality of the representatives. There are many possible ways to build a cluster representative. For example, document clusters can be represented by the set of the features most common for all the documents in a given cluster. The representative can be general or specific, which is very important in the context of inference efficiency. General representatives as a short type description may be easy to analyse but take more time to find a given document. Specific representatives contain usually many features in their descriptions and thus it takes much more time to analyse one representative, but usually we can easily find a given document.

In this project, the author works with rules in a knowledge base which are a very specific data type and thus require a specific way to manage them properly. They may have different lengths and may contain not only different attribute values but, above all, completely different attributes, which significantly affect the ability to compare them and to look for similarities.

4.4. Rule Clusters' Representatives. When a set of clusters has been generated, it is possible to construct a representative classification vector for each cluster, called a *centroid vector*, such that the property assignment of the centroid reflects the typical, or average, values of the corresponding property values for all elements within each given cluster. Various methods can be used to generate the centroid vectors. Considering the fact that rules in a knowledge base are a specific type of data and most of the time those rules are recorded with various types of data, the author proposes an approach which considers both nominal and numeric features in a representative's description. To find out which form of a representative (general or detailed) provides a greater effectiveness of the resulting structure and inference processes, the author proposes several different approaches. It should be noticed that in her previous research [11], the author analysed also other methods of generating cluster representatives. Each rule cluster $R_q \in \mathrm{PR}$ is assigned a representative called a *profile* ($\mathrm{Profile}(R_q)$). In the basic approach (further referred to as the threshold approach), a representative consists of all such attributes which have appeared in k% of rules in a given group (default $k = 30\%$):

$$\mathrm{Threshold}(R_q) = \cup\{p_s : \mathrm{frequency}(\mathrm{getAttr}(p_s)) \geq k \text{ for } p_s \in \mathrm{cond}(r_i), r_i \in R_q\}, \tag{2}$$

where $\mathrm{frequency}(\mathrm{getAttr}(p_s))$ returns the number of times when the attribute of a given premise p_s appears in the conditional part of all rules in the group R_q. If a given attribute reaches a set threshold then, depending on its type, its value (for symbolic features) or a mean (for numeric features) is added to the representative.

As this method analyses only the attribute part in pairs (attribute, value), the accuracy of the searching process may not be as precise as it is for other methods. Finding similar representatives with this technique means only that a rule cluster containing a given attribute has been found.

The conditional and decision parts of every rule are created from a given set of pairs (attribute, value). For the following set of attribute $A = \{a, b, c, d, e, dec\}$ and their values $V_a = \{a_1, a_2, a_3\}$, $V_b = \{b_1, b_2\}$, $V_c = \{c_1, c_2\}$, $V_d = \{d_1, d_2\}$, $V_e = \{e_1, e_2\}$, and $V_{dec} = \{A, B, C\}$, we may consider a few different scenarios (for simplicity's sake, in the example let us assume that all the attributes are at a nominal scale). For the knowledge base $\mathrm{KB} = \{r_1, r_2, r_3, r_4\}$, the following rules are

$$\begin{aligned} r_1 &: (a, a_1) \wedge (c, c_2) \longrightarrow (dec, A), \\ r_2 &: (a, a_2) \wedge (c, c_2) \longrightarrow (dec, B), \\ r_3 &: (b, b_1) \longrightarrow (dec, C), \\ r_4 &: (a, a_3) \longrightarrow (dec, A). \end{aligned} \tag{3}$$

We may say that rule r_3 is unlike the others (it is described by other attributes) while rules r_1 and r_2 are quite similar because besides the same premise (c, c_2), they also contain a similar premise with an attribute a. Rule r_4 is (like rule r_3) unlike others, but looking only at the attribute part, we may say that it is more similar to rules r_1 and r_2 than rule r_3, containing an attribute a.

Assuming that the selected clustering algorithm will first join the rules r_1 and r_2 and then include rule r_4 in the same cluster, the representative created with the use of the threshold method (with a k parameter set to value 50%) is $\mathrm{Profile}(r_1, r_2, r_4) = \{(a, a_1), (a, a_2), (a, a_3), (c, c_2)\}$. Undeniable advantages of approximation of sets based on the rough set theory can be found in numerous papers such as [16–18]. The rough set is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximations. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest,

whereas the upper approximation is a description of the objects which possibly belong to the subset. Using the notions of lower and upper set approximation, a representative is created with the use of the lower/upper approximation method. The lower approximation method defines a cluster's representative as all pairs (attribute, value) which appear in the conditional part of each rule in the analysed cluster. Conversely, a cluster's representative designated with the upper approximation method shall contain all such pairs (attribute, value) which have appeared in the conditional part of at least one rule in the cluster. The definition of a lower approximation for a group's profile R_q is as follows:

$$\text{LowerApp}(R_q) = \cup\left\{p_s : \underset{r_i \in R_q}{\wedge} p_s \in \text{cond}(r_i)\right\}, \quad (4)$$

and an analogical definition for the upper approximation method is

$$\text{UpperApp}(R_q) = \cup\left\{p_s : \underset{r_i \in R_q}{\vee} p_s \in \text{cond}(r_i)\right\}, \quad (5)$$

where $\text{cond}(r_i)$ means the conditional part of the $r_i - \text{th}$ rule, and p_s is a single premise in this rule r_i. The representative for rule cluster r_1, r_2, and r_4 using the lower approximation-based method regrettably contains an empty set, while using the upper approximation-based approach it contains the following features: $\text{Profile}(r_1, r_2, r_4) = \{(a, a_1), (a, a_2), (a, a_3), (c, c_2)\}$. It is imprecise as it contains the features which cover less that 30% of the rules in a given group.

Hence, it seems justifiable to control the level of coverage of features selected for group representatives. It has led to an alternative way of creating cluster representatives, namely, the weighted representative method. In this method, giving weight (expressed as k%), a representative is created from all pairs (attribute, value) which have appeared at least in k% of rules in a given group.

$$\text{Weighted}(R_q) = \cup\left\{p_s : \underset{r_i \in R_q}{\vee} p_s \in \text{cond}(r_i) \& \text{frequency}((p_s) \geq k\right\}. \quad (6)$$

The representative of a group of rules r_1, r_2, and r_4 selected with the use of this approach (with a value of the k parameter set at 50%) is $\text{Profile}(r_1, r_2, r_4) = \{(c, c_2)\}$ because only this particular premise appears in at least 50% of the rules in this group. This clearly shows the difference between the threshold and weighted approach. It must be emphasized that representatives of clusters are created promptly with clusters of rules, and as a result, there might be empty/blank representatives even though a cluster has been created. This happens when the representative designation method is too restrictive (capture conditions for some features in a representative are relatively high and difficult to fulfil) and simultaneously a stop condition has not been reached as the created structure still has more groups than the assumed threshold and the groups are continuously clustered. Such restrictive requirements are the traits of the lower approximation method. This method stipulates that a feature included in a representative's description is concurrently a common feature of all rules that constitute a cluster. This condition is usually too difficult to fulfil, especially when rules in a knowledge base are short and rarely have common premises. In consequence, at some stage (when groups are clustered into groups at a higher level of hierarchy), there are clusters without representatives. Such structures have to be avoided as they hinder a review of such group and making use of clustering as a tool in the exploration of knowledge bases. An excessive reduction of the conditions examined in the course of designation of representatives makes them too detailed and often inadequate for the described clusters. For instance, using the upper approximation method or setting up too low a threshold for the designation of representatives in the weighted or threshold representative methods (e.g., a 25% threshold) for a cluster of four rules, when a given feature is included as a premise in at least one rule, it is sufficient to be included in the cluster's representative.

4.5. Inference Process in a Hierarchical Knowledge Base. At the core of big data analytics is data science (deep knowledge discovery through data inference and exploration). A knowledge representation requires some process that, given a description of a situation, can use the knowledge to make conclusions. When the knowledge is properly represented, the inference reaches appropriate conclusions in a timely fashion. Thus, the knowledge must be adapted to the inference strategy to ensure that certain inferences are made from the knowledge. Inference in classic knowledge bases matches the entire set of rules to the known facts to deduce new facts. It is impossible to work on the entire set of rules and facts in case of big knowledge bases. Therefore, in this and previous research tasks [9], the author defines the model of the hierarchical knowledge base with rule clusters and rule clusters' representatives.

Inference in a hierarchical knowledge base involves using hierarchy properties to optimize the search of clusters of rules. The results of inference and the course of the inference process itself depend strongly on the goal of inference.

When considering the forward inference (data-driven), we need to take into account the inference with a given hypothesis to prove or without it. In the first case, we review the representatives of clusters of rules at each level and eventually select the rule or rule cluster most relevant to the given facts. If a selected rule can be activated, the result leads to the addition of a new fact to the knowledge base. When this new fact is simultaneously the goal of the inference, the process should end successfully. When the goal of the inference is not specified, we proceed as long as there are any rules that can be activated. Thus, as a result, the implemented inference algorithm leads to the exploration of a number of new facts, and one of the measures of inference efficiency is, among others, a percentage of new facts compared to the ones given at the beginning. The more new facts, the more effective the reasoning process is.

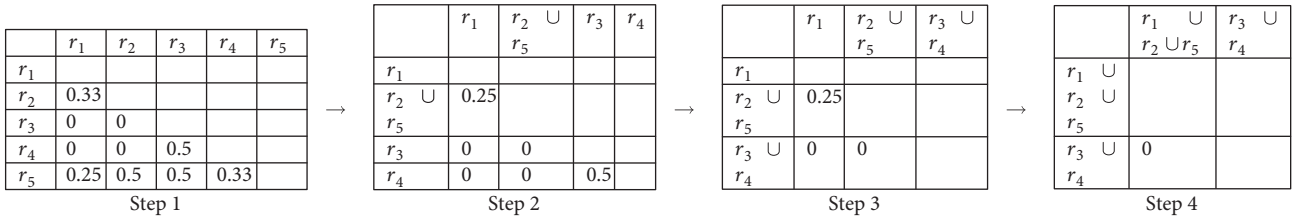


Figure 2: The course of the AHC clustering algorithm for a given knowledge base.

In the classic approach, premises of each rule are examined to see whether they match the set of facts. If they do, the rule is activated and its conclusion is added to the set of facts. If this new fact is a given hypothesis to be proved, the process ends successfully. If there is no given goal of inference, the process is repeated until there is at least one rule to be activated.

In the approach proposed in this research, only representatives of the created rule clusters are analysed, which significantly shortens the time of inference. Usually, the number of the created rule clusters is significantly smaller than the number of rules being clustered. However, the success of the inference process depends on the quality clustering and the approach to creating the representatives. For the structure of K clusters with their representatives, the inference process looks as follows. For the given set of input facts, we are looking at the representative clusters from the highest level in the created hierarchical structure, and at every level of the hierarchy, going from the root to the leaves, we choose the cluster most relevant to the facts. If the selected group is already a single rule, and all its premises match a given set of facts, then the rule is activated and its conclusion is added as a new fact to the knowledge base. If the new fact is simultaneously a given goal to be proved, the inference process is successful. Otherwise, the search process continues until the requested goal of the inference is confirmed or there are any rules to activate. It is easy to see that in the most optimistic case the process lasts only one iteration, during which one rule is activated and its conclusion matches the given goal of inference which ends the process successfully. Of course, the inference process succeeds also if the given hypothesis is proved in more than one iteration, or if any rule was activated (when no hypothesis was specified). For this reason, in the experimental stage, the author examined the following cases: was the goal specified, was it achievable, and was it eventually achieved? It was additionally examined whether any rule had been activated, how many rule clusters had been searched, and if an empty representative had occured during the searching process.

Verification of the correctness of the proposed solution consists of comparing the result of the inference for a hierarchical knowledge base with rule clusters with the result obtained for a classic knowledge base (without rule clusters) and classic inference (analyzing all the rules one by one). In the course of verification, it was checked how frequently the specified goal of inference had been confirmed or any new knowledge had been deduced from the rules and facts.

The pseudocode of the data-driven inference algorithm for rule clusters is presented as Algorithm 2 in Figure 1. The most important procedure is the one which makes it possible to find the most relevant (to the set F) rule cluster first and then the most relevant rule in the selected group. For each cluster R_i, its representative $\mathrm{Profile}(R_i)$ is compared to the set of facts F, and as a result, a group with the maximum similarity is selected ($i = 1, 2, \ldots, K$). The review time needed in the classic approach to search every rule is reduced to the time needed to search cluster representatives. Most of the time, K (number of clusters) is significantly smaller than N (number of rules). The selected rule is activated, and the inference process is finished successfully if the new fact is a requested goal of inference. If not, the process is continued.

4.6. Analysis of the Proposed Idea. For a structure containing about a thousand clusters of rules, about a dozen or so representatives will be compared to find the group which is most similar to the given information. Due to the logarithmic computational complexity of the algorithm, the more rules we group, the greater the time gain from browsing the cluster structure is. This is undoubtedly the biggest advantage of using this approach. Especially with big data sets, such solutions are particularly useful. The disadvantage may be the omission of other rules relevant to the given facts. This approach is more optimal in relation to the approach presented in the author's previous research [9, 10]. The optimization arises from the fact that if, at a given level of analysed structure of rule clusters, the group selected as more relevant contains other clusters (which means additional subsequent searches), we check if the other cluster (omitted at this level, less relevant) is not a single rule. If that is the case, and the premises of this rule match the facts, such rule is activated and makes it possible to finish the inference process earlier.

4.7. Example of Rule Clustering and the Inference Process for Rule Clusters. Let us assume that a given knowledge base contains five rules:

$$\begin{aligned} r_1 &: (a, a_1) \wedge (b, b_1) \wedge (c, c_1) \longrightarrow (dec, A), \\ r_2 &: (a, a_1) \longrightarrow (dec, B), \\ r_3 &: (d, d_1) \longrightarrow (dec, C), \\ r_4 &: (d, d_1) \wedge (e, e_1) \longrightarrow (dec, C), \\ r_5 &: (a, a_1) \wedge (d, d_1) \longrightarrow (dec, B). \end{aligned} \tag{7}$$

The course of the AHC clustering algorithm for this knowledge base, in case of using the wSMC similarity measure, is presented in Figure 2.

Table 1: The course of knowledge exploration for an example of knowledge base.

Step	LowerApp(R_i)	UpperApp(R_i)	Threshold(R_i)/Weighted(R_i)
Representative generation	$R_1 = \{\phi\}$ $R_2 = \{(d, d_1)\}$	$R_1 = \{(a, a_1), (a, a_2), (b, b_1), (c, c_1)\}$ $R_2 = \{(d, d_1), (e, e_1)\}$	$R_1 = \{(a, a_1), (b, b_1)\}$ $R_2 = \{(d, d_1), (e, e_1)\}$
Similarity between F and Profiles	$\mathrm{Sim}(F, R_1) = 0$ $\mathrm{Sim}(F, R_2) = 0$	$\mathrm{Sim}(F, R_1) = 0.25$ $\mathrm{Sim}(F, R_2) = 0$	$\mathrm{Sim}(F, R_1) = 0.5$ $\mathrm{Sim}(F, R_2) = 0$
Choosing the most relevant group	ϕ	R_1	R_1
Finding rule for activation	ϕ	$\mathrm{Sim}(F, r_1) = 0.33$ $\mathrm{Sim}(F, r_2) = 1$ $\mathrm{Sim}(F, r_3) = 0$	$\mathrm{Sim}(F, r_1) = 0.33$ $\mathrm{Sim}(F, r_2) = 1$ $\mathrm{Sim}(F, r_3) = 0$
Activated rule	ϕ	r_2	r_2
New facts	ϕ	(dec, B)	(dec, B)

Table 2: Inference efficiency vs. representative generation methods.

Representative generation method	New knowledge		Goal not achieved[a]	Goal achieved
	Less than 100%	At least 100%		
Threshold	23,145 (48.71%)	24,375 (51.29%)	40,657 (85.56%)	6863 (14.44%)
LowerApp	5692 (47.91%)	6188 (52.09%)	10,459 (88.04%)	1421 (11.96%)
UpperApp	6377 (53.68%)	5503 (46.32%)	9277 (78.09%)	2603 (21.91%)
Weighted	22,901 (48.19%)	24,619 (51.81%)	41,036 (86.36%)	6484 (13.64%)

[a]Empty representative found during inference.

As a result, two clusters of rules are generated: R_1 which contains rules r_3 and r_4 and R_2 which contains r_1, r_2, and r_5. The lower and upper approximation-based representatives for these groups are as follows:

$$\begin{aligned}
&\mathrm{LowerApp}(\mathrm{Profile}(R_1)) = \{(d, d_1)\},\\
&\mathrm{UpperApp}(\mathrm{Profile}(R_1)) = \{(d, d_1), (e, e_1)\},\\
&\mathrm{LowerApp}(\mathrm{Profile}(R_2)) = \{(a, a_1)\},\\
&\mathrm{UpperApp}(\mathrm{Profile}(R_2)) = \{(a, a_1), (b, b_1), (c, c_1), (d, d_1)\},
\end{aligned} \tag{8}$$

and there is also a given input set of facts $F = \{(a, a_1), (b, b_1)\}$. The course of the inference, taking into account the type of representatives, is presented in Table 1.

This basic example clearly illustrates how a representative generation method influences the efficiency of the inference process, producing different results. In case of the LowerApp method, no rule would be activated and no new knowledge would be extracted. When considering big data sets, one should bear in mind that the chosen cluster representation method can significantly affect the amount of new knowledge extracted from the knowledge base of hundreds or thousands of rules. The lower approximation method (producing general descriptions for rule clusters) unfortunately can make the process of discovering new knowledge from rules and facts impossible (because of empty representatives).

5. Experiments

The experiments were aimed at investigating whether the presented clustering methods (SL, CL, AL, and CoL) and representative generation methods (Threshold, LowerApp, UpperApp, and Weighted) influence the efficiency of inference and the quality of created rule clusters. The subjects of the experiments are four datasets: *heart*, *libra*, *weather*, and *krukenberg*, with various numbers of attributes and rules [14]. The smallest knowledge base contains 5 attributes and 5 rules and the greatest number of rules is two hundred, while the greatest number of attributes is 165 elements. In the experiments, many possible combinations were examined for each knowledge base: nine similarity measures, four clustering methods, and four representative generation methods with three various percentage thresholds and various numbers of clusters. The total number of experiment equals 178,200, and it results from the necessity of using all possible combinations of different similarity measures, clustering methods, cluster number, representative generation methods (with various values of threshold k), and the additional parameters related to the inference process such as a different number of facts and the cases with a given hypothesis to be proved or without any hypothesis. All tables summarize the results obtained for the whole 178,200 of the experiments performed.

Tables 2–4 present the results of the analysis of the influence of using various methods for representatives of rule clusters on the inference efficiency.

Table 3: The quality of rules clusters vs. representative generation methods.

Representative generation Method	BCS		O		ARL			BRL		
	Mean	SD	Mean	SD	Mean	SD	Min–Max	Mean	SD	Min–Max
Threshold	76.68	59.18	3.93	5.66	4.05	3.08	0.0–9.75	5.85	3.63	0.0–14.0
LowerApp	78.46	60.45	3.70	5.43	1.39	0.60	0.6–3.75	2.71	1.90	1.0–9.0
UpperApp	80.94	61.34	3.97	5.98	25.94	37.60	2.2–279.0	86.79	94.28	4.0–279.0
Weighted	77.72	59.21	3.85	5.57	4.13	3.59	0.0–14.5	6.83	6.83	0.0–19.0

Table 4: Description of inference efficiency vs. representative generation methods.

Representative generation method	Fired rules		Empty representative		New facts		Searched clusters	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Threshold	5.31	21.69	0.0	0.0	0.92	1.95	54.53	102.35
LowerApp	5.65	23.31	71.13	60.55	0.79	1.67	62.05	111.50
UpperApp	11.53	31.68	0.0	0.0	1.32	2.63	95.14	121.03
Weighted	4.64	19.93	30.45	48.67	0.82	1.69	52.52	101.17

Table 5: Inference efficiency vs. clustering methods.

Clustering method	New knowledge		Goal not achieved[a]	Goal achieved
	Less than 100%	At least 100%		
SL	14,721 (49.57%)	14,979 (50.43%)	24,941 (83.98%)	4759 (16.02%)
CL	14,182 (47.75%)	15,518 (52.25%)	25,122 (84.59%)	4578 (15.41%)
AL	14,517 (48.88%)	15,183 (51.12%)	25,795 (86.85%)	3905 (13.15%)
CoL	14,695 (49.48%)	15,005 (50.52%)	25,571 (86.10%)	4129 (13.90%)

[a]Empty representative found during inference.

Table 2 presents the frequency of finishing the inference successfully (the goal of the inference has been reached or/and any new fact was induced from rules and facts already known) and the frequency of exploration of at least 100% of new knowledge (new facts) in accordance with the input knowledge. Table 3 presents a description of created clusters dependent on different representative generation methods in the form of the following factors: BCS (biggest cluster's size), O (the number of outliers), and ARL/BRL (average/biggest representative's length). Table 4 contains a description of inference efficiency presented as an average number of fired rules, an average number of empty representatives, and the average number of new facts as well as the number of the searched clusters. It is easy to observe that the representative generation method which allows confirming a given goal most often is the UpperApp method (in 21.91% of cases while the LowerApp method allows us to confirm the goal only in 11.96% of cases). If we aim to achieve a lot of new facts (new knowledge), then the representative generation method which allows to get the new knowledge exceeding 100% of input knowledge is the LowerApp method (in 52.09% of cases). The *New knowledge* column with the value *At least 100%* corresponds to the case where for a given set of input facts, at least the same number of new facts was generated during the inference process.

The UpperApp method generates the biggest cluster size, the greatest number of outliers, and a much wider range of representatives than it is in case of other representative generation methods. Only for the UpperApp and Threshold representative generation method are empty representatives not generated at all.

Tables 5–7 contain similar information as Tables 2–4 but for various clustering methods.

The SL clustering method makes it possible to confirm a given goal of inference most often. This method also generates the smallest size of the biggest cluster, the smallest number of outliers, and the shortest lengths of the generated representatives for the created rule clusters. The above-mentioned method also yields the smallest number of fired rules, the earliest time of achieving empty representatives, and the smallest number of searched clusters.

6. Conclusions

The decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. Effective exploration of new knowledge in every domain of human life requires new algorithms of knowledge organization and searching of created data structures. Optimization proposed by the author in this paper is based on the cluster analysis method and modification of

Table 6: Quality of rule clusters vs. clustering methods.

Clustering method	BCS		O		ARL		BRL	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SL	50.78	52.60	2.81	5.14	6.24	14.31	11.99	33.11
CL	83.01	64.37	3.14	5.49	6.26	14.06	14.22	40.03
AL	83.56	55.36	4.84	5.52	5.69	13.76	14.70	39.52
CoL	93.46	56.36	4.72	6.06	5.84	13.73	15.18	41.37

Table 7: Description of inference efficiency vs. clustering methods.

Clustering method	Fired rules		Empty representative		New facts		Searched clusters	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SL	4.92	19.08	18.60	41.91	0.95	1.94	43.48	84.77
CL	5.60	22.26	20.06	43.39	0.91	1.88	63.75	110.60
AL	5.30	22.54	19.16	42.62	0.91	2.00	46.31	100.04
CoL	6.98	25.54	19.36	42.64	0.87	1.83	80.62	119.48

the inference algorithm, which searches within representatives of the created rule clusters instead of rules. This article presents both the description of the proposed approach and the results of the experiments carried out for the chosen knowledge bases.

Among various clustering algorithms, the agglomerative hierarchical clustering algorithm was selected with a modification proposed by the author in which rule clusters are built until a given number of clusters is reached. For every rule cluster, a representative is created. During the inference process, only representatives are analysed, and at every level of the created hierarchical structure, the most relevant representative is selected and further analysed. This means it is possible to search only a small part of the whole knowledge base with the same accuracy that would be achieved when the whole knowledge base is searched. During the previous experiments, it was shown that for big knowledge bases (with more than a thousand of rules), only 1.5% of the whole KB has to be analysed to finish the inference process successfully. For every combination of the clustering parameters such as similarity measures, number of clusters, and others—Tables 2–4 present the results of the described and examined methods of the cluster representative generation. Tables 5–7 present the results for four different clustering methods, respectively.

As expected, the UpperApp representative method corresponds with creating the biggest size and the largest representatives of the created clusters. As a result, this method leads to a successful conclusion more frequently. Therefore, it is recommended to consider further analysis of both the representative generation methods and the inference algorithm in order to propose new optimizations and achieve a higher efficiency.

Data Availability

The readers can access the data through the link: http://zsi.tech.us.edu.pl/~nowak/data.rar where original four datasets and four report files generated during the experimental stage are uploaded. The original knowledge bases and associated files with set of facts were used as input data for the CluVis software (implemented by the author) to build a hierarchical structure of every knowledge base and then to run the inference process. The results are report CSV-type files with inference efficiency measures such as factors calculated during the experiments.

Conflicts of Interest

The author declares that she has no conflicts of interest.

References

[1] http://news.mit.edu/2014/big-fast-weird-data.

[2] https://www.cnbc.com/2014/02/12/inside-the-wacky-world-of-weird-data-whats-getting-crunched.html.

[3] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, 2016.

[4] K. M. Wiig, *Expert Systems: a Manager's Guide*, International Labour Office, Geneva, Switzerland, 1990.

[5] R. Simiński and A. Nowak-Brzezińska, "Goal-driven inference for web knowledge based system," in *Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part IV*, vol. 432 of *Advances in Intelligent Systems and Computing*, pp. 99–109, Karpacz, Poland, 2015.

[6] T. Breidenstein, I. Bournaud, and F. Woliński, "Knowledge discovery in rule bases," in *Knowledge Acquisition, Modeling and Management*, vol. 1319, Lecture Notes in Computer Science, pp. 329–334, Springer, 1997.

[7] A. Hashizume, B. Yongguang, X. Du, and N. Ishii, "Generating representative from clusters of association rules on numeric attributes," in *Intelligent Data Engineering and Automated Learning*, vol. 2690, Lecture Notes in Computer Science, pp. 605–613, Springer, 2003.

[8] F. Ye, J. Wang, S. Wu, H. Chen, T. Huang, and L. Tao, "An integrated approach for mining meta-rules," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 3587

of Lecture Notes in Computer Science, , pp. 549–557, Springer, 2005.

[9] A. Nowak, A. Wakulicz-Deja, and S. Bachliński, "Optimization of speech recognition by clustering of phones," *Fundamenta Informaticae*, vol. 72, no. 1–3, pp. 283–293, 2006.

[10] A. Nowak and A. Wakulicz-Deja, "The concept of the hierarchical clustering algorithms for rules based systems," in *Intelligent Information Processing and Web Mining*, pp. 565–570, Springer, 2005.

[11] A. Nowak-Brzezińska, "Mining rule-based knowledge bases inspired by rough set theory," *Fundamenta Informaticae*, vol. 148, no. 1-2, pp. 35–50, 2016.

[12] Ł. Wróbel, M. Sikora, and M. Michalak, "Rule quality measures settings in classification, regression and survival rule induction — an empirical approach," *Fundamenta Informaticae*, vol. 149, no. 4, pp. 419–449, 2016.

[13] J. Stefanowski, "On rough set based approaches to induction of decision rules," in *Rough Sets in Data Mining and Knowledge Discovery*, L. Polkowski and A. Skowron, Eds., pp. 500–529, Physica, Verlag, Heidelberg, 1998.

[14] M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Sciences, Irvine, CA, USA, 2013.

[15] J. W. Grzymala-Busse, "Rule induction," in *Data Mining and Knowledge Discovery Handbook*, pp. 249–265, Springer, Boston, MA, USA, 2nd edition, 2010.

[16] R. Slowinski, S. Greco, and B. Matarazzo, "Rough sets in decision making," in *Encyclopedia of Complexity and Systems Science*, pp. 7753–7787, Springer, 2009.

[17] A. Skowron, "Extracting laws from decision tables: a rough set approach," *Computational Intelligence*, vol. 11, no. 2, pp. 371–388, 1995.

[18] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets," *Communications of the ACM*, vol. 38, no. 11, pp. 88–95, 1995.

[19] J. G. Bazan, M. S. Szczuka, and J. Wróblewski, "A new version of rough set exploration system," in *Rough Sets and Current Trends in Computing*, pp. 397–404, Springer, 2002.

[20] R. Detrano, A. Janosi, W. Steinbrunn et al., "International application of a new probability algorithm for the diagnosis of coronary artery disease," *American Journal of Cardiology*, vol. 64, no. 5, pp. 304–310, 1989.

[21] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., 1988.

[22] B. G. Buchanan and E. H. Shortliffe, *Rule Based Expert Systems: the Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley Series in Artificial Intelligence)*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[23] C. L. Forgy, "Rete: a fast algorithm for the many pattern/many object pattern match problem," in *Expert Systems*, pp. 324–341, IEEE Computer Society Press, 1990.

[24] W. K. Loh and Y. H. Park, "A survey on density-based clustering algorithms," in *Ubiquitous Information Technologies and Applications*, Y. S. Jeong, Y. H. Park, C. H. Hsu, and J. J. Park, Eds., vol. 280 of Lecture Notes in Electrical Engineering, pp. 775–780, Springer, Berlin, Heidelberg, 2014.

[25] H. K. Kanagala and V. V. Jaya Rama Krishnaiah, "A comparative study of K-means, DBSCAN and OPTICS," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, Coimbatore, India, 2016.

[26] A. Dudek, "A comparison of the performance of clustering methods using spectral approach," in *Data Analysis Methods and Its Applications*, pp. 143–156, C.H. Beck, Warszawa, Poland, 2012.

[27] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: a comparative evaluation," in *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 243–254, Atlanta, GA, USA, 2008.

[28] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, p. 857, 1971.

[29] A. Nowak-Brzezińska and T. Rybotycki, "Comparison of similarity measures in context of rules clustering," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 235–240, Gdynia, Poland, 2017, IEEE Conference Publications.

[30] A. Nowak-Brzezińska and T. Rybotycki, "Impact of clustering parameters on the efficiency of the knowledge mining process in rule-based knowledge bases," *Schedae Informaticae*, vol. 25, pp. 85–101, 2017.

[31] Y. Jung, H. Park, D. Z. Du, and B. L. Drake, "A decision criterion for the optimal number of clusters in hierarchical clustering," *Journal of Global Optimization*, vol. 25, no. 1, pp. 91–111, 2003.

[32] S. Still and W. Bialek, "How many clusters? An information-theoretic perspective," *Neural Computation*, vol. 16, no. 12, pp. 2483–2506, 2004.

[33] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: the Concepts and Technology Behind Search*, Addison Wesley, 2011.

[34] J. J. Rocchio, *Document Retrieval systems – Optimization and Evaluation, [Ph.D. Thesis]*, Harvard University, 1966.

Hindawi
Complexity
Volume 2018, Article ID 9837462, 10 pages
https://doi.org/10.1155/2018/9837462

Research Article

Stability Analysis of the Bat Algorithm Described as a Stochastic Discrete-Time State-Space System

Janusz Piotr Paplinski

Department of Computer Architectures and Teleinformatics, West Pomeranian University of Technology Szczecin, ul. Zolnierska 52, 71-210 Szczecin, Poland

Correspondence should be addressed to Janusz Piotr Paplinski; janusz.paplinski@zut.edu.pl

Received 19 April 2018; Revised 19 September 2018; Accepted 1 October 2018; Published 3 December 2018

Guest Editor: Ireneusz Czarnowski

The main problem with the soft-computing algorithms is a determination of their parameters. The tuning rules are very general and need experiments during a trial and error method. The equations describing the bat algorithm have the form of difference equations, and the algorithm can be treated as a stochastic discrete-time system. The behaviour of this system depends on its dynamic and preservation stability conditions. The paper presents the stability analysis of the bat algorithm described as a stochastic discrete-time state-space system. The observability and controllability analyses were made in order to verify the correctness of the model describing the dynamic of BA. Sufficient conditions for stability are derived based on the Lyapunov stability theory. They indicate the recommended areas of the location of the parameters. The analysis of the position of eigenvalues of the state matrix shows how the different values of parameters affect the behaviour of the algorithm. They indicate the recommended area of the location of the parameters. Simulation results confirm the theory-based analysis.

1. Introduction

In recent years, the nature-inspired metaheuristic algorithms for optimization problems become very popular. Though these algorithms do not guarantee the optimal solution, they generally have a tendency to find a good solution and become powerful methods for solving many difficult optimization problems [1–3]. The heuristic methods are based on the many different mechanisms occurring in nature. The genetic algorithms [4] are based on the biological fundamentals, tabu search is based on the social behaviour [5], and ant colony optimization [6, 7] or particle swarm optimization (PSO) [8] is based on the swarm behaviour. The bat algorithm (BA) proposed by Yang [9] belongs to the last. The bats use some type of sonar, called echolocation, to detect prey or to avoid obstacles. The echolocation guides their search and allows discrimination of different types of insects, even in complete darkness.

There are some much powerful modifications of BA, for instance, BA based on differential operator and Lévy flight trajectory (DLBA) proposed by Xie et al. [10], the improved bat algorithm (IBA) proposed by Yilmaz and Küçüksille [11], and enhanced bat algorithm proposed also by Yilmaz and Küçüksille [12], but the simple BA is a base and is more popular than other modifications. For this reason, the analysis for simple BA is made in the paper.

The main problem with the soft-computing algorithms is a determination of their parameters. The tuning rules are very general and need experiments during a trial and error method. The main idea in this process is to balance the running algorithm between exploration and exploitation. In the case of too little exploration and intensive exploitation, the algorithm can converge to a local optimum. Otherwise, too much exploration and too little exploitation can give the algorithm with a very small convergence [13–15].

The behaviour of the algorithm and ability to converge to the global optimum depend on its dynamic, which is described by difference equations. This behaviour depends on the stability works of algorithms especially. There are some stability analyses of PSO algorithm based on the location of the roots

[16–18] and the Lyapunov function [19–21]. We made a similar study for BA in [22].

The present paper is the extended version of the conference paper [22] published in INnovations in Intelligent SysTems and Applications (INISTA) 2017. The way of the approach presented in the paper is largely new and much more general. New in the paper is an extension of the dynamic description of the bat algorithm to the third order by taking into account all variables used by individuals in the population. Thereby, the description of BA becomes more complete and general. Then the use of a description in the form of the state-space and checking the controllability and observability allow reducing the order of dynamic. The method of stability analysis proposed in [22] is based on the location of the roots of the difference equation which is appropriate for linear time-invariant systems. This method was used after omitting the randomness of parameters and treatment of the algorithm as stationary. In the present paper, the new approach of the stability analysis of the dynamics of BA by using the Lyapunov stability theorem and Sylvester's criterion is made. It provides to obtain the desired range of parameters providing stability work of the algorithm. The Lyapunov theory defines the necessary and sufficient conditions for absolute stability of nonlinear and/or time-varying systems. Any simplifications are not needed during stability analysis, which is important because the obtained results are more reliable than the results from the previous work [22]. Both methods give the similar solution, then the present paper confirms the correctness of simplicity used in preceding work [22]. As an illustrative example, the same four benchmark functions were used, but the graphs were made for not presented earlier Griewank function.

The paper is organized as follows. In Section 2, the BA algorithm is described. The next section details the Lyapunov stability theory. After this, the dynamic of BA is described and analysed with a special focus on the stability condition. The example experiments and discussion are presented in the last section.

2. Bat Algorithm

The bats have fascinating abilities such as finding prey and discriminating different types of insects even in complete darkness. Bats use echolocation by emitting high-frequency audio signals and receiving a reflection of those. The time delay between emission and detection of the echo and its variation of loudness allow bats to recognize surroundings.

The metaheuristic BA uses some simplicity and idealized rules:

(1) All bats use echolocation to appoint a distance and direction to the food. They also can recognize the difference between food/prey and background barriers

(2) The i-th bat is at position x_i and flies randomly with a velocity v_i. It emits an audio signal with a variable frequency between $[f_{\min}, f_{\max}]$, a varying wavelength λ_i, and loudness A_i to search for food. It can automatically adjust the wavelength (or frequency) of its emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of its target

(3) The loudness can vary in many ways. We assume that the loudness varies from a large (positive) L_0 to the smallest constant value $L_{\min}$

Each of the artificial bats in the k-th step has a position vector x_i^k, velocity vector v_i^k, and frequency vector f_i^k which is updated during iterations by using the below relations, from (1) to (3). The position vector of the bat represents some specific solution of the optimization problem. Every bat emits an audio signal with a randomly assigned frequency f_i^k, which is drawn uniformly from the range $[f_{\min}, f_{\max}]$:

$$f_i^k = f_{\min} + (f_{\max} - f_{\min})\beta^k, \tag{1}$$

where $\beta^k \in [0, 1]$ is a random vector with a uniform distribution. The velocity of the i-th bat in the k-th step v_i^k depends on the position of the current global best solution achieved so far x_{bsf}^k:

$$v_i^k = v_i^{k-1} + \left(x_i^{k-1} - x_{\text{bsf}}^k\right) f_i^k. \tag{2}$$

The new position of the bat and thus the new solution of the problem follow from his earlier position and velocity:

$$x_i^k = x_i^{k-1} + v_i^k. \tag{3}$$

The local search procedure is also used. A new solution for a bat is generated locally using current best solution and local random walk:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon L^k, \tag{4}$$

where $\varepsilon \in [-1, 1]$ is a random number with a uniform distribution, while L^k is the average loudness of all bats at the k-th time step.

We can consider BA as a balanced combination of exploration, realized by an algorithm similar to the standard particle swarm optimization and exploitation realized by an intensive local search. The balance between these techniques is controlled by the loudness L and emission rate r, updated as follows:

$$L_i^{k+1} = \alpha L_i^k, \tag{5}$$

$$r_i^{k+1} = r_i^0 (1 - \exp(-\gamma k)), \tag{6}$$

where the coefficients α and γ are constants. In the simplification case, $\alpha = \gamma$ is often used. We can consider the parameter α as similar to the cooling factor in simulated annealing. The loudness and the pulse emission rate are updated only

1. Initialize the bat population $x_i^0 (i = 1, 2 \ldots, n)$ and v_i^0
2. Initialize pulse frequency f_i^0, pulse rates r_i^0, and the loudness L_i^0
3. While (the stop condition is not fulfilled)
4. Generate new solutions by adjusting frequency,
5. Updating velocities and locations (eq. (2,3))
6. If $(rand > r_i^k)$
7. Select a solution among the best solutions
8. Generate a local solution (eq. (4))
9. End if
10. Generate a new solution by flying randomly
11. If $(rand < L_i^k \,\&\, f(x_i^k) < f(x_{bsf}^k)$
12. Accept the new solutions
13. Reduce L_i^{k+1} and Increase r_i^{k+1} (eq. (5,6))
14. End if
15. Rank bats and find current best x_{bsf}^k
16. End while

Pseudocode 1: The pseudocode of BA.

when the new solution is improved. It means when the bat is moving toward the best solution.

The operation of BA can be described as follows. In the beginning, the metaheuristic BA initializes a population of bats, by assigning to individuals values of its parameters. They are most commonly defined randomly. Every bat will move from initial solutions toward the global best solution with each iteration using the position of the current global best solution attained so far. If any bat finds a better solution after moving, the best so far solution, pulse emission and loudness are updated. This process is repeated continuously until the termination criteria are satisfied. The best so far solution achieved is considered the final best solution. The pseudocode of BA is presented in Pseudocode 1.

3. Lyapunov Stability Theory

The basic theorems for system stability are the Lyapunov stability theorems [23–25]. The second Lyapunov criterion (direct method) allows proving the local and global stabilities of an equilibrium point using proper scalar functions, called Lyapunov functions, defined in the state-space. This criterion refers to particular "positive definite" or "positive semidefinite" scalar functions, which often have the meaning of "energy functions." By looking at how this energy-like function changes over time, we might conclude that a system is stable or asymptotically stable without solving the nonlinear differential equation. The necessary and sufficient condition for the Lyapunov function, described by the matrix, to be positive definite is determined by Sylvester's criterion.

Theorem 1 (see [26]). *Consider the equilibrium point $x = 0$ of the stochastic discrete-time system, defined by the state-space equation:*

$$x^{k+1} = A^k x^k, \tag{7}$$

where x^k is a state vector at time k and $A^k \in R^{n \times n}$ is a nonsingular matrix with stochastic values. The equilibrium point is asymptotically stable if there is a nonnegative scalar Lyapunov function $V(x^k)$ defined as

$$V\left(x^k\right) = \left(x^k\right)^T P x^k, \tag{8}$$

where P is a positive definite symmetric matrix, with $V(0) = 0$, which satisfies that the expected value of changes of the Lyapunov function $E(\Delta V)$ is greater than zero:

$$E(\Delta V) = E\left(V\left(x^{k+1}\right) - V\left(x^k\right)\right) < 0. \tag{9}$$

We can write it as

$$E(\Delta V) = E\left(\left(x^{k+1}\right)^T P x^{k+1} - \left(x^k\right)^T P x^k\right) < 0, \tag{10}$$

which after simple transformation, using (7) and (10), leads to a useful formula:

$$E(\Delta V) = E\left(\left(x^k\right)^T \left(A^T P A - P\right) x^k\right) < 0. \tag{11}$$

Remark 1. The stochastic discrete-time system (7) is asymptotically stable if and only if for any positive definite matrix Q there exists a positive definite symmetric matrix P that satisfies the Lyapunov equation [27, 28]:

$$E\left(A^T P A - P\right) = -Q. \tag{12}$$

The symmetric matrix P is positive definite if it fulfils Sylvester's criterion.

Theorem 2 (see [29]). *The necessary and sufficient condition for the matrix to be positive definite is that the determinants of all the successive principal minors of the matrix are positive.*

For the symmetric matrix P defined as

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}, \tag{13}$$

where $P = P^T$; the successive principal minors must be positive:

$$p_{11} > 0, \begin{vmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{vmatrix} > 0, \ldots, \begin{vmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{vmatrix} > 0. \tag{14}$$

4. Analysis of the Bat Algorithm as a Dynamic System

By assuming the velocity of the bat v_i^k, its position x_i^k and loudness L_i^k, as state variables $\xi^k = \begin{bmatrix} v_i^k & x_i^k & L_i^k \end{bmatrix}^T$ and the position of the current global best solution achieved so far x_{bsf}^k as an input u^k, the dynamics of the BA described by (2), (3), and (5) can be presented in the state-space form:

$$\begin{bmatrix} v_i^{k+1} \\ x_i^{k+1} \\ L_i^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & f_i^k & 0 \\ 1 & 1+f_i^k & \varepsilon \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} v_i^k \\ x_i^k \\ L_i^k \end{bmatrix} + \begin{bmatrix} -f_i^k \\ -f_i^k \\ 0 \end{bmatrix} x_{\text{bsf}}^k,$$
$$y_i^k = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_i^k \\ x_i^k \\ L_i^k \end{bmatrix}, \tag{15}$$

where the locus of the i-th bat is treated as an output y_i^k.

Equation (6) describing the emission rate affects only on the control of the algorithm in step 6 of its pseudocode (Pseudocode 1). For some random iteration, the local search around the best individual is made. It has no influence on the trajectory of individuals and was omitted in the description of the dynamics of BA in relations (15).

Each bat, the i-th dimension of (15), updates independently from the others; thus, without losing the generality, the analysis of the algorithm can be reduced to the one-dimensional case. Therefore, consequently to the general form of the state-space:

$$\xi^{k+1} = A^k \xi^k + B u^k,$$
$$y^k = C\xi^k + D, \tag{16}$$

and a one-dimensional case of (15), we obtain the state matrix A, the input matrix B, the output matrix C, and the feedforward matrix D:

$$A^k = \begin{bmatrix} 1 & f^k & 0 \\ 1 & 1+f^k & \varepsilon \\ 0 & 0 & a \end{bmatrix},$$
$$B = \begin{bmatrix} -f^k \\ -f^k \\ 0 \end{bmatrix}, \tag{17}$$
$$C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},$$
$$D = 0.$$

For the dynamic system described by the state-space form, the analyses of its observability and controllability are essential. Without losing the generality of this analysis, we can assume the constant value of frequency f^k equals its expected value $f^k = E(f^k) = f$.

Theorem 3 (see [30]). *The system is completely observable if any initial state vector $x(t_0)$ can be reconstructed by examining the output of the system $y(t)$ over the finite period of time from t_0 to t_f. Then the system is completely observable if and only if the set of vectors $\begin{bmatrix} C & CA & \ldots & CA^{n-1} \end{bmatrix}^T$ is literary independent, i.e.,* rank $\begin{bmatrix} C & CA & \ldots & CA^{n-1} \end{bmatrix}^T = n$, *where n is a size of state-space vector $\xi^k \in R^{n\times 1}$.*

The rank of the matrix of observability for BA dynamics (15) is equal:

$$\text{rank} \begin{bmatrix} CCACA^2 \end{bmatrix}^T = \text{rank} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1+f & \varepsilon \\ (1+f)^2+f & 1+2f & \varepsilon(1+f+a) \end{bmatrix} = 3, \tag{18}$$

which means the system is observable.

Theorem 4 (see [23]). *The system is completely controllable if there exists a control signal $u(t)$ defined over a finite interval $0 \le t \le t_f$, which can force system states $x(t_f)$ to any desired value. Then the system is completely controllable if and only if the set of vectors $\begin{bmatrix} B & AB & \ldots & A^{n-1}B \end{bmatrix}$ is literary independent, i.e.,* rank $\begin{bmatrix} B & AB & \ldots & A^{n-1}B \end{bmatrix} = n$.

The rank of the matrix of controllability for BA dynamics described by (15) is equal:

$$\text{rank} \begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} -f & -f^2-f & -f^3-3f^2-f \\ -f & -f^2-2f & -f^3-4f^2-3f \\ 0 & 0 & 0 \end{bmatrix} = 2. \tag{19}$$

The obtained rank of matrix A equals 2 and is less than the size of the state-space vector $n = 3$. The system (15) is uncontrollable. It results that the order of the system is reduced from third to second. We can see it as simplifying the common expression $(z - a)$ from the numerator and denominator while calculating the transfer function describing this system:

$$\begin{aligned} G(z) &= C(zI - A)^{-1}B + D \\ &= \frac{-fz(z-a)}{(z^2 - (2+f)z + 1)(z-a)} \\ &= \frac{-fz}{(z^2 - (2+f)z + 1)}. \end{aligned} \tag{20}$$

The uncontrollable state-variable is the loudness L^k, but if we assume $\alpha \in (0, 1)$, then L^k is decreasing during iteration and can be treated by the system as some disturbance. After leave out the earlier simplification and take into account the variability of the value of frequency f^k and present the state-space form as

$$\begin{bmatrix} v_i^{k+1} \\ x_i^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & f^k \\ 1 & 1 + f^k \end{bmatrix} \begin{bmatrix} v_i^k \\ x_i^k \end{bmatrix} + \begin{bmatrix} -f^k \\ -f^k \end{bmatrix} x_{\text{bsf}} + \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix} L_i^k,$$
$$y_i^k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} v_i^k \\ x_i^k \end{bmatrix}. \tag{21}$$

The description of the system can be modified to the autonomous form by taking new state variables and neglecting disturbances with the equilibrium point at $\xi^k = [0\,0]^T$:

$$\tilde{x}^k = x^k - x_{\text{bsf}}, \tag{22}$$

then state-space take the form

$$\begin{bmatrix} v_i^{k+1} \\ \tilde{x}_i^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & f^k \\ 1 & 1 + f^k \end{bmatrix} \begin{bmatrix} v_i^k \\ \tilde{x}_i^k \end{bmatrix},$$
$$y_i^k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} v_i^k \\ \tilde{x}_i^k \end{bmatrix}. \tag{23}$$

According to Theorem 1, for any positive definite matrix Q, there must exist symmetric positive definite matrix P fulfilling the Lyapunov function (8). We can define matrix Q as

$$Q = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix}, \tag{24}$$

with $c_1, c_2 > 0$. The symmetric matrix P equals

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}. \tag{25}$$

Substituting the matrices P and Q into relation (12), we obtain

$$E\left(\begin{bmatrix} p_1 + 2p_2 + p_3 & f^k p_1 + \left(1 + 2f^k\right)p_2 + \left(1 + f^k\right)p_3 \\ f^k p_1 + \left(1 + 2f^k\right)p_2 + \left(1 + f^k\right)p_3 & \left(f^k\right)^2 p_1 + 2f^k\left(1 + f^k\right)p_2 + \left(1 + f^k\right)^2 p_3 \end{bmatrix}\right) = \begin{bmatrix} -c_1 & 0 \\ 0 & -c_2 \end{bmatrix}, \tag{26}$$

which leads to the system of relations:

$$\begin{aligned} p_1 + 2p_2 + p_3 &= -c_1, \\ E\left(f^k\right)p_1 + \left(1 + 2E\left(f^k\right)\right)p_2 & \\ + \left(1 + E\left(f^k\right)\right)p_3 &= 0, \\ E\left(f^k\right)^2 p_1 + 2E\left(f^k\right)\left(1 + E\left(f^k\right)\right)p_2 & \\ + \left(1 + E\left(f^k\right)\right)^2 p_3 &= -c_2. \end{aligned} \tag{27}$$

Assuming $f = E(f^k)$, we can easily calculate that

$$\begin{aligned} p_2 &= \frac{f}{2}(p_1 - c_1), \\ p_3 &= -fp_1 + (-1 + f)c_1. \end{aligned} \tag{28}$$

Matrix P must be positive definite. According to Sylvester's criterion in Theorem 2, the successive principal minors must be positive, which leads to equations

$$\begin{aligned} p_1 &> 0, \\ p_1 p_3 - {p_2}^2 &> 0, \end{aligned} \tag{29}$$

where

$$\begin{aligned} p_1 p_3 - {p_2}^2 = -\left(\frac{f^2}{4} + f\right){p_1}^2 + \left(\frac{f^2}{2} + f - 1\right)c_1 p_1 & \\ - \frac{f^2}{4}{c_1}^2 > 0. & \end{aligned} \tag{30}$$

Considering the quadratic polynomial $a{p_1}^2 + bp_1 + c$, we obtain

$$a = -\left(\frac{f^2}{4} + f\right),$$
$$b = \left(\frac{f^2}{2} + f - 1\right)c_1, \tag{31}$$
$$c = -\frac{f^2}{4}{c_1}^2 < 0,$$

and well-known relations

$$\Delta = b^2 - 4ac,$$
$$\lambda_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}. \tag{32}$$

We will look for a range of parameter f for which the polynomial (30) is greater than zero. We can divide the range of value of frequency f into some subsets.

(I) $\boldsymbol{a} > 0$

This is met for the mean value of frequency $f \in (-4, 0)$. Assuming the big enough value of $p_{11} > \lambda_{1,2}$ and $p_{11} > 0$, we obtain that BA is stable in this range of f.

(II) $\boldsymbol{a} < 0$

The condition $p_{11} > 0$ can be satisfied only when $\Delta > 0$, which is met for $f < 0.5$. We need to examine two conditions:

(i) $\boldsymbol{f} < -4$

Because of $a < 0$ and $c < 0$, $b^2 > \Delta$, and consequently, $\lambda_{1,2} < 0$. The condition $\lambda_1 > p_{11} > \lambda_2 > 0$ indicate unstable dynamics of BA.

(ii) $0 < \boldsymbol{f} < 0.5$

For this range of f, the value of $b < 0$ and consequently $\lambda_{1,2} < 0$ which indicate unstable dynamics of BA.

Lyapunov theory leads to the conclusion that the dynamic of BA is stable for the frequency belonging to the range $f \in (-4, 0)$.

Theorem 5. *A linear discrete-time system described by the state (16) is asymptotically stable if and only if all eigenvalues of A have magnitude smaller than one. For eigenvalues lying on the unit circle, the system is on the stability border.*

The eigenvalues of A are calculated from the characteristic equation defined as

$$\det(zI - A) = 0, \tag{33}$$

where $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, or they are defined as the roots of the denominator of the transfer function (20) equivalently, which leads to the equation

$$\left(z^2 - (2 + f)z + 1\right) = 0. \tag{34}$$

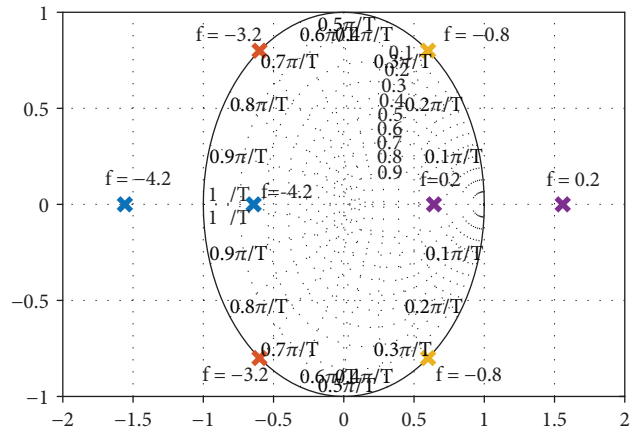


Figure 1: The eigenvalues of the matrix A on the z-plane.

The roots are equal:

$$z_{1,2} = \frac{2 + f \pm \sqrt{f^2 + 4f}}{2}. \tag{35}$$

Figure 1 presents the eigenvalues of matrix A on the z-plane. For $f \in [-4, 0]$, the eigenvalues are complex, and their absolute value equals one $|z_{1,2}| = 1$. In this case, they lie on the stability border, and the algorithm behaves as an undamped oscillator. The samples of the responses of the algorithm are presented in Figures 2(b) and 2(c). The response of the algorithm is periodic and oscillates around the best individuals. For $f < -2$, the ringing occurs, and the unit changes of the position of individuals are big. It is visible in Figure 2(b) as a high oscillation. For $f \in (-2, 0)$ and especially for $f \to 0$, the changes are smaller, and the algorithm systematically scans the solution space. It is visible in Figure 2(c) as intermediate values between the greatest and the smallest values of the response.

For $f < -4$ and $f > 0$, the eigenvalues have only a real part, and at least one of them lies outside the unit circle $|z_{1,2}| > 1$; the algorithm is unstable. The samples of the response of the algorithm for $f = -4.2$ and $f = 0.2$ are presented in Figures 2(a) and 2(d), respectively. For the frequency f smaller than $f < -4$, the response has an oscillatory character with amplitude exponential growing; the faster, the farther from the border frequency $f = -4$. For the positive frequency $f > 0$, the response is aperiodic and growing exponentially, the faster, the farther from the $f > 0$. For already shown points, lying close to the limit values, the value of the amplitude is near 10^3 after only 15 iterations. This causes that the individuals often are going beyond search space, in the case of constrained optimization. The procedure of repair infeasible individuals becomes important. This procedure is not predefined in the algorithm and strongly depends on the designer of the algorithm. The simple replacing by the limit value is often used. It is always a type of heuristic, and it causes that the echolocation does not work correctly. The procedure of mutation of the best individuals dominates in that algorithm.

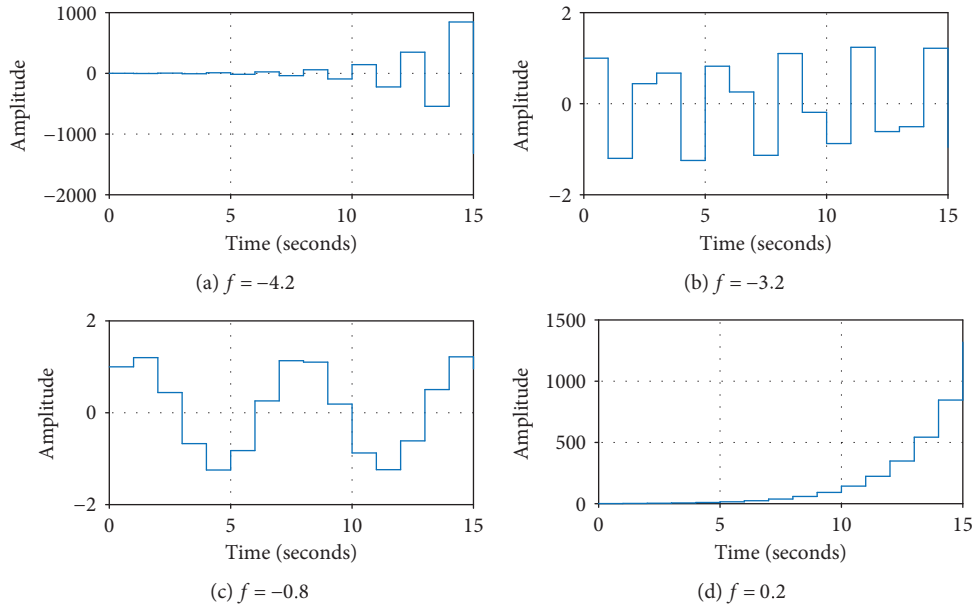


Figure 2: The responses of the algorithm with eigenvalues from Figure 1 to initial conditions.

Table 1: Benchmark functions used for calculating the quality function J and analysing the efficiency of BA with specified parameters (the loudness L and emission rate r) used during experiments.

Function	Search range	Type of function	min	Loudness L	Emission rate r
Sphere	[−5.12–5.12]	Unimod.	0	0.01	0.1
Griewank	[−600–600]	Multim.	0	4	0.1
Ackley	[−32,768–32,768]	Multim.	0	0.7	0.6
Schwefel	[−500–500]	Multim.	0	1.1	0.1

Table 2: The type of constraints of the frequency f as a function of the absolute value B ($B > 0$).

Type of constraints C	Lower bound $f_{\min}$	Upper bound $f_{\max}$
I	$-B$	0
II	$-B$	B
III	0	B

5. The Example Experiments and Discussion

The size of the population of BA was equal $N = 40$; the largest number of function evaluations for each call of BA has been set as $50{\cdot}10^3$. Step 7 of the code of BA in Pseudocode 1 ("Select a solution among the best solutions") is not predefined and can be used any, like a roulette wheel, stochastic universal sampling or other. The first one of the abovementioned methods was used during the experiments. Four benchmark functions, presented in Table 1, were used in order to check and analyse the efficiency of the BA. They are used as a quality function J during searching for global minimum value by BA. The BA was run 50 times for each benchmark function. The limitation of search space was presented in Table 1. This table also includes the values of parameters, the loudness L and emission rate r, used during experiments.

The range of variability of frequency f was divided into three types C presented in Table 2. The frequency was determined by (1) using the lower and upper bounds appropriately for each range and value of parameter B. In the second type of constraints, the mean value of frequency is equal to zero, and the only range of variability of frequency f is changed which influences the variance of the distribution function. The first and third types of constraints lie fully on the stable or unstable region, and both the mean value and variance are changing.

The main influences of frequency f on the stability of algorithms can be seen as a number of individuals out of the boundaries of the seeking area. The problem of crossing the border can exist in both, for the location and speed. The per cent of new individuals with location and velocity outside the permitted area for the Griewank function is presented in Figures 3(a) and 3(b). Similar figures for the Schwefel and the Ackley functions were presented in [22]. A low absolute value of B gives generally less per cent of new individuals crossing the bounds. The number of individuals crossing the bounds for the stable area is smaller even than for the frequency with a mean value equal to zero, i.e., for the type II of the constraints of frequency. In this second case, the variation of the sign of the frequency f results in balancing of positive and negative values of frequency and consequently the lower number of new individuals with speed and position out of the permissible value. Figure 4 presents the mean value of quality $J_{\min}$ as a function of the lower

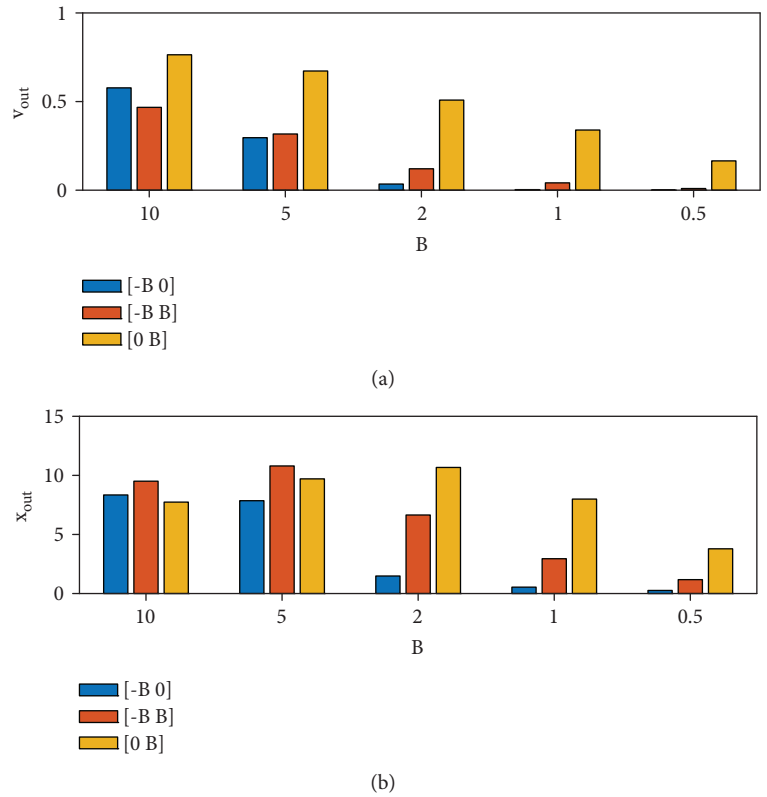


Figure 3: The per cent of new individuals with (a) location and (b) velocity, outside the permitted area for the Griewank function.

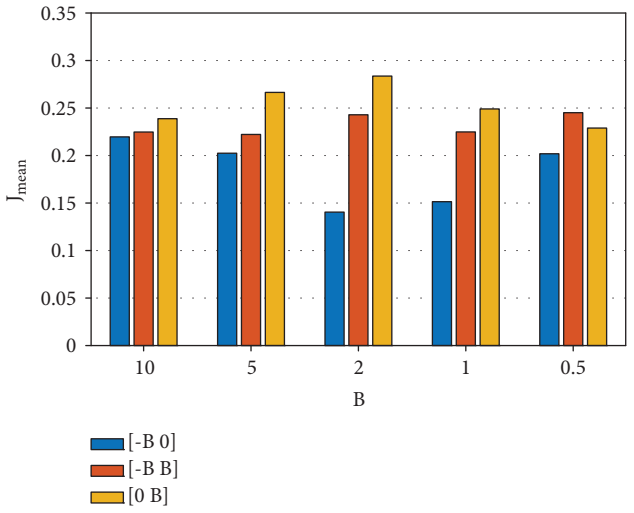


Figure 4: The mean value of the quality J_{mean} of the algorithm as a function of the absolute value of the lower and upper bounds of frequency for Griewank function.

f_{min} and upper f_{max} bounds of frequency. The best results are obtained for the negative frequency $f^k \in [-2, 0]$, which fulfils the stability condition $f \in [-4, 0]$. The best mean value of the frequency equals $f_{mean} = -1$. Table 3 presents, obtained during experiments, the optimal mean values of frequency and the best mean normalized value of the fitness function for different bounds of frequency. The mean fitness functions J_{mean} for all types of constraints C of frequency, from Table 2, are normalized using the best of them $J_{mean\,best}$ as a normalizing factor:

$$\overline{J_{mean}}(C) = \frac{J_{mean}(C)}{J_{mean\,best}}. \tag{36}$$

Table 3: The mean normalized value of the performance $\overline{J_{mean}}$ for the benchmark function for different bounds of frequency and the best found mean values of the frequency.

	Frequency symmetry			The best mean value of frequency
	Negative $[-B\,0]$	Symmetric $[-B\,B]$	Positive $[0\,B]$	f_{mean}
Sphere	**1.00**	$1.06E+05$	$1.16E+06$	−4
Griewank	**1.00**	1.50	2.26	−1
Ackley	**1.00**	2.33	3.74	−1
Schwefel	**1.00**	$2.67E+08$	$3.96E+09$	−2.5

The best solutions are bolded in the table. The Sphere and the Schwefel functions are very sensitive to the value of frequency and the stability of BA. The Ackley and the Griewank functions have the best solution for negative frequency, but the dominance of these frequencies is not significant.

6. Conclusions

In the paper, we described the dynamic of BA by the state-space form. The analyses of observability and controllability

allow reducing the dynamics of BA from third to second order. The Lyapunov stability theory and Sylvester's criterion applied to the stochastic dynamic of BA determine the condition for asymptotic stability and convergence to the equilibrium point. We also made the analysis of the location of eigenvalues of the state transition matrix and its influence on the response of the algorithm. Presented in the paper results can be used during the process of design and tuning the BA. As an illustrative example, four benchmark functions were used, with particular emphasis on the behaviour of BA used for Griewank function.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

References

[1] M. Seyedmahmoudian, S. Mekhilef, R. Rahmani, R. Yusof, and A. Asghar Shojaei, "Maximum power point tracking of partial shaded photovoltaic array using an evolutionary algorithm: a particle swarm optimization technique," *Journal of Renewable and Sustainable Energy*, vol. 6, no. 2, article 023102, 2014.

[2] A. Moeed Amjad and Z. Salam, "A review of soft computing methods for harmonics elimination PWM for inverters in renewable energy conversion systems," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 141–153, 2014.

[3] R. Talebi, M. M. Ghiasi, H. Talebi et al., "Application of soft computing approaches for modeling saturation pressure of reservoir oils," *Journal of Natural Gas Science and Engineering*, vol. 20, pp. 8–15, 2014.

[4] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization (Vol. 7)*, John Wiley & Sons, 2000.

[5] F. Glover, E. Taillard, and E. Taillard, "A user's guide to tabu search," *Annals of Operations Research*, vol. 41, no. 1, pp. 1–28, 1993.

[6] J. Paplinski, "Continuous ant colony optimization for identification of time delays in the linear plant," in *Swarm and Evolutionary Computation*, pp. 119–127, Springer Berlin Heidelberg, 2012.

[7] J. P. Paplinski, "Time delays identification by means of a hybrid interpolated ant colony optimization and Nelder-Mead algorithm," in *2013 International Conference on Process Control (PC)*, pp. 42–46, Strbske Pleso, Slovakia, 2013.

[8] J., Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 60–766, Springer US, 2010.

[9] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74, Springer Berlin Heidelberg, 2010.

[10] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Lévy flights trajectory," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 453812, 13 pages, 2013.

[11] S. Yılmaz and E. U. Küçüksille, "Improved bat algorithm (IBA) on continuous optimization problems," *Lecture Notes on Software Engineering*, vol. 1, no. 3, pp. 279–283, 2013.

[12] S. Yılmaz and E. U. Küçüksille, "A new modification approach on bat algorithm for solving optimization problems," *Applied Soft Computing*, vol. 28, pp. 259–275, 2015.

[13] X. S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-Inspired Harmony Search Algorithm*, pp. 1–14, Springer Berlin Heidelberg, 2009.

[14] S. ShabnamHasan and F. Ahmed, "Balancing explorations with exploitations in the artificial bee colony algorithm for numerical function optimization," *International Journal of Applied Information Systems*, vol. 9, no. 1, pp. 42–48, 2015.

[15] F. Xue, Y. Cai, Y. Cao, Z. Cui, and F. Li, "Optimal parameter settings for bat algorithm," *International Journal of Bio-Inspired Computation*, vol. 7, no. 2, pp. 125–128, 2015.

[16] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[17] M. Clerc and J. Kennedy, "The particle swarm—explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[18] J. L. Fernandez-Martinez and E. Garcia-Gonzalo, "Stochastic stability analysis of the linear continuous and discrete PSO models," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 3, pp. 405–423, 2011.

[19] N. R. Samal, A. Konar, S. Das, and A. Abraham, "A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence," in *2007 IEEE Congress on Evolutionary Computation*, pp. 1769–1776, Singapore, 2007.

[20] H. M. Emara and A. Fattah, "Continuous swarm optimization technique with stability analysis," *Proceedings of the 2004 American Control Conference*, pp. 2811–2817, 2004.

[21] Q. Jia and Y. Li, "The parameter selection of PSO using Lyapunov theory," *International Journal of Applied Mathematics and Statistics™*, vol. 52, no. 5, pp. 69–75, 2014.

[22] J. P. Paplinski and M. Lazoryszczak, "The stability analysis of bat algorithm," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Gdynia, Poland, 2017.

[23] F. Garofalo, G. Celentano, and L. Glielmo, "Stability robustness of interval matrices via Lyapunov quadratic forms," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 281–284, 1993.

[24] M. C. de Oliveira, J. Bernussou, and J. C. Geromel, "A new discrete-time robust stability condition," *Systems & Control Letters*, vol. 37, no. 4, pp. 261–265, 1999.

[25] R. D. DeGroat, L. R. Hunt, D. A. Linebarger, and M. Verma, "Discrete-time nonlinear system stability," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 10, pp. 834–840, 1992.

[26] Y. Li, W. Zhang, and X. Liu, "Stability of nonlinear stochastic discrete-time systems," *Journal of Applied Mathematics*, vol. 2013, Article ID 356746, 8 pages, 2013.

[27] J. M. Ortega, *Matrix Theory: A Second Course*, Springer Science & Business Media, 2013.

[28] S. N. Elaydi, *Discrete Chaos: With Applications in Science and Engineering*, CRC Press, 2007.

[29] G. T. Gilbert, "Positive definite matrices and Sylvester's criterion," *The American Mathematical Monthly*, vol. 98, no. 1, pp. 44–46, 1991.

[30] R. Kalman, "On the general theory of control systems," *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, 1959.

Hindawi
Complexity
Volume 2018, Article ID 5967604, 11 pages
https://doi.org/10.1155/2018/5967604

Research Article

Approximate Method to Evaluate Reliability of Complex Networks

Petru Caşcaval

Department of Computer Science and Engineering, "Gheorghe Asachi" Technical University of Iaşi, Dimitrie Mangeron Street, 27, 700050 Iaşi, Romania

Correspondence should be addressed to Petru Caşcaval; cascaval@cs.tuiasi.ro

Received 20 April 2018; Revised 12 July 2018; Accepted 31 July 2018; Published 12 November 2018

Academic Editor: Ireneusz Czarnowski

This paper deals with the issue of reliability evaluation in complex networks, in which both link and node failures are considered, and proposes an approximate method based on the minimal paths between two specified nodes. The method requires an algorithm for transforming the set of minimal paths into a sum of disjoint products (SDP). To reduce the computation burden, in the first stage, only the links of the network are considered. Then, in the second stage, each term of the set of disjoint link-products is separately processed, taking into consideration the reliability values for both links and adjacent nodes. In this way, a reliability expression with a one-to-one correspondence to the set of disjoint products is obtained. This approximate method provides a very good accuracy and greatly reduces the computation for complex networks.

1. Introduction

The network reliability theory is extensively applied in many real-world systems that can be modeled as stochastic networks, such as communication networks, sensor networks, social networks, etc. A variety of tools are used for system modeling and computation of reliability or availability indices that describe in a certain way the ability of a network to carry out a desired operation. Most tools are based on algorithms described in terms of minimal path set or minimal cut set (see, for example, [1–8]). Unfortunately, the problem of computing the network reliability based on the set of the minimal paths or cuts is NP-hard [7, 9]. For this reason, in case of more complex networks, other techniques for approximate reliability evaluation are also applied, such as those based on network decomposition or on Monte Carlo simulations (see, for example, [10–16]).

In this work, we deal with the problem of evaluation of two-terminal reliability or availability indices in medium-to-large networks, based on SDP algorithms, in which both link and node failures are considered.

Many authors address this problem assuming that the nodes of the system are perfectly reliable (see, for example, [1, 4–6]). However, in a communication system, nodes also have certain probability of failure so that the reliability evaluation assuming perfect nodes is not realistic.

The failure of a node inhibits the work of all links connected to it. Based on this concept, starting from the given network with unreliable nodes, reduced models with perfect nodes but with links having increased failure probabilities can be obtained. This method is simple, but not so accurate. Because the failure of a node inhibits the work of all adjacent links, the work of the links connected to it depends on the state of this common node. However, a reduced model is solved under the hypothesis according to which the failures that affect the network are independent. For this reason, the reliability estimation must be accepted with caution. Indeed, the estimation error of two-terminal network reliability could be unacceptable in many cases, especially when the failure probabilities of the nodes have high values.

To highlight this aspect, let us consider a simple network with unreliable nodes as presented in Figure 1(a). The reliability of the connection between nodes 1 and 4 has to be evaluated. These two terminal nodes are considered in series with the rest of the network. Three reduced models with perfect nodes and links having increased probabilities of failure are presented in Figures 1(b)–1(d). With dashed line, it is indicated that the failure of a node can be modeled by a cut

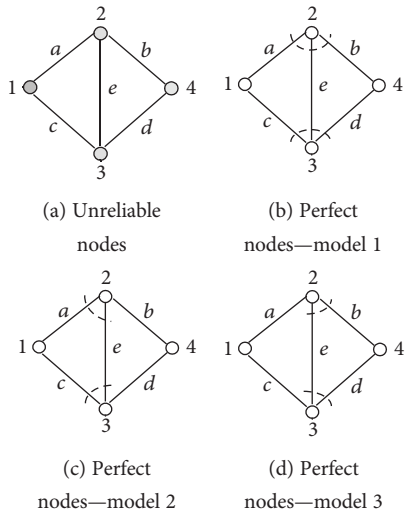


Figure 1: A simple network: (a) initial model; (b–d) reduced models.

of the links connected to it. Consequently, in case of a reduced model with perfect nodes, the reliability of the links in the network must be adjusted accordingly. It is easy to observe that other reduced models are also possible.

For the reduced models presented in Figure 1, Table 1 shows how the reliability of each link in the network is adjusted to capture the fact that the nodes of the given network are also unreliable.

For a numerical evaluation of these approximate models, let us consider a network with the following reliability values for the nodes and the links: $p_1 = 0.99$, $p_2 = 0.98$, $p_3 = 0.97$, $p_4 = 0.96$, $p_a = 0.99$, $p_b = 0.98$, $p_c = 0.97$, $p_d = 0.96$, and $p_e = 0.95$. The numerical results expressing the reliability estimation of the connection between nodes 1 and 4 $(R_{1\text{-}4})$ are presented in Table 2.

These numerical results show that the reduced models with perfect nodes might be useful in a way, but the reliability estimation is not so accurate, even for this simple network.

A better solution that makes a link-based reliability evaluation algorithm adaptable to communication systems is given by Aggarwal et al. [2]. Thus, based on a SDP expression obtained with the assumption of perfect nodes, the node reliability values are taken into account in a specific mode for each term of the set of disjoint link-products. However, the authors do not completely address aspects of the influence of a node on the links connected to it, as it will be seen in Section 5. Moreover, the method is limited to the SDP expressions generated with a so-called "single variable inversion" (SVI) technique. But, for complex networks, "multiple variable inversion" (MVI) techniques are required [1, 4, 14, 16].

In this work, a new approximate method for two-terminal network reliability evaluation with a much better accuracy is proposed. The method is based on algorithms described in terms of minimal path set and covers both SVI and MVI expressions. Just like in [2], in the first stage, the method is focused only on the links of the network. For the any two given nodes, all the minimal paths are enumerated,

Table 1: Adjusted reliability values for the links in the network.

Reduced model	New reliability values
Model 1	$p'_a = p_a p_2, p'_b = p_b p_2, p'_c = p_c p_3, p'_d = p_d p_3, p'_e = p_e p_2 p_3$
Model 2	$p'_a = p_a p_2, p'_b = p_b, p'_c = p_c p_3, p'_d = p_d, p'_e = p_e p_2 p_3$
Model 3	$p'_a = p_a, p'_b = p_b p_2, p'_c = p_c, p'_d = p_d p_3, p'_e = p_e p_2 p_3$

Table 2: Numerical results$(R_{1\text{-}4})$.

Exact result	Approximate results obtained based on the reduced models		
	Model 1	Model 2	Model 3
0.9467	0.9466	0.9477	0.9476

and then this set of minimal paths is transformed into a set of disjoint products. In the second stage, each term of the sum of disjoint products including state variables associated to the links is processed distinctly by considering both links and adjacent node reliability values.

This new approximate method reduces the computation time for large networks to a great extent, compared with an exact method. This reduction in computation time is explained by the fact that the node failures are taken into account only in the second stage when the computation process is simpler, belonging to the $O(n \times m)$ class of complexity, where n is the number of disjoint link-products and m is the number of the network components.

The rest of this paper is organized as follows. Section 2 introduces notations, assumptions, and a short nomenclature, while Section 3 presents general issues regarding the problem of network reliability evaluation. Section 4 provides a method for exact evaluation of two-terminal network reliability when both node and link failures are considered. Section 5, the most extensive one, presents a new approximate method that reduces the complexity of this problem in medium-to-large networks. Section 6 presents some obtained numerical results. The paper ends with some final remarks presented in Section 7.

2. Notations and Preliminary Considerations

2.1. Nomenclature

(a) *Reliability*: the two-terminal reliability of a stochastic network expresses the probability that there exists at least one path between any two specified nodes (let us say a source node and a target one) which operate successfully

(b) *Connected nodes*: two nodes which can communicate with each other are connected; otherwise, they are disconnected

(c) *Minimal path*: a minimal set of links and their adjacent nodes whose good operation ensures that two given nodes are connected. For a minimal path, any proper subset is no longer a path

(d) *Uniproduct*: Boolean product composed only of distinct uncomplemented variables

(e) *Subproduct*: part of a Boolean product that is a complemented or an uncomplemented uniproduct

(f) *Mixproduct*: product of one uncomplemented subproduct and one or more complemented subproducts

(g) *Disjoint products*: a set of products expressing mutually exclusive states

2.2. Notations

(a) $G(V, E)$ is a network model with node set $V = \{y_1, y_2, \dots, y_k\}$ and link set $E = \{x_1, x_2, \dots, x_m\}$

(b) $s, t \in V, s \neq t$, are the source and target nodes

(c) p_x is the reliability of node $x \in V$ or link $x \in E$, and $q_x = 1 - p_x$

(d) R_{s-t} is the two-terminal reliability of network $G(V, E)$ with s and t the source and target nodes ($s - t$ network reliability)

(e) $P(A)$ denotes the probability of the event A

2.3. Assumptions

(a) Each component in the network (i.e., node or link) is either operational or failed, so a logical variable is used to indicate its state. The same notations $y_1, y_2, \dots, y_k$ and $x_1, x_2, \dots, x_m$ are used to denote these logical variables

(b) The events of failure that affect the nodes or the links in network are stochastically independent

3. Considerations on Network Reliability Evaluation

Consider $G(V, E)$ the network under study and $s, t \in V, s \neq t$, the source and target nodes. For this model, consider the minimal path set MPS $= \{P_1, P_2, \dots, P_{np}\}$. Note that a minimal path $P_i \in$ MPS is expressed by a product of distinct logical variables associated with some links or nodes of the network, and the reliability of this path is given by

$$P(P_i) = \prod_{c \in P_i} p_c. \tag{1}$$

Starting from this minimal path set, a structure function $S = \bigcup_{i=1}^{np} P_i$ is defined, and the two-terminal network reliability of this model is calculated by

$$R_{s-t} = P(S) = P\left(\bigcup_{i=1}^{np} P_i\right). \tag{2}$$

Efficient methods for enumerating all minimal paths are presented in [14, 17, 18]. To compute the network reliability R_{s-t} based on (2), the well-known rule of sum of disjoint products is recommended:

$$P\left(\bigcup_{i=1}^{n} A_i\right) = P(A_1) + P\left(\bar{A}_1 \cap A_2\right) + P\left(\bar{A}_1 \cap \bar{A}_2 \cap A_3\right) + \cdots + P\left(\bar{A}_1 \cap \bar{A}_2 \cap \cdots \cap \bar{A}_{n-1} \cap A_n\right). \tag{3}$$

For this purpose, the structure function S is transformed into an equivalent form S', composed only of disjoint products (DP), so that the two-terminal network reliability R_{s-t} is given by

$$R_{s-t} = P\left(S'\right) = \bigcup_j \mathrm{DP}_j = \sum_j P\left(\mathrm{DP}_j\right). \tag{4}$$

Observe that (4) is easy to compute, so that the problem of computing the two-terminal network reliability essentially boils down to generating a new set of disjoint products starting from the set MPS of minimal paths. Unfortunately, this task falls in the NP-hard category.

The first computerized SDP algorithm was proposed by Aggarwal et al. [3], but one of the best known SDP algorithms for transforming the structure function to a sum of disjoint products is given by Abraham [4].

If P and Q are two undisjoint products, and $x_1, x_2, \dots, x_s \in P \setminus Q$, according to Abraham's theorem, the following logical expression can be written as follows:

$$P + Q = P + \bar{x}_1 Q + x_1 \bar{x}_2 Q + x_1 x_2 \bar{x}_3 Q + \cdots + x_1 x_2 \cdots x_{s-1} \bar{x}_s Q. \tag{5}$$

Note that, to ensure that two products are disjoint, only a single complemented variable is added with each new term. Abraham's algorithm is a reference for the so-called SVI algorithms. Two improved SVI algorithms are presented in [19, 20].

To reduce the computation time, other approaches based on the so-called MVI technique have been devised (see, for example, [5, 6, 21–23]).

When an MVI technique is applied, a product may contain distinct logical variables (complemented or not) but also one or more complemented subproducts. For instance, take seven variables representing a network state where links 2 and 4 are not both operational, link 6 is operational, link 7 is in the failed state, and links 1, 3, and 5 are in a do-not-care state. In an MVI approach, this network state is represented by the Boolean expression $\overline{x_2 x_4} x_6 \overline{x_7}$, whereas in an SVI approach, by the expression $\overline{x_2}\, x_6\, \overline{x_7} + x_2\, \overline{x_4}\, x_6\, \overline{x_7}$, so that the advantage of the MVI approach is obvious.

An excellent survey on MVI techniques can be found in [16]. A new MVI technique, called NMVI, is proposed by Cașcaval and Floria in [1].

According to the NMVI method, in order to expand a product Q in relation to a given uniproduct P, so that any new generated product to be disjoint with P, the following two MVI rules are applied.

Rule 1. Type I expansion
If $x_1, x_2, \ldots, x_s \in P \setminus Q$, the following equation can be written as follows:

$$P + Q = P + x_1 x_2 \cdots x_s Q + \overline{x_1 x_2 \cdots x_s} Q. \tag{6}$$

When P and Q are both uniproducts, for the new term $x_1 x_2 \cdots x_s Q$, the absorption law is applicable, so that a reduced logical expression with two disjoint products is obtained:

$$P + Q = P + \overline{x_1 x_2 \cdots x_s} Q. \tag{7}$$

Rule 2. Type II expansion
Consider $P = x_1 x_2 \cdots x_i R_1$, and $Q = \overline{x_1 x_2 \cdots x_i x_{i+1} \cdots x_s} R_2$. By applying the Boolean rule $\overline{xy} = \bar{x} + x\bar{y}$, the following logical expression results are as follows:

$$\begin{aligned} P + Q &= P + \overline{x_1 x_2 \cdots x_i x_{i+1} \cdots x_s} R_2 \\ &= P + \overline{x_1 x_2 \cdots x_i} R_2 + x_1 x_2 \cdots x_i \overline{x_{i+1} \cdots x_s} R_2. \end{aligned} \tag{8}$$

When $R_1 \in R_2$, the term $x_1 x_2 \ldots x_i \overline{x_{i+1} \cdots x_s} R_2$ is absorbed by product P, so that a reduced logical expression composed of two disjoint products is obtained:

$$P + Q = P + \overline{x_1 x_2 \cdots x_i} R_2. \tag{9}$$

As shown in [1], NMVI is an efficient method, providing fewer disjoint products compared with other well-known MVI methods, as CAREL [5], VT [6], or KDH88 [21].

In the next two sections, we address the problem of two-terminal network reliability evaluation, in which both link and node failures are considered. First, an exact method of reliability evaluation is discussed. Then, a new approximate method is presented, with the advantage of being much faster and able to offer a very good accuracy.

4. Exact Evaluation of Network Reliability

For two given nodes, s and t, an exact evaluation of two-terminal network reliability can be obtained based on the set of minimal paths that include both links and adjacent nodes. Compared with the case in which the study is limited to the links of the network, when the nodes are also considered, the number of the minimal paths is unchanged, but any term is extended by also including the adjacent nodes. To illustrate this method, let us analyze the network N_1 presented in Figure 2(a), where the source and target nodes are 1 and 5. These two terminal nodes are considered in series with the rest of the network. For these two given nodes, the set of minimal paths is

$$\begin{aligned} \text{MPS} = \{&3bf, 34beg, 24adg, 23acf, \\ &234aceg, 234adef, 234bcdg\}. \end{aligned} \tag{10}$$

By applying the NMVI method, the following set of disjoint products results as follows:

$$\begin{aligned} \text{DPS} = \Big\{&3bf, 34beg\bar{f}, 24adfg\overline{3b}, 24adgf\overline{\overline{3b}e}, 23acf\bar{b}\overline{\overline{4dg}}, \\ &234aceg\bar{b}\bar{d}\bar{f}, 234adef\bar{b}\bar{c}\bar{g}, 234bcdg\bar{a}\bar{e}\bar{f}\Big\}. \end{aligned} \tag{11}$$

Finally, the reliability R_{1-5} is given by

$$\begin{aligned} R_{1-5} = p_1 p_5 \Big(& p_3 p_b p_f + p_3 p_4 p_b p_e p_g (1 - p_f) \\ & + p_2 p_4 p_a p_d p_g \Big(p_f (1 - p_3 p_b) \\ & + (1 - p_f)(1 - p_3 p_b p_e) \Big) \\ & + p_2 p_3 p_a p_c p_f (1 - p_b)(1 - p_4 p_d p_g) \\ & + p_2 p_3 p_4 \Big(p_a p_c p_e p_g (1 - p_b)(1 - p_d)(1 - p_f) \\ & + p_a p_d p_e p_f (1 - p_b)(1 - p_c)(1 - p_g) \\ & + p_b p_c p_d p_g (1 - p_a)(1 - p_e)(1 - p_f) \Big) \Big). \end{aligned} \tag{12}$$

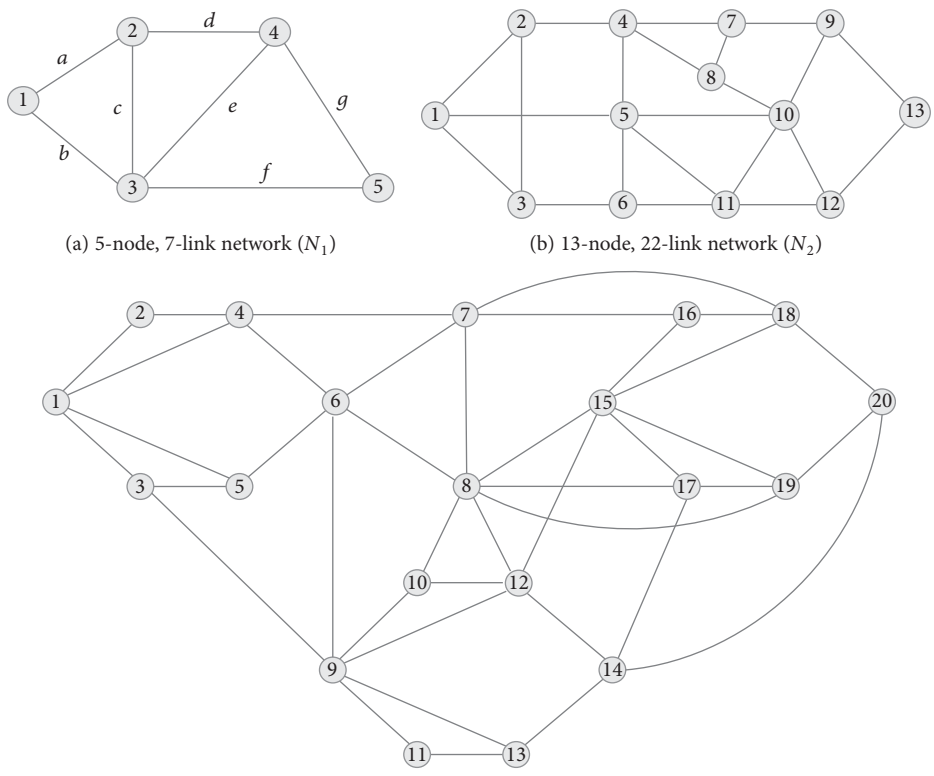
The same network is analyzed in [2], example 2. So, we compared the numerical result obtained based on this equation with the result generated with equation (21) presented in [2]. These results are identical. For example, assuming for all the nodes a reliability of 0.98 and for all the links a reliability of 0.95, both methods give a reliability value $R_{1-5} = 0.969611$.

To cover both nodes and links, much more logical variables are used. The problem that arises in this case is that the number of disjoint products increases to a large extent when complex networks are evaluated. To highlight this aspect, comparative results with respect to the network models N_2 and N_3 given in Figure 2 are presented in Table 3.

Compared with the case in which the study is limited to the links of the network, when the adjacent nodes are also considered, the number of disjoint products increases significantly. The relative growth with respect to the number of disjoint products is about 39% for network N_2, but for the more complex network N_3, this relative growth reaches 86%.

5. Approximate Approach for Network Reliability Evaluation

The process of generating the set of disjoint products is a difficult one, of NP-hard complexity. In order to reduce the computation time, the process of enumerating the minimal paths and their development as a sum of disjoint products is focused only to the links of the network. For this purpose, for a link $x_i \in E$, let X_i be a logical variable that reflects the event of successful communication through that branch—that means that the link x_i and the two adjacent nodes are operational. Thus, the structure function S can be expressed in terms of these logical variables $X_1, X_2, \ldots, X_m$.



(a) 5-node, 7-link network (N_1)

(b) 13-node, 22-link network (N_2)

(c) 20-node, 40-link network (N_3)

FIGURE 2: Network models with unreliable nodes: (a) 5-node, 7-link network (N_1); (b) 13-node, 22-link network (N_2); (c) 20-node, 40-link network (N_3).

TABLE 3: The number of disjoint products. Comparative results.

Network models	Number of minimal paths	Number of disjoint products generated by NMVI, including	
		Only links	Both links and nodes
N_2	281	2269	3151
N_3	16618	1799888	3353457

In the second stage, each term of the sum of disjoint products is processed distinctly by considering both links and adjacent node reliability values. The node reliability values are taken into account in a specific mode for each term of the set of disjoint link-products, when only the adjacent nodes of the links that compose the current product are considered. This is the starting point for this approximate approach.

Based on the set of disjoint products DPS = $\{DP_1, DP_2, \ldots, DP_n\}$, the two-terminal network reliability is computed by applying (4).

A term DP in the set of disjoint products is a mixproduct that includes one uniproduct, noted with U, and one or more complemented subproducts. Figure 3 shows such a complex mixproduct.

As illustrated in Figure 3, the uniproduct U reflects a state of operability of a part of the network that ensures the connection between the source and target nodes. Let SAN be the set of all adjacent nodes of the links that compose the uniproduct U. All these links and all the nodes that belong to SAN are operational. Consequently, the probability of the network state described by U is given by

$$P(U) = \prod_{x \in U} p_x \prod_{y \in \text{SAN}} p_y. \tag{13}$$

The main problem is how to compute or at least evaluate with a good accuracy the probability of a network state described by a complemented subproduct (such a subproduct is illustrated in Figure 3 with a dashed line).

A complemented subproduct reflects a state of inoperability of a branch or of a bigger portion of the network. To begin with, consider the case where such a portion of the network is independent of that portions described by the other complemented subproducts. Under these circumstances, the current term can be independently evaluated. Two cases are distinguished.

Case 1 (a single complemented variable (an SVI term)). Consider a single complemented variable $\overline{X_i}$ (an SVI term) associated with the link x_i that connects two nodes; let us say y_i and y_j (for instance, the variable $\overline{X_{11}}$ in Figure 3). The probability of this event is

$$P\left(\overline{X_i}\right) = 1 - p'_{x_i}, \tag{14}$$

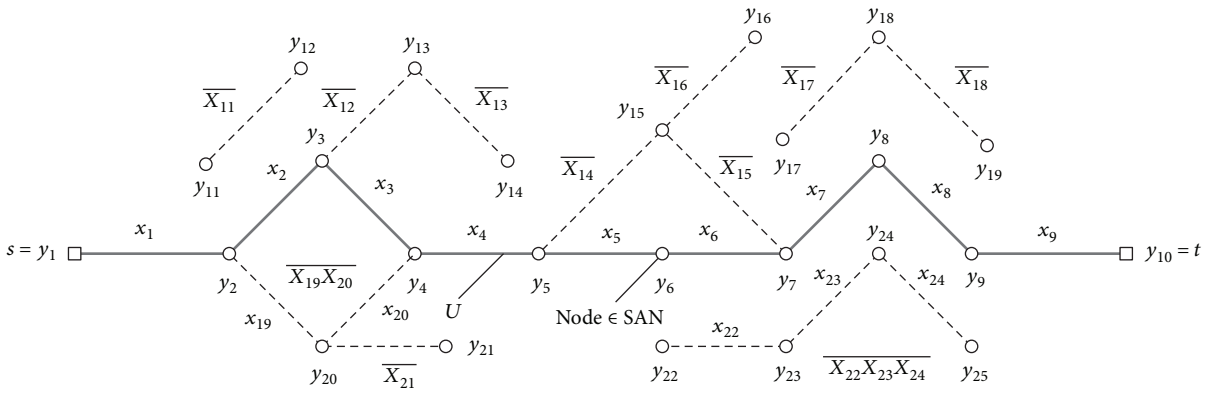


Figure 3: Illustration of a complex mixproduct: $\mathrm{DP} = X_1X_2 \cdots X_9\overline{X_{11}}\,\overline{X_{12}} \cdots \overline{X_{18}}\,\overline{X_{19}X_{20}}\,\overline{X_{21}}\,\overline{X_{22}X_{23}X_{24}}$.

where

$$p'_{x_i} = \begin{cases} p_{x_i}p_{y_i}, & \text{if } y_j \in \mathrm{SAN}, y_i \notin \mathrm{SAN}, \\ p_{x_i}p_{y_j}, & \text{if } y_i \in \mathrm{SAN}, y_j \notin \mathrm{SAN}, \\ p_{x_i}p_{y_i}p_{y_j}, & \text{if } y_i, y_j \notin \mathrm{SAN}. \end{cases} \tag{15}$$

Equations (13) and (14) are found in another form in [2] where the same problem of network reliability evaluation is treated. Remember that the method presented in [2] covers only SDP expressions composed of SVI terms.

Case 2 (an MVI term). Consider a complemented subproduct $\overline{X_1X_2 \cdots X_k}$ (an MVI term) that describes a state of inoperability of a portion of the network as illustrated in Figure 4. The probability that this portion of the network to be inoperable is

$$P\left(\overline{X_1X_2 \cdots X_k}\right) = 1 - Q, \tag{16}$$

where the product Q includes not only the reliability of the corresponding links but also the reliability of the adjacent nodes that do not belong to SAN, considered only once. More exactly, the probability Q is computed by the following sequence of steps presented in Pseudocode 1.

Even though the two portions of the network described by two complemented subproducts may not have any common link, they may have one or even more common nodes. Consequently, the state of inoperability of these two portions of the network may be due to the failure of such a common node. Of course, we refer to a common node that does not belong to SAN.

In the first stage, the analysis of these dependencies given by the common nodes is limited to the level of pairs of complemented subproducts. The following three cases are distinguished.

Case 3 (two SVI terms with a common node). Consider two complemented variables $\overline{X_i}$ and $\overline{X_j}$ that describe a state of inoperability for two branches x_i and x_j that have

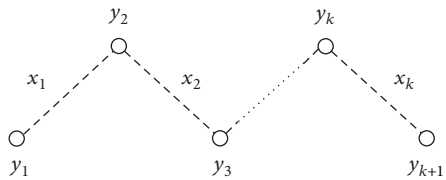


Figure 4: Illustration of a MVI term: $\overline{X_1X_2 \cdots X_k}$.

```
Q = 1;
for i = 1 : k
    Q = Q* p_{x_i};
    if y_i ∉ SAN then Q = Q* p_{y_i};
end
if y_{k+1} ∉ SAN then Q = Q* p_{y_{k+1}};
```

Pseudocode 1: Computing the product Q.

y_k as a common node, as illustrated in Figure 5. The node $y_k \notin \mathrm{SAN}$.

Let us define the probabilities p'_{x_i} and p'_{x_j} associated with the links x_i and x_j, given as follows:

$$p'_{x_i} = \begin{cases} p_{x_i}, & \text{if } y_i \in \mathrm{SAN}, \\ p_{x_i}p_{y_i}, & \text{if } y_i \notin \mathrm{SAN}, \end{cases}$$
$$p'_{x_j} = \begin{cases} p_{x_j}, & \text{if } y_j \in \mathrm{SAN}, \\ p_{x_j}p_{y_j}, & \text{if } y_j \notin \mathrm{SAN}. \end{cases} \tag{17}$$

By applying the theorem of total probability to the event space $\{y_k, \bar{y}_k\}$, the following equation can be written as follows:

$$P\left(\overline{X_i}\,\overline{X_j}\right) = P(X) = p_{y_k}P(X \mid y_k) + \left(1 - p_{y_k}\right)P(X \mid \bar{y}_k). \tag{18}$$

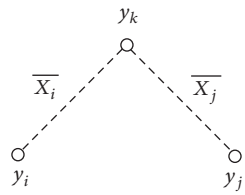


Figure 5: Illustration of two inoperable links with a common node.

As

$$\begin{aligned} P(X \mid \bar{y}_k) &= 1, \\ P(X \mid y_k) &= \left(1 - p'_{x_i}\right)\left(1 - p'_{x_j}\right), \end{aligned} \tag{19}$$

finally, the equation becomes

$$P\left(\overline{X_i}\,\overline{X_j}\right) = 1 - p_{y_k}\left(p'_{x_i} + p'_{x_j} - p'_{x_i}p'_{x_j}\right). \tag{20}$$

Note that this case is not treated in [2].

Case 4 (an MVI term and an SVI one with a common node). Consider, for example, $\overline{X_1X_2X_3}$ and $\overline{X_4}$ to be two terms in DP describing a state of inoperability of two portions of the network as illustrated in Figure 6. The common node $y_2 \notin$ SAN.

Let us define the probabilities p'_{x_1}, p'_{x_2}, p'_{x_3}, and p'_{x_4} given as follows:

$$\begin{aligned} p'_{x_1} &= \begin{cases} p_{x_1}, & \text{if } y_1 \in \text{SAN}, \\ p_{x_1}p_{y_1}, & \text{if } y_1 \notin \text{SAN}, \end{cases} \\ p'_{x_2} &= \begin{cases} p_{x_2}, & \text{if } y_3 \in \text{SAN}, \\ p_{x_2}p_{y_3}, & \text{if } y_3 \notin \text{SAN}, \end{cases} \\ p'_{x_3} &= \begin{cases} p_{x_3}, & \text{if } y_4 \in \text{SAN}, \\ p_{x_3}p_{y_4}, & \text{if } y_4 \notin \text{SAN}, \end{cases} \\ p'_{x_4} &= \begin{cases} p_{x_4}, & \text{if } y_5 \in \text{SAN}, \\ p_{x_4}p_{y_5}, & \text{if } y_5 \notin \text{SAN}. \end{cases} \end{aligned} \tag{21}$$

By applying the theorem of total probability to the event space $\{y_2, \bar{y}_2\}$, the following equation can be written as follows:

$$P\left(\overline{X_1X_2X_3}\,\overline{X_4}\right) = P(X) = p_{y_2}P(X \mid y_2) + \left(1 - p_{y_2}\right)P(X \mid \bar{y}_2). \tag{22}$$

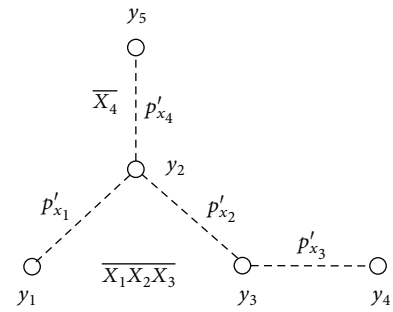


Figure 6: Illustration of an MVI term and an SVI one with a common node.

As

$$\begin{aligned} P(X \mid \bar{y}_2) &= 1, \\ P(X \mid y_2) &= \left(1 - p'_{x_1}p'_{x_2}p'_{x_3}\right)\left(1 - p'_{x_4}\right), \end{aligned} \tag{23}$$

finally, the following equation results in

$$P\left(\overline{X_1X_2X_3}\,\overline{X_4}\right) = 1 - p_{y_2}\left(p'_{x_1}p'_{x_2}p'_{x_3} + p'_{x_4} - p'_{x_1}p'_{x_2}p'_{x_3}p'_{x_4}\right). \tag{24}$$

Note that, if the two events were treated independently, the following equation would result in

$$\begin{aligned} P\left(\overline{X_1X_2X_3}\right)P\left(\overline{X_4}\right) &= \left(1 - p'_{x_1}p'_{x_2}p'_{x_3}p_{y_2}\right)\left(1 - p'_{x_4}p_{y_2}\right) \\ &= 1 - p_{y_2}\left(p'_{x_1}p'_{x_2}p'_{x_3} + p'_{x_4} - p'_{x_1}p'_{x_2}p'_{x_3}p'_{x_4}p_{y_2}\right) \\ &< P\left(\overline{X_1X_2X_3}\,\overline{X_4}\right). \end{aligned} \tag{25}$$

Remark 1. Equation (25) shows that when a common node is not taken into account, the reliability estimation is a pessimistic one.

Case 5 (two MVI terms with a common node). Consider, for example, $\overline{X_1X_2}$ and $\overline{X_3X_4}$ to be the two MVI terms describing a state of inoperability of two parts of the network, as illustrated in Figure 7, where the communication between nodes 1 and 2 and between nodes 3 and 4 is not possible. The common node $y_5 \notin$ SAN.

Let us define the probabilities p'_{x_1}, p'_{x_2}, p'_{x_3}, and p'_{x_4} given by the following:

$$p'_{x_i} = \begin{cases} p_{x_i}, & \text{if } y_i \in \text{SAN}, \\ p_{x_i}p_{y_i}, & \text{if } y_i \notin \text{SAN}, \\ i = 1, 2, 3, 4. \end{cases} \tag{26}$$

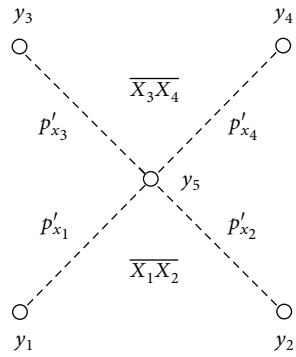


Figure 7: Illustration of two MVI terms with a common node.

The probability of this state, $P(\overline{X_1X_2}\,\overline{X_3X_4})$, can be determined by applying the rule of total probability to the event space $\{y_5, \bar{y}_5\}$.

If $X = \overline{X_1X_2}\,\overline{X_3X_4}$, the following equation can be written as follows:

$$P(X) = p_{y_5}P(X \mid y_5) + \left(1 - p_{y_5}\right)P(X \mid \bar{y}_5). \tag{27}$$

Obviously, $P(X/\bar{y}_5) = 1$.

$$\begin{aligned} P(X \mid y_5) &= \left(1 - p'_{x_1}p'_{x_2}\right)\left(1 - p'_{x_3}p'_{x_4}\right) \\ &= 1 - p'_{x_1}p'_{x_2} - p'_{x_3}p'_{x_4} + p'_{x_1}p'_{x_2}p'_{x_3}p'_{x_4}. \end{aligned} \tag{28}$$

Finally, the equation becomes

$$P\left(\overline{X_1X_2}\,\overline{X_3X_4}\right) = 1 - p_{y_5}\left(p'_{x_1}p'_{x_2} + p'_{x_3}p'_{x_4} - p'_{x_1}p'_{x_2}p'_{x_3}p'_{x_4}\right). \tag{29}$$

To exemplify these 5 rules previously defined, consider the mixproduct

$$\mathrm{DP} = X_1X_2 \cdots X_9\overline{X_{11}}\,\overline{X_{12}} \cdots \overline{X_{18}}\,\overline{X_{19}X_{20}}\,\overline{X_{21}}\,\overline{X_{22}X_{23}X_{24}}, \tag{30}$$

as illustrated in Figure 3. The mixproduct DP includes the uniproduct $U = X_1X_2 \cdots X_9$ and for this operable path, the set of adjacent nodes is $\mathrm{SAN} = \{y_1, y_2, \ldots, y_{10}\}$.

Taking into account the common nodes for the SVI and MVI terms, the probability of the mixproduct DP can be evaluated with a good accuracy by the following:

$$\begin{aligned} P(\mathrm{DP}) = P(U)P\left(\overline{X_{11}}\right)P\left(\overline{X_{12}}\,\overline{X_{13}}\right)P\left(\overline{X_{14}}\,\overline{X_{15}}\right)P\left(\overline{X_{16}}\right) \\ \times P\left(\overline{X_{17}}\,\overline{X_{18}}\right)P\left(\overline{X_{19}X_{20}}\,\overline{X_{21}}\right)P\left(\overline{X_{22}X_{23}X_{24}}\right). \end{aligned} \tag{31}$$

By applying the rules presented before, the following equations result in

$$\begin{aligned} P(U) &= p_{x_1}p_{x_2} \cdots p_{x_9}p_{y_1}p_{y_2} \cdots p_{y_{10}}, \\ P\left(\overline{X_{11}}\right) &= 1 - p_{y_{11}}p_{x_{11}}p_{y_{12}}, \\ P\left(\overline{X_{12}}\,\overline{X_{13}}\right) &= 1 - p_{y_{13}}\left(p_{x_{12}} + p_{x_{13}}{}' - p_{x_{12}}p_{x_{13}}{}'\right), \end{aligned} \tag{32}$$

where $p'_{x_{13}} = p_{x_{13}}p_{y_{14}}$.

$$\begin{aligned} P\left(\overline{X_{14}}\,\overline{X_{15}}\right) &= 1 - p_{y_{15}}\left(p_{x_{14}} + p_{x_{15}} - p_{x_{14}}p_{x_{15}}\right), \\ P\left(\overline{X_{16}}\right) &= 1 - p_{y_{15}}p_{x_{16}}p_{y_{16}}\,(\text{an approximate evaluation}), \\ P\left(\overline{X_{17}}\,\overline{X_{18}}\right) &= 1 - p_{y_{18}}\left(p'_{x_{17}} + p'_{x_{18}} - p'_{x_{17}}p'_{x_{18}}\right), \end{aligned} \tag{33}$$

where $p'_{x_{17}} = p_{x_{17}}p_{y_{17}}$ and $p'_{x_{18}} = p_{x_{18}}p_{y_{19}}$.

$$P\left(\overline{X_{19}X_{20}}\,\overline{X_{21}}\right) = 1 - p_{y_{20}}\left(p_{x_{19}}p_{x_{20}} + p'_{x_{21}} - p_{x_{19}}p_{x_{20}}p'_{x_{21}}\right), \tag{34}$$

where $p'_{x_{21}} = p_{x_{21}}p_{y_{21}}$.

$$P\left(\overline{X_{22}X_{23}X_{24}}\right) = 1 - p_{y_{22}}p_{x_{22}}p_{y_{23}}p_{x_{23}}p_{y_{24}}p_{x_{24}}p_{y_{25}}. \tag{35}$$

Observe that, related to the probability of this mixproduct, an approximation is made with respect to the terms $\overline{X_{14}}$, $\overline{X_{15}}$, and $\overline{X_{16}}$, because the links x_{14}, x_{15}, and x_{16} have a common node, $y_{15} \notin \mathrm{SAN}$. This case is discussed in more detail below.

Case 6 (many terms with a common node). Consider three links x_1, x_2, and x_3 with a common node and the network state reflected by the SVI terms $\overline{X_1}$, $\overline{X_2}$, and $\overline{X_3}$, as illustrated in Figure 8.

Suppose that $y_4 \notin \mathrm{SAN}$. Let us first define the probabilities p'_{x_1}, p'_{x_2}, and p'_{x_3} by the following:

$$p'_{x_i} = \begin{cases} p_{x_i}, & \text{if } y_i \in \mathrm{SAN}, \\ p_{x_i}p_{y_i}, & \text{if } y_i \notin \mathrm{SAN}, \\ i = 1, 2, 3. \end{cases} \tag{36}$$

In order to evaluate the probability $P(\overline{X_1}\,\overline{X_2}\,\overline{X_3})$, the theorem of total probability is applied to the event space $\{y_4, \bar{y}_4\}$. The following equation results in

$$P\left(\overline{X_1}\,\overline{X_2}\,\overline{X_3}\right) = P(X) = p_{y_4}P(X \mid y_4) + \left(1 - p_{y_4}\right)P(X \mid \bar{y}_4). \tag{37}$$

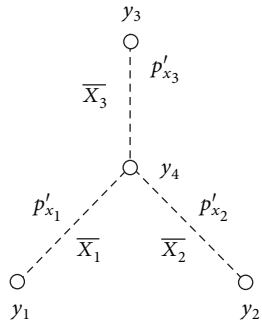


Figure 8: Illustration of three SVI terms with a common node.

Obviously, $P(X/\bar{y}_4) = 1$. When the node y_4 is operational, $\overline{X_1}$, $\overline{X_2}$, and $\overline{X_3}$ reflect independent events, so that

$$P\left(\overline{X_1}\,\overline{X_2}\,\overline{X_3}\right) = P\left(\overline{X_1}\right)P\left(\overline{X_2}\right)P\left(\overline{X_3}\right). \tag{38}$$

Consequently, we have

$$\begin{aligned} P(X \mid y_4) &= \left(1 - p'_{x_1}\right)\left(1 - p'_{x_2}\right)\left(1 - p'_{x_3}\right) \\ &= 1 - p'_{x_1} - p'_{x_2} - p'_{x_3} + p'_{x_1}p'_{x_2} + p'_{x_1}p'_{x_3} \\ &\quad + p'_{x_2}p'_{x_3} - p'_{x_1}p'_{x_2}p'_{x_3}. \end{aligned} \tag{39}$$

Finally, the equation becomes

$$\begin{aligned} P\left(\overline{X_1}\,\overline{X_2}\,\overline{X_3}\right) = 1 - p_{y_4}\Big(&p'_{x_1} + p'_{x_2} + p'_{x_3} - p'_{x_1}p'_{x_2} \\ &- p'_{x_1}p'_{x_3} - p'_{x_2}p'_{x_3} + p'_{x_1}p'_{x_2}p'_{x_3}\Big). \end{aligned} \tag{40}$$

Observe that, in case of an exact evaluation, the equation is composed of $2^3 = 8$ terms. This result can be generalized for the case with more events $\overline{X_1}, \overline{X_2}, \ldots, \overline{X_m}$ that depend on the state of a common node. The equation for computing the probability $P(\overline{X_1}\,\overline{X_2}\cdots\overline{X_m})$ comprises 2^m terms. To reduce the computation time, the method we propose is limited to the pairs of events depending on the state of a common node, for which the equation (20, 24, or 29 as is the case) comprises only four terms.

This is the approximation that may slightly affect the result given by the method we propose. Also, this work does not treat those cases when two MVI terms have two or more common nodes, considering that these cases are very rare. Nevertheless, this method provides a network reliability evaluation with a very good accuracy, as can be seen in the next section. In all the checks we have made, this method has generated exact values or slightly pessimistic results. So, the reliability value given by this method can be interpreted as a lower limit and it can be explained on the basis of (25) and Remark 1. This aspect is very important for a reliability study.

Before ending this section, an answer to the next question is required: why did this method focus only on minimal paths and not on minimal cuts at all? Indeed, it is well known that for networks with high reliability, the approaches based on minimal cuts are generally more appropriate for an approximate evaluation. However, in our case, an approach similar to that applied to the minimal paths is no longer appropriate, because when both node and link failures are considered, one must take into account a set of cuts consisting of nodes, a set of cuts consisting of links, and another one that comprises both nodes and links. So, when the nodes are also considered, the number of minimal cuts increases very much. For this reason, the proposed method is focused only on minimal paths.

6. Numerical Results

To illustrate the efficiency of this approximate method, we consider the network models N_2 and N_3 presented in Figure 2. For these networks, Table 4 presents comparative results, by assuming for all the nodes a reliability value of 0.98 and for all the links a reliability value of 0.95. Observe that the proposed method gives an accurate result for the network N_2, and it gives a slightly approximate value with five accurate decimal places for the larger network N_3.

Computing time for reliability evaluation is presented in Table 5. Compared to the exact method presented in Section 5, for network model N_3, the proposed approximate method greatly reduces the computational time, from 57 min 17 s to 18 min 43 s.

The values presented in Table 5 highlight the rapid growth of the computation time for the reliability evaluation with an increasing network size. The methods of network reliability evaluation based on SDP algorithms fall in the NP-hard category and, consequently, are difficult to apply for very large networks, such as social networks. In these cases, other techniques for approximate evaluation can also be applied, especially the Monte Carlo simulation (see, for example, [24–27]). Even so, the methods based on SDP algorithms are still necessary for the validation of simulation programs.

7. Final Remarks

In this work, the problem of two-terminal network reliability evaluation in which both link and node failures are considered is discussed. An approximate method that provides a very good accuracy is proposed. Compared to an exact method, this approximate method greatly reduces the computation time for complex networks.

The proposed solution can be applied with any SDP algorithm, but the accuracy of the reliability estimation depends on the method used for transforming the set of minimal paths into a set of disjoint products. When the number of disjoint products is lower, the reliability estimation is better. For this reason, efficient MVI algorithms as NMVI or the hybrid algorithm given by Chaturvedi and Misra [23] are recommended. Another approach based on binary decision diagrams (BDDs) is also recommended [28].

Table 4: Network reliability evaluation (R_{s-t}). Comparative results.

Network model	Source and target nodes	Exact method based on NMVI	The proposed method
N_2	$s = 1$, $t = 13$	0.982883	0.982883
N_3	$s = 1$, $t = 20$	0.959579	0.959575

Table 5: Computing time for reliability evaluation.

Network model	Exact method based on NMVI	The proposed method
N_2	0.1 s	0.06 s
N_3	57 min 17 s	18 min 43 s

Data Availability

The paper presents a method for network reliability evaluation for which mathematical proofs are included. It can be applied for any network, and therefore, it does not depend on specific data.

Disclosure

The same issue of two-terminal reliability evaluation in large networks is addressed by the paper "SDP Algorithm for Network Reliability Evaluation", authors P. Caşcaval and S. A. Floria, presented at the IEEE Conference INISTA 2017 [1]. In that paper, an efficient SDP method (called NMVI) for transforming algebraically a structure function (expressed in terms of minimal paths or cuts) into a sum of disjoint products is proposed. This new method is based on an MVI technique and provides better solutions, with fewer disjoint products, compared with other well-known MVI methods. The author asserts that some general issues, such as notations, nomenclature, or other general considerations on network reliability evaluation, are similar to those outlined in [1].

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The author thanks his colleague Sabina-Adriana Floria for the useful and fruitful discussions. Also, many thanks are due to Dr. Florin Leon and Dr. Marius Kloetzer for the helpful suggestions which helped improve the readability of this paper.

References

[1] P. Caşcaval and S. A. Floria, "SDP algorithm for network reliability evaluation," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Gdynia, Poland, July 2017.

[2] K. Aggarwal, J. Gupta, and K. Misra, "A simple method for reliability evaluation of a communication system," *IEEE Transactions on Communications*, vol. 23, no. 5, pp. 563–566, 1975.

[3] K. K. Aggarwal, K. B. Misra, and J. S. Gupta, "A fast algorithm for reliability evaluation," *IEEE Transactions on Reliability*, vol. R-24, no. 1, pp. 83–85, 1975.

[4] J. A. Abraham, "An improved algorithm for network reliability," *IEEE Transactions on Reliability*, vol. R-28, no. 1, pp. 58–61, 1979.

[5] S. Soh and S. Rai, "CAREL: computer aided reliability evaluator for distributed computing networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, no. 2, pp. 199–213, 1991.

[6] M. Veeraraghavan and K. S. Trivedi, "An improved algorithm for symbolic reliability analysis," *IEEE Transactions on Reliability*, vol. 40, no. 3, pp. 347–358, 1991.

[7] J. S. Provan and M. O. Ball, "Computing network reliability in time polynomial in the number of cuts," *Operations Research*, vol. 32, no. 3, pp. 516–526, 1984.

[8] M. O. Ball and J. S. Provan, "Disjoint products and efficient computation of reliability," *Operations Research*, vol. 36, no. 5, pp. 703–715, 1988.

[9] K. S. Trivedi, *Probability and Statistics with Reliability*, Queueing and Computer Science Applications, John Wiley & Sons, New York, NY, USA, 2002.

[10] F. Beichelt and L. Spross, "An effective method for reliability analysis of complex systems," *Journal of Information Processing and Cybernetics*, vol. 23, pp. 227–235, 1987.

[11] P. Caşcaval and B. F. Romanescu, "Complementary approaches for the network reliability evaluation: network decomposition and Monte Carlo simulation," in *Bul. Inst. Polit. Iaşi, Tomul L (LIV), Fasc. 1–4*, pp. 123–131, Automatică şi Calculatoare, 2004.

[12] P. Caşcaval and A. R. Macovei, "Reliability evaluation by network decomposition," in *Bul. Inst. Polit. Iaşi, Tomul XLIX (LIII), Fasc. 1–4*, pp. 56–65, Automatică şi Calculatoare, 2003.

[13] P. Caşcaval and B. A. Botez, "Recursive algorithm for two-terminal network reliability evaluation," in *Bul. Inst. Polit. Iasi, LI (LV), Fasc. 1–4*, pp. 137–146, Automatică şi Calculatoare, 2005.

[14] K. B. Misra, *Reliability Analysis and Prediction: A Methodological Oriented Treatment*, Elsevier, Amsterdam, Oxford, New York, Tokyo, 1992.

[15] M. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*, John Wiley & Sons, New York, NY, USA, 2002.

[16] S. K. Chaturvedi, *Network Reliability: Measures and Evaluation*, Scrivener Publishing-Wiley, Hoboken, NJ, USA, 2016.

[17] Y. Shen, "A new simple algorithm for enumerating all minimal paths and cuts of a graph," *Microelectronics Reliability*, vol. 35, no. 6, pp. 973–976, 1995.

[18] R. Mishra, M. A. Saifi, and S. K. Chaturvedi, "Enumeration of minimal cutsets for directed networks with comparative reliability study for paths or cuts," *Quality and Reliability Engineering International*, vol. 32, no. 2, pp. 555–565, 2016.

[19] F. Beichelt and L. Spross, "An improved Abraham-method for generating disjoint sums," *IEEE Transactions on Reliability*, vol. R-36, no. 1, pp. 70–74, 1987.

[20] M. O. Locks, "A minimizing algorithm for sum of disjoint products," *IEEE Transactions on Reliability*, vol. R-36, no. 4, pp. 445–453, 1987.

[21] K. D. Heidtmann, "Smaller sums of disjoint products by subproduct inversion," *IEEE Transactions on Reliability*, vol. 38, no. 3, pp. 305–311, 1989.

[22] T. Luo and K. S. Trivedi, "An improved algorithm for coherent-system reliability," *IEEE Transactions on Reliability*, vol. 47, no. 1, pp. 73–78, 1998.

[23] S. K. Chaturvedi and K. B. Misra, "A hybrid method to evaluate reliability of complex networks," *International Journal of Quality & Reliability Management*, vol. 19, no. 8/9, pp. 1098–1112, 2002.

[24] K. F. Tee, L. R. Khan, and H. Li, "Application of subset simulation in reliability estimation of underground pipelines," *Reliability Engineering & System Safety*, vol. 130, pp. 125–131, 2014.

[25] K. M. Zuev, S. Wu, and J. L. Beck, "General network reliability problem and its efficient solution by subset simulation," *Probabilistic Engineering Mechanics*, vol. 40, pp. 25–35, 2015.

[26] H.-S. Li, Y.-Z. Ma, and Z. Cao, "A generalized subset simulation approach for estimating small failure probabilities of multiple stochastic responses," *Computers & Structures*, vol. 153, pp. 239–251, 2015.

[27] A. Birolini, *Reliability Engineering*, Theory and practice, Springer-Verlag, Berlin Heidelberg, 2014.

[28] X. Zang, H.-R. Sun, K. S. Trivedi, and D. R. Avresky, Eds., "A BDD approach to dependability analysis of distributed computer systems with imperfect coverage," in *Dependable Network Computing*, pp. 167–190, Kluwer Academic Publishers, Amsterdam, Netherlands, 1999.

Hindawi
Complexity
Volume 2018, Article ID 6794067, 12 pages
https://doi.org/10.1155/2018/6794067

Research Article

Incremental Gene Expression Programming Classifier with Metagenes and Data Reduction

Joanna Jedrzejowicz [1] **and Piotr Jedrzejowicz** [2]

[1] *Institute of Informatics, Faculty of Mathematics, Physics and Informatics, University of Gdansk, 80-308 Gdansk, Poland*
[2] *Department of Information Systems, Gdynia Maritime University, 81-225 Gdynia, Poland*

Correspondence should be addressed to Joanna Jedrzejowicz; jj@inf.ug.edu.pl

Received 27 March 2018; Revised 8 October 2018; Accepted 24 October 2018; Published 7 November 2018

Academic Editor: Vincent Labatut

The paper proposes an incremental Gene Expression Programming classifier. Its main features include using two-level ensemble consisting of base classifiers in form of genes and the upper-level classifier in the form of metagene. The approach enables us to deal with big datasets through controlling computation time using data reduction mechanisms. The user can control the number of attributes used to induce base classifiers as well as the number of base classifiers used to induce metagenes. To optimize the parameter setting phase, an approach based on the Orthogonal Experiment Design principles is proposed, allowing for statistical evaluation of the influence of different factors on the classifier performance. In addition, the algorithm is equipped with a simple mechanism for drift detection. A detailed description of the algorithm is followed by the extensive computational experiment. Its results validate the approach. Computational experiment results show that the proposed approach compares favourably with several state-of-the-art incremental classifiers.

1. Introduction

Learning from the environment through data mining remains an important research challenge. Numerous approaches, algorithms, and techniques have been proposed during recent years to deal with the data mining tasks. An important part of these efforts focuses on mining big datasets and data streams. Barriers posed by a sheer size of the real-life datasets, on one side, and constraints on the resources available for performing the data mining task, including time and computational resources, on the other, are not easy to overcome. Additional complications, apart from the above-mentioned complexity issues, are often encountered due to the nonstationary environments.

One of the most effective approaches to mining big datasets and data streams is using online or incremental learners. Online learning assumes dealing strictly with data streams. Online learners should have the following properties [1]:

(i) Single-pass through the data.

(ii) Each example is processed very fast and in a constant period of time.

(iii) Any-time learning: the classifier should provide the best answer at every moment of time.

The incremental learning is understood as a slightly wider concept, as compared with the online learning one. Incremental learners can deal not only with data streams but also with big datasets stored in databases for which using the "one-by-one" or "chunk-by-chunk" approach could be more effective than using the traditional "batch" learners, even if no concept drift has been detected. An important feature of the incremental learners is their ability to update the currently used model using only newly available individual data instances, without having to reprocess all of the past instances.

In fact, using incremental learners is, quite often, the only possible way to extract any meaningful knowledge. Usual for the contemporary databases is a constant inflow of new data instances. Hence, the knowledge discovered in databases needs to be constantly updated, which is usually an infeasible task for classic learners. Data streams, and even stored datasets, may be affected by the so-called concept drift. In the above cases, online or incremental learners are needed.

In the paper, we propose a new version of the incremental classifier based on Gene Expression Programming (GEP) with data reduction and a metagene as the final, upper-level, classifier. Classifiers using the GEP-induced expression trees are known to produce satisfactory or very good results in terms of the classification accuracy. Our approach uses GEP-induced expression trees to construct learners with the ability to deal with large datasets environment and with a concept drift phenomenon. The rest of the paper is organized as follows. In Section 2 a brief survey of the related results is offered. In Section 3 we describe a new version of the proposed approach. Section 4 contains a detailed description of the validating computational experiment and a discussion of its results including suggestions on how to deal with the real-life datasets through the Orthogonal Experiment Design technique. Section 5 includes conclusions and ideas for future research.

2. Related Work

To meet the required properties of the online learners several approaches and techniques have been proposed in the literature. The most successful ones include sampling, windowing, and drift detecting. Sampling assumes using only some data instances or some part of instances out of the available dataset. In [14] random sampling strategy with a probabilistic removal of some instances from the training set was proposed. Later on, the idea was extended in [15]. Some more advanced sampling strategies were proposed in [16]. Effects of sampling strategy on classification accuracy were investigated in [17].

As it has been observed in the review of [18], data sampling methods for machine learning have been investigated for decades. According to the above paper, in recent years progress has been made in methods that can be broadly categorized into random sampling including density-biased and nonuniform sampling methods, active learning methods, which are the type of semisupervised learning, and progressive sampling methods, which can be viewed as a combination of the above two approaches.

Closely related to sampling is the sliding window model. Sliding window can be seen a subset that runs over an underlying collection. Several versions of the approach can be found in [19–21]. The idea is that analysis of the data stream is based on recent instances only and a limited number of the data instances, usually equal to the window size, are used to induce a classifier. In machine learning, the concept can be used for incremental mining of association rules [22]. Another interesting application of the sliding window technique is known as the high utility pattern mining [23].

For noisy environments or environments with a concept drift the key question is when and how the current model should be adopted. Possible solutions include explicit drift detection models (see the survey by Ditzler et al. [24]) or explicit partitioning approaches (see, for example, [25]).

One of the most successful approaches to incremental mining of data streams is using the drift detection techniques. The aim of the drift detection is to identify changes in statistical properties of data distribution over time. Such changes are often referred to as the concept drift. To minimize deterioration of learners accuracy caused by the concept drift, one can apply change detection tests and modify or replace a learner upon discovering the drift (see, for example, [26, 27]). The above-described approach is known as an active solution as opposed to a passive one, where the model is constantly retrained based on the most recent sample. More recently several Extreme Learning Machine (ELM) approaches to incremental learning have been discussed. For example, [28] proposed a forgetting parameters concept named FP-ELM. Recent surveys on data stream mining can be found in [24, 29].

Among incremental models, there are also those based on exploiting a power of the ensemble classifiers. Ensemble learners involve a combination of several models. Their predictions can be combined in some manner like, for example, averaging or voting to arrive at the final prediction. Ensemble learners for the data stream mining were proposed, among others, in [30–34].

One of techniques used to construct incremental classifiers is Gene Expression Programming (GEP). Gene Expression Programming was introduced in [35]. In GEP programs are represented as linear character strings of a fixed length called chromosomes which, in the subsequent fitness evaluation, evolve into expression trees without any user intervention. This feature makes GEP-induced expression trees a convenient model for constructing classifiers [36].

An improvement of the basic GEP classifiers can be achieved by combining GEP-induced weak classifiers into a classifier ensemble. In [37] two well-known ensemble techniques, bagging and boosting, were used to enhance the generalization ability of GEP classifiers. Yet another approach to building GEP-based classifier ensembles was proposed in [38]. The idea was to construct weak (base) classifiers from different subsets of attributes controlling the diversity among these subsets through applying a variant of niching technique. Further extensions and variants of GEP-induced ensemble classifiers were discussed in [39] where ideas of incremental learning and cluster-based learning were proposed. Approaches to constructing ensemble classifiers from GEP-induced weak classifiers were also studied in [40].

3. The Proposed Incremental GEP-Based Classifier

In this paper, we extend and improve the incremental GEP-based classifier proposed in [41]. In the above paper, GEP was used to induce base classifiers. Base classifiers serve to construct an ensemble of classifiers. Such an ensemble requires the application of some integration techniques like for instance majority voting, bagging or boosting. Review of the ensemble construction methods for the online learning can be found in [42]. Alternatively, a metaclassifier can be constructed following the idea of the stacked generalization [43]. In our case, such a metaclassifier is called a metagene.

Our approach follows steps proposed in [41] as far as the construction of base classifiers and respective metagenes are concerned. The algorithm for learning the best classifier using GEP works as follows. Suppose that a training dataset is given and each vector in the dataset has a correct label representing

0	1	2	3	4
OR	(>, 1, 0.57)	NOT	(≤, 10, 0.16)	···

Figure 1: Single gene example.

the class. In the initial step, the minimal and maximal values of each attribute are calculated and a random population of chromosomes is generated. Each chromosome is composed of a single gene divided into two parts as in the original head-tail method [35]. The size of the head (h) is determined by the user with the suggested size not less than the number of attributes in the dataset. The size of the tail (t) is computed as $t = h + 1$. The size of the chromosome is $h + t = 2h + 1$. For each gene, the symbols in the head part are randomly selected from the set of functions AND, OR, NOT, XOR, and NOR and the set of terminals of type (op; $attrib$; $const$), where the value of $const$ is in the range of attribute $attrib$ and op is a relational operator. The symbols in the tail part are all terminals. In Figure 1 an example of a gene is given. The start position (position 0) in the chromosome corresponds to the root of the expression tree (OR, in the example). Then, below each function branches are attached and there are as many of them as the arity of the function, 2 in our case. The following symbols in the chromosome are attached to the branches on a given level. The process is complete when each branch is completed with a terminal. The number of symbols from the chromosome to form the expression tree is denoted as the termination point. For the discussed example, the termination point is 4; therefore further symbols are not meaningful and are denoted by ··· in Figure 1. The rule corresponding to the chromosome from Figure 1 is

IF ($attribute1 > 0.57$) OR NOT ($attribute10 \leq 0.16$) THEN Class 1.

To introduce variation in the population the following genetic operators are used:

(i) mutation,

(ii) transposition of insertion sequence elements (IS transposition),

(iii) root transposition (RIS transposition),

(iv) one-point recombination,

(v) two-point recombination.

Mutation can occur anywhere in the chromosome. We consider one-point mutation which means that with a probability, called mutation rate, one symbol in a chromosome is changed. In case of a functional symbol it is replaced by another randomly selected function; otherwise for $g = (op, attrib, const)$ a random relational operator op', an attribute $attrib'$, and a constant $const'$ in the range of $attrib'$ are selected. Note that mutation can change the respective expression tree since a function of one argument may be mutated into a function of two arguments or vice versa.

Transposition stands for moving part of a chromosome to another location. Here we consider two kinds of transposable elements. In the case of transposition of insertion sequence (IS) three values are randomly chosen: a position in the chromosome (start of IS), the length of the sequence and the target site in the head, a bond between two positions. Then a cut is made in the bond defined by the target site and the insertion sequence is copied into the site of the insertion. The sequence downstream from the copied IS element loses, at the end of the head, as many symbols as the length of the transposon. Observe that since the target site is in the head, the newly created individual is always syntactically correct though it can reshape the tree quite dramatically. In the case of root transposition, a position in the head is randomly selected, the first function following this position is chosen; it is the start of the RIS element. If no function is found, then no change is performed. The length of the insertion sequence is chosen. The insertion sequence is copied at the root position and at the same time the last symbols of the head (as many as RIS length) are deleted.

For both kinds of recombination two parent chromosomes P_1, P_2 are randomly chosen and two new child chromosomes C_1, C_2 are formed. In the case of one-point recombination, one position is randomly generated and both parent chromosomes are split by this position into two parts. Child chromosomes C_1 (respectively, C_2) is formed as containing the first part from P_1 (respectively, P_2) and the second part from P_2 (and P_1). In two-point recombination two positions are randomly chosen and the symbols between recombination positions are exchanged between two parent chromosomes forming two new child chromosomes. Observe that again, in both cases, the newly formed chromosomes are syntactically correct no matter whether the recombination positions were taken from the head or tail.

During GEP learning, the individuals are selected and copied into the next generation based on their fitness and the roulette wheel sampling with elitism which guarantees the survival and cloning of the best chromosome in the next generation.

Further details on GEP operators and GEP learning can be found in [39, 40, 44].

For a fixed training set TR and fixed gene g the fitness function counts the proportion of vectors from TR classified correctly:

$$fit_{TR}(g) = \frac{\sum_{rw \in TR,\, g(rw)\ is\ true} sg(\text{rw is from class 1})}{|TR|} \quad (1)$$

where

$$sg(\varphi) = \begin{cases} 1 & \text{if } \varphi \text{ is true} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Having generated a population of genes it is possible to create a population of metagenes which corresponds to creating an ensemble classifier. The idea is as follows. Let pop be a population of genes, with each gene identified by its id. To create metagenes from pop we define the set of functions again as Boolean ones as above and set terminals equal to identifiers

0	1	2	3	4
OR	AND	$g1$	$g2$	$g3$

Figure 2: Single metagene example.

of genes. For example, the metagene mg shown in Figure 2 makes use of three genes g1, g2, and g3.

$$g1: \begin{array}{c|c|c} 0 & 1 & 2 \\ \hline \text{AND} & (=,2,0.8) & (=,3,2.5) \end{array}$$

$$g2: \begin{array}{c|c|c|c} 0 & 1 & 2 & 3 \\ \hline \text{AND} & \text{NOT} & (=,3,0) & (<,2,0) \end{array} \quad (3)$$

$$g3: \begin{array}{c|c} 0 & 1 \\ \hline (=,1,0) & \ldots \end{array}$$

For a fixed attribute vector rw each terminal (i.e., gene) has a Boolean value and thus the value of metagene can be computed. For the metagene mg from Figure 2 and $rw = (1.2, 0.8, 2.5)$ we have

$$\begin{aligned} g1(rw) &= true, \\ g2(rw) &= false, \\ g3(rw) &= false, \\ mg(rw) &= true \end{aligned} \quad (4)$$

Similarly as in (1), for a fixed training set TR and fixed metagene mg the fitness function counts the proportion of vectors from the testing set classified correctly:

$$FIT_{TR}(mg) = \frac{\sum_{rw \in TR,\, mg(rw)\ is\ true} sg(\text{rw is from class 1})}{|TR|} \quad (5)$$

The incremental GEP classifier with metagenes works in rounds. In each round, a chunk of data is used to induce genes and another chunk to induce metagenes. Chunk size is one of the incremental classifier parameters. Its role is to control the frequency with which the model is updated with a view to adapt to a possible concept drift. Main assumptions for such an approach are as follows:

(i) Class labels of instances belonging to the first and second chunks are known at the outset

(ii) Class labels of instances belonging to the chunk number 3, and to all the following chunks, are immediately revealed after the class of each instance has been predicted

(iii) All instances except those belonging to the first two chunks are classified one by one in the "natural" order

Based on the above assumptions, in [2], the following procedure was implemented. In each round a chunk of training data c_1 is used to create a population of genes, next chunk of data c_2 is used to create the population of metagenes and to choose one best-fitted metagene denoted mg, and the following chunk c_3 is tested by metagene mg. In the next round, $c_1 := c_2$, $c_2 := c_3$ and next chunk is used as c_3. For further comparisons, the incremental classifier from [2] is denoted as Inc-GEP1.

Computational experiments confirmed that Inc-GEP1 performs quite well. Comparison with the state-of-the-art incremental classifiers showed that the approach outperforms, in the majority of cases, the existing solutions in terms of the classification accuracy. Unfortunately, Inc-GEP1 suffers from a high demand on computational resources which, in many situations, might prevent it from mining data streams and datasets from the big data environment. One of the reasons behind the above situation is that Inc-GEP1 has not been equipped with any adaptation mechanism providing for updating the model only upon detecting a concept drift. Instead, the model is induced anew each time after classifying a chunk of instances.

To offer more flexibility and to shorten the computation time as compared with Inc-GEP1 we propose two measures. The first is an extensive data reduction option, and the second is providing some adaptation mechanism with a view to decreasing the number of required learner updates during computations. Following the idea of the random sampling proposed for the classic (nonincremental) learners [41], in the proposed incremental learner, the user has an option to set values of the following main parameters:

(i) Chunk size (ch)

(ii) Number of the base classifiers (NB)

(iii) Number of attributes used to induce the base genes (NA)

(iv) Percent of instances used to induce the base genes (RB)

(v) Percent of instances used to induce metagenes (RM)

Each of the above options can be used to control and effectively decrease or increase the computation time of the whole process, including learning models and predicting class labels of the incoming instances. Setting value of the chunk size determines how often the learner is updated. Smaller size results in increasing the number of updates. In our case, this number can be decreased through the proposed adaptation mechanism described later in this section. The number of base classifiers used to induce metagenes influences computation time needed to perform the job. A smaller number of the base classifiers may, however, decrease the accuracy of the resulting metagenes. The number of attributes used to induce base genes should be smaller than the number of original attributes in each instance of the considered dataset. Once set, it results in selecting randomly as many attributes as required from the set of all data attributes. The random draw of attributes takes place each time when one of the base classifiers is induced. This means that for inducing each base classifier a combination of attributes is repeatedly randomly drawn. Setting percent of instances used to induce the base genes and metagenes results in randomly sampling chunks used to induce the base genes and metagenes, respectively.

Input: chunk C, number of base classifiers NB, number of attributes NA, percent of instances RB
Output: the population of base classifiers BC
(1) $BC \longleftarrow \emptyset$
(2) **for** $i \longleftarrow 1$ **to** NB **do**
/* prepare chunk for learning */
(3) $CF \longleftarrow C$ filtered onto NA attributes chosen randomly
(4) $NI \longleftarrow$ size(CF)×RB
(5) $CN \longleftarrow \emptyset$
(6) **for** $i \longleftarrow$**to** NI **do**
(7) select random row r from CF
(8) add row r to CN
(9) apply GEP learning to CN ([2])
(10) add the best gene to base classifiers BC
(11) **return** BC

Algorithm 1: Inducing base classifiers.

Input: chunk C, base classifiers BC, percent of instances RM
Output: best metagene mg
/* prepare chunk for learning metagene */
(1) $NI \longleftarrow$ size(C)×RM
(2) $CN \longleftarrow \emptyset$
(3) **for** $i \longleftarrow 0$ **to** NI **do**
(4) select random row r from C
(5) add row r to CN
(6) apply metagene learning to CN and base classifiers BC ([2])
(7) select best metagene mg
(8) **return** mg

Algorithm 2: Inducing metagene.

Such filtering results in diminishing the number of instances used to induce each of the base classifiers and each of metagenes, by a given percentage.

Apart from the data reduction measures, we also propose to introduce a simple adaptation mechanism reducing unnecessary learner updates. After having used the first two data chunks to induce the initial set of base classifiers and the current metagene (mg), the following scheme is used. Class labels of instances belonging to the third chunk c_3 are predicted using mg and the average accuracy of class prediction for that chunk (av_3) is recorded. In the next step, mg is used to predict class labels of the fourth chunk c_4 and the average accuracy of prediction av_4 is calculated and recorded. If $av_4 < av_3$, then the learner is updated using c_3 and c_4 producing new current mg. Else, the current metagene is used to predict class labels of instances belonging to the next incoming chunk. The procedure is repeated until instances in all chunks have been classified. Wherever the inequality $av_i < av_{i-1}$ holds, the current metagene is replaced by a new one induced using chunks c_i and c_{i-1}. The above adaptation mechanism is denoted as ADAPT1. Alternatively, the second version of the adaptation mechanism, denoted as ADAPT2, can be used. Under ADAPT2 the current metagene is replaced by a newly induced one only after the average classification accuracy for two consecutive chunks is worse than the accuracy produced by the metagene induced for their predecessor chunk. The procedure using ADAPT1 is shown as Algorithm 3 and the case for ADAPT2 is omitted, as being similar. The incremental classifier with data reduction and ADAPT1 mechanism is further on referred to as Inc-GEP2. Such classifier equipped with ADAPT2 mechanism is further on referred to as Inc-GEP3.

Procedures for inducing base classifiers and metagenes are shown as Algorithms 1 and 2, respectively. In both cases, the fitness function is an accuracy of the class label prediction calculated over the respective chunk of data.

4. Computational Experiment Results

To evaluate the performance of the proposed approach we have carried out the computational experiment over a representative group of the publicly available 2-classes benchmark datasets including large datasets and datasets often used to test incremental learning algorithms. Datasets used in the experiment are shown in Table 1.

In Table 2 experiment settings used in Inc-GEP2 and Inc-GEP3 are shown. There are 4 main parameters affecting the proposed classifiers performance. Chunk size refers to the number of instances classified one by one without interruption using the current metagene. The number of attributes

```
Input: dataset D, chunk size ch, number of base classifiers NB
Output: overall prediction accuracy
/* induce NB base classifiers using the first chunk and best metagene
   using the second chunk                                              */
(1) dataTrain ⟵ first ch rows from D
(2) dataTrainM ⟵ next ch rows from D
(3) apply Algorithm 1 to dataTrain to induce NB base classifiers BC
(4) apply Algorithm 2 to dataTrainM and BC to induce metagene mg
(5) dataTest ⟵ next ch rows from D
(6) ac ⟵ accuracy of classification performed on dataTest by metagene mg
(7) acV ⟵ ac
(8) while rows in D not considered yet do
(9)     dataTestN ⟵ next ch rows from D
(10)    acN ⟵ accuracy of classification performed on dataTestN by metagene mg
(11)    acV ⟵ acV + acN
(12)    dataTrain ⟵ dataTrainM
(13)    dataTrainM ⟵ dataTest
(14)    dataTest ⟵ dataTestN
(15)    if acN < ac then
            /* metagene updated by new learning                        */
(16)        apply Algorithm 1 to dataTrain to induce base classifiers BC
(17)        apply Algorithm 2 to dataTrainM to induce metagene mg
(18)    ac ⟵ acN
(19) noC ⟵ number of chunks −2
(20) acV ⟵ acV/noC
(21) return acV
```

Algorithm 3: Incremental classifier with data reduction and ADAPT1 adaptation mechanism.

refers to the number of randomly selected attributes used to induce each gene. Reduction rate reflects the percent of both instances used to induce genes and instances used to induce metagenes. Number of classifiers refers to the number of base classifiers (genes). Method of setting values of the above parameters is explained later. Other settings including the number of iterations in GEP (set at 100) and probabilities of applying genetic operators (set as in [2]) have been the same throughout the whole experiment.

In Table 3 mean classification accuracy of Inc-GEP1, Inc-GEP2, and Inc-GEP3 is shown. Accuracy and standard deviation have been calculated as mean values obtained over 20 runs with parameter settings as shown in Table 2. For the Inc-GEP1 chunk size and the number of attributes are identical as in the case of the Inc-GEP2 and Inc-GEP3. In Inc-GEP1, however, there is no reduction with respect to the percentage of genes used to induce base classifiers and metagenes. Additionally, in Inc-GEP1 base classifiers and metagenes are induced using the full set of attributes.

Parameter values shown in Table 2 have been selected through the Orthogonal Experimental Design (OED) method. Since there are four main factors affecting classifier performance, it has been decided to use an L9 orthogonal array to identify the influence of 4 different independent variables on classifier performance. For each variable 3 level values have been set. Selection of the level values was arbitrary, albeit based on common sense.

The decision to use the OED method has been preceded by a comparison of mean classification accuracy values for each dataset and each combination of main factors out of 9 combinations under analysis. Thus, for each dataset, we had 9 groups of samples, each containing 10 classification accuracies obtained by running the considered classifier for 10 times for each combination of factors. The one-way ANOVA with the null hypotheses stating that samples in all groups are drawn from populations with the same mean values has

Table 1: Benchmark datasets used in the experiment. The table is reproduced from [2] (under the Creative Commons Attribution License/public domain).

Dataset	Source	Instances	Attributes
Airlines	[3]	539383	8
Bank M.	[4]	45211	10
Banknote Auth	[4]	1372	5
Breast Cancer	[4]	263	10
Chess	[5]	503	9
Diabetes	[4]	768	9
Electricity	[6]	45312	6
Heart	[4]	303	14
Image	[4]	2086	19
Internet Adv	[4]	3279	1559
Ionosphere	[4]	351	35
Luxemburg	[5]	1901	32
Sea	[7]	5000	4
Usenet2	[7]	1500	100

Table 2: Experiment settings for algorithms Inc-GEP2 and Inc-GEP3.

Dataset name	Chunk size	No of attrib.	Reduction rate (%)	No of classifiers
Airlines	10000	4	50	20
Bank M	400	13	10	20
Banknote Auth.	120	4	80	50
Breast cancer	30	4	20	20
Chess	50	7	50	20
Diabetes	60	3	80	30
Electricity	4000	4	50	30
Heart	30	11	90	20
Image	100	15	50	20
Internet Adv.	500	500	70	60
Ionosphere	60	30	50	20
Luxemburg	50	11	80	30
Sea	2500	2	10	30
Usenet2	20	60	50	20

Table 3: Computational experiment results (mean accuracy and standard deviation, %).

Dataset name	Inc-GEP1		Inc-GEP2		Inc-GEP3	
	Accuracy	+/-	Accuracy	+/-	Accuracy	+/-
Airlines	62.56	2.339	63.79	2.029	61.24	3.108
Banknote auth	93.07	1.436	93.01	1.356	93.41	1.199
Bank M	93.31	1.867	88.79	0.461	89.98	0.985
Breast cancer	82.13	0.720	74.38	0.654	76.37	0.536
Chess	85.94	1.676	84.16	0.897	77.34	0.549
Diabetes	85.01	0.932	62.13	0.245	65.33	0.453
Electricity	88.13	3.247	95.39	1.298	92.67	1.541
Heart	83.91	1.127	78.95	1.457	77.33	0.914
Image	86.6	2.358	79.04	1.298	75.58	1.598
Internet Adv.	91.47	0.793	95.67	0.033	94.92	0.055
Ionosphere	92.9	1.002	89.61	0.972	87.06	0.839
Luxemburg	100.00	0.000	100.00	0.000	100.00	0.000
Sea	81.12	1.393	83.28	0.356	84.56	0.541
Usenet2	78.15	2.362	74.24	1.885	69.78	1.177

shown that, for all considered datasets with the exception of the Bank Marketing dataset, null hypotheses should be rejected. This finding assures sensibility of searching for the best combination of factor values for each of the considered datasets.

The procedure of the orthogonal experiment and selection of the parameter values is shown below on the example of the Sea dataset. The similar procedure has been applied to all considered datasets.

In Table 4 factor (term) levels for the orthogonal array used in the experiment with the Sea dataset are shown. In Table 5 response values representing classification accuracy using the Inc-GEP2 classifier are displayed. The first column shows factor level numbers. Next ten columns contain response values. The last column contains the average of responses.

Response table for signal-to-noise ratio shown in Table 6 indicates that key role in maximizing the discussed ratio plays the number of attributes while data in Table 7 showing the response table for classification accuracy means indicate that key factor in maximizing accuracy plays the number of classifiers and next the number of attributes. The response table for signal-to-noise ratios contains a row for the average signal-to-noise ratio for each factor level, Delta, and rank. Delta is the difference between the maximum and minimum average response for the factor. The response table for means shows the size of the effect by taking the difference between the highest and lowest characteristic average for a factor. Ranks in a response table allow to quickly identify which factors have the largest effect. All factors, however, have statistically significant effects on response. This is confirmed by the main effect plot for means shown in Figure 3. Main effect plot is constructed by plotting the means for each value of a variable. A line connects the points for each variable. When the line is horizontal (parallel to the x-axis), there is no main effect present. The response mean is the same across

Table 4: Factor levels for the orthogonal array used in the experiment with the Sea dataset.

Factor level	Window size	No of attrib.	Reduction rate (%)	No of classifiers
1	5000	3	30	30
2	2500	2	20	20
3	100	1	10	10

Table 5: Experiment response results (Sea dataset, Inc-GEP2 classifier).

Levels	1	2	3	4	5	6	7	8	9	10	AV
1,1,1,1	0.816	0.802	0.798	0.803	0.806	0.797	0.820	0.815	0.820	0.791	0.807
1,2,2,2	0.815	0.813	0.804	0.800	0.802	0.794	0.803	0.820	0.811	0.816	0.808
1,3,3,3	0.736	0.720	0.744	0.690	0.735	0.735	0.737	0.743	0.711	0.710	0.726
2,1,2,3	0.823	0.837	0.839	0.818	0.828	0.834	0.837	0.832	0.821	0.803	0.827
2,2,3,1	0.837	0.835	0.832	0.831	0.833	0.826	0.831	0.837	0.830	0.833	0.833
2,3,1,2	0.756	0.727	0.742	0.758	0.759	0.736	0.742	0.758	0.742	0.761	0.748
3,1,3,2	0.796	0.812	0.799	0.801	0.809	0.812	0.807	0.812	0.791	0.793	0.803
3,2,1,3	0.793	0.771	0.780	0.804	0.809	0.802	0.808	0.819	0.800	0.805	0.799
3,3,2,1	0.759	0.771	0.785	0.765	0.740	0.758	0.761	0.752	0.774	0.737	0.760

Table 6: Response Table for signal-to-noise ratios: Sea dataset.

Level	Window	Attributes	Reduction	Class.no
1	-2.080	-2.563	-2.092	-2.128
2	-1.920	-1.798	-1.962	-2.093
3	-2.166	-1.806	-2.113	-1.946
Delta	0.246	0.765	0.151	0.182
Rank	2	1	4	3

Table 7: Response table for means: Sea dataset.

Level	Window	Attributes	Reduction	Class.no
1	0.9214	0.9142	0.9242	0.9145
2	0.9402	0.9366	0.9334	0.9329
3	0.9254	0.9362	0.9294	0.9396
Delta	0.0189	0.0224	0.0092	0.0251
Rank	3	2	4	1

all factor levels. On the other hand, when the line is not horizontal, there is a main effect present and the response mean is not the same across all factor levels. The steeper the slope of the line, the greater the magnitude of the main effect.

As data in Table 5 indicate, the best combination of factor levels for the Sea dataset is window (chunk) size 2500, 2 attributes for inducing base classifiers, 10% of instances used to induce genes and, respectively, metagenes, and 30 classifiers. Similar analysis has been performed for all considered datasets with a view to find out the best combination of parameter (factor) values.

Orthogonal array analysis can be also carried out with respect to the computation times. For example, in the case of the Sea dataset the respective response table for computation times means indicates that key role in minimizing computation time plays the window size and number of classifiers used to construct the ensemble. The respective main effects plot displaying how the considered factors affect computation times for the Sea dataset is shown in Figure 4. In this Figure "means" refers to times in seconds needed to classify a single instance. In Table 8 comparison between mean computation times for all the considered dataset and for settings of parameters from Table 2 is shown. Respective values refer to times in seconds needed to classify 100 instances by the considered algorithms run on Dell Precision 3520 workstation with Xeon processor and 16 GB RAM. Columns Speed-up1 and Speed-up2 contain speed-up factors comparing Inc-GEP1 with Inc-GEP2 and Inc-GEP1 with Inc-GEP3, respectively. As can be observed from Table 8, there are significant differences in computation times needed to run algorithms under comparison. On average, the proposed Inc-GEP2 classifier is over 2 times quicker as compared with the incremental Gene Expression Programming with metagenes without data reduction (Inc-GEP1). Moreover, the proposed Inc-GEP3 classifier is, on average, over 7 times quicker than the control algorithm Inc-GEP1. To properly evaluate both Inc-GEP2 and Inc-GEP3 one has to evaluate also their performance in terms of the classification accuracy. Assuming equal variances, one-way ANOVA allows observing that null hypothesis stating that all three mean accuracies are equal under the confidence level 0.05 holds. Hence, the alternative hypothesis stating that not all the considered means are equal should be rejected. The above finding is confirmed by Fisher and Tukey tests.

In Table 9 comparison of the proposed GEP-based incremental classifiers with some literature reporting state-of-the-art incremental classifiers in terms of the mean classification accuracy is shown. The abbreviations used for incremental classifiers are as follows: FTDD, Fisher Test Drift Detection; IncSVM, Incremental SVM; EDDM, Early Drift Detection Method; IncN-B, Incremental Naïve Bayes; KFCM, Online distance based classifier with Kernel Fuzzy C-means; IncEnsemble, Incremental Ensemble; and FISH, Unified Instance Selection Algorithm.

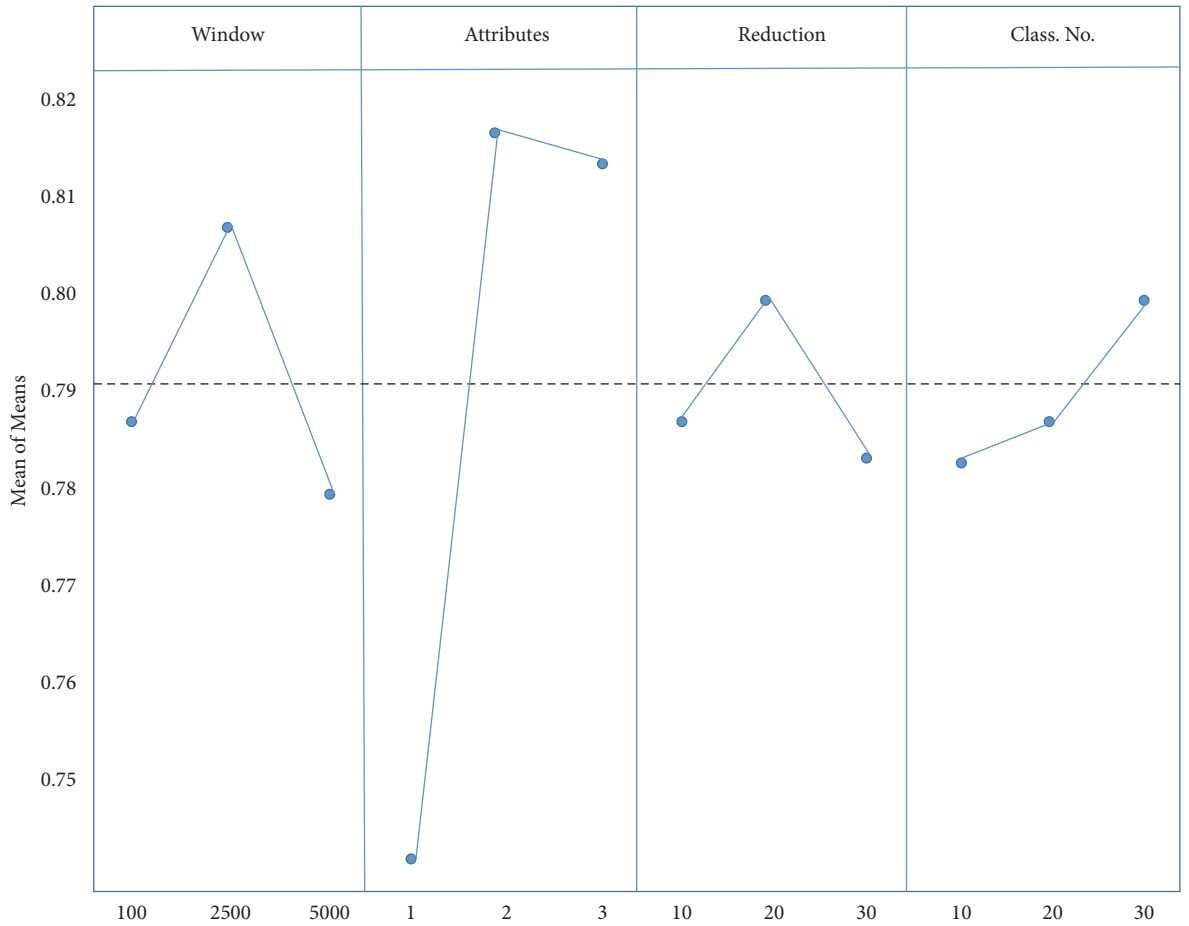


Figure 3: Main effects plot for classification accuracy means: Sea dataset.

Table 8: Mean computation times (seconds per 100 instances).

Dataset	Inc-GEP1	Inc-GEP2	Inc-GEP3	Speed-up1	Speed-up2
Airlines	1.07	0.39	0.16	2.7	6.7
Banknote auth.	6.63	4.30	1.09	1.5	6.1
Bank M.	0.75	0.39	0.20	1.9	3.8
Breast	26.24	11.03	1.14	2.4	23.0
Chess	5.37	3.58	2.39	1.5	2.3
Diabetes	4.56	3.65	2.86	1.3	1.6
Electricity	7.27	2.53	0.50	2.9	14.5
Heart	18.81	10.56	5.61	1.8	3.4
Image	14.43	2.92	0.91	4.9	15.8
Internet Ad	13.08	5.58	2.10	2.3	6.2
Ionosphere	10.54	7.69	3.70	1.4	2.8
Luxemburg	0.79	0.26	0.16	3.0	5.0
Sea	0.76	0.34	0.18	2.3	4.3
Usenet2	17.80	8.07	3.13	2.2	5.7

Table 9: Comparison of the proposed GEP-based incremental classifiers with some literature reporting incremental classifiers in terms of the mean classification accuracy.

Dataset	Inc-GEP1	Inc-GEP2	Literature reported acc	Incremental classifier	Source
Airlines	63.79	61.24	65.44	FTDD	[8]
Bank M.	88.79	89.98	86.90	IncSVM	[9]
Breast C.	74.38	76.37	72.20	IncSVM	[10]
Chess	84.16	77.34	71.80	EDDM	[11]
Diabetes	62.13	65.33	75.70	IncN-B	[12]
Electricity	95.39	92.67	90.70	KFCM	[13]
Heart	78.95	77.33	83.80	IncSVM	[10]
Ionosphere	89.61	87.06	92.40	IncEnsemble	[12]
Luxemburg	100.00	100.00	88.11	FISH2	[5]

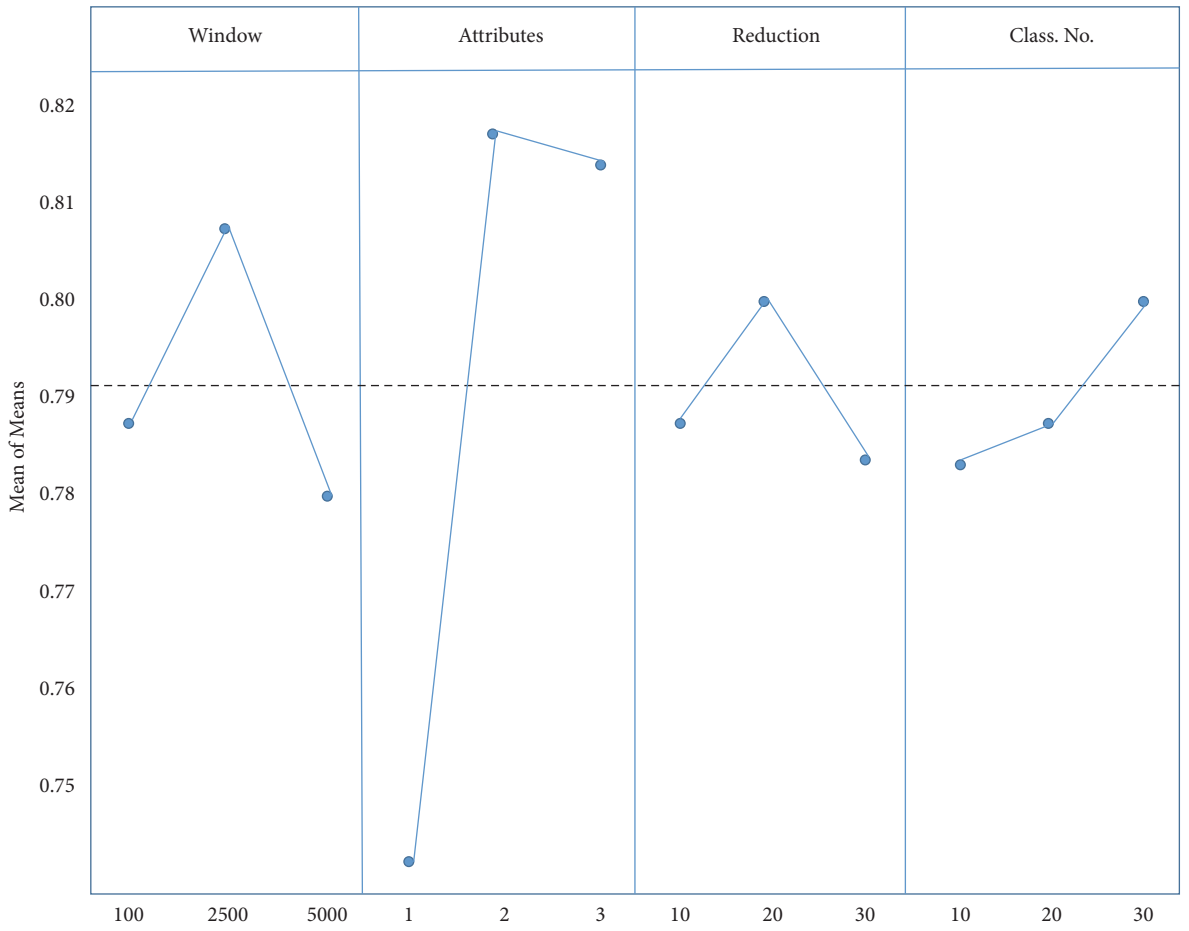


Figure 4: Main effects plot for classification time means: Sea dataset.

From Table 9 it can be seen that the proposed classifiers perform well and are competitive to several other approaches. In several cases, GEP-based incremental classifiers outperform earlier available solutions.

5. Conclusions

The main contribution of the paper is to propose the incremental Gene Expression Programming classifier with metagenes and data reduction. The concept of metagenes increases the classification accuracy while data reduction allows controlling computation time. The proposed approach extends earlier incremental GEP-based classifier [2]. Additionally, the extended version contains a simple drift detection mechanism allowing dealing more effectively with data streams.

Another important novelty introduced in the paper is using the Orthogonal Experimental Design principles to set up classifier parameters values. The approach allows us to easily evaluate the statistical importance of main parameters (factors) showing through main effects plots and the

respective response tables key factors and their influence on classifier performance and signal-to-noise ratios.

An extensive computational experiment confirms that the proposed classifier offers better performance in respect to the required computation times as compared with its earlier version. At the same time, it provides similar results in terms of classification accuracy. The algorithm offers also scalability through the possibility of adjusting computation times to the user needs, which might be a useful feature even at a cost of possibly a bit lower classification accuracy.

Comparison of the proposed GEP-based incremental classifiers with some literature reporting state-of-the-art incremental classifiers in terms of the mean classification accuracy proves that our approach offers quite satisfactory solutions, outperforming in many cases the existing methods. The proposed approach can be useful in data analytics and big data processing where single-pass limited-memory models enabling a treatment of big data within a streaming setting are increasingly needed [45].

Future research would concentrate on incorporating more sophisticated drift detection mechanisms and to further improve efficiency by implementing the algorithm in a parallel environment.

Acronyms and Abbreviations

GEP:	Gene Expression Programming
IS:	Sequence insertion
RIS:	Root transposition
TR:	Training set
pop:	Population of genes
mg:	Metagene
fit:	Fitness function for genes
FIT:	Fitness function for metagenes
ch:	Chunk size
NB:	Number of base classifiers
NA:	Number of base classifiers
RB:	Percent of instances used to induce base classifiers
RM:	Percent of instances used to induce metagenes
BC:	Base classifiers
ADAPT1, ADAPT2:	Adaptation procedure in two versions
Inc-GEP1:	Incremental classifier
Inc-GEP2:	Incremental classifier with adaptation ADAPT1
Inc-GEP3:	Incremental classifier with adaptation ADAPT2.

Data Availability

Previously reported datasets data were used to support this study and are publically available at UCI Machine Learning Repository (see [36]) with respect to Bank Marketing, Banknote Authentication, Breast Cancer, Diabetes, Heart, Image, Internet Advertisement, and Ionosphere. Airlines dataset is publically available at Open Machine Learning site (https://www.openml.org/). Chess and Luxemburg datasets are available from Indre Zliobaite (see [4]). Electricity dataset is publically available from UCI Repository–Massive Online Analysis (see [15]). SEA and Usenet2 datasets are publically available from Joaquin Vanschoren et al. (see [41]). These datasets are cited at relevant places within the text.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. I. Kuncheva, "Classifier ensembles for changing environments," in *Proceedings of the 5th International Workshop Multiple Classifier Systems MCS '04*, F. Roli, J. Kittler, and T. Windeatt, Eds., vol. 3077 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Berlin, Germany, 2004.

[2] J. Jedrzejowicz and P. Jedrzejowicz, "Incremetal GEP-Based Ensemble Classifier," in *Intelligent Decision Technologies 2017*, vol. 72 of *Smart Innovation, Systems and Technologies*, pp. 61–70, Springer International Publishing, Cham, 2018.

[3] Airlines dataset, 2017.

[4] M. Lichman, "Uci machine learning repository," 2013.

[5] I. Žliobait, "Combining similarity in time and space for training set formation under concept drift," *Intelligent Data Analysis*, vol. 15, no. 4, pp. 589–611, 2011.

[6] Massive Online Analysis, Uci machine learning repository, 2013.

[7] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, Openml: networked science in machine learning, 2014.

[8] D. R. Cabral and R. S. Barros, "Concept drift detection based on Fisher's exact test," *Information Sciences*, vol. 442/443, pp. 220–234, 2018.

[9] K. Wisaeng, "A comparison of different classification techniques for bank direct marketing," *International Journal of Soft Computing and Engineering*, vol. 3, no. 4, pp. 116–119, 2013.

[10] L. Wang, H.-B. Ji, and Y. Jin, "Fuzzy Passive-Aggressive classification: A robust and efficient algorithm for online classification problems," *Information Sciences*, vol. 220, pp. 46–63, 2013.

[11] I. Žliobaitė, "Controlled Permutations for Testing Adaptive Classifiers," in *Discovery Science*, vol. 6926 of *Lecture Notes in Computer Science*, pp. 365–379, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[12] S. B. Kotsiantis, "An incremental ensemble of classifiers," *Artificial Intelligence Review*, vol. 36, no. 4, pp. 249–266, 2011.

[13] J. Jędrzejowicz and P. Jędrzejowicz, "Distance-based online classifiers," *Expert Systems with Applications*, vol. 60, pp. 249–257, 2016.

[14] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software*, vol. 11, no. 1, pp. 37–57, 1985.

[15] S. Chaudhuri, R. Motwani, and V. Narasayya, "On Random Sampling over Joins," *SIGMOD Record*, vol. 28, no. 2, pp. 263–273, 1999.

[16] J. Xu, Y. Y. Tang, B. Zou, Z. Xu, L. Li, and Y. Lu, "The generalization ability of online svm classification based on markov sampling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 628–639, 2015.

[17] M. S. Esfahani and E. R. Dougherty, "Effect of separate sampling on classification accuracy," *Bioinformatics*, vol. 30, no. 2, pp. 242–250, 2014.

[18] A. ElRafey and J. Wojtusiak, "Recent advances in scaling-down sampling methods in machine learning," *Wiley Interdisciplinary Reviews. Computational Statistics (WIREs)*, vol. 9, no. 6, e1414, 13 pages, 2017.

[19] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Sliding window-based frequent pattern mining over data streams," *Information Sciences*, vol. 179, no. 22, pp. 3843–3865, 2009.

[20] M. Deypir, M. H. Sadreddini, and S. Hashemi, "Towards a variable size sliding window model for frequent itemset mining over data streams," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 161–172, 2012.

[21] H. Chen, L. Shu, J. Xia, and Q. Deng, "Mining frequent patterns in a varying-size sliding window of online transactional data streams," *Information Sciences*, vol. 215, pp. 15–36, 2012.

[22] C. Lee, C. Lin, and M. Chen, "Sliding-window filtering: an efficient algorithm for incremental mining," in *Proceedings of the 10th International Conference on Information and Knowledge Management, CIKM '01*, pp. 263–270, ACM, New York, NY, USA, 2001.

[23] H. Ryang and U. Yun, "High utility pattern mining over data streams with sliding window technique," *Expert Systems with Applications*, vol. 57, pp. 214–231, 2016.

[24] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: a survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[25] J. P. Fan, J. Zhang, K. Z. Mei, J. Y. Peng, and L. Gao, "Cost-sensitive learning of hierarchical tree classifiers for large-scale image classification and novel category detection," *Pattern Recognition*, vol. 48, no. 5, pp. 1673–1687, 2015.

[26] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers - Part I: Detecting nonstationary changes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 7, pp. 1145–1153, 2008.

[27] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence—SBIA 2004: Proceedings of the 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29–Ocotber 1, 2004*, vol. 3171 of *Lecture Notes in Computer Science*, pp. 286–295, Springer, Berlin, Germany, 2004.

[28] D. Liu, Y. Wu, and H. Jiang, "FP-ELM: An online sequential learning algorithm for dealing with concept drift," *Neurocomputing*, vol. 207, pp. 322–334, 2016.

[29] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[30] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 139–147, Paris, France, July 2009.

[31] P. Zhang, X. Zhu, J. Tan, and L. Guo, "Classifier and cluster ensembles for mining concept drifting data streams," in *Proceedings of the IEEE International Conference on Data Mining*, pp. 1175–1180, 2010.

[32] I. Czarnowski and P. Jędrzejowicz, "Ensemble classifier for mining data streams," in *Proceedings of the 18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES '14*, P. Jedrzejowicz, L. C. Jain, R. J. Howlett, and I. Czarnowski, Eds., vol. 35 of *Procedia Computer Science*, pp. 397–406, Elsevier, Gdynia, Poland, 2014.

[33] X.-C. Yin, K. Huang, and H.-W. Hao, "De2: Dynamic ensemble of ensembles for learning nonstationary data," *Neurocomputing*, vol. 165, pp. 14–22, 2015.

[34] D. Mejri, R. Khanchel, and M. Limam, "An ensemble method for concept drift in nonstationary environment," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1115–1128, 2013.

[35] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[36] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, vol. 21 of *Studies in Computational Intelligence*, Springer, Berlin, Germany, 2006.

[37] Q. Li, W. Wang, S. Han, and J. Li, "Evolving classifier ensemble with gene expression programming," in *Proceedings of the 3rd International Conference on Natural Computation, ICNC '07*, vol. 3, pp. 546–550, China, 2007.

[38] W. Jiang, T. Changjie, Z. Jun et al., "An attribute-oriented ensemble classifier based on niche gene expression programming," in *Proceedings of the 3rd International Conference on Natural Computation, ICNC '07*, vol. 3, pp. 525–529, China, August 2007.

[39] J. Jędrzejowicz and P. Jędrzejowicz, "A family of gep-induced ensemble classifiers," in *Proceedings of the 1st International Conference Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems, ICCCI '09*, N. T. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds., vol. 5796 of *Lecture Notes in Computer Science*, pp. 641–652, Springer Berlin Heidelberg, 2009.

[40] J. Jędrzejowicz and P. Jędrzejowicz, "Experimental evaluation of two new GEP-based ensemble classifiers," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10932–10939, 2011.

[41] J. Jedrzejowicz and P. Jedrzejowicz, "Gene expression programming ensemble for classifying big datasets," in *Proceedings of the Computational Collective Intelligence - 9th International Conference, ICCCI '17*, T. N. Ngoc, A. George, P. Jedrzejowicz, B. Trawinski, and G. Vossen, Eds., vol. volume 10449 of *Lecture Notes in Computer Science*, pp. 3–12, Springer, Berlin, Germany, 2017.

[42] A. Fern and R. Givan, "Online ensemble learning: an empirical study," *Machine Learning*, vol. 53, no. 1-2, pp. 71–109, 2003.

[43] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[44] J. Jedrzejowicz and P. Jedrzejowicz, "Gep-induced expression trees as weak classifiers," in *Proceedings of the 8th Industrial Conference Advances in Data Mining, Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, ICDM '08*, P. Perner, Ed., vol. 5077 of *Lecture Notes in Computer Science*, pp. 129–141, Springer, Berlin, Germany, 2008.

[45] B. Hammer, H. He, and T. Martinetz, "Learning and modeling big data," in *Proceedings of the 22th European Symposium on Artificial Neural Networks, ESANN '14*, 2014.

Hindawi
Complexity
Volume 2018, Article ID 7404627, 13 pages
https://doi.org/10.1155/2018/7404627

Research Article

An Approach to Data Reduction for Learning from Big Datasets: Integrating Stacking, Rotation, and Agent Population Learning Techniques

Ireneusz Czarnowski and Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland

Correspondence should be addressed to Ireneusz Czarnowski; irek@am.gdynia.pl

Received 26 June 2018; Revised 3 September 2018; Accepted 16 September 2018; Published 5 November 2018

Academic Editor: Sergio Gómez

In the paper, several data reduction techniques for machine learning from big datasets are discussed and evaluated. The discussed approach focuses on combining several techniques including stacking, rotation, and data reduction aimed at improving the performance of the machine classification. Stacking is seen as the technique allowing to take advantage of the multiple classification models. The rotation-based techniques are used to increase the heterogeneity of the stacking ensembles. Data reduction makes it possible to classify instances belonging to big datasets. We propose to use an agent-based population learning algorithm for data reduction in the feature and instance dimensions. For diversification of the classifier ensembles within the rotation also, alternatively, principal component analysis and independent component analysis are used. The research question addressed in the paper is formulated as follows: does the performance of a classifier using the reduced dataset be improved by integrating the data reduction mechanism with the rotation-based technique and the stacking?

1. Introduction

Big data, so far, does not have a formal definition, although it is generally accepted that the concept refers to datasets that are too large to be processed using conventional data processing tools and techniques. Contemporary information systems produce data in huge quantities that are difficult to be measured [1]. It means that we already have found ourselves in the "big data era," and the question of how to solve large-scale machine learning problems is open and requires a lot of research efforts. Dealing with huge datasets poses a lot of the processing challenges. The big data sources including contemporary information systems and databases contain inherently complex data characterized by the well-known 5V properties: huge volume, high velocity, much variety, big variability, low veracity, and high value [2].

The big data applications involve four major phases: data generation, data management, data analytics, and data application. The data analytics is the most important phase, where the aim is to discover patterns from data. However, in the big data era, the task is not trivial and much more complicated than normal-sized data analytics [3]. This becomes especially troublesome in numerous critical domains like security, healthcare, finance, and environment protection, where obtaining a dependable knowledge of different processes and their properties is crucial to the social welfare.

Learning from data is an example of the most important data analytics problem, where machine learning algorithms are used. The aim of the machine learning is to expand algorithms that are able to learn through experience [4]. The algorithms, called learners, can improve their performance based on analysis of the collected data, which are called examples [5], and which are collected from the environment. Today, machine learning offers a wide range of tools and methods that can be used to solve a variety of data mining problems. Their common weakness is, however, the

so-called dimensionality curse, making them inefficient or even useless when solving large-scale problems. Thus, achieving scalability, low computational complexity, and efficient performance of the machine learning algorithms have become hot topics for the machine learning community.

Since traditional techniques used for analytical processing are not fit to effectively deal with the massive datasets, searching for new and better techniques, methods, and approaches suitable for big data mining is a hot area for the machine learning community. Considering the above facts and observing current trends in the machine learning research, it can be observed that among main contemporary challenges, the most important one is a search for improvements with respect to scalability and performance of the available algorithms. Among techniques for dealing with massive datasets are different parallel processing approaches aiming at achieving a substantial speed-up of the computation. Examples of such techniques are Hadoop and MapReduce techniques which have proven suitable for the computation and data intensive tasks [6].

The scalability and performance issues lead to the two simple questions: "how fast?" and "how large?," that is, how fast one can get a solution and how large is a dataset one can effectively deal with. In this paper, we focus on the question of "how large?," and we analyze approaches to deal with big data. In reference to a short discussion on fundamental strategies for big data analytics included in [3], the following approaches are currently considered as the most promising ones:

(i) Divide-and-conquer

(ii) Parallelization

(iii) Sampling

(iv) Granular computing

(v) Feature selection

Divide-and-conquer is a well-known strategy based on processing small chunks of data and then fusing separated results together.

Parallelization concerns dividing a large problem into several smaller ones which can be solved concurrently in parallel, producing, in the end, the final result.

Sampling is a well-known statistical technique based on the probability theory. The approach is based on identifying a relationship between the sample and the population. With the advent of the big data era, many new sampling techniques have emerged or have been modified including simple random sampling, systematic sampling, stratified sampling, cluster sampling, quota sampling, and minimum-maximum sampling [7].

Granular computing is a technique using granules to build an efficient computational model for complex applications in the big data environment. Examples of these granules are classes, clusters, subsets, groups, and intervals. From the implementation point of view, the technique reduces the data size through analyzing data at different levels of granularity [8].

Feature selection is a technique for dimensionality reduction in a feature space [9]. The aim of the feature selection is to obtain a representative subset of features that has fewer features in comparison to the original feature set. Several different techniques have been proposed for feature selection, so far. The feature extraction technique is one of the possible approaches.

The above-described strategies are in line with techniques proposed to achieve better scalability of the machine learning algorithms. In [10], such techniques were classified into the three categories. The first includes extensions and modification of the traditional machine learning tools. The second is based on the problem decomposition into a set of smaller or computationally less complex problems. The third involves using parallel processing where possible. In this paper, we use the idea of the problem decomposition. The paper is an extension of the earlier research results included in [11] and presented during the 2017 IEEE INISTA Conference. The extension involves an improvement of the stacking and rotation procedures allowing for either deterministic or random transformations in the feature space. The above option improves the performance of the procedure. The paper also refers to and offers some extensions of the research results included in other papers of the authors' [12–14].

The paper considers an approach dedicated to reducing the dimensionality in data, so this also means that it is dedicated to working with large datasets, with a view to enabling an efficient machine learning classification in terms of a high classification accuracy and an acceptable computation time. To achieve the above, the following techniques are used:

(i) Data reduction based on the prototype selection through the instance and feature selections from clusters of instances

(ii) Stacking

(iii) Rotation-based

(iv) Agent-based population learning algorithm for data reduction

The research question addressed in the paper is formulated as follows: does the performance of a classifier over the reduced dataset be improved by integrating the data reduction mechanism with the rotation-based technique and the stacking? In [11], to diversify the classifier ensembles, the rotation-based techniques using principal component analysis for feature selection have been implemented. In this paper, the alternatively independent component analysis method and feature selection based on an agent-based population learning algorithm implementation is used. We also propose to use an agent-based population learning algorithm for data reduction in the instance dimension. The techniques used have been integrated, and an adaptive approach to constructing the machine classifiers is proposed. The approach is validated experimentally by solving selected classification problems over benchmark datasets from UCI and the KEEL repositories [15, 16].

The paper is organized as follows. A brief review of the stacking, rotation, and data reduction is included in the next section. The following section provides a detailed description of the proposed approach. Next, computational experiment carried out, including its plan and results, is described and discussed. The final section focuses on conclusions and ideas for further research.

2. Techniques for Improving Performance for Big Data Analytics

In this section, a brief review of the data reduction techniques, the rotation-based technique and the agent-based population learning algorithm (PLA), as a background for further consideration, is offered.

2.1. Data Reduction. Reducing the quantity of data aims at selecting pertinent information only as an input to the data mining algorithm. Thus, data reduction identifies and, eventually, leads to discarding information which is irrelevant or redundant. Ideally, after data reduction has been carried out, the user has to do with datasets of smaller dimensions representing the original dataset. It is also assumed that the reduced dataset carries the acceptable or identical amount of information as the original dataset.

Data reduction aim is not losing extractable information but to increase the effectiveness of the machine learning when the available datasets are large [4]. It is the most critical component in retrieving information from big data in many data mining processes [17].

Reducing data size may cover for the unwanted consequences of scaling up. Among such consequences, specialists list excessive memory requirements, increasing computational complexity and deteriorating learning performance [17].

In practice, data dimensionality reduction is concerned with selecting informative instances and features from the training dataset. In the literature on data reduction, quite often, instance and feature selections are addressed separately. There exist also approaches where both tasks are solved simultaneously as a dual selection problem [18]. Data reduction can be also merged with the problem of the prototype extraction.

The prototype extraction problem also aims at reducing the dimensionality of the training set by replacement of the existing instances by the extracted ones. Extracting prototypes from the original dataset may also include constructing new features. In such case, a smaller number of features are constructed from the original feature set through certain transformation operations [19]. A well-known tool for carrying such transformation is the principal component analysis (PCA) [20].

More on the data reduction problem as well as a review of the proposed approaches including instance selection, feature selection, and the dual dimension data reduction can be found among others in [2, 9, 14, 21–23].

Formally, the data reduction process aims at finding the reduced dataset S_{opt}, which is the subset of the original dataset D, such that the performance criterion of the machine learning algorithm L is maximized. From the above perspective, the performance of the classifier induced from the reduced dataset should be better or at least not worse than the classifier induced from the original dataset [24].

In this paper, the approach to data reduction is proposed as a tool for dimensionality reduction of the original dataset and is carried out in both dimensions (instance and feature). Moreover, in this paper, the implementation of data reduction is an example of the idea of data partitioning (as suggested in [10]), as well as an exemplification of the strategy of granular computing (in a sense proposed in [3]).

2.2. Stacked Generalization. Stacked generalization also known as stacking was proposed by Wolpert [25]. The technique was designed to improve classification algorithm performance.

Stacking is an example of a sampling strategy and is one of the ensemble learning techniques. The idea of stacking is based on combining the multiple classifications or regression models via a metaclassifier or a metaregressor.

In stacking, the base learners consist of different learning algorithms, so the stacking ensembles are often heterogeneous. Performance of the stacking-based classifiers is competitive in comparison with learners using bagging and boosting techniques. Besides, stacking allows for combining learners of the different types, which is not the case in bagging and boosting. Stacked generalization can be implemented using one of the two modes for combining, the so-called, base classifiers or combining their output. In the first mode, outputs from base classifiers are combined to obtain the final classification decision. In the second mode, base classifiers are used to construct the metamodel used for predicting unknown class labels.

In the vast literature on the stacked generalization, there are two basic approaches to combining base classifiers. The first one assumes combining, at a higher level, outputs from the base classifiers to obtain classification decision. Alternatively, at a higher level, base classifiers are integrated into the metamodel, subsequently used to predict unknown class labels.

In the standard stacking approach, at first q, different instance subsets of equal size are generated using a random generator. It is assumed that the subsets will be generated in such a way that assures relative proportion of instances from the different classes like it is observed in the original dataset. In the next step, omitting one of the subsets in each iteration, the so-called level-0 classifiers are generated from the remaining subsets. The process is repeated q times following the pattern of the q-fold cross-validation procedure. At each iteration, the omitted subset of instances is used to generate the so-called level-1 set of instances. Thus, the level-0 models produce predictions that form the input to the level-1 model. They are used to predict the class label for new instances with unknown class labels. In the approach, the metaclassifier in the form of relative weight for each level-0 classifier is created by assigning weights to classifiers proportional to their performance. The schema for metaclassifier induction has a form of the so-called leave-one-out cross-validation [25].

Thus, combining classifiers under the umbrella of stacking can be seen as follows. Supposing that there are q different learners $L_1, \ldots, L_q$ and q different training sets, $D_1, \ldots, D_q$, where $D = D_1 \cup D_2 \ldots \cup D_q$ and D is the original training set. Each learner is induced from training sets $D_1, \ldots, D_q$, respectively. As the result, we have the output hypotheses $h_1, \ldots, h_q$, where $\forall h_{i:i=1,\ldots,q} \in H$ and H is a hypothesis space, which is defined as a set of all possible hypothesis, that the learner can draw. Thus, the goal of stacking is to learn a well-combined classifier h such that the final classification will be computed from $h_1(x), \ldots, h_q(x)$ as shown in the equation:

$$h(x) = \sum_{i=1}^{q} w_i h_i(x), \tag{1}$$

where vector w represents the respective weights.

Different variants of stacking have been proposed so far. A review of the stacking algorithms is included, for example, in [25] or [26].

In this paper, the stacking technique used has been inspired by Skalak's proposal [27], where the prototype selection on the level-0 of the stacking is carried out as the mean for data reduction. Next, the outputs of the level-0 are used for generating the metaclassifier at the level-1. In this paper, we also assume that the data reduction is carried out through prototyping and that prototypes are selected from the clusters, which are induced during the carried out data analysis. Stacking plays the role of the sampling strategy paradigm and helps with achieving a diversification of the level-0 models.

2.3. Rotation-Based Technique. The rotation-based technique belongs to the family of the ensemble methods, while in turn, the ensemble methods can be seen as meta-algorithms. The rotation-based technique combines several machine learning techniques into one predictive model aiming at improving the machine learning performance. The rotation-based technique belongs to the class of the multiple classifier systems (MCS) described in [28]. The idea behind the rotation-based ensembles (RE) is to use the rotation operator to project or transform the original dataset into a new feature space. From such feature space, new features are extracted. To implement the approach, the following two steps are executed. First, the original dataset is projected into a new feature space. From such space, at the second step, feature subsets are selected, and base individual classifiers are induced. The procedure is expected to improve the classification accuracy as compared with the traditional approach. It is known that the approach is usually effective when classifying high dimensional data [29].

Well-known example of the RE is the rotation forest (RF) algorithm. Rotation forest extends the idea of the random forest, which combines the bagging and the random subspace methods [30]. Random forest consists of a number of decision trees trained based on the example bootstraps sampled from the original training set. Each subset of the training dataset is modified by selecting randomly a subset of features. The RF procedure starts with the feature extraction from the input data followed by training of each decision tree in a different new rotated space. The process results in achieving, at the same time, a high individual accuracy and the required diversity among the ensemble members. Four feature extraction methods, principal component analysis (PCA), maximum noise fraction (MNF), independent component analysis (ICA), and local fisher discriminant analysis (LFDA), have been applied in the rotation forest [30, 31].

In [23], feature extraction and data transformation were based on the principal component analysis (PCA). How exactly to apply PCA depends on the user. One possible way is to apply it to a subset of features only. In such case, one has to split the original set of features to a number of subsets associating with each subset a subset of instances through the axis rotation [32]. The approach suffers from one drawback. Since PCA is a deterministic algorithm, it may generate the ensemble with members characterized by the identical set of features. To avoid such a situation, some diversification mechanisms like, for example, removing some instances from the dataset are often used [33].

The experimental results show that the rotation forest is on the average superior and can produce more accurate results than bagging, AdaBoost, random subspace, and random forest [29, 31].

In the proposed approach, generation of base classifiers through feature rotation has been integrated with stacking and data reduction. It is shown experimentally that such an integration assures better diversification of the base classifier ensemble and, consequently, better classification performance. Two approaches are applied to the feature space modification. In the first one, the original RF algorithm is used. In the second case, the feature space is modified through solving the respective optimization problem using the agent-based population learning algorithm (described in the next subsection).

2.4. Agent-Based Population Learning Algorithm. The agent-based population learning algorithm seems to be a promising tool for solving complex computational problems arising in the big data environment. During the last years, the idea of implementing the agent-based approaches for the big data analytics is a hot topic. Examples and exchange of ideas in the above respect can be found in a special issue of *Web Intelligence and Agent Systems: An International Journal* [34]. The subject has been also discussed during the international conferences (for example, Metaheuristics International Conference, IEEE/WIC/ACM International Conference on Intelligent Agent Technology). The implementation of the agent-based approach has been also a subject of the paper [35]. An agent-based paradigm and the example case study have been also discussed in the context of applying the big data analytics in retailing [36].

Recent advances in distributed problem solving and agent-based data mining confirm that both techniques can help work with data extracted from the distributed and online environments. Agent-based technologies offer a variety of mechanisms that can significantly reduce costs of processing a large volume of data and improve data processing

Generate the initial population of solutions (individuals) and store them in the common memory
Implement different improvement procedures executed by the optimization agents
Activate optimization agents
While (stopping criterion is not met) **do** {in parallel}
  Read randomly selected individual from the common memory
  Execute improvement algorithm
  Store the improved individual back in the common memory
End while
Take the best solution from the population as the final result.

Algorithm 1: Agent-based population learning algorithm.

quality. A number of emerging technologies have been proposed for processing huge datasets by employing the multiagent systems. For example, the agent-based techniques can help solve the information overload problems [37]. Furthermore, agent-based applications can be of help in evaluating the quality of big data [38].

In [35], as well as in the following papers of the authors (see, for example, [39, 40]), it has been shown that the agent-based approach can help in solving difficult optimization problems. It is well known that data reduction belongs to the class of the combinatorial optimization problems and as such is computationally difficult. Hence, to solve the problem, some metaheuristics or other approximate algorithms are required. Numerous approaches to data reduction through instance selection have been based on using genetic or evolutionary algorithms (see, for example, [15, 41–43]).

A brief review of different approaches for instance selection can be found [4]. A broad review of the evolutionary algorithm applications to feature selection is available in [44].

This paper deals with the implementation of the agent-based population learning algorithm (PLA) to data reduction. The agent-based population learning algorithm has been proposed in [39] and belongs to the family of metaheuristics.

The PLA has been already used for solving problems of learning from data [35]. In [14], the stacking ensemble approach has been proposed for the purpose of improving the quality of the agent-based data reduction algorithm. In [11], the implementation has been extended using the rotation-based techniques. In the mentioned paper, the goal was to find the effective classification tool, which uses data reduction and which guarantees the maximization of the classification quality criterion.

The agent-based population learning algorithm is based on the A-Team architecture. The A-Team concept was originally introduced in [45]. It was motivated by several approaches like blackboard systems and evolutionary algorithms, which have proven to be able to successfully solve some difficult combinatorial optimization problems.

The functionality of the algorithm based on the implementation of the agent-based population learning approach can be defined as the organized and incremental search for the best solution. The agent-based population learning algorithm involves a specialized team of agents working asynchronously and in parallel, executing various improvement procedures with a view to solving the problem at hand. Agents working in the A-Team achieve an implicit cooperation by sharing the population of solutions to the problem to be solved. A-Team can be also defined as a set of agents and a set of memories, forming a network in which every agent remains in a closed loop. Agents cooperate to construct, find, and improve solutions which are read from the shared common memory. More information on the PLA and more details on the implementation of A-Teams can be found in [39]. The pseudocode of the agent-based population learning approach is shown as Algorithm 1.

3. The Proposed Approach to Learning from Big Datasets

3.1. Problem Formulation. It is well known that data reduction belongs to the class of the combinatorial optimization problems and as such is computationally difficult. Hence, to solve the problem, some metaheuristics or other approximate algorithms are required. Numerous approaches to data reduction through instance selection have been based on using genetic or evolutionary algorithms (see, for example, [15, 42, 43]).

In this paper, to enable dealing with huge datasets and to make the learning process more effective, it has been decided to apply the dual data reduction, that is, a reduction in the feature and instance spaces. It has been assumed that the resulting classifier will perform better either in terms of the computational resources required or in terms of classification accuracy or in respect to both criteria. Formally, dual data reduction can be viewed as searching for the dataset S which is the subset of the set D and $|S| < |D|$ (possibly $S = S_{\text{opt}}$), where each data instance belonging to S is represented by the set of original or transformed features A' with $|A'| < |A|$.

The proposed approach is based on the integration of the data reduction and learning stages with a view to improving the final classifier performance. Such an integration allows introducing some adaptation mechanisms into the learning process. The idea has been described in a more detailed manner in [35]. Such integrated learning has proven effective for assuring the required diversification among prototypes using the stacking technique [13]. A general model of the integrated learning is shown in Figure 1.

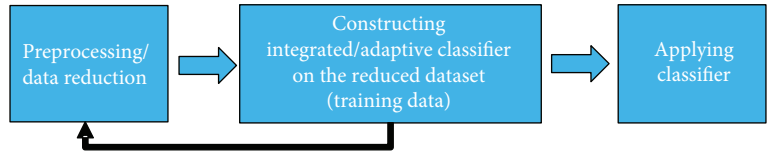


Figure 1: Integrated and adaptive learning from examples with data reduction.

Integrated and adaptive learning uses the positive feedback whereby more effective data reduction leads to a higher learning accuracy, and in return, higher learning accuracy results in even more effective data reduction.

Assume that the goal of learning from examples is to find a hypothesis h. The learner used to produce h requires the setting of some parameters decisive from the point of view of its performance. Let parameters g describe the way the training set should be transformed before training. Thus, it can be said that the goal of learning from examples is to find a hypothesis $h = L(D, g)$, where parameters g affect the learning process and influence the performance measure f. In such case, the learning task takes the following form:

$$h = \arg \max_{h \in H, g \in G} f(h = L(D, g)), \tag{2}$$

where G is the parameter space.

3.2. The Proposed Approach. In the proposed approach, it has been assumed that the learner is induced from prototypes. Prototypes, also referred to as reference instances, are represented by instances from the original dataset which have been selected in the evolutionary process. Before the selection process is activated, instances from the original dataset are grouped into clusters, and each cluster has its own reference instances in the final compact representation of the original dataset. In fact, each cluster has exactly one representative (reference instance) in the final dataset.

The above assumptions make a selection of the clustering algorithm crucial to the effectiveness of the resulting learner. We use two such algorithms—clustering guided by the similarity coefficient (SC) and the kernel-based C-means clustering algorithm (KFCM).

Similarity coefficient–based clustering was proposed in [40]. The algorithm assumes that for each instance from the original dataset, a similarity coefficient is calculated. Instances with identical coefficient are grouped into a cluster. The number of clusters is determined by the number of different similarity coefficients among the original dataset instances.

The second clustering algorithm—KFCM—was proposed to deal with problems caused by the noise and sensitivity to outliers characterizing the classic fuzzy C-means clustering algorithm. KFCM transforms input data into a higher dimensional kernel space through a nonlinear mapping [46]. The procedure has been already successfully used for the prototype selection [14].

To further increase chances for achieving a satisfactory performance of the learner induced over the reduced dataset, it has been decided to use the stacked generalization method using stratified sampling with replacement.

To improve performance and generalization ability of the prototype-based machine learning classification, it was decided to use the stacking technique. The implementation of the stacking technique in the discussed approach means that the process of classification with data reduction is carried out within the procedure that at first creates q different subsets of the training data using stratified sampling with replacement. All subsets are generated assuring relative proportion of the different classes as in the original dataset. However, to assure the required diversity, at first, q-1 training sets are split into the independent subsets with different feature subsets. Next, using q-1 subsets of the training sets, the process of the feature space modification is run.

Another diversifying factor is using the rotation technique or, alternatively, selecting features applying the population learning algorithm. In this paper, the first method is named as deterministic, while the second one as nondeterministic. In the case of the deterministic variant of the approach, based on rotation, two feature extraction techniques including principal component analysis (PCA) or independent component analysis (ICA) have been proposed.

After the above steps have been carried out, the learner is induced from the reduced (final) dataset transformed and diversified through applying stacking and rotation procedures. The process is executed by the set of agents cooperating and acting within the agent-based population learning algorithm. After the clusters have been produced followed by generation of the diversified subsets of the training data through stacking and rotation, potential solutions, forming their initial population, are generated through randomly selecting exactly one single instance from each of the considered clusters. Thus, a potential solution is represented by the set of prototypes, i.e., by the compact representations of the original dataset. A feasible solution to the data reduction problem is encoded as a string consisting of numbers of the selected reference vectors.

Selection procedure of the representation of instances through population-based search is carried out by the team of optimizing agents. Each agent is an implementation of the local search procedure and operates on individuals. The instance selection is carried out for each cluster, and removal of the remaining instances constitutes the basic step of the instance selection process. In case of feature selection, the potential solutions are improved by removing or adding an attribute to the solution that constitutes a basic step of the feature selection process. More precisely, the implemented improvement procedures include local search with the tabu list for instance selection, simple local search for instance selection, local search with the tabu list for feature selection,

Input: Dataset D with the feature set A; number of iterations q (i.e. the number of stacking folds); natural number T (defined by the user); *option* – the Boolean parameter determining the type of the transformation in the feature space (deterministic or nondeterministic)
Output: $h_{it(i:i=1,\ldots,q;t:t=1,\ldots T)}$ – set of the base classifiers
Begin
Allocate randomly instances from D into q disjoint subsets $D_1, \ldots, D_q$.
For $i = 1$ **to** q **do**
Let $D_i^{'} = D - D_i$
Partition randomly the feature set A into T subsets $\{A_{it}: t \leq T\}$ obtaining subsets D'_{it}, each with the identical number of features, smaller than the number of features in the original dataset.
For $t = 1$ **to** T **do**
Generate training set D'_{it} with features A_{it}, through bootstrapping with the size of 75% of the original dataset.
If *option* **then**
Run **PCA** or **ICA** over the transformed D'_{it} and produce new training datasets D''_{it}, with features A'_{it}, using the axis rotation;
Else
Run the **PLA** for feature selection on D'_{it} and produce new training datasets D''_{it} described on the set A'_{it}.
End If
Partition D'_{it} into clusters using the **KFCM** procedure or **SC** procedure.
Run **PLA** for the prototype selection obtaining $S'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ (i.e. subsets of the selected prototypes).
Induce base classifier h_{it} based on $S'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ using D_i with features A'_{it}, as the testing set.
End for
End for
Return $h_{it}, \ldots, h_{qT}$.
End.

Algorithm 2: Stacked generalization with rotation.

and simple local search for instance and feature selections. The detailed description and the background of these procedures can be found in [35].

To sum up, the optimizing agent task is to search for a better solution upon receiving a current one. To perform such search, each optimizing agent is equipped with some heuristic or local search algorithm which is activated immediately after the solution to be improved has been received. In case the agent is not able to find a better solution, the current one is returned. Otherwise, an improved solution is returned to the common memory. Quality of solutions also referred to as their fitness is evaluated through estimating the performance of the base classifier under evaluation. This performance is measured in terms of the classification accuracy provided the learner has been induced using instances and features of the reduced dataset.

From the implementation point of view, the above-described process of searching for the best results using the agent-based population learning algorithm is carried out in parallel for q independent data reduction problems. In each stream of such search, a different dataset is processed, and the independent set of the prototypes is selected.

The process of searching for solutions is iterative with q iterations. In each iteration, the reference instances are selected, and the respective decision tree is induced. Such tree plays the role of the base classifier. Its evaluation is carried out using the subset of instances which, in the current iteration, has been removed from the original dataset with a view to serving as the temporary testing set. The procedure produces a number of heterogeneous base classifiers forming an ensemble. The final decision as to the unknown class label is taken at the upper level of the stacking scheme through a majority vote. Pseudocode of the proposed scheme for stacked generalization with rotation is shown as Algorithm 2.

The diversity of the obtained set of the base classifiers is assured by application of stacking and rotation methods resulting in varying the training sets at the learning stage. The majority voting paradigm leads to the final decision as to the class label of the considered instance. It is computed as shown in

$$h = \arg \max_{h_{it} \in H,\, g_{it} \in G} \sum_{i=1}^{q} \sum_{t=1}^{T} w_{it} f(h_{it} = L(D_{it}, g_{it})), \qquad (3)$$

where g_{it} are the reduced instances produced by stacking and rotation procedures for $D'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)} \subset D$, $h_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ are output hypotheses induced from training sets $D'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$, respectively, and w_{it} represents weights of the base classifiers induced at respective stacking levels.

The framework of the considered approach is shown in a graphic form in Figure 2.

However, the proposed approach is based on decomposition and involves the stacking and rotation, which can be carried out at random or in a deterministic way in feature space; the complexity of the approach is the sum of

(i) complexity depending on the number of iteration, i.e., the number of stacking folds—q

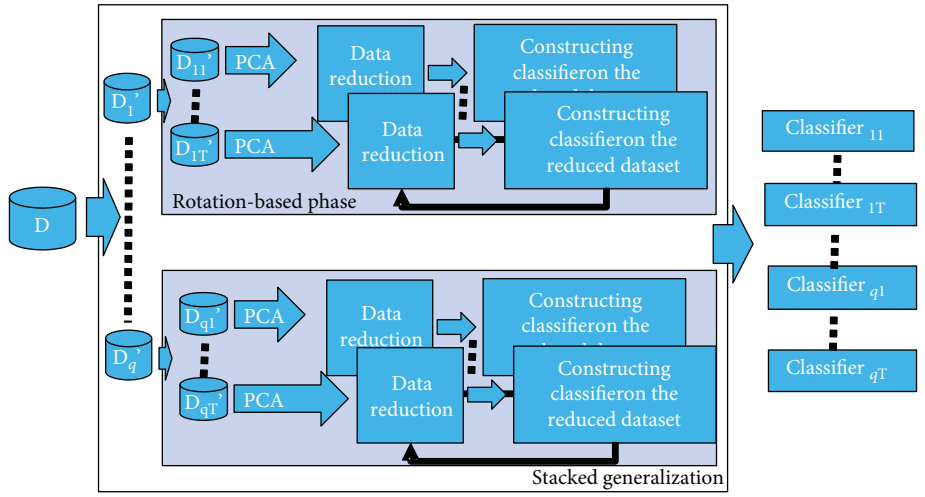


Figure 2: A framework of the proposed approach based on the PCA feature extraction.

(ii) the complexity of the selected rotation procedure, i.e.,

(a) the computational complexity of SC: $O(nN) + O(N \log N)$

(b) the computational complexity of KFCM: $O(tN^2n)$ [46]

(c) the computational complexity of the PLA, which especially depends on the computational complexity of the implemented improvement procedures,

where n denotes the number of features, N denotes the number of instances in the dataset, and t is the so-called number of eigenvalues

(iii) the complexity of the PLA implemented for the prototype selection—the complexity also depends on the computational complexity of the implemented improvement procedures

(iv) the complexity of the machine learning algorithm used to induce base classifier

4. Computational Experiment

4.1. Computational Experiment Setting. To validate the proposed approach, an extensive computational experiment has been planned and carried out. The experiment goals include searching for answers to the following questions:

(i) Can the instance selection be strengthened by using the rotation-based technique and the stacking?

(ii) How competitive is the proposed approach in comparison with the performance of the state-of-the-art classifiers?

(iii) Does the proposed approach produces, on average, better results than those produced by the earlier version of the algorithm introduced in [11], as well in [12–14, 35]?

In experiment, six following versions of the proposed algorithm have been considered:

(i) ABDRStEr$_{PCA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and PCA for feature extraction—introduced in [11]

(ii) ABDRkfStEr$_{PCA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and PCA for feature extraction—introduced in [11]

(iii) ABDRStEr$_{ICA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and ICA for feature extraction—a new version of the algorithm

(iv) ABDRkfStEr$_{ICA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and ICA for feature extraction—a new version of the algorithm

(v) ABDRStEr$_{PLA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and PLA for feature extraction—a new version of the algorithm

(vi) ABDRkfStEr$_{PLA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and PLA for feature extraction—a new version of the algorithm

Among other algorithms proposed by the authors and compared in this paper are

(i) ABInDRkfStE: agent-based integrated data reduction based on the KFCM with the stacking ensemble learning—introduced in [13]

Table 1: Characteristics of the datasets used in this paper.

Dataset	Source of data	Instances	Attributes	Classes	Best reported results classification accuracy
Heart	[15]	303	13	2	90.0% [15]
Diabetes	[15]	768	8	2	77.34% [15]
WBC	[15]	699	9	2	97.5% [15]
ACredit	[15]	690	15	2	86.9% [15]
GCredit	[15]	1000	20	2	77.47% [15]
Sonar	[15]	208	60	2	97.1% [15]
Shuttle	[15]	58,000	9	7	95.6% [42]
Connect-4	[16]	67,557	42	3	—
Magic	[16]	19,020	10	2	—
Census	[16]	142,521	41	3	—

(ii) ABInDRStE: agent-based integrated data reduction based on the similarity coefficient with the stacking ensemble learning—introduced in [13]

(iii) ABDRkfStE: agent-based data reduction based on the KFCM with stacking ensemble learning and without feature selection—introduced in [14]

(iv) ABDRStE: agent-based data reduction based on the similarity coefficient with stacking ensemble learning and without feature selection—introduced in [14]

(v) ABIS: agent-based instance selection—proposed in [35]

(vi) ABDRE: agent-based data reduction with ensemble with RM-RR (random move and replace randomly strategy)—proposed in [12]

(vii) ABDRE with RM-RW (random move and replaces first worst strategy): proposed in [12]

All proposed and all above-mentioned algorithms belong to the family of the integrated-based learning paradigm.

Computational experiment results produced by the proposed approach have been also compared with some other approaches based on different ensemble techniques (AdaBoost, bagging, and random subspace) proposed in [11]. In the experiment, several benchmark datasets from the UCI and KEEL repositories [15, 16] have been used (for details see Table 1). The criterion for fitness evaluation has been the classification accuracy (Acc.) understood as the correct classification ratio. The experiment involved several runs. The number of stacking folds has been set from 3 to 10, respectively. The number of bootstraps has been set to 4. For each considered dataset, the experiment plan has required 10 repetitions over the 10-cross-validation (10-C-V) scheme induced using C.45 or CART algorithms. Each set of the 10-C-V of runs has repeated for 50 times.

For experiment where searching for a solution has been carried out by A-Teams, the following A-Team parameters have been used: population size (40) and stopping criterion (100 iterations without an improvement or one minute of computations without such improvement). In the case of bagging and random subspace, the size of bags has been set to 50% of the original training set. The number of base models in ABDRE with RM-RR and ABDRE with RM-RW has been set to 40.

4.2. Experiment Results. Classification accuracies produced by the investigated approaches using all considered data sets are shown in Table 2. The results have been reported as averages over all runs of each algorithm and for each problem, respectively.

In general, in the case of the proposed approach, results shown refer to the number of stacking folds producing the best results. Among the proposed models, best performers, on the average, are approaches using the integrated learning paradigm and stacking (Algorithms 1–16). This conclusion is valid independent of the clustering procedure used. Only in one case, we notice better results obtained by AdaBoost (please see the result for Connect-4). It can be observed that the proposed algorithms are competitive in comparison to others, among them to the DROP4 algorithm. This observation answers positively the second question asked at the beginning of this section.

The experiment also confirms that the rotation technique can improve the quality of results (the rotation technique has been implemented within algorithms from 1 to 12). Although the algorithms based on the rotation assured the best results in four cases, we can conclude that the rotation can improve the learning based on instance selection with stacking. On the other side, observing all algorithms proposed by the authors, among them their earlier versions, we can conclude that the instance selection can be strengthened by using the rotation-based technique and the stacking, which answers positively the first question asked at the beginning of this section.

The aim of the paper was also to verify the benefits from the diversification by the rotation technique, and two deterministic methods have been used (i.e., PCA and ICA). Alternatively, selecting features applying the population

Table 2: Classification accuracy (%) and comparison of different classifiers.

#	Algorithm	Heart	Diabetes	WBC	ACredit	GCredit	Sonar	Shuttle	Connect-4	Magic	Census
1	$ABDRStEr_{PCA}$ (C4.5) [11]	92.4	79.15	**98.25**	91.31	**80.42**	90.02	98.15	47.95	72.5	**83.54**
2	$ABDRkfStEr_{PCA}$ (C4.5) [11]	92.78	80.12	97.04	92.61	79.03	89.54	98.65	48.04	72.05	82.67
3	$ABDRStEr_{ICA}$ (C4.5)	91.21	78.21	96.2	91.6	78.2	87.01	98.5	47.62	71.9	81.52
4	$ABDRkfStEr_{ICA}$ (C4.5)	90.91	80.3	97.04	90.48	77.16	86.5	98.2	46.7	70.56	80.3
5	$ABDRStEr_{PLA}$ (C4.5)	91.94	80.23	95.21	91.24	78.69	88.6	97.84	46.8	71.69	81.62
6	$ABDRkfStEr_{PLA}$ (C4.5)	92.05	79.21	94.47	92	80.1	88.14	98.63	47.56	72.23	82.6
7	$ABDRStEr_{PCA}$ (CART)	90.56	78.21	96.21	95.2	78.65	87.5	97.45	45.21	71.86	82.15
8	$ABDRkfStEr_{PCA}$ (CART)	91.52	79.05	97.06	**96.4**	79.8	86.9	97.63	46.63	70.03	82.42
9	$ABDRStEr_{ICA}$ (CART)	90.72	78	95.28	94.26	77.76	85.71	98.05	45.77	72	81.62
10	$ABDRkfStEr_{ICA}$ (CART)	91.32	79.11	95.98	92.42	76.58	86.65	97.17	46.34	70.69	80.45
11	$ABDRStEr_{PLA}$ (CART)	90.06	80.6	96.02	91.3	77.91	86	98.26	46.04	71.62	82.41
12	$ABDRkfStEr_{PLA}$ (CART)	91.3	79.33	95.72	91.61	77.08	85.59	98.48	46.9	72.5	81.98
13	ABInDRkfStE [13]	**93.01**	**80.71**	98.08	92.04	78.45	90.57	98.41	46.98	**72.59**	82.68
14	ABInDRStE [13]	92.87	79.84	98.13	91.89	80.24	**91.15**	98.73	47.23	71.84	82.05
15	ABDRkfStE [14]	90.45	75.15	96.91	90.78	77.41	80.42	**99.66**	46.07	71.6	81.57
16	ABDRStE [14]	92.12	79.12	96.91	91.45	80.21	85.63	98.75	46.14	71.08	81.07
17	ABDRE with RM-RR [12]	92.84	80.4	96.4	90.8	78.2	83.4	97.51	45.67	70.96	81.65
18	ABDRE with RM-RW [12]	90.84	78.07	97.6	89.45	76.28	81.75	97.74	44.6	69.84	80.45
19	ABIS [35]	91.21	76.54	97.44	90.72	77.7	83.65	95.48	45.02	70.02	81.69
20	AdaBoost	82.23	73.55	63.09	91.05	73.01	86.09	96.13	**54.16**	68.57	80.6
21	Bagging	79.69	76.37	95.77	85.87	74.19	76.2	95.27	44.68	70.69	80.04
22	Random subspace method	84.44	74.81	71.08	82.14	75.4	85.18	92.81	43.58	70.56	79.05
23	C 4.5	77.8	73	94.7	84.5	70.5	76.09	95.6	45.89	69.13	80.61
24	SVM	81.5	77	97.2	84.8	72.5	90.4	—	—	—	—
25	DROP4 [20]	80.9	72.4	96.28	84.78	—	82.81	—	—	—	—

learning algorithm (PLA) have been implemented. These diversification techniques have been implemented within algorithms from 1 to 12. Analyzing the experiment results, we can observe that in nine cases out of ten, the best results have been obtained by PCA. Only in one case, the best result has been obtained using the PLA. Comparing results obtained using PLA and ICA allows one to observe that in seven cases, the better results have been obtained by PLA. In three cases, the better results have been assured by ICA. Thus, it can be observed that the rotation technique based on the PCA performs better than using PLA and ICA, even if PLA outperforms ICA.

The performance of the proposed approach has been also evaluated with respect to the kind of method used for inducing the base classifier. The computational experiment results show that the C4.5 as a machine learning tool used for ensemble induction assured better generalization than algorithm CART.

The question of the performance of the proposed methods can be also formulated with respect to the kind of the clustering methods. As it has been mentioned before, the clustering algorithm can be crucial to the effectiveness of the resulting learner. In this case, the computational experiment results show that the most effective is clustering guided by the similarity coefficient (SC). SC has been six times more effective in comparison to the kernel-based C-means clustering algorithm. We also observe that the agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning has been more effective when the PCA for feature extraction was used.

To confirm and verify the obtained results, Friedman and Iman-Davenport's nonparametric ranking test has been carried out for comparison of the results. Results have been ranked, and the ranking of the results has been computed assigning to the best of them rank 1 and rank 23 to the worst one (the statistical analysis does not include results for SVM and DROP4). Figure 3 depicts average weights for each compared algorithm obtained by Friedman's test.

The tests have been carried out under the following hypotheses:

(i) H_0—null hypothesis: all of the 23 compared algorithms are statistically equally effective regardless of the kind of the problem

(ii) H_1—alternative hypothesis: not all algorithms are equally effective

Both analyses have been carried out at the significance level of 0.05. The respective value of the x^2 statistics for

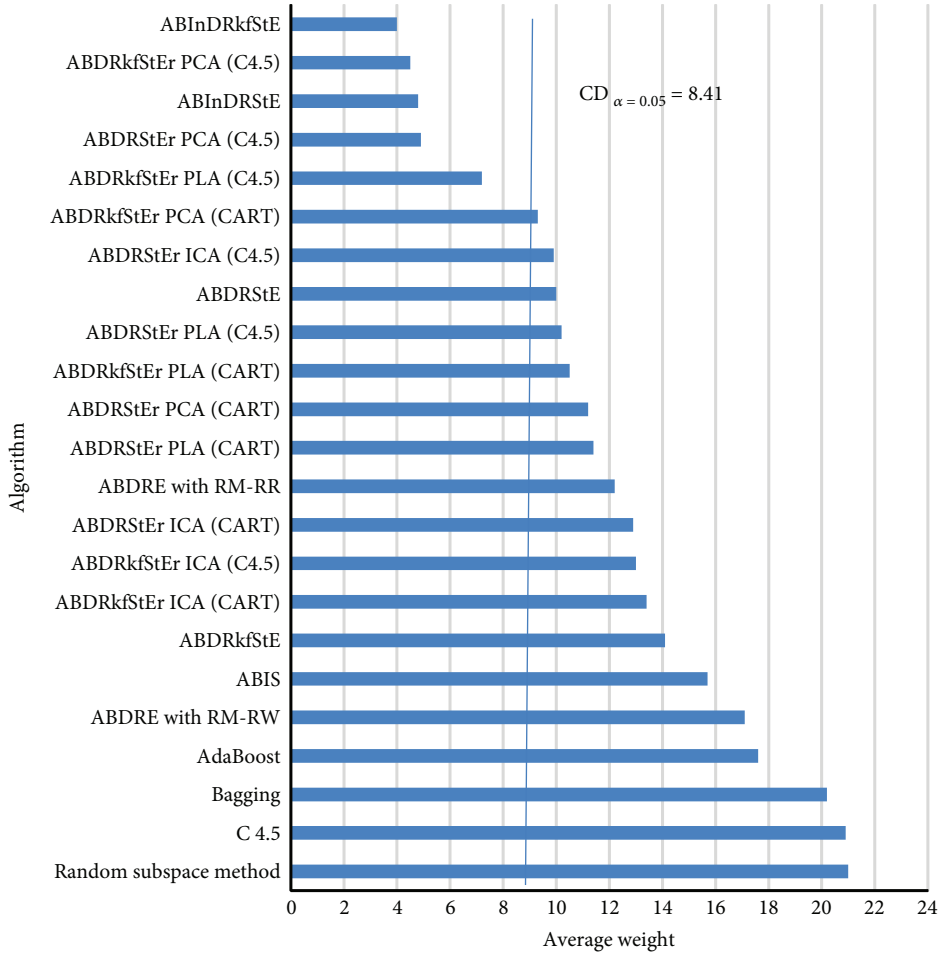


Figure 3: The average Friedman test weights and Bonferroni-Dunn's graphics corresponding to the obtained ranking.

Friedman's test with 23 algorithms and 10 instances of the considered problems is 124.8826087; the value of χ^2 of the distribution is equal to 33.92443847 for 22 degrees of freedom. The respective value F_F of Iman-Davenport's test is 9.763455; the critical value of $F(22.198)$ degrees of freedom is 1.59630281. For both tests, the p values are lower than the considered significance level $\alpha = 0.05$; thus, there are significant differences among the analyzed results, and the null hypothesis should be rejected. This means that not all algorithms are equally effective regardless of the kind of problem which instances are being solved.

A post hoc statistical analysis, based on Bonferroni-Dunn's test, to detect significant differences between the compared algorithms has been carried out. The critical difference (CD) of Bonferroni-Dunn's procedure is shown in Figure 3. The vertical cut line represents the threshold for the best performing algorithms. These bars which exceed the threshold are associated with algorithms displaying the worst performance with respect to the first five algorithms (ABInDRkfStE, $\text{ABDRkfStEr}_{\text{PCA}}$ (C4.5), ABInDRStE, $\text{ABDRStEr}_{\text{PCA}}$ (C4.5), and $\text{ABDRkfStEr}_{\text{PLA}}$ (C4.5)). These algorithms are better than the other versions with $\alpha = 0.05$.

To sum up the results of the statistical analysis, it can be concluded that the best results have been obtained

(i) by data reduction algorithms based on stacking and without rotation transformation in the feature space

(ii) by data reduction algorithms with stacking rotation ensemble learning and based on PCA for feature extraction independently on the cluster method used in the process of data reduction; however, when the KFCM has been used, the PLA was preferred

(iii) by data reduction based on integrated learning, which confirms our previous observation

The important factor of the research is that the proposed approach is based on decomposition by stacking, and the process of learning on the decomposition strategy is assured by the multiple agent system. It should be also underlined that the success of the learning process of the learning based on the PLA algorithm depends on the improvement procedures employed by the optimization agents.

5. Conclusions

The main scientific contribution of the paper is to propose an improvement to the core procedure of the proposed data reduction approach. The procedure integrates stacked generalization and rotation-based methods. The proposed algorithm allows for either deterministic or random transformations in the feature space. This feature was not available in the earlier algorithm proposed in [11]. It has been shown experimentally that the above option improves the performance of the procedure. The paper contributes also by proposing and evaluating a family of the hybrid classifiers based on data reduction, stacking, feature space rotation, and multiple agent environments. The proposed approach can be applied to mine huge datasets owing to quite radical data reduction mechanism and inherent parallelization typical for the multiple agent systems. It has been experimentally shown that merging stacking, rotation-based ensemble techniques, and data reduction with machine classification may bring the added value with respect to the accuracy of the classification process.

Future research will concentrate on searching for more effective local search procedures employed by the optimization agents. It is also envisaged to investigate different learners and different strategies with respect to the decision making within the classification ensemble. Finally, it would be also interesting to detect experimentally scaling up barriers for the proposed approaches.

Data Availability

The data used in this study are available at

(i) UCI Machine Learning Repository—http://archive.ics.uci.edu/ml/index.php

(ii) KEEL-dataset repository—http://sci2s.ugr.es/keel/datasets.php

Disclosure

The paper includes an extension of the research results presented earlier during the 2017 IEEE INISTA Conference. The earlier presented approach has been modified, new datasets have been used, and a broader analysis of the results has been carried out.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] N. García-Pedrajas and A. de Haro-García, "Scaling up data mining algorithms: review and taxonomy," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 71–87, 2012.

[2] M. H. U. Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, "Big data reduction methods: a survey," *Data Science and Engineering*, vol. 1, no. 4, pp. 265–284, 2016.

[3] X. Wang and Y. He, "Learning from uncertainty for big data: future analytical challenges and strategies," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 2, no. 2, pp. 26–31, 2016.

[4] S.-W. Kim and B. J. Oommen, "A brief taxonomy and ranking of creative prototype reduction schemes," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 232–244, 2003.

[5] T. Mitchell, *Machine Learning*, McGraw-Hill, New York, 1997.

[6] I. Triguero, M. Galar, H. Bustince, and F. Herrera, "A first attempt on global evolutionary unsampling for imbalanced big data," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2054–2061, San Sebastian, Spain, June 2017.

[7] S. L. Lohr, *Sampling: Design and Analysis*, Cengage. Learning, Boston, MA, USA, 2nd edition, 2009.

[8] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*, CRC Press/Francis Taylor, Boca Raton, FL, 2013.

[9] Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 551–577, 2017.

[10] N. García-Pedrajas and A. de Haro-García, "Boosting instance selection algorithms," *Knowledge-Based Systems*, vol. 67, pp. 342–360, 2014.

[11] I. Czarnowski and P. Jędrzejowicz, "Stacking and rotation-based technique for machine learning classification with data reduction," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 55–60, Gdynia, Poland.

[12] I. Czarnowski and P. Jędrzejowicz, "Agent-based data reduction using ensemble technique," in *Computational Collective Intelligence. Technologies and Applications*, C. Badica, N. T. Nguyen, and M. Brezovan, Eds., pp. 447–456, Springer, Berlin, Heidelberg, 2013.

[13] I. Czarnowski and P. Jędrzejowicz, "An approach to machine classification based on stacked generalization and instance selection," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4836–4484, Budapest, Hungary, October 2016.

[14] I. Czarnowski and P. Jędrzejowicz, "Learning from examples with data reduction and stacked generalization," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 2, pp. 1401–1411, 2017.

[15] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[16] J. Alcalá-Fdez, A. Fernández, J. Luengo et al., "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, 2011.

[17] A. A. Yıldırım, C. Özdoğan, and D. Watson, "Parallel data reduction techniques for big datasets," in *Big Data Management, Technologies, and Applications*, W.-C. Hu and N. Kaabouch, Eds., pp. 72–93, IGI Global, 2014.

[18] J. Derrac, N. Verbiest, S. García, C. Cornelis, and F. Herrera, "On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection," *Soft Computing*, vol. 17, no. 2, pp. 223–238, 2013.

[19] J. C. Bezdek and L. I. Kuncheva, "Nearest prototype classifier designs: an experimental study," *International Journal of Intelligence Systems*, vol. 16, no. 12, pp. 1445–1473, 2001.

[20] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithm," in *Machine Learning*, D. R. Wilson and T. R. Martinez, Eds., vol. 38, no. 3pp. 257–286, Kluwer Academic Publishers, 2000.

[21] Á. Arnaiz-González, J. F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Instance selection of linear complexity for big data," *Knowledge-Based Systems*, vol. 107, pp. 83–95, 2016.

[22] C. Liu, W. Wang, M. Wang, F. Lv, and M. Konan, "An efficient instance selection algorithm to reconstruct training set for support vector machine," *Knowledge-Based Systems*, vol. 116, pp. 58–73, 2017.

[23] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[24] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[25] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[26] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 271–289, 1999.

[27] D. B. Skalak, *Prototype Selection for Composite Neighbor Classifiers*, University Of Massachusetts Amherst, 1997, https://web.cs.umass.edu/publication/docs/1996/UM-CS-1996-089.pdf.

[28] J. A. Benediktsson, J. Chanussot, and M. Fauvel, "Multiple classifier systems in remote sensing: from basics to recent developments," *Lecture Notes in Computer Science*, vol. 4472, pp. 501–512, 2007.

[29] J. Xia, J. Chanussot, P. Du, and X. He, "Rotation-based ensemble classifiers for high-dimensional data," in *Fusion in Computer Vision*, B. Ionescu, J. Benois-Pineau, T. Piatrik, and G. Quénot, Eds., pp. 135–160, Springer, 2014.

[30] R. Blaser and P. Fryzlewicz, "Random rotation ensembles," *The Journal of Machine Learning Research*, vol. 2, pp. 1–15, 2015.

[31] J. Xia, *Multiple classifier systems for the classification of hyperspectral data, [Ph. D. Thesis]*, University de Grenoble, 2014.

[32] I. H. Witten and D. J. Merz, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2nd edition, 2005.

[33] I. Czarnowski and P. Jędrzejowicz, "Cluster-dependent rotation-based feature selection for the RBF networks initialization," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, Gdynia, Poland, June 2015.

[34] H. L. Zhang and H. C. Lau, "Agent-based problem-solving methods in big data environment," *Web Intelligence and Agent Systems: An International Journal*, vol. 12, pp. 343–345, 2014.

[35] I. Czarnowski, "Distributed learning with data reduction," in *Transactions on Computational Collective Intelligence IV*, pp. 3–121, Springer, 2011.

[36] F. D. Ahmed, A. N. Jaber, and M. B. A. Majid, "Agent-based big data analytics in retailing: a case study," in *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pp. 67–72, Kuantan, Malaysia, August 2015.

[37] A. Amato, B. Di Martino, and S. Venticinque, "Agent-based decision support for smart market using big data," *Lecture Notes in Computer Science*, vol. 8286, pp. 251–258, 2013.

[38] S.-H. Chen and R. Venkatachalam, "Agent-based modelling as a foundation for big data," *Journal of Economic Methodology*, vol. 24, no. 4, pp. 362–383, 2017.

[39] D. Barbucha, I. Czarnowski, P. Jędrzejowicz, E. Ratajczak-Ropel, and I. Wierzbowska, "e-JABAT—an implementation of the web-based A-Team," in *Intelligence Agents in the Evolution of Web and Applications. Studies in Computational Intelligence 167*, N. T. Nguyen and L. C. Jain, Eds., pp. 57–86, Springer, Berlin, Heidelberg, 2009.

[40] I. Czarnowski, "Cluster-based instance selection for machine classification," *Knowledge and Information Systems*, vol. 30, no. 1, pp. 113–133, 2012.

[41] J. R. Cano, F. Herrera, and M. Lozano, "On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining," *Applied Soft Computing*, vol. 6, no. 3, pp. 323–332, 2006.

[42] R. Sikora and O. H. Al-Laymoun, "A modified stacking ensemble machine learning algorithm using genetic algorithms," *Journal of International Technology and Information Management*, vol. 23, no. 1, pp. 1–11, 2014.

[43] S. H. Lee and J. S. Lim, "Evolutionary instance selection algorithm based on Takagi-Sugeno fuzzy model," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, pp. 1307–1312, 2014.

[44] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[45] S. Talukdar, L. Baerentzen, A. Gove, and P. de Souza, "Asynchronous teams: cooperation schemes for autonomous," *Computer-Based Agents, Technical Report EDRC 18–59-96*, Carnegie Mellon University, Pittsburgh, 1996.

[46] S. Zhou and J. Q. Gan, "Mercer kernel fuzzy c-means algorithm and prototypes of clusters," *Proceedings of the International Conference on Data Engineering and Automated Learning*, , pp. 613–618, Lecture Notes in Computer Science, 2004.

Hindawi
Complexity
Volume 2018, Article ID 7408431, 9 pages
https://doi.org/10.1155/2018/7408431

Research Article

Using Deep Learning to Predict Sentiments: Case Study in Tourism

C. A. Martín, J. M. Torres, R. M. Aguilar, and S. Diaz

Department of Computer and Systems Engineering, Universidad de La Laguna, 38200 La Laguna, Tenerife, Spain

Correspondence should be addressed to R. M. Aguilar; raguilar@ull.edu.es

Received 19 April 2018; Accepted 23 September 2018; Published 23 October 2018

Guest Editor: Ireneusz Czarnowski

Technology and the Internet have changed how travel is booked, the relationship between travelers and the tourism industry, and how tourists share their travel experiences. As a result of this multiplicity of options, mass tourism markets have been dispersing. But the global demand has not fallen; quite the contrary, it has increased. Another important factor, the digital transformation, is taking hold to reach new client profiles, especially the so-called third generation of tourism consumers, digital natives who only understand the world through their online presence and who make the most of every one of its advantages. In this context, the digital platforms where users publish their impressions of tourism experiences are starting to carry more weight than the corporate content created by companies and brands. In this paper, we propose using different deep-learning techniques and architectures to solve the problem of classifying the comments that tourists publish online and that new tourists use to decide how best to plan their trip. Specifically, in this paper, we propose a classifier to determine the sentiments reflected on the http://booking.com and http://tripadvisor.com platforms for the service received in hotels. We develop and compare various classifiers based on convolutional neural networks (CNN) and long short-term memory networks (LSTM). These classifiers were trained and validated with data from hotels located on the island of Tenerife. An analysis of our findings shows that the most accurate and robust estimators are those based on LSTM recurrent neural networks.

1. Introduction

That the world of tourism is changing is not news. There are more and more data, both structured and nonstructured, being generated at ever higher rates, which once transformed into information, which provide a tangible value to businesses. Opinion mining and sentiment analysis is a very active field of study in recent years [1, 2]. The problem is determining how to benefit from these data and how to use them to generate value. Although future data is out of reach, it is possible to predict what will happen based on past data through a process known as predictive analytics.

The use of automatic tools in social networks for the tourism sector has generated ample literature due to the importance of influencing the consumer's participation and affecting the way in which tourists perceive their experience [3]. Thus, for a tourism company to grow, it is essential that it crafts forecasts to optimize its marketing campaigns and the performance of its corporate website in order to improve the feedback from its users. The predictive scores obtained from the models for each client indicate the steps that should be taken to achieve the goals of retaining the client, upselling a product, or offering him a new service. Reliable and consistent information is needed in order to forecast client behavior. The work presented in this paper builds on this idea, testing methods to analyze the reviews clients provide on digital platforms concerning the service they received.

In contrast to the traditional conversations that take place in specific physical locations, the digital conversation is shaped using new methods and tools for engaging the public, whose social interaction characteristic is the focus of its dynamic [4, 5]. Basically, it is the so-called electronic word of mouth (eWOM), the features of which lead to completely different consequences [6] from traditional conversations. These features focus on the ease with which the message is distributed via social media, which are online communications platforms where users create their own content using Web 2.0 technologies that enable the writing, publication,

and exchange of information (Kaplan and Haenlein, 2010). The incorporation of eWOM into social media or the Web 2.0 has the following properties:

(i) Great ability to disseminate, where users can access opinions from strangers

(ii) Massive engagement by users of different ages and groups, all sharing different points of view

(iii) The message spreading quickly in several ways: blogs, websites, social media, messages posted in online groups, etc.

(iv) Multidirectional discussion among users, who play an active role by answering questions on the information presented

(v) Persistence over time, since the discussions are uploaded for the current and future reference

(vi) Credibility, since the information is offered by users spontaneously and, in theory, with no profit motive

These features make monitoring of eWOM by tourism companies particularly relevant [7].

According to [7], an eWOM is "any positive or negative statement made by potential, current, or former customers about a product or company that will be made available to a large number of people and institutions through the Internet." The main goal of this paper is to classify the positive and negative statements made by tourists by using various deep-learning techniques.

The tourism industry is of great importance in the Canary Islands, as it is the real engine of growth and development in the archipelago, accounting for a high percentage of its GDP. This has a knock-on effect on the remaining industries and services in the islands, especially in the development of trade, transportation, food production, and industry. Tourism also comprises a very important component in creating jobs in the service sector of the archipelago, which encompasses direct employment in the sun and beach sector, as well as workers in activities that support tourism, such as restaurants, hotels, travel agencies, passenger transport, car rental, and recreational, cultural, and sports activities.

In 2016, Tenerife, one of the Canary Islands, received over seven million tourists, most of them from the United Kingdom, which accounted for thirty-one percent of the passengers arriving at Tenerife's airports in 2016. These figures indicate that the opinion that English tourists have of Tenerife is particularly relevant. As a result, we will conduct our experiments based on the comments in English made about the hotels in Tenerife.

2. Materials and Methods

Below, we describe the estimators implemented, the data used, and the procedures employed to train, evaluate, and compare the estimators.

2.1. Data Sets. The techniques used, which will be detailed in the sections that follow, fall under the category of supervised learning methods. This requires having a set of previously classified data before the prediction system can be trained and a test sample in order to validate how accurately the technique behaves. To satisfy this requirement and in order to use real data in the methods, we extracted information from reviews in English on the http://booking.com and http://tripadvisor.com websites for hotels on the island of Tenerife. The structure of the information varies depending on the portal that is used as the source of data:

(i) Booking

  (a) The comment will have a general score between 0 and 10

  (b) The comment's author is able to separate positive and negative aspects

(ii) TripAdvisor

  (a) The comment features a rating system using bubbles or stars to assign a score between 1 and 5

  (b) There is a single field for expressing an opinion

In order to extract the information, we developed Python scripts based on the Scrapy framework and adapted them to the domain using the scripts offered by the MonkeyLearn project [8].

As a result of this process, we obtained more than 40,000 records with different fields, depending on the source portal for the data: title, comment, score, date, and location of the visitor.

The preparation of the data set so that it can be used in the deep-learning techniques studied in this paper is an important part of the work. In the initial phase and in an effort to standardize the information taken from the two sources used, we developed programs in Python to build a CSV file with two columns:

(i) Comment (free text)

(ii) Label ("Bad" or "Good")

Depending on the portal from which the data were sourced, the scripts had different functionalities.

In the case of the samples taken from TripAdvisor, the original title and comment were used to comprise a single text containing the visitor's full opinion. Those reviews scoring three or higher were labeled "Good," and those scoring two or below were labeled "Bad." Any samples with an intermediate rating were discarded so as not to hamper the training.

In the case of the samples taken from Booking, we evaluated the number score awarded by the visitor. For scores higher than six, which were labeled "Good," the comment was generated by concatenating the title and comment fields. For scores of four and below, the title and negative comment were concatenated and labeled "Bad."

Once the structure of the samples was standardized, the data set was divided into three parts. For the first set, a random balanced selection (half of the samples labeled "Good"

Figure 1: Negative review words.

Figure 2: Positive review words.

and the other half labeled "Bad") of 9640 samples was carried out. The second set was used to evaluate the models after each training epoch, and finally, the test sample was created using 2785 samples in which the scores assigned by the tourists were known and which we used to compare the accuracy of the various models.

The second phase to prepare the data sets for the deep-learning models involved adapting them to the data domain that can be input into the models. Each comment used from the training set was subject to preprocessing before it could be used. Fortunately, Python offers an ecosystem of libraries that can be used in several machine learning applications [9] like the Python NLTK library [10] that allow us to perform this task. The comments were processed as follows: separation into words, deletion of punctuation and any nonalphabetic terms, and lastly deletion of words identified as stop words, which provide no information when determining if a comment is positive or negative.

The figures below offer a visual representation of those words that appeared most frequently in the positive and negative comments:

As we can see by analyzing Figures 1 and 2, there are certain words that clearly identify a polarized sentiment like "Good" or "Bad." However, this does not apply to many other words, which are sometimes even included in comments with the exact opposite meaning (note that the group of words used in negative comments contains words like "better," "nice," or "clean"). As a result, a simple classification of the comment based on the appearance or absence of certain words is not sufficient; rather, machine-learning techniques, such as the ones used in this paper, are needed to analyze the relationships between the words.

The algorithms used require as an input a fixed-length vector in which each component is a number. In the technique for coding text into number vectors known as bag of words (BoW), a dictionary is created with the words found most frequently in all of the training comments. Each comment is then coded into a fixed-length vector corresponding to the number of words in the dictionary created. In BoW, a comment is coded into a vector in which each component counts how many times each word in the dictionary appears in the comment. We ruled out this coding method because even though it represents the frequency of words in the comment, it discards information involving the order in which those words appear in the comment.

The word embedding technique is currently one of the best for representing texts as number vectors. It is a learned representation in which words with similar meanings are given a similar representation. Each vocabulary word is represented with a vector, and its representation is learned based on the use it is given in the training comments. As a result, words that are used in a similar way will have a similar representation. The learning process for embedding is carried out in this paper by adding a layer at the front of the neural network in each of the models. In order to be able to generate our models in Python, we resorted to the Keras library [11].

In order to use this library and add the embedding layer to our models, we have to first transform the tourists' comments into integer vectors in which each word is represented by an index in a list of words. We do so by using the Tokenizer class in Keras, creating an instance based on the training data set and limiting the size of the vocabulary, in our case, to the 5000 most common words. By using the Tokenizer instance, we transform each comment into a vector of variable length in which each word is an integer where the value i corresponds to the i-th most used word in the samples. Since the tourist comments do not all have the same length, the comment vectors are filled in with zeros until the specified fixed length is reached that matches the number of words in the longest comment (582 in this work). Figure 3 shows this process.

The embedding layer is initialized with random weights and trained at the same time as the rest of the models, with the training supplied by the training data set. In every model implemented for this work, the embedding layer was used as the first layer in the model, with the following characteristics:

(i) Input dim (the maximum integer value of the vector component input): its value, based on how we coded the comments, is 5000

(ii) Output dim (the length of the vectors that will represent the words after embedding): in most of the experiments, this length is set to 300

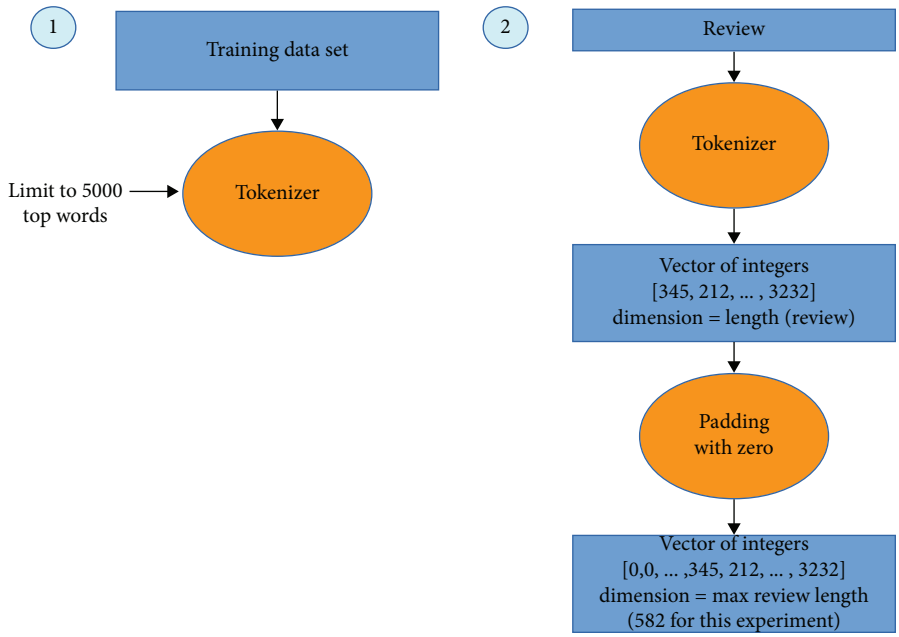


Figure 3: Converting review to fixed-length vectors.

(iii) Input length (the length of the vectors in the layer): As defined earlier, it is the maximum length of the comments in words. In the experiments conducted for this work, the maximum comment length was 582 words

As shown in Figure 4, the output of the embedding layer is a 582×300 matrix in which each comment is transformed into a vector with 582 components, in which each component (representing one word) is coded into a vector of length 300.

2.2. Recurrent Neural Networks. To predict the sentiment of the comments, we use models based on neural networks. Each comment is a sequence of encoded words that can be processed as a time series. However, the most common neural networks (e.g., feed-forward neural networks) lack the memory to store information over time. Recurrent neural networks [12] (RNN) solve this problem by making the network output y_j at step j depend on previous computations through a hidden state s_j that acts as a memory for the network.

Figure 5 shows the RNN we used, unfolded into a full network. By unfolded, we simply mean that we write out the network for a complete sequence of N steps, where x_j is the j-th encoded word in the comment, which we used as the input to the network in the j-th step.

In a RNN, the relationship between output y_j, input x_j, and state s_j in step j is determined by the type of RNN cell. As Figure 5 shows, we used a kind of cell called long short-term memory [13] (LSTM). LSTM recurrent neural networks are capable of learning and remembering over long input sequences and tend to work very well for labeling sequences of word problems [14].

As (1) shows, output y_j depends on the state s_j of the LSTM cell through the activation function $\sigma_y(x)$ (which is generally $\tanh(x)$). The state s_j depends on the state of the previous step s_{j-1} and on the candidate for the new value of the state $\tilde{s}_j$. The output gate o_j controls the extent to which the state s_j is used to compute the output y_j by means of the Hadamard product $(\circ)$. The input gate i_j controls the extent to which $\tilde{s}_j$ flows into the memory, and the forget gate f_j controls the extent to which s_{j-1} remains in memory.

The o_j, f_j, and i_j gates and the candidate for the new value of the state of the cell $\tilde{s}_j$ can be interpreted as the outputs of conventional artificial neurons whose inputs are the input to cell x_j at step j and the output of cell y_{j-1} at $j-1$. The activation function for the gates σ_g is usually the sigmoid function, while for σ_s it is usually $\tanh(x)$.

$$\begin{aligned} f_j &= \sigma_g\left(W_f x_j + U_f y_{j-1} + b_f\right), \\ i_j &= \sigma_g\left(W_i x_j + U_i y_{j-1} + b_i\right), \\ o_j &= \sigma_g\left(W_o x_j + U_o y_{j-1} + b_o\right), \\ \tilde{s}_j &= \sigma_s\left(W_s x_j + U_s y_{j-1} + b_s\right), \\ s_j &= f_j \circ s_{j-1} + i_j \circ \tilde{s}_j, \\ y_j &= o_j \circ \sigma_y\left(s_j\right). \end{aligned} \tag{1}$$

As Figure 5 shows, each input x_j in (1) is the coded value of the successively coded sequence of words $(x_j)_{j=0}^{N-1}$ in the comment. The RNN cell provides an output at each step j,

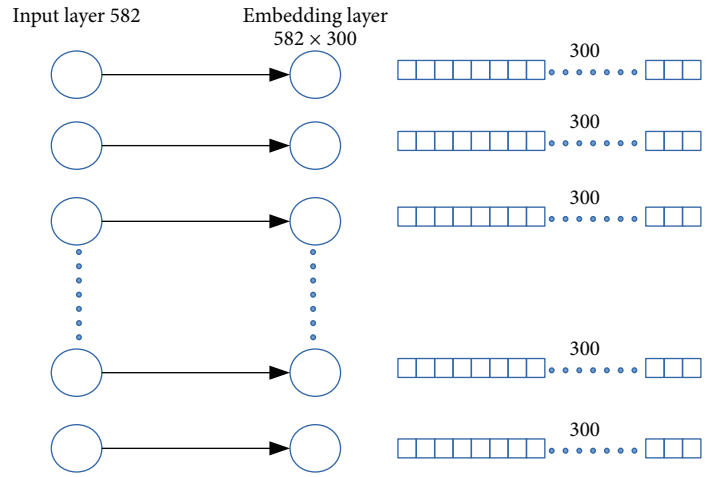


Figure 4: Embedding layer.

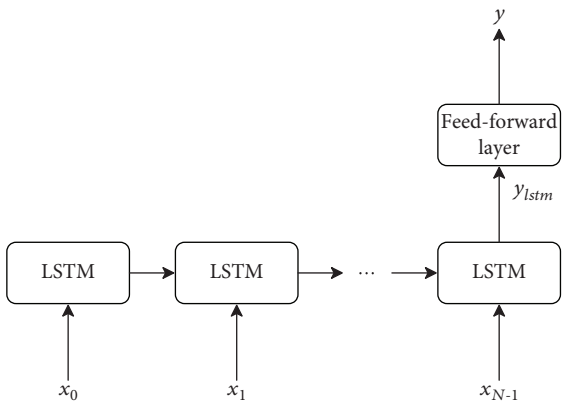


Figure 5: LSTM unfolded into a full network.

but for the prediction, only output y_{N-1} of the network is considered when the last word x_{N-1} is input into the network. This output is what we refer to as y_{lstm} in Figure 5.

Output y_{lstm} is used as an input to a one-neuron feed-forward layer with a sigmoid activation function, the output of which, between 0 and 1, is the network's prediction for whether the sentiment is positive or negative. The output y of that single neuron can be expressed as indicated in

$$y = \sigma_{\text{sigmoid}}\left(\sum_k w_k y_{\text{lstm}} + b\right)\sigma_{\text{sigmoid}}(x) = \frac{e^x}{e^x + 1}, \quad (2)$$

where w_k is used to denote the weight of the k-th input, b is the bias, and $\sigma_{\text{sigmoid}}(x)$ is the sigmoid activation function of the output neuron in the layer.

2.3. Convolutional Neural Networks. Another type of neural network that can be used to predict time series is a convolutional neural network (CNN). These are biologically inspired variants of feed-forward neural networks used primarily in computer vision problems [15], although their ability to exploit spatially local correlation in images can also be used in time-series problems, like sentiment analysis.

In these models, the output of each neuron a_j^l is generated from the output of a subset of spatially adjacent neurons. Every neuron in the same layer shares the same weight and bias, meaning the layer can be expressed in terms of a filter that is convoluted with the output of the previous layer. Equation (3) shows the output a_j^{lk} of the j-th neuron of the l-th convolutional layer:

$$a_j^{lk} = \sigma^{lk}\left(\left(W^{lk*}a^{l-1}\right)_j + b^{lk}\right), \quad (3)$$

where $\left(W^{lk*}a^{l-1}\right)_j$ is the j-th element resulting from the convolution of the filter defined by W^{lk} with the output of the previous layer a^{l-1}, $\sigma^{lk}(x)$ is the activation function for the convolutional layer, and $k \in [0, \ldots, K]$ indicates that it is the output of the k-th channel of the layer. In each convolutional layer, different filters can be applied to the output of the previous layer to generate different representations or channels k, thus yielding a fuller representation of the data.

For the $\sigma^{lk}(x)$ activation function, we used the ReLU function because it does not suffer from the vanishing gradient problem [7] when training the neural network. Equation (4) shows the expression for the activation function.

$$\sigma^{lk}(x) = \sigma_{\text{relu}}(x) = \max(0, x). \quad (4)$$

Figure 6 shows a general diagram of the CNN developed in this paper to analyze sentiment. In order to apply a CNN to a time series, we arranged the encoded words in the same order as they appear in the comment, such that adjacent words in the comment are spatially adjacent at the input to the neural network. Moreover, each word embedding dimension is a different input channel to the network. In this way, the convolutional layers can exploit the local correlation between words in each comment.

After each convolutional layer with a ReLU, activation function is a max-pooling layer, which partitions the input into a set of nonoverlapping ranges and, for each range,

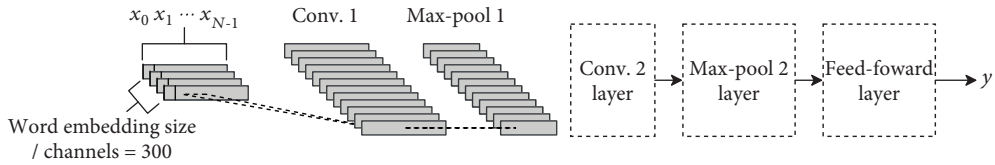


Figure 6: General diagram of the CNNs used.

Table 1: Model structure.

Model	Model description
1	(582) embedding→(582 × 300) LSTM→(30) dense [sigmoid]→(1)
2	(582) embedding→(582 × 300) LSTM→(50) dense [sigmoid]→(1)
3	(582) embedding→(582 × 300) LSTM→(70) dense [sigmoid]→(1)
4	(582) embedding→(582 × 300) LSTM→(100) dense [sigmoid]→(1)
5	(582) embedding→(582 × 300) LSTM→(200) dense [sigmoid]→(1)
6	(582) embedding→(582 × 300) LSTM→(300) dense [sigmoid]→(1)
7	(582) embedding→(582 × 300) LSTM→(500) dense [sigmoid]→(1)
8	(582) embedding→(582 × 300) Conv1D→(575 × 64) MaxPooling1D→(287 × 64) flatten→(18,368) dense [relu]→(10) dense [sigmoid]→(1)
9	(582) embedding→(582 × 300) Conv1D→(575 × 128) MaxPooling1D→(287 × 128) flatten→(36,736) dense [relu]→(10) dense [sigmoid]→(1)
10	(582) embedding→(582 × 300) Conv1D→(575 × 32) MaxPooling1D→(287 × 32) Conv1D→(280 × 64) flatten→(17,920) dense [relu]→(10) dense [sigmoid]→(1)
11	(582) embedding→(582 × 300) Conv1D→(582 × 32) MaxPooling1D→(291 × 32) LSTM→(100) dense [sigmoid]→(1)

Table 2: Training data set.

Training data set	
Positive reviews	4820
Negative reviews	4820
Total reviews	9640
Mean review length (chars)	53
Max review length (chars)	582

outputs the maximum value. Following the convolutional and max-pooling layers is a feedforward layer (as described in (2)) to yield the output of the entire network.

3. Results and Discussion

In order to compare some of the deep-learning techniques mentioned in this paper, we conducted a series of experiments on different models based on LSTM neural networks and CNN. Table 1 shows the structure of each of the models used with the lengths of the inputs at each layer. The first column will be used to refer to the models in subsequent references.

Table 3: Test data set.

Test data set	
Positive reviews	1408
Negative reviews	1377
Total reviews	2785

Table 4: Test results.

Model N	Training time	Good hits	Bad hits	False good	False bad	Accuracy end
1	2208	1227	1228	149	181	88.15
2	2734	1211	1239	138	197	87.97
3	4658	1215	1237	140	193	88.04
4	4406	1198	1261	116	210	88.29
5	4388	1213	1256	121	195	88.65
6	10,630	1221	1263	114	187	89.19
7	13,574	1190	1261	218	116	88.01
8	6994	1210	1247	130	198	88.22
9	9139	1206	1235	142	202	87.65
10	1765	1166	1268	109	242	87.40
11	1602	1187	1272	105	221	88.29

First, we prepared the data as explained in Section 2.1 + 0.1667 emA. The same comment coding technique was used for each test, which included training an embedding layer.

To make the models comparable, we used the same training data set (Table 2) with a total of 9640 reviews. The classification was conducted independently using another data set containing 2785 reviews that were not used during the training. As Table 3 shows, a test data set was employed that was fully balanced between positive and negative reviews.

So as not to use a different number of training epochs based on the model, we decided to use a fixed number, 10, since, for every model, the loss value did not improve significantly with longer training.

Table 4, containing the general results, shows the number of correctly predicted positive and negative reviews, as well as the false positives and negatives.

Models 1 to 6 have the same structure, the number of memory units varying as shown in Table 1. As we can see in Figure 7, the results improve as the number of memory cells is increased, reaching a maximum accuracy of 89.19% in model 6.

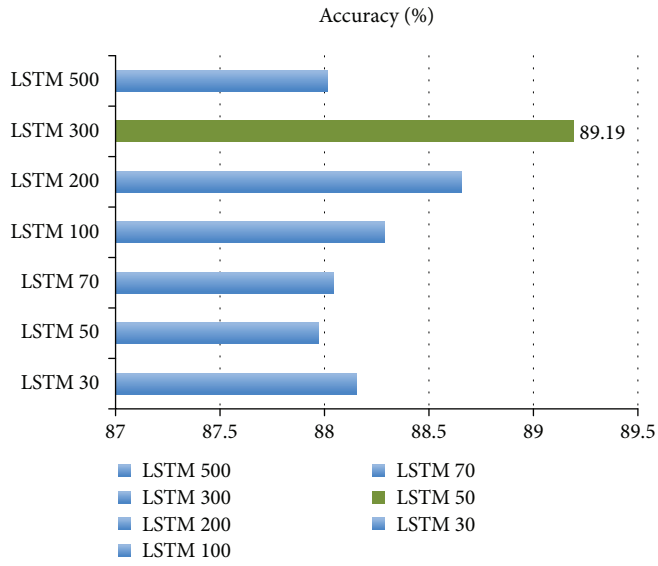


Figure 7: LSTM comparison.

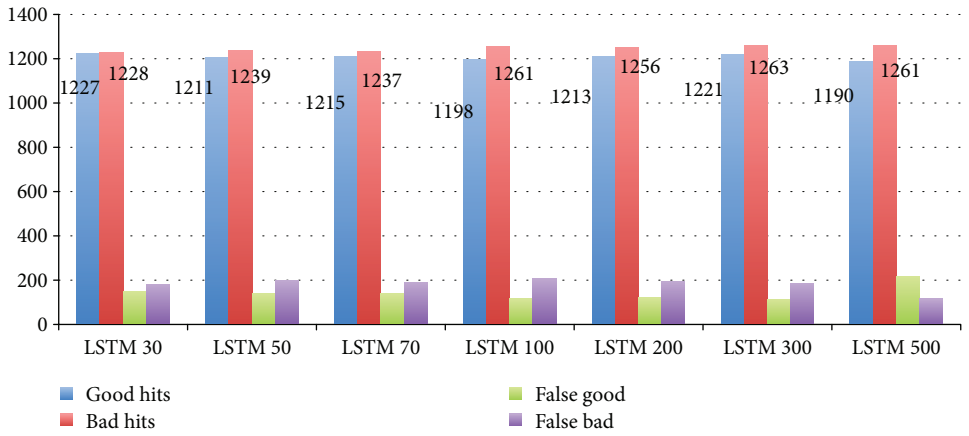


Figure 8: LSTM comparison.

Figure 8 shows the results of the test separated by hits and false positives. The model with the best result improved on the predictions from previous models for negative reviews.

The second part of the experiment consisted of checking for a significant variation when the number of filters was changed in the CNN models. Specifically, we compared models 8 and 9 with one another, yielding the results shown in Figure 9. As we can see, the difference is not significant, and increasing the number of filters does not provide any improvement.

To complete the study, we compared the results for the previous models that yielded the best outcome (model 6 LSTM and model 8 CNN) with models 10 (two-layer CNN) and 11 (CNN and LSTM). As Figure 10 shows, the LSTM models exhibit the highest accuracy, a result that is not improved by adding a convolutional layer.

Figure 11 shows that the best overall result for the LSTM neural networks is based primarily on their better prediction of positive results, in comparison to the other networks trained.

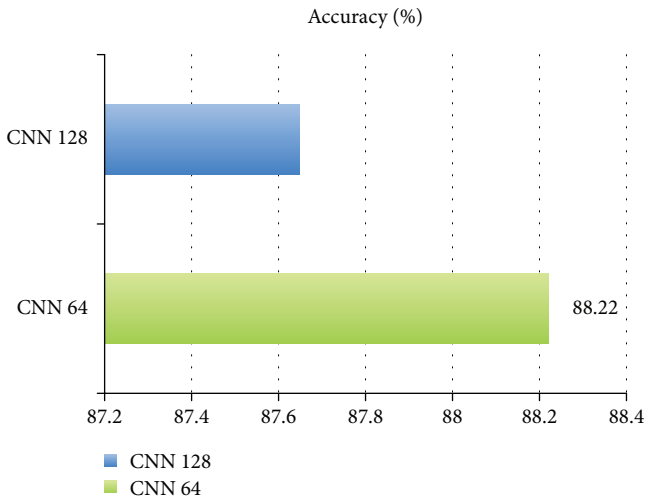


Figure 9: CNN comparison.

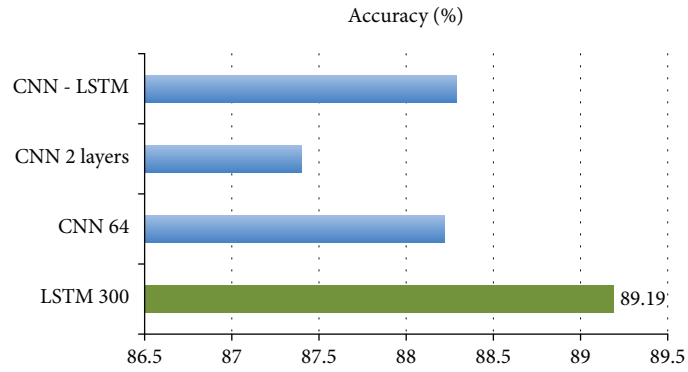


Figure 10: Comparison of LSTM and CNN models.

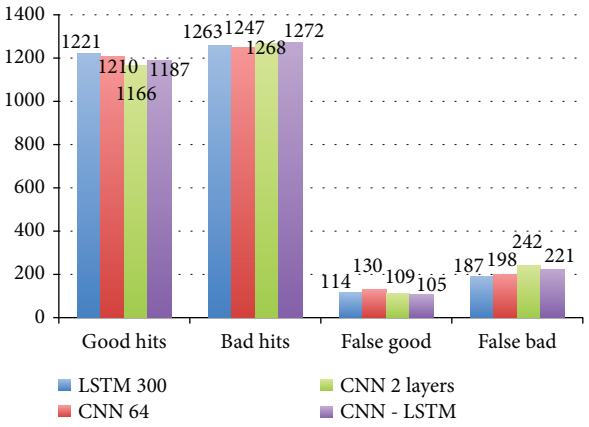


Figure 11: LSTM comparison.

4. Conclusions

In this paper, we considered the problem of predicting sentiment in tourist reviews taken from eWOM platforms for hotels at an important international tourist destination. The use of techniques and automatic tools such as those considered in this paper are very useful for tourism industry practitioners [3]. Specifically, the prediction of positive or negative sentiment expressed by a tourist can be used in different marketing and service management applications:

(i) Comparison of feeling about another local competitor or tourist destination (market positioning)

(ii) Performing a proactive customer service management, generating a job ticket when a negative review is detected (customer management)

(iii) As a measure of indicators to start a campaign to improve the reputation (marketing management)

(iv) As a measure of risk indicators that affect the hotel or destination image (risk management)

Once the models studied in this article have been trained, they can be used in combination with other tools (review extraction or dashboards).

We used deep-learning techniques to devise different predictors based on neural networks, which were trained with the extracted data to compare the accuracies of each.

The predictors evaluated were based on recursive neural networks with cell LSTM and convolutional neural networks. Different designs were considered for each. The methodology was checked by training and validating the model with samples taken from Booking and TripAdvisor.

The results show that LSTM neural networks outperform CNN. The optimum result for CNN is attained with a single convolutional layer and 64 channels. More layers or more channels result in symptoms of overfitting. The LSTM neural networks yield higher accuracies, with one LSTM with a vector length of 300 for the internal state yielding an accuracy just over 89%, the highest for any model.

Finally, the results show that the better results of the neural networks are due primarily to their advantage when classifying the positive comments. They also show that combining convolutional layers with recurrent LSTM layers does not yield any advantages.

Data Availability

The comment tourist review data used to support the findings of this study have been deposited in the GitHub repository https://github.com/camartinULL/Deep_Learning_Predict_Sentiments.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is sponsored by the "VITUIN: Vigilancia Turística Inteligente de Tenerife en Redes Sociales" project, through research funds of the Fundación CajaCanarias.

References

[1] R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Information Processing and Management*, vol. 53, no. 1, pp. 122–150, 2017.

[2] J. A. Balazs and J. D. Velásquez, "Opinion mining and information fusion: a survey," *Information Fusion*, vol. 27, pp. 95–110, 2016.

[3] M. D. Sotiriadis, "Sharing tourism experiences in social media: a literature review and a set of suggested business strategies," *International Journal of Contemporary Hospitality Management*, vol. 29, no. 1, pp. 179–225, 2017.

[4] M. D. Sotiriadis and C. van Zyl, "Electronic word-of-mouth and online reviews in tourism services: the use of twitter by tourists," *Electronic Commerce Research*, vol. 13, no. 1, pp. 103–124, 2013.

[5] R. A. King, P. Racherla, and V. D. Bush, "What we know and don't know about online word-of-mouth: a review and synthesis of the literature," *Journal of Interactive Marketing*, vol. 28, no. 3, pp. 167–183, 2014.

[6] L. de Vries, S. Gensler, and P. S. H. Leeflang, "Popularity of brand posts on brand fan pages: an investigation of the effects of social media marketing," *Journal of Interactive Marketing*, vol. 26, no. 2, pp. 83–91, 2012.

[7] T. Hennig-Thurau, K. P. Gwinner, G. Walsh, and D. D. Gremler, "Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the Internet?," *Journal of Interactive Marketing*, vol. 18, no. 1, pp. 38–52, 2004.

[8] "Hotel review analysis. monkey learn," 2018, https://github.com/monkeylearn/hotel-review-analysis.

[9] R. M. Aguilar Chinea, J. M. Torres Jorge, and C. A. M. Galán, "Aprendizaje automático en la identificación de sistemas. un caso de estudio en la generación de un parque eólico," *Revista Iberoamericana de Automática e Informática industrial*, vol. 16, 2018.

[10] "NLTK python library," 2018, https://www.nltk.org.

[11] "Keras python library," 2018, https://keras.io.

[12] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Computation*, vol. 4, no. 2, pp. 234–242, 1992.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] A. Graves, *Supervised sequence labelling with recurrent neural networks*, Springer, 2012.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010*, Chia Laguna Resort, Sardinia, Italy, 2010.

Hindawi
Complexity
Volume 2018, Article ID 8487641, 9 pages
https://doi.org/10.1155/2018/8487641

Research Article

An Efficient Method for Mining Erasable Itemsets Using Multicore Processor Platform

Bao Huynh [1,2] **and Bay Vo** [3]

[1]*Division of Data Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[2]*Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[3]*Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City, Vietnam*

Correspondence should be addressed to Bay Vo; vd.bay@hutech.edu.vn

Received 19 April 2018; Revised 10 July 2018; Accepted 30 July 2018; Published 22 October 2018

Academic Editor: Ireneusz Czarnowski

Mining erasable itemset (EI) is an attracting field in frequent pattern mining, a wide tool used in decision support systems, which was proposed to analyze and resolve economic problem. Many approaches have been proposed recently, but the complexity of the problem is high which leads to time-consuming and requires large system resources. Therefore, this study proposes an effective method for mining EIs based on multicore processors (pMEI) to improve the performance of system in aspect of execution time to achieve the better user experiences. This method also solves some limitations of parallel computing approaches in communication, data transfers, and synchronization. A dynamic mechanism is also used to resolve the load balancing issue among processors. We compared the execution time and memory usage of pMEI to other methods for mining EIs to prove the effectiveness of the proposed algorithm. The experiments show that pMEI is better than MEI in the execution time while the memory usage of both methods is the same.

1. Introduction

Data mining is an interesting field that has attracted many experts because of the huge amounts of data that were collected every day and the need to transfer such data into useful information to use in intelligence systems such as recommendation systems, decision making, and expert systems. Data mining has been widely used in market basket analysis, manufacturing engineering, financial banking, bioinformatics and future healthcare, and so on. The mining frequent pattern (FP) has a vital position in many data mining fields including association rule mining [1], clustering [2], and text mining [3]. Mining FP is to find all patterns that have the frequency satisfying the user-given threshold. There are many methods [4, 5] for mining FPs in recent years. In addition, some issues related to FP mining has been proposed such as maximal frequent patterns [6], top-k cooccurrence items with sequential pattern [7], weighted-based patterns [8], periodic-frequent patterns [9], and their applications [10, 11].

In 2009, the erasable itemset mining was first introduced [12], which comes from predicting merchandises of the production scheming as an exciting alteration of pattern mining. For an example, a factory needs to produce several new products, each of which requires some amount of raw materials to produce. However, the factory does not have enough budget to purchase all materials. Therefore, the factory managers need to determine what essential materials are needed to produce while the profit is not affected. The main problem is how to efficiently find these materials, without which the loss of the profit is less than the given threshold. These elements are also called as erasable itemsets. Based on these erasable itemsets, the consulting team can give the managers several suggestions about production plans in the near future. It has attracted a lot of research and become an ideal topic in recent years. There are many approaches, which were summarized in [13], to mine such patterns including META [12], MERIT [14], MEI [15], and EIFDD [16]. Several related problems of mining erasable closed itemsets [17], top-rank-k

erasable itemsets [18], erasable itemsets with constraints [19], and weighted erasable itemsets [20, 21] have also been developed. Erasable closed itemset [17], a condensed representation of EIs without information loss, was proposed to reduce the computational cost. Top-rank-k erasable itemset [18] is to merge the mining and ranking phases into one phase that only returns a small number of EIs to use in intelligent systems. Erasable itemset with constraints [19] is another approach only producing a small number of EIs, which delight a special requirement. In addition, mining weighted erasable itemset [20, 21] is a framework for mining erasable itemsets with the weight conditions for each item.

The existing algorithms for mining EIs have high computational complexity. It leads to very long execution time, especially on huge datasets and inefficiently intelligent systems. Therefore, this study proposes a parallel approach named parallel mining erasable itemsets (pMEI) using a multicore processor platform to improve the execution time for mining EIs. The major contributions of this paper are as follows:

(i) A parallel method, namely, pMEI for mining EIs, splits the jobs into several duties to lessen the operating cost.

(ii) Applying the difference pidset (dPidset) structure for quickly determining EIs information.

(iii) A dynamic mechanism is used for load balancing the workload among cores when some processes are free.

The experiment results show that pMEI algorithm is better than MEI in execution time for the most of experimental datasets.

The remaining parts of the paper are arranged as follows: the preliminaries and related works, comprising the erasable itemset mining problem, several methods for mining EIs, as well as multicore processor architecture are presented in Section 2. Section 3 presents the pMEI algorithm proposed. The experiments on runtime and memory usage of pMEI and MEI methods for mining EIs are shown in Section 4. Finally, Section 5 reviews the outcomes and proposes some potential study topics.

2. Related Works

2.1. Preliminaries. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n distinct items, which are the conceptual descriptions of elements of products created in a manufactory. For example, assuming that $I = \{\text{sugar, milk, cheese, cafe, snack, and wine}\}$, then the included items are sugar, milk, cheese, cafe, snack, and wine. A product dataset D consists of m products, $\{P_1, P_2, \ldots, P_m\}$, where $P_k (1 \le k \le m)$ is a product shown in the pair of (items, value). In this pair, items are the items that compose product P_k, and value is the profit value of the product P_k. Table 1 shows the example datasets that have five items including $\{a, b, c, d, e, f\}$ and the set of products $\{P_1, P_2, \ldots, P_{10}\}$.

Table 1: An example of product dataset D.

Product	Items	Value (USD)
P_1	a, b, c	3100
P_2	a, b	1500
P_3	a, c	1200
P_4	b, c, e	250
P_5	b, e	150
P_6	c, e	110
P_7	c, d, e, f	250
P_8	d, e, f	100
P_9	d, f	350
P_{10}	b, f	50

Definition 1 (itemset). An itemset $Y = (i_1, i_2, \ldots, i_u)$, such that $i_k \in I (1 \le k \le u)$, is called a k-itemset.

Definition 2 (profit of itemset). Given an itemset $Y \subseteq I$, the profit of itemset Y (denoted $\text{pro}(Y)$) is calculated by

$$\text{pro}(Y) = \sum_{\{P_k \in D | Y \cap P_k \text{ items} \neq \emptyset\}} P_k \text{ value}. \quad (1)$$

For example, let $Y = \{ad\}$ the set of products that contains $\{a\}$, $\{d\}$, or $\{ad\}$ from Table 1 which are $\{P_1, P_2, P_3, P_7, P_8, P_9\}$. Therefore, $\text{pro}(Y) = P_1$ value + P_2 value + P_3 value + P_7 value + P_8 value + P_9 value = 6500 USD.

An itemset Y in product dataset D is said to be erasable iff $\text{pro}(Y) \le \theta \times \delta$, where δ is the minimum threshold defined by user; θ is the total value of D determined by this formula $\theta = \sum_{P_k \in D} P_k$ value.

Definition 3 (EI mining problem). The EI mining problem is to discover all itemsets (Y) that have $\text{pro}(Y)$ not greater than $\theta \times \delta$.

For the example dataset in Table 1 and $\delta = 20\%$, we have $S = 7060$ USD by summing the value of all products. The itemset d is an erasable itemset due to $\text{pro}(d) = 700 \le 7060 \times 20\% = 1412$.

2.2. Erasable Itemset Mining. The erasable itemset (EI) mining problem was first introduced in 2009 [12], which comes from predicting merchandises of the production industry. It supports managers to decide their manufacturing strategy to guarantee the development of the company. The managers can decide which new merchandises are suitable for the factory without affecting the company's profit. For example, a company that makes many types of goods, each good that is created will have a profit value. To create all products, the factory has to buy all essential materials. Currently, the company has no enough budget to purchase all materials. Hence, the managers of this company should deliberate their manufacturing strategies to make sure the

steadiness of the company. The challenge is to obtain the itemsets that can be excluded but do not significantly change the company's profit.

2.3. Methods for Erasable Itemset Mining. Many methods have been suggested to mine EIs such as META [12], MERIT [14], MEI [15], and EIFDD [16]. Firstly, META, an Apriori-based algorithm, generates candidate itemsets using level-wise approach. Let S be the set of erasable $k-1$-itemsets. An itemset $Y \in S$ is verified with $Z \in S \wedge Y \neq Z$ for coherency to produce applicant erasable k-itemsets. However, only a small number of $Z \in S \wedge Z \neq Y$ that have the same prefix as Y are joined. Later, MERIT based on the NC_sets is structured to decrease memory manipulation, which is its main improvement. The performance of MERIT is better than that of META significantly. However, storing NC_sets structure leads to high computational cost including memory usage and runtime. Therefore, MEI uses a divide-and-conquer approach associated with the difference of pidsets (dPidset) for mining EIs to improve the memory usage and runtime. It only scans the dataset one time to determine the total profit (S), the index of gain (pro), and the erasable 1-itemsets with their pidsets. Although the runtime and memory consumption are enhancing than those of META and MERIT, however, the MEI's performance from several dense datasets is quite weak. To resolve this drawback of MEI for dense datasets, EIFDD is proposed by using the subsume theory. This concept helps to quickly determine the information of EIs, without the generation cost. In brief, EIFDD is regularly applied to mine EIs for dense datasets, while MEI is applied to mine EIs for the other kinds of datasets.

Although existing methods improved the computation of mining EIs, however, these methods still consume more time in large datasets or large thresholds. Hence, in this paper, we develop a new technique to improve the computational cost for mining EIs.

2.4. dPidset Structure. The dPidset structure was suggested by Le and Vo [15] to lessen memory consumption by using an index of profit for effectively mining EIs. This structure is summarized as follows.

Definition 4. Given an itemset Y and an item $X \subseteq Y$, the pidset of itemset Y is denoted as

$$p(Y) = \bigcup_{X \in Y} p(X), \tag{2}$$

where $p(X)$ is the pidset of item X, that is, the set of product identifiers (IDs) which contain item X.

Definition 5. Let $\text{pro}(Y)$ be the profit of itemset Y that is computed as follows:

$$\text{pro}(Y) = \sum_{P_k \in p(Y)} \text{value}(P_k). \tag{3}$$

Theorem 1 [15]. *Two given itemsets YA and YB have the same prefix which is Y. $p(YA)$ and $p(YB)$ are pidsets of YA and YB, correspondingly. $p(YAB)$ is determined as follows:*

$$p(YAB) = p(YB) \cup p(YA). \tag{4}$$

Example 1. Consider the illustration datasets in Table 1, $p(de) = \{4, 5, 6, 7, 8\}$ and $p(df) = \{7, 8, 9, 10\}$. We have $p(def) = p(dfe) = p(de) \cup p(df) = \{4, 5, 6, 7, 8, 9, 10\}$.

Definition 6. The dPidset of itemset YAB signified by $dP(YAB)$ is defined as follows:

$$dP(YAB) = p(YB) - p(YA), \tag{5}$$

where $p(YB) - p(YA)$ is the list of itemset IDs that only exist in $p(YB)$.

Example 2. We have $p(de) = \{4, 5, 6, 7, 8\}$ and $p(df) = \{7, 8, 9, 10\}$, so $dP(def) = p(df) - p(de) = \{9, 10\}$.

Theorem 2 [15]. *Two given itemsets YA and YB have the dPidsets which are $dP(YA)$ and $dP(YB)$. The $dP(YAB)$ is determined as follows:*

$$dP(YAB) = dP(YB) - dP(YA). \tag{6}$$

Example 3. We have $p(d) = \{7, 8, 9\}$, $p(e) = \{4, 5, 6, 7\}$, and $p(f) = \{7, 8, 9, 10\}$. As in Definition 6, $dP(de) = p(e) - p(d) = \{4, 5, 6\}$ and $dP(df) = p(f) - p(d) = \{10\}$. As in Theorem 2, $dP(def) = dP(df) - dP(de) = \{10\}$.

Theorem 3 [15]. *The profit of YAB, denoted by* pro *(YAB), is calculated by YA as follows:*

$$\text{pro}(YAB) = \text{pro}(YA) + \sum_{P_k \in dP(YAB)} \text{value}(P_k), \tag{7}$$

where $\text{pro}(YA)$ *is the profit of Y and* $\text{value}(P_k)$ *is the profit of P_k.*

Example 4. We have $p(d) = \{7, 8, 9\}$ and $p(e) = \{4, 5, 6, 7\}$, and thus, $\text{pro}(e) = 700$, $\text{pro}(e) = 860$, and $\text{pro}(f) = 750$. According to Example 3, $dP(de) = \{4, 5, 6\}$ and $dP(df) = \{10\}$, and thus, $\text{pro}(d) + \sum_{P_k \in dP(de)} \text{value}(P_k) = 1050$ and $\text{pro}(df) = \text{pro}(d) + \sum_{P_k \in dP(df)} \text{value}(P_k) = 1400$. $dP(def) = \{10\}$, so $\text{pro}(def) = \text{pro}(de) + \sum_{P_k \in dP(def)} \text{value}(P_k) = 1400$.

2.5. Multicore Processor Platform. A multicore processor (MCP) is a physical chip including many separate cores in the same circuit [22]. MCPs enable executed multiple missions concurrently to increase the performance of applications. In a multicore processor, each core has a distinct L1 cache and execution module and uses a public L2 cache for the whole processor. This makes the greatest use of the resources and optimizes the communication between inter-cores. If several tasks carry on separate cores of the same circuit and if they portion data that match in the cache, then the public last-level cache between cores will reduce the data

Input: product database D and a minimum given threshold δ
Output: E_{result}, the list of EIs
1 root=NULL
2 Scan D to calculate the whole profit of $D(S)$, the index of profit (pro), and erasable 1-itemsets with their pidsets (E_1)
3 Sort E_1 by the size of their pidsets in descending order
4 $root$ = root $\cup$ E_1, where E_i is a child node
5 For each (k in $root$) do
6 Start new task t_k
7 **pMEI_Ext** (k, t_k)
8 End for

Algorithm 1: pMEI method.

replication. Hence, it is further effective in interaction. The major improvement of multicore processors is decreasing the temperature occurring off CPU and to extensively growing the speedup of processors while it is low cost than distributed systems, so it broadly applied in many fields such as computer vision, social network, image processing, and embedded system.

In data mining, there are many studies using this architecture to enhance the performance. Nguyen et al. [23] implement a technique for parallel mining class association rules on a computer that has the multicore processor platform which uses SCR-tree (single constraint rule-tree) structure and task parallel mechanism on .NET framework. Huynh et al. in [24] utilize multicore processors for mining frequent sequential patterns and frequent closed sequential patterns which use DBV-tree (dynamic bit vector-tree) structure and data parallel strategy based on TPL (task parallel library). Laurent et al. [25] proposed PGP-mc, which uses parallel gradual pattern mining used by the POSIX thread library. Flouri et al. [26] implement GapMis-OMP, a tool for pairwise short read alignment used by the OpenMP application programming interface, and Sánchez et al. [27] propose SW (Smith-Waterman), a method comparing sequence lengths based on the CellBE hardware. Recently, Kan et al. [28] proposed the parallelization and acceleration of the shuffled complex evolution utilizing the multicore CPU and many-core GPU. In 2018, Le et al. [29] suggested a parallel approach for mining intersequence patterns with constraints, which used DBV-PatternList structure and task parallel approach on multicore architecture.

3. A Parallel Method for Erasable Itemset Mining

Our proposed approach (Algorithm 1) in this study first determines the total profit of dataset, the erasable 1-itemsets (E_1) with their pidsets (line 2). Then, pMEI will sort E_1 by the size of their pidsets in not ascending order (line 3) and add them to child node of the root node (line 4). Finally, for each node in root, pMEI will start a new task (line 6) and call pMEI_Ext procedure to process this node with the created task in parallel that is a light-weight object in .NET framework for handling a parallel

Input: node p, the task T
1 **for** i $\leftarrow$ 0 to$|E_v|$
2 $E_{next} \leftarrow \varnothing$
3 **for** j $\leftarrow$ (i + 1) to$|E_v|$ **do**
4 E.Items=E_v [i].Items $\cup$ E_v [j].Items
5 (E.pidset, pro) $\leftarrow$ E_v [i].pidset\E_v [j].pidset
6 E.pro=E_v [i].pro+pro
7 **if** E.pro<$T\times\delta$ **then**
8 $E_{next} \leftarrow E$
9 $E_{result} \leftarrow E$
10 **if** $|E_{next}| > 1$ **then**
11 **pMEI_Ext**(E_{next})

Algorithm 2: Procedure pMEI_Ext.

element of work. It can be used when we would like to perform something in parallel. The works (or jobs) are stretched across multiple processors to maximize performance of computer. Tasks are adjusted for leveraging multicore processors to improve performance.

For procedure pMEI_Ext, the algorithm will combine each node in root together and create the next level of candidates. This strategy will be used until there is no new EI to create. The entire procedure pMEI_Ext task will be performed in parallel to achieve the good performance.

An outstanding point of pMEI algorithm is that it uses a dynamic load balancing mechanism. pMEI uses a queue to store jobs (a list of work to perform), and if the queue is not empty and there exists a task that is idle, then task is assigned a job to execute. In contrast, if the queue is full, the task will perform a job until completion. After a task completes a job or is idle, then it will be immediately assigned a new job and this job will be removed from the queue, and the process continues until the queue is empty. This mechanism is effective and can help avoid both task idling and achieve balanced workloads.

In addition, one of the differences between the pMEI and parallel methods in [24, 29] is in the sorting strategy to balance the search space. For mining frequent patterns, we can sort patterns according to their supports. In mining EIs, we do not need to compute the support of itemsets, so

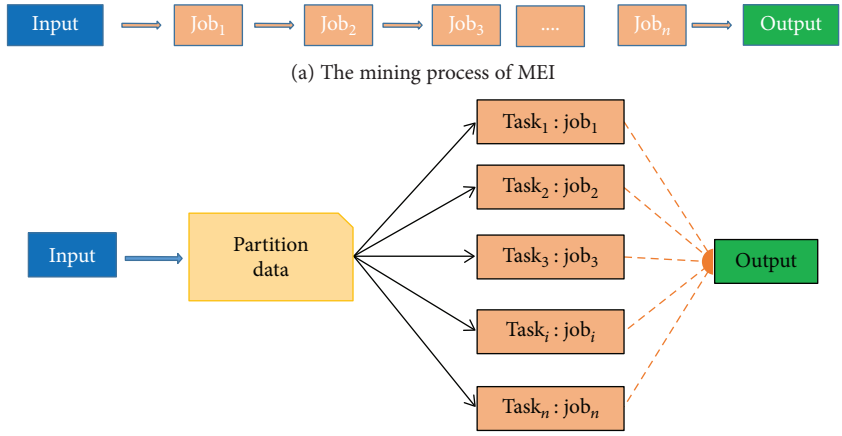


Figure 1: Illustration of the mining process of MEI and pMEI.

Table 2: Dataset descriptions.

Dataset	Number of products	Number of items	Average length	Density
Accidents	340183	468	33.8	0.072
Chess	3196	76	35	0.460
Connect	67557	130	43	0.333
Mushroom	8124	120	23	0.192
Pumsb	49046	2113	74	0.035
T10I4D100K	100000	870	10	0.011

pMEI sorts itemsets according to the size of their pidsets to balance the search space.

The illustration of the mining process of MEI and pMEI is shown in Figure 1.

pMEI executed in depth-first search; the result (EIs) is writing to global memory. Thus, it does not need to merge or synthesize process. In addition, pMEI uses a global queue for parent task and local queue for child task; both are accessed in LIFO order. The synchronization between tasks is not necessary because of the task using local queue does not involve any shared data.

4. Experimental Results

This section presents the results of the experiments that were executed on a PC with Intel Core I5-6200U (2.30 GHz, 4 threads) with 4 GB RAM and implemented in C# in Visual Studio 2015.

The experiments have been performed on Accidents, Chess, Connect, Mushroom, Pumsb, and T10I4D100K datasets which were downloaded from UCI datasets (http://fimi.ua.ac.be/data/). We have added a new column to hold the value associated to each product because the value of products has not existed in these datasets. The value was calculated based on the normal distribution $N(100, 50)$. The characteristics of these datasets are exhibited in Table 2 and are accessible at http://sdrv.ms/14eshVm.

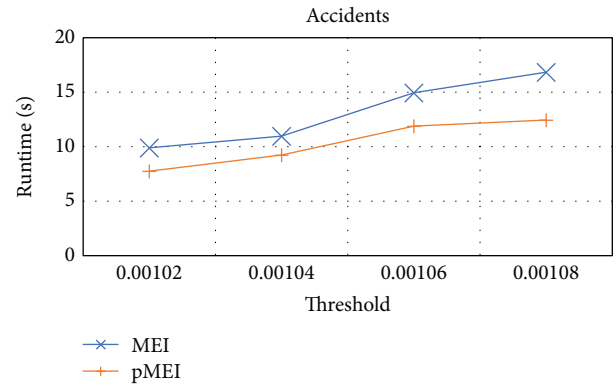


Figure 2: Runtimes of the MEI and pMEI methods on the Accidents.

In this part, we will evaluate the memory manipulation and running time of the suggested algorithm with MEI algorithm [15] to show the effectiveness of pMEI algorithm.

4.1. Running Time. We evaluate the execution times between MEI and pMEI algorithms on six experimental datasets (Figures 2–7). Note that the running times are averaged across five runs.

For dense datasets, such as Chess, Connect, and Mushroom, the execution time of pMEI is much better than MEI (Figures 3–5). In detail, for Chess dataset at $\delta = 30\%$, pMEI

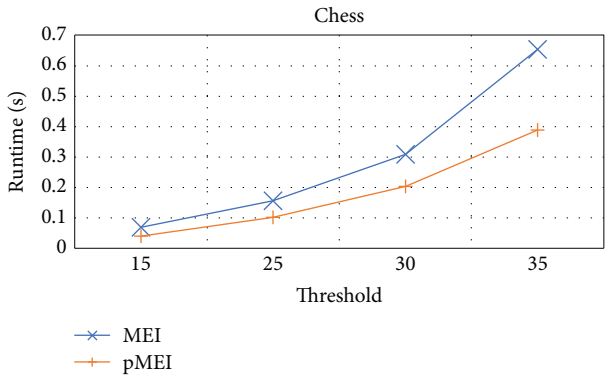


Figure 3: Runtimes of the MEI and pMEI methods on the Chess.

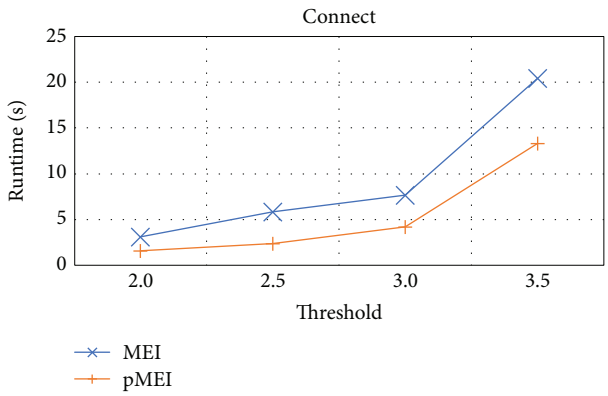


Figure 4: Runtimes of the MEI and pMEI methods on the Connect.

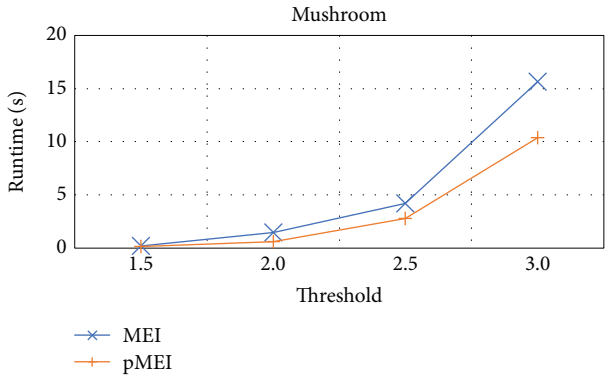


Figure 5: Runtimes of the MEI and pMEI methods on the Mushroom.

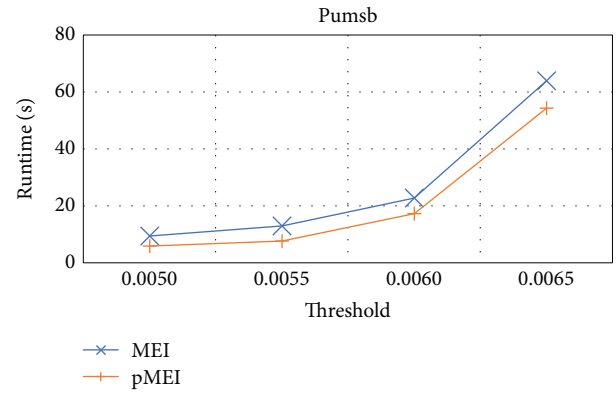


Figure 6: Runtimes of the MEI and pMEI methods on the Pumsb.

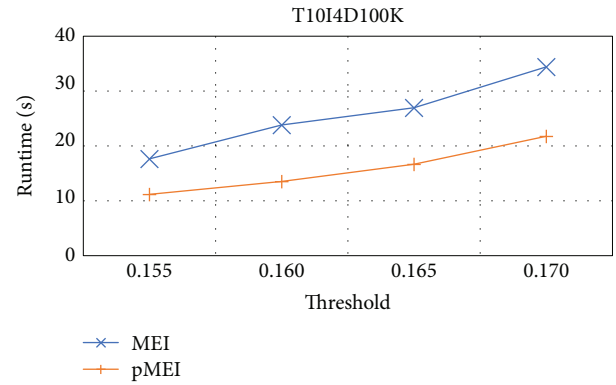


Figure 7: Runtimes of the MEI and pMEI methods on the T10I4D100K.

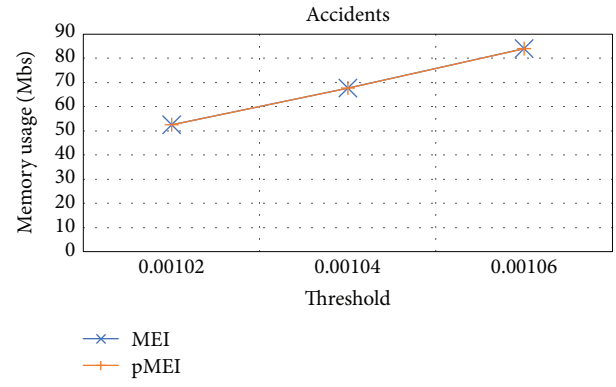


Figure 8: Memory usage of the MEI and pMEI methods on the Accidents.

only requires 0.203 s while MEI requires 0.31 s; and at $\delta =$ 35%, pMEI only requires 0.389 s while MEI requires 0.654 s, respectively. Specifically, this gap will be increased as the threshold increases. Like Connect dataset, at $\delta = 3\%$, the time gap (Δs) between the execution time of pMEI and that of MEI is 3.49 s while at $\delta = 3.5\%$, the time gap is 7.07 s.

For sparse datasets, such as Accidents, Pumsb, and T10I4D100K, the time gaps between pMEI and MEI are small (Figures 2, 6, and 7). Therefore, pMEI outperforms MEI in terms of the execution times for all experimental datasets especially for dense datasets.

4.2. Memory Usage. For all experimental datasets, pMEI and MEI have the same memory usage (see Figures 8–13). In summary, pMEI improves the execution times for mining

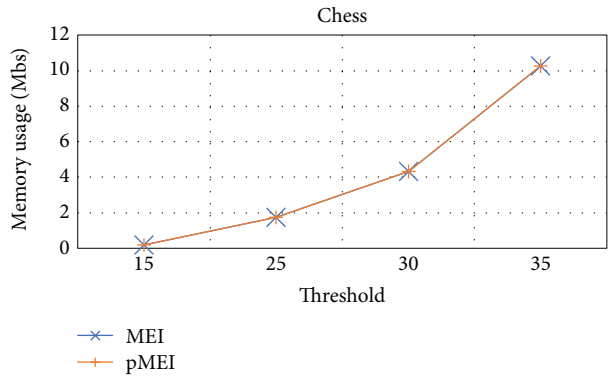


Figure 9: Memory usage of the MEI and pMEI methods on the Chess.

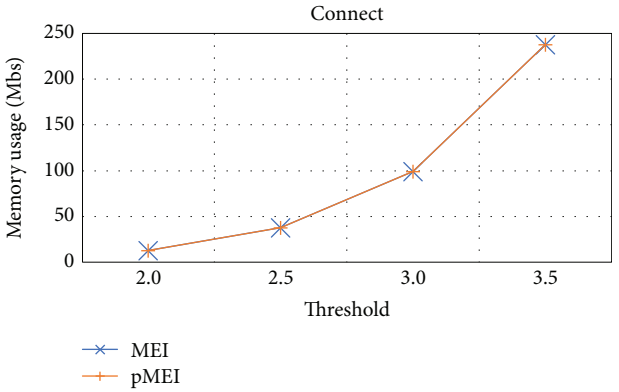


Figure 10: Memory usage of the MEI and pMEI methods on the Connect.

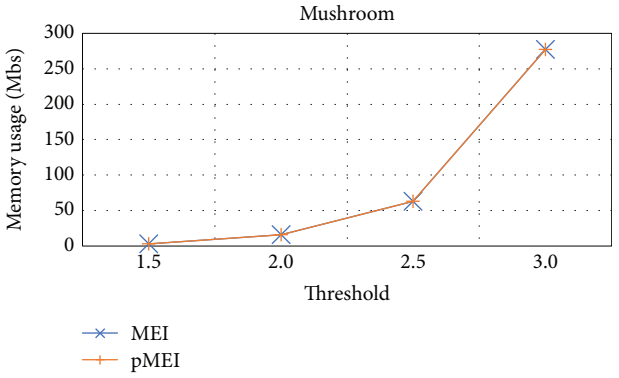


Figure 11: Memory usage of the MEI and pMEI methods on the Mushroom.

EIs on all experimental datasets while keeping the memory usage compared with MEI algorithm.

To evaluate the effectiveness of multicore systems, we executed the proposed method with the various numbers of cores (Figure 14). The speedup increases nearly two times on 2 cores and increases nearly four times on 4 cores. The

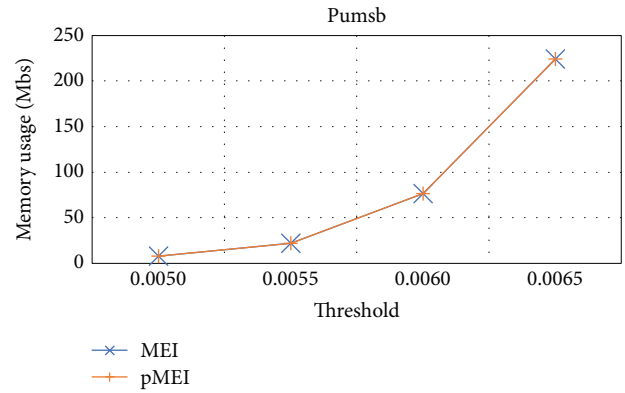


Figure 12: Memory usage of the MEI and pMEI methods on the Pumsb.

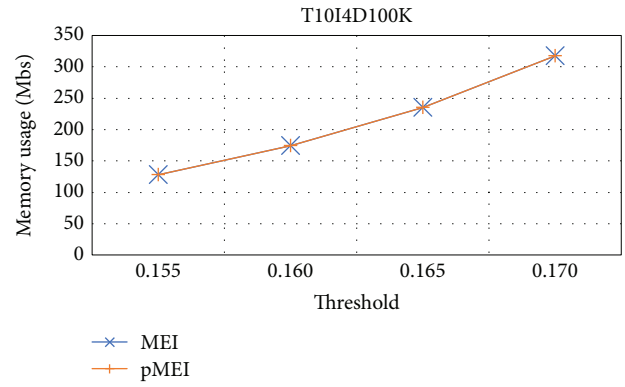


Figure 13: Memory usage of the MEI and pMEI methods on the T10I4D100K.

average speedup rate is 1.94, 1.95, 1.95, 1.98, 1.99, and 2.06 on 2 cores and 3.82, 3.87, 3.82, 3.80, 3.82, and 3.93 on 4 cores with Accidents, Chess, Connect, Mushroom, Pumsb, and T10I4D100K datasets, respectively. Generally, the speedup is always proportional to the number of cores of the computer.

5. Conclusions

This study proposed a proficient technique for mining EIs, namely, pMEI based on multicore computers to enhance the performance. This method overcomes these drawbacks of parallel computing approaches including the interactive expense cost, synchronization, and data duplication. A dynamic mechanism for load balancing the processor workloads was also used. Experiments show that pMEI is better than MERIT for mining EIs in execution time.

In the future, we will study mining the top-rank-k erasable closed itemsets and maximal erasable itemsets. In addition, we will expand the study of EI mining associated with some kinds of item constraints. Besides, approaches for mining such patterns on distributed computing systems will be developed.

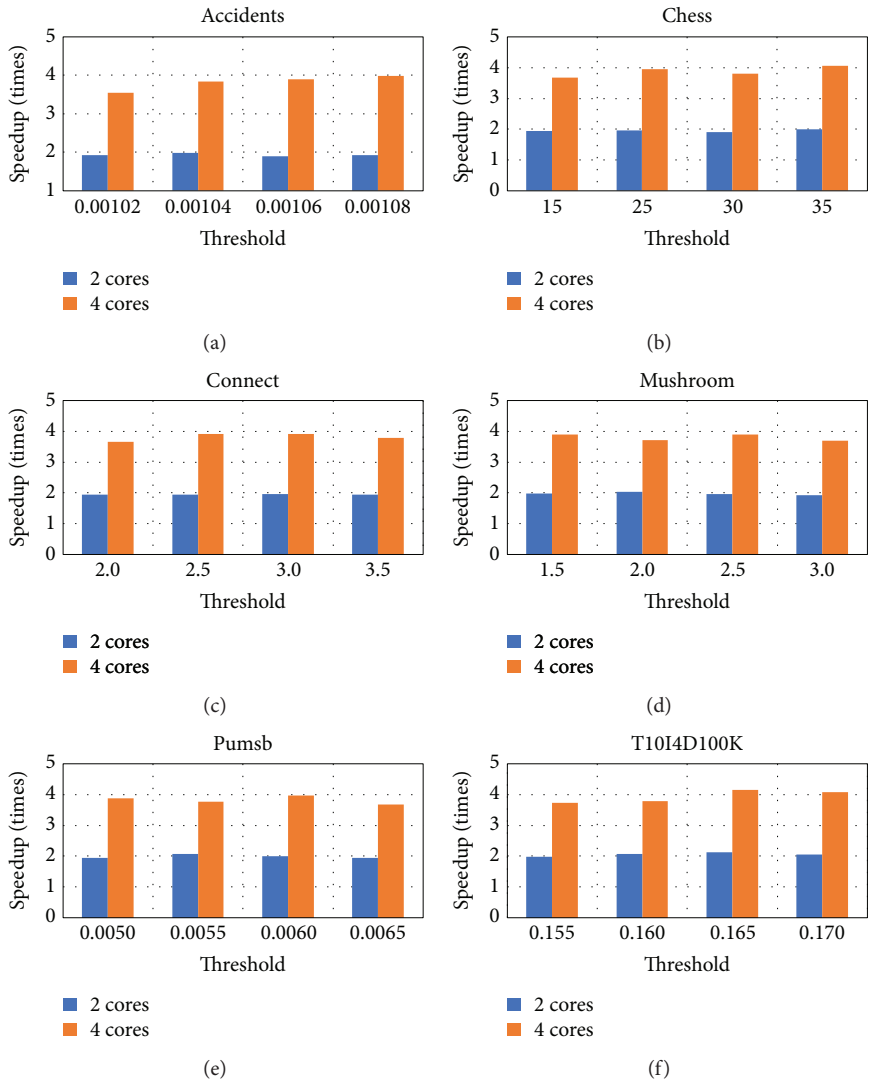


Figure 14: Speedup of the pMEI with the different numbers of cores on various datasets.

Data Availability

The product data used to support the findings of this study have been deposited in the Frequent Itemset Mining Dataset Repository (http://fimi.ua.ac.be/data/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors wish to thank Tuong Le for his valuable comments and suggestions.

References

[1] T. Mai, B. Vo, and L. T. T. Nguyen, "A lattice-based approach for mining high utility association rules," *Information Sciences*, vol. 399, pp. 81–97, 2017.

[2] D. T. Hai, L. H. Son, and T. L. Vinh, "Novel fuzzy clustering scheme for 3D wireless sensor networks," *Applied Soft Computing*, vol. 54, pp. 141–149, 2017.

[3] N. Indurkhya, "Emerging directions in predictive text mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 4, pp. 155–164, 2015.

[4] Z.-H. Deng, "Diffnodesets: an efficient structure for fast mining frequent itemsets," *Applied Soft Computing*, vol. 41, pp. 214–223, 2016.

[5] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and J. Zhan, "Mining of frequent patterns with multiple minimum supports," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 83–96, 2017.

[6] B. Vo, S. Pham, T. Le, and Z.-H. Deng, "A novel approach for mining maximal frequent patterns," *Expert Systems with Applications*, vol. 73, pp. 178–186, 2017.

[7] T. Kieu, B. Vo, T. Le, Z.-H. Deng, and B. Le, "Mining top-*k* co-occurrence items with sequential pattern," *Expert Systems with Applications*, vol. 85, pp. 123–133, 2017.

[8] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, J. M.-T. Wu, and J. Zhan, "Extracting recent weighted-based patterns

from uncertain temporal databases," *Engineering Applications of Artificial Intelligence*, vol. 61, pp. 161–172, 2017.

[9] R. U. Kiran, J. N. Venkatesh, M. Toyoda, M. Kitsuregawa, and P. K. Reddy, "Discovering partial periodic-frequent patterns in a transactional database," *Journal of Systems and Software*, vol. 125, pp. 170–182, 2017.

[10] A. Hellal and L. Ben Romdhane, "Minimal contrast frequent pattern mining for malware detection," *Computers & Security*, vol. 62, pp. 19–32, 2016.

[11] J. Wen, M. Zhong, and Z. Wang, "Activity recognition with weighted frequent patterns mining in smart environments," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6423–6432, 2015.

[12] Z.-H. Deng, G.-D. Fang, Z.-H. Wang, and X.-R. Xu, "Mining erasable itemsets," in *2009 International Conference on Machine Learning and Cybernetics*, pp. 67–73, Hebei, China, 2009.

[13] T. Le and B. Vo, "The lattice-based approaches for mining association rules: a review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 140–151, 2016.

[14] Z.-H. Deng and X.-R. Xu, "Fast mining erasable itemsets using NC_sets," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4453–4463, 2012.

[15] T. Le and B. Vo, "MEI: an efficient algorithm for mining erasable itemsets," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 155–166, 2014.

[16] G. Nguyen, T. Le, B. Vo, and B. Le, "EIFDD: an efficient approach for erasable itemset mining of very dense datasets," *Applied Intelligence*, vol. 43, no. 1, pp. 85–94, 2015.

[17] B. Vo, T. le, G. Nguyen, and T.-P. Hong, "Efficient algorithms for mining erasable closed patterns from product datasets," *IEEE Access*, vol. 5, no. 1, pp. 3111–3120, 2017.

[18] T. Le, B. Vo, and S. W. Baik, "Efficient algorithms for mining top-rank-*k* erasable patterns using pruning strategies and the subsume concept," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 1–9, 2018.

[19] B. Vo, T. le, W. Pedrycz, G. Nguyen, and S. W. Baik, "Mining erasable itemsets with subset and superset itemset constraints," *Expert Systems with Applications*, vol. 69, pp. 50–61, 2017.

[20] G. Lee, U. Yun, H. Ryang, and D. Kim, "Erasable itemset mining over incremental databases with weight conditions," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 213–234, 2016.

[21] U. Yun and G. Lee, "Sliding window based weighted erasable stream pattern mining for stream data applications," *Future Generation Computer Systems*, vol. 59, pp. 1–20, 2016.

[22] A. Vajda, "Multi-core and many-core processor architectures," in *Programming Many-Core Chips*, pp. 9–43, Springer, Boston, MA, USA, 2011.

[23] D. Nguyen, B. Vo, and B. Le, "Efficient strategies for parallel mining class association rules," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4716–4729, 2014.

[24] B. Huynh, B. Vo, and V. Snasel, "An efficient method for mining frequent sequential patterns using multi-core processors," *Applied Intelligence*, vol. 46, no. 3, pp. 703–716, 2017.

[25] A. Laurent, B. Négrevergne, N. Sicard, and A. Termier, "Efficient parallel mining of gradual patterns on multicore processors," in *Advances in Knowledge Discovery and Management*, vol. 398 of Studies in Computational Intelligence, pp. 137–151, Springer, Berlin, Heidelberg, 2012.

[26] T. Flouri, C. S. Iliopoulos, K. Park, and S. P. Pissis, "GapMis-OMP: pairwise short-read alignment on multi-core architectures," in *Artificial Intelligence Applications and Innovations*, vol. 382 of IFIP Advances in Information and Communication Technology, pp. 593–601, Springer, Berlin, Heidelberg, 2012.

[27] F. Sánchez, F. Cabarcas, A. Ramirez, and M. Valero, "Long DNA sequence comparison on multicore architectures," in *Euro-Par 2010 - Parallel Processing*, vol. 6272 of Lecture Notes in Computer Science, pp. 247–259, Springer, Berlin, Heidelberg, 2010.

[28] G. Kan, T. Lei, K. Liang et al., "A multi-core CPU and many-core GPU based fast parallel shuffled complex evolution global optimization approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 332–344, 2016.

[29] T. Le, A. Nguyen, B. Huynh, B. Vo, and W. Pedrycz, "Mining constrained inter-sequence patterns: a novel approach to cope with item constraints," *Applied Intelligence*, vol. 48, no. 5, pp. 1327–1343, 2018.

Hindawi
Complexity
Volume 2018, Article ID 2520706, 11 pages
https://doi.org/10.1155/2018/2520706

Research Article

Integrating Correlation-Based Feature Selection and Clustering for Improved Cardiovascular Disease Diagnosis

Agnieszka Wosiak and Danuta Zakrzewska

Institute of Information Technology, Lodz University of Technology, 90-924, Poland

Correspondence should be addressed to Danuta Zakrzewska; danuta.zakrzewska@p.lodz.pl

Received 20 April 2018; Accepted 17 September 2018; Published 14 October 2018

Guest Editor: Ireneusz Czarnowski

Based on the growing problem of heart diseases, their efficient diagnosis is of great importance to the modern world. Statistical inference is the tool that most physicians use for diagnosis, though in many cases it does not appear powerful enough. Clustering of patient instances allows finding out groups for which statistical models can be built more efficiently. However, the performance of such an approach depends on the features used as clustering attributes. In this paper, the methodology that consists of combining unsupervised feature selection and grouping to improve the performance of statistical analysis is considered. We assume that the set of attributes used in clustering and statistical analysis phases should be different and not correlated. Thus, the method consisting of selecting reversed correlated features as attributes of cluster analysis is considered. The proposed methodology has been verified by experiments done on three real datasets of cardiovascular cases. The obtained effects have been evaluated regarding the number of detected dependencies between parameters. Experiment results showed the advantage of the presented approach compared to other feature selection methods and without using clustering to support statistical inference.

1. Introduction

Nowadays, data play a very important role in medical diagnostics since, due to equipment development, an increasing amount of data can be collected and thus, a huge volume of information concerning patient characteristics can be acquired. However, the possibilities of using data in medical diagnosis depend on the efficacy of the applied techniques. In practise, medical diagnostics are mainly supported by statistical inference, though in many cases it does not appear effective enough. It is worth emphasising that in medicine, the results of analysis are expected to be implemented in real life and thus the efficiency and usefulness of the methods should be taken into consideration. To obtain valuable recommendations for diagnostic statements, more sophisticated analytical methods are required. Including data mining algorithms to the process seems to be appropriate. Those techniques were recognized as efficient by Yoo et al. [1], who indicated that the application of descriptive and predictive methods are useful in biomedical as well as healthcare areas. In addition, stand-alone statistical analysis cannot be supportive in many cases, especially when correlations between attributes, considered as important by physicians, cannot be found. Such situation usually takes place for datasets of great standard deviation values [2]. What is more, dissimilarities or inconsistencies within the datasets can appear due to incorrect measurements or distortions. The presence of these kinds of deviations may lead to the rejection of true hypothesis; for example, such situation takes place when datasets are of small sizes. In these cases, supporting medical diagnosis becomes a complicated task, particularly when the number of attributes exceeds the number of records.

Integrating statistical analysis and data mining may not only improve the effectiveness of the obtained results, but also, by finding new dependencies between attributes, enable a multiperspective approach to medical diagnosis.

The research concerning the integration of cluster analysis and statistical methods on medical data, for defining the phenotypes of clinical asthma, has been presented in [3].

The research was proposed against other models of asthma classification and, according to authors, it might have played a supporting role for different phenotypes of a heterogeneous asthma population. Data mining methods have been used in several clinical data systems. A survey of these systems and the applied techniques has been presented in [4]. Data mining techniques have been also considered in different clinical decision support systems for heart disease prediction and diagnosis in [2]. However, in the investigation results, the authors stated that the examined techniques are not satisfactory enough. Moreover, a solution for the identification of treatment options for patients with heart diseases is still lacking. Statistical inference of heart rate and blood pressure was investigated in [5]. The authors examined the correlation between raw data, then they examined the correlation between filtered data, and finally they applied the least squares approximation. In all the cases, the obtained correlation coefficients seemed to be unpredictable random numbers.

In this paper, we examine combining statistical inference and cluster analysis as a methodology supporting cardiovascular medical diagnosis. Including clustering in the preprocessing phase allows identifying groups of similar instances, for which respective parameters can be evaluated efficiently and thus statistical models of good quality can be created. Such an approach has been proposed in [6] to improve the performance of statistical models in hypertension problems in cardiovascular diagnosis. In the paper [7], a new reversed correlation algorithm (RCA) of an automatic unsupervised feature selection complemented the methodology. The RCA algorithm consisted of choosing subsequent features as the least correlated with their predecessors.

In the current research, we introduce a modification to the RCA that concerns the choice of the first attribute. Moreover, we extend the study [7] by comparing the performance of the considered algorithm with two other feature selection methods: correlation-based CFS and ReliefF. We also examine the effectiveness of the presented methodology regarding not only the statistical approach, but also the deterministic clustering algorithm with elbow criterion for determining the best number of clusters. Additionally, during the experiments we broaden the range of patients involved by changing the considered datasets. In the current research, instead of one of the three datasets gathered from children [7], we use a reference "CORONARY" dataset with a higher number of patient records. The dataset was derived from the UCI repository [8].

In this paper, we validate the performance of the investigated methodology applied to datasets of real patient records via numerical experiments. We consider three datasets of different proportions between the numbers of instances and attributes. The experimental results are evaluated by statistical inference performed on clusters. The results demonstrate that the statistical inference performed on clusters enable detection of new relationships, which have not been discovered in the whole datasets; thus, significant benefits of using the proposed hybrid approach for improving medical diagnosis can be recognized. The proposed feature selection algorithm outperforms the effects obtained by other considered techniques. As in all the analysed cases, we attained the best results regarding the numbers of discovered dependencies.

The remainder of the paper is organised as follows. In the next section, the cardiovascular disease diagnosis problem is introduced and the whole methodology is described including its overview, the RCA feature selection, and all the considered algorithms. Next, the experiments carried out for the methodology evaluation are presented regarding the dataset characteristics, and the results obtained at all the stages of the proposed method are discussed. The final section presents the study's conclusions and delineates future research.

2. Materials and Methods

2.1. Heart Disease Diagnosis Problem. The detection and diagnosis of heart diseases are of great importance due to their growing prevalence in the world population. Heart diseases result in severe disabilities and higher mortality than other diseases, including cancer. They cause more than 7 million deaths every year [9, 10].

Heart diseases include a diverse range of disorders: coronary artery diseases, stroke, heart failure, hypertensive heart disease, rheumatic heart disease, heart arrhythmia, and many others. Therefore, the detection of heart diseases from various factors is a complex issue, and the underlying mechanisms vary, depending on the considered problem and the conditions that affect the heart and the whole cardiovascular system. Moreover, there are many additional socioeconomic, demographic, and gestational factors that affect heart diseases, and are considered as their main reasons [11–13].

To improve early detection and diagnosis of heart abnormalities, new factors and dependencies that may indicate cardiovascular disorders are searched. Statistical data analysis supports the evaluation of the characteristics of the parameters in medical datasets and helps in discovering their mutual dependencies. However, in some situations the significance of statistical inference between medical attributes may be interfered by a wide range of values, subsets of relatively dissimilar instances, or outliers. Thus, there is a strong need for new techniques that will support statistical inference in finding parameter dependencies and thereby improve medical diagnosis.

2.2. The Method Overview. The considered methodology for supporting the process of medical diagnosis by patient dataset analysis consists of three main steps. They are preceded by data preparation, which aims at adjusting original datasets to analysis needs. The proposed steps can be presented as follows:

(1) Feature selection, based on statistical analysis of correlation coefficients, which enables appointing the set of attributes for clustering

(2) Finding groups of similar characteristics, including a validation technique used to determine the appropriate number of clusters

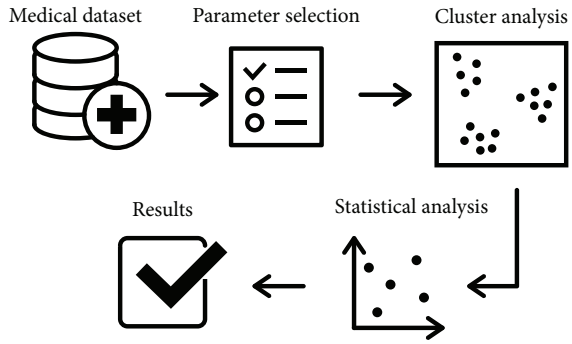


Figure 1: Overview of the methodology.

(3) Statistical analysis performed in clusters to find new dependencies between all the considered parameters

The general overview of the method is shown in Figure 1. We assume that clustering and statistical analysis are applied on the separate subsets of attributes. The descriptions of the main steps of the methodology are presented in Subsections 2.3, 2.4, and 2.5.

2.3. Feature Selection. Patient records usually contain many attributes that may be used for supporting medical diagnosis. However, the performance of the diagnostics process may depend on the choice of the attributes in all the phases of the considered methodology. The quality of results obtained in the final step depends not only on the choice of parameters used for finding correlations, but also depends on the quality of patient groups and thus on the subset of attributes used in the clustering process. Therefore, the process of feature selection for cluster analysis is crucial for the whole presented methodology of medical diagnosis.

Regarding the main supporting tool, which is a statistical inference according to physician preferences, we propose the reversed correlation algorithm (RCA) that uses correlation coefficients but in a reversed order. This means that we look for features that are the least correlated with all their predecessors.

First, we start building a subset of features with the attribute that is the least correlated with the others. Then, correlation coefficients between the chosen feature and the rest of the parameters are calculated. The attribute with the lowest correlation value is indicated as the second feature. The obtained subset of two features is further extended by adding the attribute of the correlation coefficient with the lowest value between the subset and the rest of the parameters. The process of appending the features of the lowest correlation values is repeated unless all the correlation coefficients indicate statistically significant dependencies (respective values exceed thresholds) or the number of features in the subset is equal to the determined percentage of the total number of attributes. The whole procedure is presented in the Algorithm 1.

In order to compare the results of the proposed feature selection algorithms, two other techniques have been considered: the opposite approach represented by the correlation-based feature selection (CFS) and an extension of the relief algorithm called ReliefF.

Correlation-based feature selection (CFS) ranks attributes according to a heuristic evaluation function based on correlations [14]. The function evaluates subsets made of attribute vectors, which are correlated with the class label, but independent of each other. The CFS method assumes that irrelevant features show a low correlation with the class and therefore should be ignored by the algorithm. On the other hand, excess features should be examined, as they are usually strongly correlated with one or more of the other attributes. The criterion used to assess a subset of l features can be expressed as follows:

$$M_S = \frac{l\overline{t_{cf}}}{\sqrt{l + l(l-1)\overline{t_{ff}}}}, \quad (1)$$

where M_S is the evaluation of a subset of S consisting of l features, $\overline{t_{cf}}$ is the average correlation value between features and class labels, and $\overline{t_{ff}}$ is the average correlation value between two features.

There exist different variations of CFS that employ different attribute quality measures, such as Symmetrical Uncertainty, normalized symmetrical Minimum Description Length (MDL), or Relief.

Relief algorithm, described in [15], concerns the evaluation of attributes based on the similarity of the neighbouring examples in the set of analysed instances [16]. For the given set of training instances, sample size, and the relevancy threshold τ, Relief detects features that are statistically consistent with the target task. Relief picks an instance X from the set and its two nearest neighbours: one of the same class—called "near-hit" and one of the opposite class—called "near-miss". Then, it updates the feature weight vector W for every triplet and uses it to determine the average relevance feature vector. The algorithm selects those features for which the value of the average weight, called relevance level, exceeds the given threshold value τ.

ReliefF algorithm has been proposed in [16]. Contrary to Relief, it is not limited to two class problems, it is more effective and can deal with noisy or incomplete data, for missing values of attributes are treated probabilistically. Similarly to Relief, ReliefF randomly selects an instance X, but it searches for the determined number of the nearest neighbours from the same class, called "nearest hits," and the same number of the nearest neighbours from every different class ("nearest misses"). Then, it updates the vector W of estimations of the qualities for all the attributes depending on their values for X and sets of hits and misses.

2.4. Cluster Analysis. Cluster analysis is an unsupervised classification technique, which can be used for grouping complex multidimensional data. Opposite to supervised methods, the profiles of obtained groups cannot be obviously stated and using additional techniques for discovering the meaning of clustering is required in many cases [17]. On the other side, statistical analysis is the most popular tool used in the medical field. Therefore, in this area, combining clustering and

Input: $F = f_1, f_2, f_3, \ldots f_n$ */* set of all the features */;*
P/ statistical significance level */;*
R/ a threshold for correlation coefficient levels */;*
N/ the maximum of features for the subset/*;*
Output: F_s */* selected subset of features */;*
(1) Initialize F_s *with feature* $f_j \in F$ *that is the least correlated with other ones;*
(2) do
(3) Compute $C_{ij}(F_s, F \setminus F_s)$ *as a vector of correlation coefficients between* F_s *and each* $f_i \in \{F \setminus F_s\}$*;*
(4) Choose $f_j \in \{F \setminus F_s\}$ *with the lowest value of correlation coefficient in a vector* $C_{ij}(F_s, F \setminus F_s)$*;*
(5) Include f_j *in* F_s
(6) while $(s < N \text{ AND } p > P \text{ AND } C_{ij}(F_s, F \setminus F_s) < R)$.

Algorithm 1: Proposed feature selection algorithm using reversed correlations

statistical inference may not only enable patient grouping, but also finding dependencies between their characteristics and thus supporting medical diagnostics.

In further investigations, which aim at evaluating the presented technique regarding its efficiency on cardiovascular data, simple popular clustering algorithms will be considered, for such techniques are expected to be comprehensible for physicians.

We will examine two different clustering approaches: deterministic and probabilistic. The first approach will be represented by k-means algorithm, which in comparison to other techniques, demonstrated good performance for medical data regarding accuracy as well as lower root mean square error [18]. The k-means algorithm is one of the most popular partitioning methods, where clusters are built around k centers, by minimizing a distance function. The goal of the algorithm is to find the set of clusters for which the sum of the squared distance values between their points and respective centers is minimal. As the distance function, the Euclidean metric is used, which has been applied in most of the cases [19, 20]. The first k centers are usually chosen at random, which does not guarantee finding optimal clusters. To increase the chance of finding the optimum, the algorithm is usually launched several times with different initial choices and the result of the smallest total squared distance is indicated [20].

The goal of a statistical model is to find the most probable set of clusters on the basis of training data and prior expectations. As a representative of these techniques, EM (expectation-maximization) algorithm, based on the finite Gaussian mixtures model, has been investigated. EM generates probabilistic descriptions of clusters in terms of means and standard deviations [17]. The algorithm iteratively calculates the maximum likelihood estimated in parametric models in the presence of missing data [21]. EM enables using cross-validation for selecting the number of clusters and thus obtaining its optimal value [20]. That feature allows avoiding the determination of the number of clusters at the beginning of the algorithm.

The choice of the optimal number of clusters is one of the most important parts of the clustering process. In the case of the k-means algorithm, the elbow technique was used. It is based on the statement that the number of clusters should increase together with the increase of the quantity of information. The last number of clusters, for which a gain value was augmented, should be indicated as optimal. On the graph, where validation measure is plotted against the number of clusters, that point is presented as an angle, and called the elbow. There are cases, when angles cannot be unambiguously identified, and the number of clusters indicated by the elbow technique should be confirmed by other methods.

Thus, considering two clustering methods equipped with different techniques for choosing the optimal number of clusters may help in confirming the right choice. However, it is worth noticing that in medicine there exists the usual intent to split the whole dataset into two groups and thus the number of clusters is very often equal to two [18]. Besides, in medical applications, the number of collected instances is very small and the high number of clusters may result in small group sizes and in less reliable medical inference, as the consequence of the lack of statistical tests of high power [19, 22].

2.5. Statistical Analysis. Before carrying out statistical inference, the assessment of measures of descriptive statistics should be performed. Such an approach allows detecting errors that were not identified during the data preparation phase. As the main descriptors, for which the evaluation is indicated, one should mention central tendency measures (arithmetic mean, median, and modal) as well as dispersion measures (range and standard deviation). Next, an appropriate test is run as a part of the statistical analysis process. The test should be chosen according to the type and the structure of analysed data regarding such characteristics as attribute types, the scale type, the number of experimental groups, and their dependencies, as well as the test power. Additionally, the selection should be consistent with the requirements of the USMLE (The United States Medical Licensing Examination). In the presented research, these are considered the tests usually applied in medical diagnostics [2]:

(i) Kolmogorov–Smirnov test, which is used to check the normality of distribution of the attributes

(ii) Unpaired two-sample Student's t-test for the significance of a difference between two normally distributed values of attributes

(iii) Mann–Whitney U test, which is a nonparametric test for the determination of significant differences, where attributes are in nominal scales

Pearson's correlation coefficient $r_P(x, y)$ is used to express the impact of one variable measured in an interval or ratio scale to another variable in the same scale. Spearman's correlation $r_S(x, y)$ test is used, in the case when one or both of the variables are measured with an ordinal scale, or variables are expressed as an interval scale, but the relationship is not a linear one.

3. Results and Discussion

The performance of the proposed methodology has been examined by experiments conducted on the real datasets collected for supporting heart disease diagnosis. The statistical analysis results obtained for clusters have been compared with the ones taken for the whole datasets.

3.1. Data Description. The experiments were carried out on three datasets:

(i) "HEART"

(ii) "IUGR"

(iii) "CORONARY"

The "HEART" dataset consisted of 30 cases collected to discover dependencies between arterial hypertension and left ventricle systolic functions. The "IUGR" dataset includes 47 instances of children born with intrauterine growth restriction (IUGR), gathered to find out dependencies between abnormal blood pressure and being born as small for gestational age. The data of both of the datasets were collected in the Children's Cardiology and Rheumatology Department of the Second Chair of Paediatrics at the Medical University of Lodz.

Each dataset was characterized by two types of parameters: the main and the supplementary ones, all of them gathered for discovering new dependencies. The attributes correspond to high blood pressure and include echocardiography and blood pressure assessment, prenatal and neonatal history, risk factors for IUGR, and family survey of cardiovascular disease, as well as nutritional status. There were no missing values within the attributes. The full medical explanations of the data are given in [13, 23].

The "CORONARY" dataset also refers to cardiovascular problems. It comes from the UCI Machine Learning Repository [8]. The dataset contains the records of 303 patients, each of which is described by 54 features. The attributes were arranged in four groups of features: demographic, symptom and examination, and ECG, as well as laboratory and echo ones [24–26].

The summary of characteristics for all the datasets was presented in Table 1. The datasets have been chosen to ensure diversification of the mutual proportion between the number of instances and the number of attributes:

Table 1: The characteristics of datasets.

Dataset	Instances	Main attributes	Supplementary attributes
HEART	30	14	35
IUGR	47	6	40
CORONARY	303	10	44

(i) The number of instances in the "HEART" dataset is smaller than the number of parameters

(ii) The number of instances in the "IUGR" dataset is comparable with the number of attributes

(iii) In the "CORONARY" dataset, the number of instances is greater than the number of parameters

Tables 2–4 describe the selection of the parameters with the main statistical descriptors: the values of range, median or mean, and standard deviation (SD).

3.2. Selecting Relevant Features. For each dataset, only parameters concerning main characteristics were considered as initial attributes used for grouping. The selection of the appropriate features for building clusters has been performed by using three different techniques:

(1) The reversed correlation algorithm (RCA)

(2) CFS method

(3) ReliefF algorithm

The parameters necessary to run the RCA algorithm were chosen according to principles commonly approved in statistics (see [24, 28]):

(i) $N = 50\%\ n$ for the maximal number of features

(ii) $R = 0.3$ for the maximal value of correlation coefficients

(iii) $P = 0.05$ for the maximal value of statistical significance p value

In the case of the ReliefF algorithm, the threshold for the number of attributes included in the subset of selected features was set to $N = 50\%\ n$.

The subsets of features presented in Table 5 were obtained as the results of the proposed feature selection process. The first column of the table represents names of datasets, the second column represents the names of the feature selection algorithms, and the following columns contain the number and names of selected features in the order indicated by the algorithms.

3.3. Data Clustering. In the next step of the experiments, the clusters for diagnosed patients were created by using two clustering algorithms: k-means and EM implemented by WEKA Open Source software [20].

Table 2: Characteristics of attributes for the "HEART" dataset.

Attribute(s)	Description	Range	Median/mean (mean range)	SD (SD range)
Main attributes				
BMI	Current body mass index	17.00 to 25.00	22.16	1.64
Birth_weight	Birth weight	2500 to 4000	3158	392.00
SBP, DBP, ABPM-SBP, ABPM-DBP	Average systolic/diastolic blood pressure taken manually and by ABPM	61 to 150	74.87 to 136.97	5.22 to 7.04
HR	Heart rate	44 to 91	75.97	11.20
Risk factors	Risk factors	True/false	—	—
Supplementary attributes				
IVSd, IVSs, PWDd, PWDs, LVDd, LVDs	Left ventricular dimensions	5.00 to 56.00	8.00 to 46.03	1.51 to 9.02
EF, SF	Systolic function	34 to 84	40 to 70	3 to 5
Sm, Sml, V/S/SR long/rad/circ	Tissue Doppler echocardiography parameters	−37 to 40.17	−27.25 to 29.64	0.42 to 6.35

Table 3: Characteristics of attributes for the "IUGR" dataset.

Attribute(s)	Description	Range	Median/mean (mean range)	Mode (N) or SD (SD range)
Main attributes				
Birth_weight	Birth weight	1980–2850	2556.70	2700 (7)
Head_circ	Head circumference	29–35	33	32 (16)
Gest_age	Gestational age	38–42	39	—
Apgar	Apgar score at 1 min	7–10	9	9 (23)
5_Percentile	Growth chart factor	True/false	—	False (25)
Supplementary attributes				
SBP, DBP	Average systolic/diastolic blood pressure	55–137	55–115	5.03–8.73
SBP load, DBP load	Blood pressure loads	0–96	9–20	10–21
LVm	Left ventricular mass (Simone, Devreux)	17.65–93.21	30.26–59.11	6.91–12.91
Risk factors	Risk factors	True/false	—	—

Clusters were built regarding the main characteristics and the parameters indicated by feature selection methods, namely RCA, CFS, and ReliefF.

In the case of the EM algorithm, the best number of clusters was indicated by using cross-validation. To choose the best number of clusters for k-means clustering, the elbow criterion has been applied and within cluster sum of squares has been considered as a validation measure. The charts of validation measures plotted against the number of clusters with marked elbow points for HEART, IUGR, and CORONARY datasets, respectively, are presented in Figures 2–4. For better result visualisation, the values of within cluster sum of squares were normalized.

The results of clustering are presented in Table 6, where the first column describes datasets, the second column contains the names of the feature selection methods, and the last two columns present the number of clusters and clustering schemes.

3.4. Statistical Inference. Correlation values obtained for the clusters were compared with the ones taken for the whole group of diagnosed patients in terms of different selection techniques. Comparison of results confirmed the effectiveness of the proposed methodology. For each dataset, we obtained a greater number of statistically significant correlations in clusters which may lead to improved medical diagnosis in the future. By significant correlations we mean values with correlation coefficient $r \geq 0.3$ and p value ≤ 0.05 ([27, 28]). The biggest growth of the number of correlations concerns the HEART dataset, where the number of instances is smaller than the number of parameters. The numbers of detected correlations are presented in Table 7.

One can easily notice that the results attained by the unsupervised RCA feature selection technique and supervised ReliefF algorithm were comparable; however, the first method outperforms the second one in the case of the IUGR dataset and k-means technique. As in many cases, the supervised technique of feature selection cannot be used due to the lack of information on labels; one can expect that the RCA method would be indicated as more often used than the ReliefF algorithm.

Table 4: Characteristics of attributes for the "CORONARY" dataset.

Attribute(s)	Description	Range	Median/mean (mean range)	Mode (N) or SD (SD range)
Main attributes				
Q wave, St elevation, St depression, Tinversion, LVH, poor R progression	ECG parameters	Yes/no	—	—
FBS	Fasting blood sugar	62–400	119	52
EF-TTE	Ejection fraction—transthoracic echocardiography	15–60	47	9
Region RWMA	Regional wall motion abnormalities	0–4	0 (217)	—
Supplementary attributes				
Age	Age	30–86	58.00	10.39
Weight	Weight	48–120	73.83	11.89
Sex	Sex	Male/female	—	Male (176)
BMI	BMI	18–41	27.25	4.10
DM, HTN, current smoker, ex-smoker, FH, obesity, CRF, airway disease, thyroid disease, CHF, DLP	Diabetes mellitus, hypertension, current smoker, ex-smoker, family history, obesity, chronic renal failure, cerebrovascular accident, airway disease, thyroid disease, congestive heart failure, dyslipidemia	Yes/no	—	—
Edema, weak peripheral pulse, lung rales, systolic murmur, diastolic murmur, typical chest pain, dyspnea	Symptom and examination parameters	Yes/no	—	—
Cr, TG, LDL, HDL, BUN, ESR, HB, K, Na, WBC, lymph, neut, PLT	Laboratory parameters (creatine, triglyceride, low density lipoprotein, high density lipoprotein, blood urea nitrogen, erythrocyte sedimentation rate, haemoglobin, potassium, sodium, white blood cell, lymphocyte, neutrophil, platelet)	0.5–18,000	1.05–7652.04	0.24–2413.74

Table 5: Feature selection results.

Dataset	FS algorithm	Size	Supplementary attributes
HEART	RCA	6	Physical_activity, fundus, BMI, HR, height, birth_weight
	CFS	1	Weight
	ReliefF	6	Physical_activity, family_interview, weight, fundus, height, BMI
IUGR	RCA	3	Apgar_score, ponderal_index, 5_percentile
	CFS	1	Birth_weight
	ReliefF	3	Head_circ, ponderal_index, birth_weight
CORONARY	RCA	4	FBS, EF-TTE, St depression, LVH
	CFS	5	Q wave, Tinversion, FBS, EF-TTE, region RWMA
	ReliefF	5	Region RWMA, Tinversion, St depression, St elevation, Q wave

4. Conclusions

The process of computer-aided medical studies is usually based on only one of the data analysis methods, most often a statistical approach. In this paper, we present an approach that integrates a feature selection technique and clustering with statistical inference, to improve medical diagnosis by finding out new dependencies between parameters. We consider using the new feature selection technique based on reversed correlations (RCA), combining it with two clustering algorithms: EM and k-means. We compare the RCA technique with two other feature selection methods: CFS and ReliefF. The comparison has been done by experiments carried out on real patient datasets. The experimental results are evaluated by a number of statistically significant correlations detected in clusters.

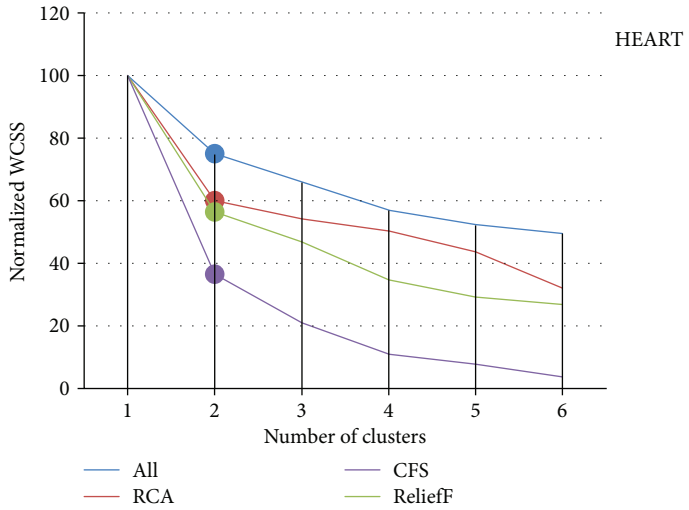


Figure 2: Validation of clustering for the HEART dataset.

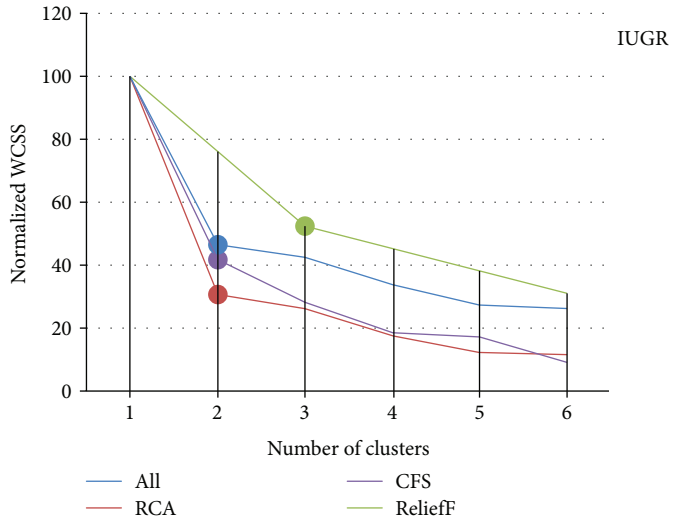


Figure 3: Validation of clustering for the IUGR dataset.

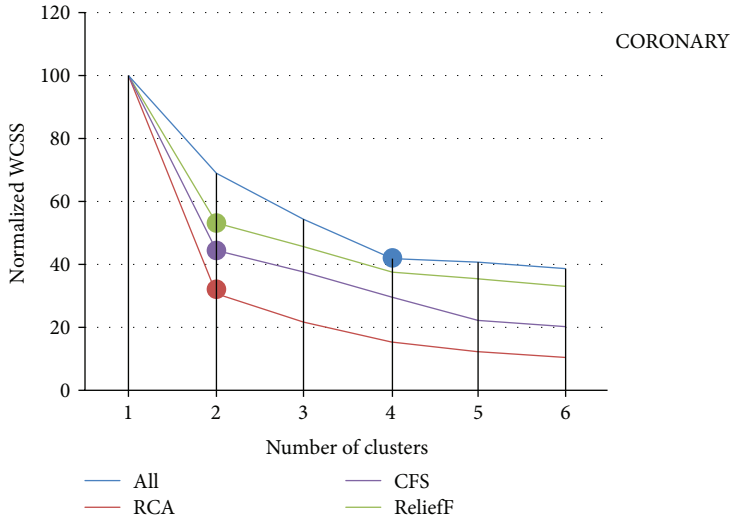


Figure 4: Validation of clustering for the CORONARY dataset.

Table 6: Clustering results.

Dataset (1)	FS algorithm (2)	Cluster algorithm (3)	No of clusters (4)	Clustering schema (5)			
HEART	Main attributes	EM	2	7	23		
		k-Means	2	8	22		
	RCA	EM	2	6	24		
		k-Means	2	6	24		
	CFS	EM	1	30			
		k-Means	2	11	19		
	ReliefF	EM	4	6	4	15	3
		EM	2	21	9		
		k-Means	2	6	24		
IUGR	Main attributes	EM	2	22	25		
		k-Means	2	22	25		
	RCA	EM	2	25	22		
		k-Means	2	25	22		
	CFS	EM	2	12	35		
		k-Means	2	16	31		
	ReliefF	EM	4	7	12	18	10
		k-Means	3	13	14	20	
CORONARY	Main attributes	EM	4	22	49	1	231
		k-Means	4	148	50	71	34
	RCA	EM	2	71	232		
		k-Means	2	232	71		
	CFS	EM	3	101	17	185	
		k-Means	2	213	90		
	ReliefF	EM	3	89	23	191	
		k-Means	2	213	90		

Table 7: Numbers of statistically significant correlations detected in the whole datasets and in clusters.

Dataset	Whole dataset	Main features		RCA		CFS		ReliefF	
		EM	*k*-Means	EM	*k*-Means	EM	*k*-Means	EM	*k*-Means
HEART	14	29	30	28	28	14	28	28	28
IUGR	11	15	15	16	16	11	11	16	15
CORONARY	14	15	20	16	16	15	16	16	16

The experiments have shown that the proposed hybrid approach provides significant benefits. The statistical inference performed in clusters enabled detection of new relationships, which have not been discovered in the whole datasets, regardless of the applied feature selection algorithm and the clustering technique. Moreover, the proposed RCA technique attained results at least as good as other considered feature selection methods, but as opposed to CFS and ReliefF, it belongs to unsupervised approaches, which implies a more flexible application. It is also worth emphasising that the presented approach has been checked using datasets of different mutual proportions between the number of instances and the number of attributes. The experimental results have shown that the proposed methodology performs well on datasets with the small number of instances and what is more, the biggest growth of the number of correlations concerns the dataset where the number of instances is smaller than the number of attributes. Such situations very often take place in the case of patient datasets.

Future research will focus on further investigations that aim at improving medical diagnostics by using hybrid approaches combining data mining and statistical inference. First, more datasets should be examined regarding different mutual proportions between the number of instances and the number of attributes. The research area should be broadened to diagnostics for the diseases of other types. Further research should also include indicating the effective integration of feature selection and clustering algorithms that will perform well combined with statistical inference.

Data Availability

The dataset "CORONARY" that supports the findings of this study is openly available at the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml. The datasets "HEART" and "IUGR" are not publicly available due to ethical restrictions. The full medical description of the data can be found in [13, 23].

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors received funding from the Institute of Information Technology, Lodz University of Technology.

References

[1] I. Yoo, P. Alafaireet, M. Marinov et al., "Data mining in healthcare and biomedicine: a survey of the literature," *Journal of Medical Systems*, vol. 36, no. 4, pp. 2431–2448, 2012.

[2] S. U. Amin, K. Agarwal, and R. Beg, "Data mining in clinical decision support systems for diagnosis," *Prediction and Treatment of Heart Disease, Int J Adv Res Comput Eng Technol (IJARCET)*, vol. 2, no. 1, pp. 218–223, 2008.

[3] P. Haldar, I. D. Pavord, D. E. Shaw et al., "Cluster analysis and clinical asthma phenotypes," *American Journal of Respiratory and Critical Care Medicine*, vol. 178, no. 3, pp. 218–224, 2008.

[4] X. Zhang, X. Zhou, R. Zhang, B. Liu, and Q. Xie, "Real-world clinical data mining on TCM clinical diagnosis and treatment: a survey," in *2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 88–93, Beijing, China, October 2012.

[5] A. Poliński, J. Kot, and A. Meresta, "Analysis of correlation between heart rate and blood pressure," in *IEEE Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 417–420, Szczecin, Poland, 2011.

[6] A. Wosiak and D. Zakrzewska, "On integrating clustering and statistical analysis for supporting cardiovascular disease diagnosis," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, IEEE 2015, Annals of Computer Science and Information Systems*, vol. 5, pp. 303–310, Lodz, Poland, 2015.

[7] A. Wosiak and D. Zakrzewska, "Unsupervised feature selection using reversed correlation for improved medical diagnosis," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, P. Jędrzejowicz, T. Yildirim, and P. Czarnowski, Eds., pp. 18–22, IEEE, Gdynia Poland, 2017.

[8] M. Lichman, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml.

[9] E. Claes, J. M. Atienza, G. V. Guinea et al., "Mechanical properties of human coronary arteries," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 3792–3795, Buenos Aires, Argentina, August-September 2010.

[10] E. D. Grech, "Pathophysiology and investigation of coronary artery disease," *BMJ*, vol. 326, no. 7397, pp. 1027–1030, 2003.

[11] C. J. Murray and A. D. Lopez, *The Global Burden of Disease: A Comprehensive Assessment of Mortality and Disability from Diseases, Injuries, and Risk Factors in 1990 and Projected to 2020: Summary*, Global Burden of Disease And Injury Series, World Health Organization, 1996.

[12] K. Niewiadomska-Jarosik, J. Zamojska, A. Zamecznik, A. Wosiak, P. Jarosik, and J. Stańczyk, "Myocardial dysfunction in children with intrauterine growth restriction: an echocardiographic study," *Cardiovascular Journal of Africa*, vol. 28, no. 1, pp. 36–39, 2017.

[13] A. Zamecznik, K. Niewiadomska-Jarosik, A. Wosiak, J. Zamojska, J. Moll, and J. Stańczyk, "Intra-uterine growth restriction as a risk factor for hypertension in children six to 10 years old: cardiovascular topic," *Cardiovascular Journal Of Africa*, vol. 25, no. 2, pp. 73–77, 2014.

[14] A. M. Hall, "Correlation-based feature selection for machine learning," Doctoral Disertation, University Of Waikato, Department of Computer Science, 1999.

[15] K. Kira and L. A. Rendell, "A practical approach to feature selection," *Machine Learning Proceedings*, pp. 249–256, 1992.

[16] I. KononenkoF. Bergadano and L. De Raedt, "Estimating attributes: analysis and extensions of RELIEF," in *European Conference on Machine Learning: ECML 1994: Machine Learning: ECML-94*, vol. 784 of Lecture Notes in Computer Science, pp. 171–182, Springer, Berlin, Heidelberg, 1994.

[17] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Elsevier, USA, 2011.

[18] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[19] S. W. Looney and J. L. Hagan, "Statistical methods for assessing biomarkers and analyzing biomarker data," in *Essential Statistical Methods for Medical Statistics*, C. R. Rao, J. P. Miller, and D. C. Rao, Eds., pp. 27–65, Elsevier, 2011.

[20] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, USA, 2011.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[22] Y. F. Wang, M. Y. Chang, R. D. Chiang, L. J. Hwang, C. M. Lee, and Y. H. Wang, "Mining medical data: a case study of endometriosis," *Journal of Medical Systems*, vol. 37, no. 2, p. 9899, 2013.

[23] J. Zamojska, K. Niewiadomska-Jarosik, A. Wosiak, P. Lipiec, and J. Stańczyk, *Myocardial Dysfunction Measured by Tissue Doppler Echocardiography in Children with Primary Arterial Hypertension*, Kardiologia Polska, 2015.

[24] R. Alizadehsani, J. Habibi, M. J. Hosseini et al., "A data mining approach for diagnosis of coronary artery disease," *Computer Methods and Programs in Biomedicine*, vol. 111, no. 1, pp. 52–61, 2013.

[25] R. Alizadehsani, M. H. Zangooei, M. J. Hosseini et al., "Coronary artery disease detection using computational intelligence methods," *Knowledge-Based Systems*, vol. 109, pp. 187–197, 2016.

[26] Z. Arabasadi, R. Alizadehsani, M. Roshanzamir, H. Moosaei, and A. A. Yarifard, "Computer aided decision making for heart disease detection using hybrid neural network-genetic

algorithm," *Computer Methods and Programs in Biomedicine*, vol. 141, pp. 19–26, 2017.

[27] D. G. Altman and J. M. Bland, "Measurement in medicine: the analysis of method comparison studies," *The Statistician*, vol. 32, no. 3, pp. 307–317, 1983.

[28] D. E. Hinkle, W. Wiersma, and S. G. Jurs, *Applied Statistics for the Behavioral Sciences*, Houghton Mifflin, Boston, 5th Ed edition, 2003.

Hindawi
Complexity
Volume 2018, Article ID 7130146, 10 pages
https://doi.org/10.1155/2018/7130146

Research Article

Deep Learning- and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification

Zeynep H. Kilimci[1] **and Selim Akyokus**[2]

[1]*Computer Engineering Department, Dogus University, Istanbul 34722, Turkey*
[2]*Computer Engineering Department, İstanbul Medipol University, Istanbul 34722, Turkey*

Correspondence should be addressed to Zeynep H. Kilimci; hkilimci@dogus.edu.tr

Received 17 April 2018; Accepted 25 September 2018; Published 9 October 2018

Guest Editor: Ireneusz Czarnowski

The use of ensemble learning, deep learning, and effective document representation methods is currently some of the most common trends to improve the overall accuracy of a text classification/categorization system. Ensemble learning is an approach to raise the overall accuracy of a classification system by utilizing multiple classifiers. Deep learning-based methods provide better results in many applications when compared with the other conventional machine learning algorithms. Word embeddings enable representation of words learned from a corpus as vectors that provide a mapping of words with similar meaning to have similar representation. In this study, we use different document representations with the benefit of word embeddings and an ensemble of base classifiers for text classification. The ensemble of base classifiers includes traditional machine learning algorithms such as naïve Bayes, support vector machine, and random forest and a deep learning-based conventional network classifier. We analysed the classification accuracy of different document representations by employing an ensemble of classifiers on eight different datasets. Experimental results demonstrate that the usage of heterogeneous ensembles together with deep learning methods and word embeddings enhances the classification performance of texts.

1. Introduction

Recently, text classification/categorization has gained remarkable attention of many researchers due to the huge number of documents and text available on the different digital platforms. Given a text or document, the main objective of text classification is to classify the given text into a set of predefined categories by using supervised learning algorithms. The supervised learning algorithms can be trained to generate a model of relationship between features and categories from samples of a dataset. Using the trained model, the learning algorithm can predict the category of a given document.

A text classification task consists of parsing of documents, tokenization, stemming, stop-word removal, and representation of documents in document-term matrix with different weighting methods, feature selection, and selection of the best classifiers by training and testing [1]. Different methods are used for each of the subtasks given above. Each document is usually represented in vector space model (also called bag of words model). In vector space model, each document is represented as a vector that includes a set of terms (words) that appears in the document. The set of documents with their vector representation forms the document-term matrix. The significance of each term in a document is computed by utilizing different term weighting methods. Common term weighting methods include Boolean, term frequency, and TF-IDF weighting schemes. In addition, there has been a recent interest to employ word embeddings to represent documents. There are many types of supervised classifiers used in text categorization. A review and comparison of supervised algorithms are presented in papers [1, 2]. Some of the commonly used supervised algorithms

include naïve Bayes (NB), k-nearest neighbours (k-NN), decision trees (DT), artificial neural networks (ANN), and support vector machines (SVM). Deep learning networks have also been attracting attention of researchers in text classification due to their high performance with less need of engineered features.

Another trend in machine learning is to increase the classification performance by using an ensemble of classifiers. In an ensemble system, a group of base classifiers is employed. If different types of classifiers are used as base learners, then such a system is called heterogeneous ensemble, otherwise homogenous ensemble. In this study, we focus on heterogeneous ensembles. An ensemble system is composed of two parts: ensemble generation and ensemble integration [3–7]. In ensemble generation part, a diverse set of models is generated using different base classifiers. Naïve Bayes, support vector machine, random forest, and convolutional neural network learning algorithms are used as base classifiers in this study. There are many integration methods that combine decisions of base classifiers to obtain a final decision [8–11]. For ensemble integration, we used majority voting and stacking methods.

In a previous paper [12], we presented an ensemble of heterogeneous classifiers to enhance the text classification performance. In this study, we try to expand our work by using deep learning methods, which have produced state-of-the-art results in many domains including natural language processing (NLP) and text classification [13]. One of the well-known deep learning-related methods is word embeddings. Word embeddings provide a vector representation of words learned from a corpus. It maps words into a vector of real numbers. While words with similar meaning are mapped into similar vectors, a more efficient representation of words with a much lower dimensional space is obtained when compared with simple bag-of-words approach. Convolutional neural network model (CNN) is another deep learning method employed in this study. CNN is added into our set of base classifiers in order to improve accuracy of the ensemble of heterogeneous classifiers. In addition, we use different document representation methods including TF-IDF weighted document-term matrix, mean of word embeddings, and TF-IDF weighted document matrix enhanced with addition of mean vectors of word embeddings as features. We have performed experiments on eight different datasets in which four of them are in Turkish. In summary, this paper utilizes and analyses an ensemble of classifiers including CNN model with word embeddings and different document representation methods to enhance the performance of text classification.

We have evaluated and discussed the performance of an ensemble of five heterogeneous base learners and two integration methods using three different document representation methods on eight different datasets on this paper. This paper is organized as follows. Section 2 gives related research on text categorization using ensemble systems, word embeddings, and CNN. Section 3 presents base learners and ensemble fusion methods used in experimental studies. Experimental setup and results are given in Section 4. Section 5 summarizes and discusses results and outlines future research directions.

2. Related Work

This section gives a brief summary of ensemble systems, pretrained word embeddings and deep neural networks related to text classification/categorization problem. High dimensionality of input feature space, sparsity of document vectors, and the presence of few irrelevant features are the main characteristics of text categorization problem that differs from other classification problems [14].

The study of Larkey and Croft [15] is one of the early works of applying ensemble systems to text categorization. They used an ensemble of three classifiers, k-nearest neighbour, relevance feedback, and Bayes classifiers to categorize medical documents. Dong and Han [16] used three different variants of naive Bayes and SVM classifiers. They compared the performance of six different homogenous ensembles and a heterogeneous ensemble classifier. Fung et al. [17] use a heterogeneous ensemble classifier that uses a dynamic weighting function to combine decisions. A pairwise ensemble approach is presented in [18] that achieves better performance than popular ensemble approaches bagging and ECOC. Keretna et al. [19] have worked on named entity recognition problem using an ensemble system. In another study done by Gangeh et al. [20], random subspace method is applied to text categorization problem. The paper emphases the estimation of ensemble parameters of size and the dimensionality of each random subspace submitted to the base classifiers. Boroš et al. [21] applied ensemble methods to multiclass text documents where each document can belong to more than one category. They evaluated the performance of ensemble techniques by using multilabel learning algorithms. Elghazel et al. [22] propose a novel ensemble multilabel text categorization algorithm, called multilabel rotation forest (MLRF), based on a combination of rotation forest and latent semantic indexing. Sentiment analysis with ensembles is currently a popular topic among researchers. For Twitter sentiment analysis, an ensemble classifier is proposed in [23] where the dataset includes very short texts. A combination of several polarity classifiers provides an improvement of the base classifiers. In a recent study, the predictive performance of ensemble learning methods on Twitter text documents that are represented by keywords is evaluated by Onan et al. [24] empirically. The five different ensemble methods that use four different base classifiers are applied on the documents represented by keywords.

Representation of text documents with the benefit of word embeddings is another current trend in text classification and natural language processing (NLP). Text representation with word embeddings has been successful in many of NLP applications [13]. Word embeddings are low-dimensional and dense vector representation of words. Word embeddings can be learned from a corpus and can be reused among different applications. Although it is possible to generate your own word embeddings from a dataset, many investigators prefer to use pretrained word embeddings generated from a large corpus. The generation of word embeddings requires a large amount of computational power, preprocessing, and training time [25]. The most commonly used pretrained word embeddings include Word2Vec

[26, 27], GloVe [28], and fastText [29]. In this study, we use pretrained Word2vec embeddings in English and Turkish.

Supervised deep learning networks can model high-level abstractions and provide better classification accuracies compared with other supervised traditional machine algorithms. For this reason, supervised deep learning architectures have received significant attention in NLP and text classification recently. Kalchbrenner et al. [30] described a dynamic convolutional network that uses dynamic k-max pooling, a global pooling operation over linear sequences for sentiment classification of movie reviews and Twitter. Kim [31] reports a series of experiments with four different convolutional neural networks for sentence-level sentiment classification tasks using pretrained word vectors. Kim achieves good performance results on different datasets and suggests that the pretrained word vectors are universal and can be used for various classification tasks. In [32], a new deep convolutional neural network is proposed to utilize from character-level to sentence-level information to implement sentiment classification of short texts. In [33, 34], text classification is realized with a convolutional neural network that accepts a sequence of encoded characters as input rather than words. It is shown that character-level coding is an effective method. Johnson and Zhang [35] propose a semisupervised convolutional network for text classification that learns embeddings of small text regions. Joulin et al. [29] proposes a simple and efficient baseline classifier that performs as well as deep learning classifiers in terms of accuracy and runs faster. Conneau et al. [36] present a new architecture called very deep (VDCNN) for text processing which operates directly at the character level and uses only small convolutions and pooling operations. Kowsari et al. [37] introduce a new approach to hierarchical document classification, called HDLTex that employs multiple deep learning approaches to produce hierarchical classifications.

3. Ensemble Learning, Word Embeddings, and Representations

This section gives a summary of learning algorithms, ensemble integration techniques, word embeddings, and document representation methods used in this study.

3.1. Base Learners. In this study, we employ multivariate Bernoulli naïve Bayes (MVNB), multinomial naïve Bayes (MNB), support vector machine (SVM), random forest (RF), and convolutional neural network (CNN) learning algorithms as base classifiers to generate a heterogeneous ensemble system.

3.1.1. MVNB and MNB. As a simple probabilistic classifier, naïve Bayes is based on Bayes' theorem with the assumption of independence of features from each other. There are two types of naïve Bayes classifier frequently used for text categorization: multivariate Bernoulli naïve Bayes (MVNB) and multinomial naïve Bayes (MNB). In MVNB, each document is represented by a vector with binary variables that can take values 1 or 0 depending upon the presence of a word in the document. In MNB, each document vector is represented by the frequency of words that appear in the document. Equation (1) defines MVNB classifier with Laplace smoothing. The occurrence of the term t in document i is indicated by B_{it} which can be either 1 or 0. $|D|$ indicates the number of labelled training documents. $P(c_j \mid d_i)$ is 1 if document i is in class j. The probability of term w_t in class c_j is as follows [38]:

$$P(w_t|c_j) = \frac{1 + \sum_{i=1}^{D} B_{it} P(c_j|d_i)}{2 + \sum_{i=1}^{D} P(c_j|d_i)}. \quad (1)$$

3.1.2. SVM. Support vector machine (SVM) is binary classifier that divides an n-dimensional space with n features into two regions related with two classes [39]. The n-dimensional hyperplane separates two regions in a way that the hyperplane has the largest distance from training vectors of two classes called support vectors. SVM can also be used for a nonlinear classification using a method called the kernel trick that implicitly maps input instances into high-dimensional feature spaces that can be separated linearly. In SVM, the use of different kernel functions enables the construction of a set of diverse classifiers with different decision boundaries.

3.1.3. RF. Random forests (RF) are a collection of decision tree classifiers introduced by Breiman [40]. It is a particular implementation of bagging in which decision trees are used as base classifiers. Given a standard training set, the bagging method generates a new training set by sampling with replacement for each of base classifiers. In standard decision trees, each node of the tree is split by the best feature among all other features using splitting criteria. To provide randomness at feature space, random forest algorithm first selects a random subset of features and then decides on the best split among the randomly selected subset of features. Random forests are strong against overfitting because of randomness applied in both sample and feature spaces.

3.1.4. CNN. Convolutional neural networks (CNNs) are a class of deep learning networks that have achieved remarkable results in image recognition and classification [41–43]. CNN models have subsequently been shown to be effective for natural language processing tasks [13]. CNN is a feed-forward network with an input and output layer and hidden layers. Hidden layers consist of convolutional layers interleaved with pooling layers. The most important block of CNN is the convolutional layer. Convolutional layer applies a convolution filter to input data to produce a feature map to combine information with data on the filter. Multiple filters are applied to input data to get a stack of feature maps that becomes the final output of the convolutional layer. The values of filters are learned during the training process. Convolution operation captures local dependencies or semantics in the regions of original data. An additional activation function (usually RELU (rectified linear unit)) is applied to feature maps to add nonlinearity to CNN. After convolution, a pooling layer reduces the number of samples in each feature map and retains the most important

information. The pooling layer shortens the training time and reduces the dimensionality of data and overfitting. Max pooling is the most common type of pooling function that takes the largest value in a specified neighbourhood window. CNN architectures consist of a series of convolutional layers interleaved with pooling layers, followed by a number of fully connected layers.

CNN is originally applied on image processing and recognition tasks where input data is in a two-dimensional (2D) structure. On image processing, CNN exploits spatial local correlation of 2D images, learns, and responds to small regions of 2D images. In natural language processing and text classification, words in a text or document are converted into a vector of numbers that has one-dimensional (1D) structure. On text processing applications, CNN can exploit the 1D structure of data (word order) by learning small text regions of data in addition to learning from a bag of independent word vectors that represents an entire document [44].

3.2. Ensemble Integration Strategies. To combine decisions of individual base leaners MVNB, MNB, SVM, RF, and CNN, majority voting and stacking methods are used in this study. In majority voting method, an unlabelled instance is classified according to the class that obtains the highest number of votes from collection of base classifiers. In stacking method, also called stacked generalization, a metalevel classifier is used to combine the decision of base-level classifiers [45]. The stacking method consists of two steps. In the first step, a set of base-level classifiers $C_1, C_2, \ldots, C_n$ is generated from a sample training set S that consists of feature examples $s_i = (\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ is feature vector and y_i is prediction (class label). A meta-dataset is constructed from the decisions of base-level classifiers. The meta-dataset contains an instance for predictions of classifiers in the original training dataset. The meta-dataset is in the form of $m_i = (d_i, y_i)$ where d_i is the prediction of individual n base classifiers. The meta-dataset can also include both original training examples and decisions of base-level classifiers in the form of $m_i = (\mathbf{x}_i, d_i, y_i)$ to improve performance. After the generation of meta-dataset, a metalevel classifier is trained with meta-dataset and used to make predictions. In our study, the meta-dataset includes both original training examples and decisions of base-level classifiers.

3.3. Word Embeddings with Word2vec. Word embeddings are an active research area that tries to discover better word representations of words in a document collection (corpus). The idea behind all of the word embeddings is to capture as much contextual, semantical, and syntactical information as possible from documents from a corpus. Word embeddings are a distributed representation of words where each word is represented as real-valued vectors in a predefined vector space. Distributed representation is based on the notion of distributional hypothesis in which words with similar meaning occur in similar contexts or textual vicinity. Distributed vector representation has proven to be useful in many natural language processing applications such as named entity recognition, word sense disambiguation, machine translation, and parsing [13].

Currently, Word2vec is the most popular word embedding technique proposed by Mikolov et al. [26, 27]. The Word2vec method learns vector representations of words from a training corpus by using neural networks. It maps the words that have similar meaning into vectors that will be close to each other in the embedded vector space. Word2vec offers a combination of two methods: CBOW (continuous bag of words) and skip-gram model. While the CBOW model predicts a word in a given context, the Skip-gram model predicts the context of a given word. Word2vec extracts continuous vector representations of words from usually very large datasets. While it is possible to generate your own vector representation model from a given corpus, many studies prefers to use pretrained models because of high computational power and training time required for large corpuses. The pretrained models have been found useful in many NLP applications.

3.4. Document Representation Methods. The effective representations of documents in a document collection have a significant role in the success performance of text processing applications. In many text classification applications, each document in a corpus is represented as a vector of real numbers. Elements of a vector usually correspond to terms (words) appearing in a document. The set of documents with their vector representation forms document-term matrix. The significance of each term in a document is computed by using different term weighting methods. Traditional term weighting methods include Boolean, term frequency, and TF-IDF weighting schemes. TF-IDF is the common weighting method used for text processing. In this representation, the term frequency for each word is multiplied by the inverse document frequency (IDF). This reduces the importance of common terms in the collection and also increases the influence of rare words which have relatively low frequencies. As explained above, word embedding is also another effective method to represent documents as a vector of numbers.

In this study, we use and compare the following document representation methods:

(i) TF-IDF. A document vector consists of words appearing in a document weighted with TF-IDF scheme

(ii) Avg-Word2vec. A document vector is obtained by taking average of all vectors of word embeddings appearing in a document by using pretrained models

(iii) TF-IDF + Avg-Word2vec. A document vector includes both TF-IDF and Avg-Word2vec vectors

4. Experiments

In this study, we have evaluated the performance of an ensemble of eight heterogeneous base learners with two integration methods using three different document representation methods on eight different datasets.

4.1. Datasets. We use eight different datasets with different sizes and properties to explore the classification performances of the heterogeneous classifier ensembles in Turkish

and English. Turkish is an agglutinative language in which words are composed of a sequence of morphemes (meaningful word elements). A single Turkish word can correspond to a sentence that can be expressed with several words in other languages. Turkish datasets include news articles from Milliyet, Hurriyet, 1150haber, and Aahaber datasets. English datasets consist of 20News-19997, 20News-18828, Mini-news, and WebKB4. The properties of each dataset are explained below.

Milliyet dataset includes text from columns of Turkish newspaper Milliyet from years 2002 to 2011. It contains 9 categories and 1000 documents for each category. The categories of this dataset are café (cafe), dünya (world), ege (region), ekonomi (economy), güncel (current), siyaset (politics), spor (sports), Türkiye (Turkey), and yaşam (life). *Hurriyet* dataset includes news from 2010 to 2011 on Turkish newspaper Hurriyet. It contains six categories and 1000 documents for each category. Categories in this dataset are dünya (world), ekonomi (economy), güncel (current), spor (sports), siyaset (politics), and yaşam (life). *1150haber* dataset is obtained from a study done by Amasyalı and Beken [46]. It consists of 1150 Turkish news texts in five classes (economy, magazine, health, politics, and sports) and 230 documents for each category. *Aahaber*, collected by Tantug [47], is a dataset that consists of newspaper articles broadcasted by Turkish National News Agency, Anadolu Agency. This dataset includes eight categories and 2500 documents for each category. Categories are Turkey, world, politics, economics, sports, education science, "culture and art", and "environment and health".

Milliyet and 1150haber include the writings of the column writers; therefore, they are longer and more formal. On the other hand, Hurriyet and Aahaber datasets contain traditional news articles. They are more irregular, much shorter than documents of the other datasets.

The 20 newsgroup is a popular English dataset used in many text classification and clustering experiments [48]. The 20 newsgroup dataset is a collection of about 20,000 documents extracted from 20 different newsgroups. The data is almost evenly partitioned into 20 categories. We use three versions of newsgroup dataset. The first one is the original 20 newsgroup dataset that includes 19,997 documents. It is called as *20News-19997*. The second one is named *20News-18828* with 18,828 documents, and it covers less number of documents than the original dataset. This dataset includes messages with only "From" and "Subject" headers with the removal of cross-post duplicate messages. The third one is a small subset of the original dataset composed of 100 postings per class, and it is called as *mini-newsgroups*. The last dataset is called *WebKB* [49] which includes web pages gathered from computer science departments of different universities. These web pages are composed of seven categories (student, faculty, staff, course, project, department, and other) and contain approximately 8300 pages. Another version of WebKB is called *WebKB4* where the number of categories is reduced to four. We use WebKB4 in our experiments.

Characteristics of the datasets without application of any preprocessing procedures are given in Table 1 where $|C|$ is the number of classes, $|D|$ is the number of documents, and $|V|$ is the vocabulary size. We only filter infrequent terms whose document frequency is less than three. We do not apply any stemming or stop-word filtering in order to avoid any bias that can be introduced by stemming algorithms or stop-word lists.

Table 1: Number of classes ($|C|$), documents ($|D|$), and words ($|V|$) in each document.

Dataset	$\|C\|$	$\|D\|$	$\|V\|$
20News-18828	20	18,828	50,570
20News-19997	20	19,997	43,553
Mini-news	20	2000	13,943
WebKB4	4	4199	16,116
1150haber	5	1150	11,040
Milliyet	9	9000	63,371
Hurriyet	8	6000	18,280
Aahaber	8	2000	14,396

4.2. Experiment Results. We use repeated holdout method in our experiments. We randomly divide a dataset into two halves where 80% of data is used for training and 20% for testing. To get a reliable estimation, we repeat the holdout process 10 times and an overall accuracy is computed by taking averages of each iteration. Classification accuracies and computation times of algorithms on different datasets are presented below.

As a first step, we calculate accuracies of individual classifiers to compare and observe results that we obtain by using an ensemble of classifiers with different representation methods. As explained before, base classifiers include MVNB, MNB, SVM, RF, and CNN. Table 2 lists the accuracies of these classifiers on eight different Turkish and English datasets. The best accuracy is obtained for each dataset and is shown in boldface. As it can be seen from Table 2, there is no best algorithm that performs well on each dataset like in many machine language problems. It seems that RF and CNN generally produce better accuracies from the rest of the algorithms. The random forest (RF) is the best algorithm in our experiments. This might be due to ensemble strategy applied in RF algorithm that uses a set of decision trees for classification. CNN also performs well. That might be because of the use of different convolutional filters that work like an ensemble system that extracts different features from datasets. The order of average classification accuracies of single classifiers can be summarized as follows: RF > CNN > MNB > SVM > MVNB.

To construct a heterogeneous ensemble system, we use MVNB, MNB, SVM, RF, and CNN algorithms as base classifiers. The decisions of each of these base classifiers are combined with majority voting (MV) and stacking (STCK) integration methods as described in Section 3.2. Each dataset is represented with traditional TF-IDF weighting scheme. Table 3 demonstrates the classification accuracies of heterogeneous ensemble systems with majority voting (Heter-MV) and stacking (Heter-STCK) together

Table 2: Classification accuracies of single classifiers on datasets represented with TF-IDF weighting scheme.

Dataset	MVNB	MNB	SVM	RF	CNN
20News-18828	75.89	91.49	**91.50**	90.36	89.23
20News-19997	63.91	**81.29**	63.14	78.56	75.56
Mini-news	77.78	82.93	**92.40**	90.44	91.7
WebKB4	79.30	86.74	91.15	89.75	**91.56**
1150haber	85.17	**95.30**	92.52	95.07	94.37
Milliyet	83.19	83.91	92.75	93.05	78.06
Hurriyet	77.14	82.29	81.44	85.11	**87.31**
Aahaber	82.06	83.26	80.89	88.26	**90.19**
Average	78.06	85.90	85.72	**88.83**	87.25
Computation time	1 h 13 m	1 h 45 m	2 h 17 m	3 h 34 m	5 h 23 m

Table 3: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with TF-IDF weighting.

Methods	20News-18828	20News-19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	75.89	63.91	77.78	79.30	85.17	83.19	77.14	82.06	1 h 13 m
MNB	91.49	81.29	82.93	86.74	95.30	83.91	82.29	83.26	1 h 45 m
SVM	91.50	63.14	92.40	91.15	92.52	92.75	81.44	80.89	2 h 17 m
RF	90.36	78.56	90.44	89.75	95.07	93.05	85.11	88.26	3 h 34 m
CNN	89.23	75.56	91.70	91.56	94.37	78.06	87.31	90.19	5 h 23 m
Heter-MV	92.39	79.63	92.60	90.85	94.98	92.34	88.41	89.94	7 h 55 m
Heter-Stck	93.25	81.29	94.07	93.57	96.77	94.09	89.55	92.70	8 h 31 m

with each of the single classifiers. As it can be seen from Table 3, an ensemble system with stacking integration strategy always performs better than single classifiers and the ensemble model with majority integration strategy. On our previous study [50], we obtain classification accuracies 85.44 and 87.73 for Turkish Hurriyet and Aahaber datasets, respectively, represented with TF (term frequency) weighting by using an ensemble of classifiers without CNN. With the inclusion of CNN to our set of base learners, improved accuracies are obtained with 89.55 and 92.70 for Turkish Hurriyet and Aahaber datasets, respectively.

In text mining applications, different words appearing in a document collection form feature set where the number of features is usually expressed in thousands. High dimensionality of feature space is a problem in text classification when documents are represented with "bag of words" model. Word2vec can be used as a feature extraction technique to reduce the number of features. The average of Word2vec vectors of words is employed to represent documents. Given a document d represented with n words $d = w_1, w_2, \ldots, w_n$, words appearing in a document are represented with Word2vec embedding vectors $e_{w_1}, e_{w_2}, \ldots, e_{w_n}$ by looking up the vector representation of a word from a pretrained embedding model.

Each document is represented by taking average of word embeddings as follows:

$$e_d = \frac{1}{n}\sum_{i=1}^{n} e_{w_i}. \quad (2)$$

Google has used Google News dataset that contains about 100 billion words to obtain pretrained vectors with the Word2vec skip-gram algorithm [26, 51]. The pretrained model includes word vectors for about 3 million words and phrases. Each vector has 300 dimensions or features. A pretrained Turkish Word2vec model is constructed with all Wikipedia articles written in Turkish [52]. We use these pretrained models in English and Turkish to represent documents with 300 dimensions or features.

Table 4 shows classification accuracies of documents represented by average of Word2vec vectors, called as Avg-Word2vec, in this study. Classification accuracies of single classifiers and heterogeneous ensemble systems are given in Table 4. Ensemble method with stacking integration strategy (Heter-Stck) produces better outcomes than the ensemble method with majority voting integration strategy (Heter-MV). We observe that there is a slight decrease in the classification accuracies of documents represented with Avg-Word2vec on the six datasets (20News-18828, 20News-19997, Mini-news, 1150-haber, Milliyet, and Hurriyet) when compared with the classification accuracies (Heter-Stck) of documents represented with TF-IDF given in Table 3. On the contrary, there is a slight improvement on two datasets (WebKB4, Aahaber). Avg-Word2vec-based classification performance of the system is competitive although a much less number of features are employed.

In the last experiment, each document is represented by concatenation of TF-IDF and average of Word2vec vectors. Although the size of the feature set is increased, we observe better classification accuracies than the other representation

Table 4: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with the average of Word2vec vectors.

Methods	20News 18828	20News 19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	74.23	61.01	76.18	79.77	87.27	81.12	77.58	83.11	56 m
MNB	89.50	80.44	81.67	87.05	96.13	81.55	82.39	84.36	1 h 17 m
SVM	88.51	62.78	90.35	91.71	93.89	89.27	82.00	82.23	1 h 50 m
RF	87.34	76.65	89.27	90.06	97.56	90.30	84.45	89.78	2 h 44 m
CNN	90.17	76.91	92.55	92.22	95.14	76.10	88.07	93.26	4 h 55 m
Heter-MV	89.62	78.20	91.45	93.45	95.81	90.93	87.18	92.78	6 h 30 m
Heter-Stck	90.85	80.94	93.78	94.56	96.34	92.05	88.96	93.80	7 h 23 m

Table 5: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with TF-IDF and the average of Word2vec vectors.

Methods	20News-18828	20News-19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	75.83	63.78	77.00	80.21	87.97	83.70	78.80	83.71	1 h 57 m
MNB	90.27	81.74	81.98	87.80	96.78	84.42	83.09	84.80	2 h 24 m
SVM	91.51	64.08	92.35	92.05	94.20	91.36	83.34	81.56	2 h 55 m
RF	91.25	79.15	91.47	90.92	97.99	92.10	85.92	90.38	4 h 5 m
CNN	91.88	77.80	92.12	93.01	95.59	79.49	89.27	94.15	5 h 48 m
Heter-MV	92.90	79.90	92.68	93.75	96.00	92.13	88.53	93.10	8 h 32 m
Heter-Stck	94.30	82.23	95.06	95.08	97.40	93.90	90.02	94.86	9 h 20 m

methods. Table 5 gives the classification accuracies of ensemble systems together with the base classifiers. Heterogeneous ensemble with stacking integration strategy (Heter-Stck) presents the best classification accuracies among the other experiments. When compared with the classification accuracies of documents represented with TF-IDF given in Table 2, we obtain better classification accuracies on seven of the datasets except Milliyet in which accuracy is decreased from 94.09 to 93.9 (0.2%). It can be concluded that the inclusion of Avg-Word2vec vectors to TF-IDF vectors as additional features boosts classification success of the ensemble system.

It is difficult to compare the performance of our results with other studies because of the lack of works with similar combinations of different datasets, representation models, and ensemble approaches. Although a comparison of baseline classifiers and heterogeneous ensemble systems with different representation techniques is given in this study, we also report the classification accuracies of a number of research works here. In [53], Pappagari et al. propose an end-to-end multiscale CNN framework for topic identification by employing 20 newsgroups and Fisher datasets. The classification success of our proposed system outperforms with 94.3% while the accuracy performance of the work [53] presents 86.1% classification accuracy.

In another study [54], a graph-based framework is presented by utilizing SVM for text classification problem. Two data representation models (TF-IDF and Word2vec) and datasets (20 newsgroups and WebKB) are studied in this work [54]. 20 newsgroup dataset with TF-IDF representation exhibits 83.0% classification accuracy while the Word2vec version of the same dataset presents 75.8% classification success in [54]. The TF-IDF representation model of our study performs 91.5% and the Word2vec representation of our work achieves 88.5% classification accuracies for 20 newsgroup dataset. Moreover, their study [54] with WebKB dataset and TF-IDF representation exhibits 89.9% classification accuracy while the Word2vec version of the same dataset presents 86.6% classification success. The TF-IDF representation model of our work performs 91.2%, and the Word2vec representation of ours represents 91.7% classification accuracies for WebKB dataset. In [55], Zheng et al. propose a bidirectional hierarchy skip-gram model to mine topic information within given texts. They use CNN and SVM as classification algorithms and 20 newsgroups and WebKB as datasets like in our study. Our proposed system performs 91.9% classification success with CNN algorithm while their CNN implementation exhibits 79.2% for 20 newsgroups in [55]. The proposed system of our study performs 91.5% classification success with SVM algorithm while SVM exhibits 85.9% for 20 newsgroups in [55]. For the WebKB dataset, CNN implementation of our work performs 93.0% classification success while their CNN method exhibits 91.8%. With SVM algorithm on WebKB dataset, we obtain 92.1% classification success while their SVM implementation exhibits 88.1%. The proposed heterogeneous ensemble systems given in this study always perform well compared with other studies reported above in terms of classification accuracies.

5. Conclusion

In this paper, we focus to enhance the overall accuracy of a text classification system by using ensemble learning, deep

learning, and effective document representation methods. It is known that the classifier ensembles boost the overall classification performance by depending on two factors, namely, individual success of the base learners and the diversity of them. The different learning algorithms, namely, variants of two naïve Bayes (MVNB and MNB), support vector machine (SVM), random forest (RF), and currently popular convolutional neural networks (CNNs), are chosen to provide diversity for the ensemble system. The majority voting and stacking ensemble integration strategies are performed to consolidate the final decision of the ensemble system. Word embeddings are also utilized to raise the overall accuracy of text classification. Word embeddings can capture contextual, semantical, and syntactical information in a textual vicinity from documents from a corpus.

A set of experiments is performed on eight different datasets represented with different methods using an ensemble of classifiers MVNB, MNB, SVM, RF, and CNN. As a result of experiments, some of the main findings of this study are as follows:

(i) RF and CNN are the best performing single classifiers among others. The order of classification accuracies of single classifiers can be summarized as RF > CNN > MNB > SVM > MVNB

(ii) An ensemble of classifiers increases the classification accuracies of texts on different datasets that have different characteristics and distributions

(iii) A set of heterogeneous ensemble of classifiers can provide slight performance increases in terms of accuracy when compared with homogenous ensemble of classifiers [12, 50]

(iv) Stacking is a better ensemble integration strategy than majority voting

(v) The inclusion of state-of-art deep learning CNN classifier to the set of classifiers of an ensemble system can provide further enhancement

(vi) The use of pretrained word embeddings is an effective method to represent documents. It can be a good feature reduction method without losing much in terms of classification accuracy

(vii) Inclusion of word embeddings to TF-IDF weighted vectors as additional features provides a further improvement in text classification because word embeddings can capture contextual, semantical, and syntactical information from text

In the future, we plan to use different pretrained word embedding models, document representation methods using word embeddings, and other deep learning algorithms for text classification and natural language processing tasks.

Data Availability

We used publicly available datasets in our experiments. If necessary, we can share those datasets.

Disclosure

This work is prepared after invitation of the paper [12] published in 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[2] C. C. Aggarwal and C. X. ZhaiC. C. Aggarwal and C. X. Zhai, "A survey of text classification algorithms," in *Mining Text Data*, pp. 163–222, Springer, 2012.

[3] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[4] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[5] S. Reid, "A review of heterogeneous ensemble methods," in *Department of Computer Science*, University of Colorado at Boulder, 2007.

[6] D. Gopika and B. Azhagusundari, "An analysis on ensemble methods in classification tasks," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 7, pp. 7423–7427, 2014.

[7] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions [review article]," *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[8] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Technical Review*, vol. 27, no. 4, pp. 293–307, 2010.

[9] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[10] G. Tsoumakas, L. Angelis, and I. Vlahavas, "Selective fusion of heterogeneous classifiers," *Intelligent Data Analysis*, vol. 9, no. 6, pp. 511–525, 2005.

[11] A. Güran, M. Uysal, Y. Ekinci, and B. Güran, "An additive FAHP based sentence score function for text summarization," *Information Technology And Control*, vol. 46, no. 1, 2017.

[12] Z. H. Kilimci, S. Akyokus, and S. İ. Omurca, "The evaluation of heterogeneous classifier ensembles for Turkish texts," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 307–311, Gdynia, 2017.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," 2018, http://arxiv.org/abs/1708.02709v5.

[14] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, C. Nédellec and C. Rouveirol, Eds., vol. 1398, Springer, Berlin, Heidelberg, 1998.

[15] L. S. Larkey and W. Bruce Croft, "Combining classifiers in text categorization," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '96*, pp. 289–297, Zurich, Switzerland, 1996.

[16] Y.-S. Dong and K.-S. Han, "A comparison of several ensemble methods for text categorization," in *IEEE International Conference onServices Computing, 2004. (SCC 2004). Proceedings. 2004*, pp. 419–422, Shanghai, China, 2004.

[17] G. P. C. Fung, Y. Jeffrey Xu, H. Wang, D. W. Cheung, and H. Liu, "A balanced ensemble approach to weighting classifiers for text classification," in *Sixth International Conference on Data Mining (ICDM'06)*, pp. 869–873, Hong Kong, 2006.

[18] Y. Liu, J. Carbonell, and R. Jin, "A new pairwise ensemble approach for text classification," in *Machine Learning: ECML 2003. ECML 2003. Lecture Notes in Computer Science*, pp. 277–288, Springer, Berlin, Heidelberg, 2003.

[19] S. Keretna, C. P. Lim, D. Creighton, and K. B. Shaban, "Classification ensemble to improve medical named entity recognition," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2630–2636, San Diego, CA, USA, October 2014.

[20] M. J. Gangeh, M. S. Kamel, and R. P. W. Duin, "Random subspace method in text categorization," in *2010 20th International Conference on Pattern Recognition*, pp. 2049–2052, Istanbul, Turkey, August 2010.

[21] M. Boroš, Franky, and J. Maršík, "Multi-label text classification via ensemble techniques," *International Journal of Computer and Communication Engineering*, vol. 1, no. 1, pp. 62–65, 2012.

[22] H. Elghazel, A. Aussem, O. Gharroudi, and W. Saadaoui, "Ensemble multi-label text categorization based on rotation forest and latent semantic indexing," *Expert Systems with Applications*, vol. 57, no. 15, pp. 1–11, 2016.

[23] M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of ensemble methods on Twitter sentiment analysis using NLP techniques," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169-170, Anaheim, CA, February 2015.

[24] A. Onan, S. Korukoglu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[25] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Improving the accuracy of pre-trained word embeddings for sentiment analysis," 2017, http://arxiv.org/abs/1711.08609v1.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, http://arxiv.org/abs/1301.3781.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, Advances in Neural Information Processing Systems Conference (NIPS 2013), 2013.

[28] J. Pennington, R. Socher, and C. Manning, "GloVe: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, http://arxiv.org/abs/1612.03651.

[30] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–666, Baltimore, MD, USA, 2014.

[31] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.

[32] C. N. D. Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *The 25th International Conference on Computational Linguistics*, pp. 69–78, Dublin, Ireland, August 2014.

[33] X. Zhang and Y. LeCun, "Text understanding from scratch," 2015, http://arxiv.org/abs/1502.01710.

[34] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, pp. 649–657, Montreal, Canada, 2015.

[35] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in Neural Information Processing Systems*, pp. 919–927, Montreal, Canada, 2015.

[36] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," 2017, http://arxiv.org/abs/1606.01781v2.

[37] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: hierarchical deep learning for text classification," http://arxiv.org/abs/1709.08267v2.

[38] A. McCallum and K. A. Nigam, "Comparison of event models for naive Bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, AAAI Press, Wisconsin, USA, 1998.

[39] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[40] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[44] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, CO, USA, 2015.

[45] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine Learning*, vol. 54, no. 3, pp. 255–273, 2004.

[46] M. F. Amasyalı and A. Beken, "Türkçe kelimelerin anlamsal benzerliklerinin ölçülmesi ve metin sınıflandırmada kullanılması," in *IEEE signal processing and communications applications conference*, Antalya, Turkey, 2009.

[47] A. C. Tantug, "Document categorization with modified statistical language models for agglutinative languages," *International Journal of Computational Intelligence Systems*, vol. 3, no. 5, pp. 632–645, 2010.

[48] https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html.

[49] M. Craven, D. DiPasquo, D. Freitag et al., "Learning to extract symbolic knowledge from the world wide web," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. Menlo Park: American Association for Artificial Intelligence*, pp. 509–516, Menlo Park, CA, USA, 1998.

[50] Z. H. Kilimci, S. Akyokus, and S. I. Omurca, "The effectiveness of homogenous ensemble classifiers for Turkish and English texts," in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–7, Sinaia, Romania, August 2016.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "English pre-trained Word2vec model," https://code.google.com/archive/p/word2vec/.

[52] A. Koksal, "Turkish pre-trained Word2vec model," https://github.com/akoksal/Turkish-Word2Vec.

[53] R. Pappagari, J. Villalba, and N. Dehak, "Joint verificationidentification in end-to-end multi-scale cnn framework for topic identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, April 2018.

[54] K. Skianis, F. Malliaros, and M. Vazirgiannis, "Fusing document, collection and label graph-based representations with word embeddings for text classification," in *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, New Orleans, Louisiana, United States, June 2018.

[55] S. Zheng, J. X. Hongyun Bao, Y. Hao, Z. Qi, and H. Hao, "A bidirectional hierarchical skip-gram model for text topic embedding," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 855–862, Vancouver, BC, July 2016.

Hindawi
Complexity
Volume 2018, Article ID 6292143, 15 pages
https://doi.org/10.1155/2018/6292143

Research Article

Scalable Multilabel Learning Based on Feature and Label Dimensionality Reduction

Jaesung Lee[1] **and Dae-Won Kim**[2]

[1]*Chung-Ang University, Seoul, Republic of Korea*
[2]*The School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea*

Correspondence should be addressed to Dae-Won Kim; dwkim@cau.ac.kr

Received 3 January 2018; Revised 15 July 2018; Accepted 16 August 2018; Published 24 September 2018

Academic Editor: Ireneusz Czarnowski

The data-driven management of real-life systems based on a trained model, which in turn is based on the data gathered from its daily usage, has attracted a lot of attention because it realizes scalable control for large-scale and complex systems. To obtain a model within an acceptable computational cost that is restricted by practical constraints, the learning algorithm may need to identify essential data that carries important knowledge on the relation between the observed features representing the measurement value and labels encoding the multiple target concepts. This results in an increased computational burden owing to the concurrent learning of multiple labels. A straightforward approach to address this issue is feature selection; however, it may be insufficient to satisfy the practical constraints because the computational cost for feature selection can be impractical when the number of labels is large. In this study, we propose an efficient multilabel feature selection method to achieve scalable multilabel learning when the number of labels is large. The empirical experiments on several multilabel datasets show that the multilabel learning process can be boosted without deteriorating the discriminating power of the multilabel classifier.

1. Introduction

Nowadays, the data-driven management of real-life systems based on a model obtained by analyzing data gathered from its daily usage is attracting significant attention because it realizes scalable control for large-scale and complex systems [1, 2]. Unfortunately, advances in the identification of important knowledge on the relation between the observed information and target concept are far from satisfactory for real-life applications such as text categorization [3], protein function prediction [4], emotion recognition [5], and assembly line monitoring [6]. This is because the underlying combinatorial optimization problem is computationally difficult. To deal with this complicated task in a scalable manner, the algorithm may need to identify essential data that carries important knowledge for building an acceptable model while satisfying practical constraints such as real-time response, limited data storage, and computational capability [7].

Although the majority of current machine learning algorithms are designed to learn the relation between information sources or features and a single concept or label, recent complex applications require that the algorithm extracts the relation to multiple concepts [8]. For example, a document can be assigned to multiple categories simultaneously [9], and protein compounds can also have multiple roles in a biological system [10]. Therefore, to identify important knowledge in this scenario, the algorithm must learn the complex relation between features and labels, formalized as the multilabel learning problem in this field. This scenario differs from that of the single-label learning problem because the problem itself offers the opportunity to improve learning accuracy by exploiting the dependency between labels [11, 12]. However, the algorithm eventually suffers as a result of the computational cost of the learning process owing to the multiple labels.

To reduce the computational burden of the algorithm, a straightforward approach is to ignore unimportant features in the training process that do not influence the learning quality [13, 14]. However, in the multilabel learning problem, this approach may be insufficient to satisfy the practical

constraints because a large number of labels can be involved in a related application. Moreover, the possible combinations of features and labels that should be considered for scoring the importance of features increases exponentially; i.e., the feature selection process can become computationally impractical. Additionally, the computational burden increases significantly because the number of features in the dataset is typically large when feature selection is considered. As a result, the number of possible combinations can increase considerably [15]. This is a serious problem because conventional multilabel learning algorithms with and without the feature selection process are unable to finish the learning process owing to the presence of too many features and the scoring process of the features, respectively.

In this study, we devise a new multilabel feature selection method that facilitates dimensionality reduction of labels from the scoring process. Specifically, our algorithm first analyzes the amount of information content in labels and reduces the computational burden by discarding labels that are unimportant to the scoring of the importance of features. Our contribution to this study compared to our previous works and the strategy to deal with the scalability issue can be summarized as follows:

(i) We propose an efficient multilabel feature selection method based on the simplest approximation of mutual information (MI) that is scalable to the number of labels; it costs constant time computations in terms of the number of labels

(ii) The computational cost of the feature selection process can be controlled easily owing to its simple form. This is an important property when the execution time is limited

(iii) The proposed method identifies a subset of labels that carries the majority of the information content compared to the original label set to preserve the quality of the scoring process

(iv) According to the characteristics of labels in terms of information content, we suggest that the size of labels be considered in the feature scoring process to preserve the majority of the information content

(v) In contrast to our previous works, the proposed method explicitly discards unimportant labels from the scoring process, resulting in a significant acceleration of the multilabel feature selection process

2. Multilabel Feature Selection

One of the most common methods of multilabel feature selection is the use of the conventional single-label feature selection method after transforming label sets into one or more labels [9, 16, 17]. In this regard, the simplest strategy is known as binary relevance, in which each label is separated and analyzed independently [18]. A statistical measure that can be used as a score function to measure feature importance can be employed after separating the label set; these measures include the Pearson correlation coefficient [19] and the odds ratio [20]. Thus, prohibitive computations may be required to obtain the final feature score if a large label set is involved. In contrast, efficient multilabel feature selection may not be achieved if the transformation process consumes excessive computational resources. For example, ELA + CHI evaluates the importance of each feature using χ^2 statistics (CHI) between the feature and a single label obtained by using entropy-based label assignment (ELA), which separates multiple labels and assigns them to duplicated patterns [9]. Thus, the label transformation process can be the bottleneck that incurs a prohibitive execution time if the multilabel dataset is composed of a large number of patterns and labels.

Although the computational cost of the transformation process can be reduced by applying a simple procedure such as a label powerset that treats each distinct label set as a class [17, 21], the feature selection process may be inefficient if the scoring process incurs excessive computational costs during the evaluation of the importance of the features [18, 22]. For example, PPT + RF identifies appropriate weight values for the features based on a label that is transformed by the pruned problem transformation (PPT) [21] and the conventional ReliefF (RF) scheme [23] for single-label feature selection [24]. Although the ReliefF method can be extended to handle multilabel problems directly [25], the execution time to obtain the final feature subset can be excessively long if the dataset is composed of a large number of patterns. This is because ReliefF requires similarity calculations for pattern pairs. Thus, the feature selection process itself should not incur a complicated scoring process to achieve efficient multilabel learning.

Instead of a label set transformation approach that may incur side effects [26], an algorithm adaptation approach that attempts to handle the problem of multilabel feature selection directly is considered [15, 27–31]. In this approach, a feature subset is obtained by optimizing a specific criterion such as a joint learning criterion involving feature selection and multilabel learning concurrently [32, 33], $l_{2,1}$-norm function optimization [31], a Hilbert–Schmidt independence criterion [28], label ranking errors [27], F-statistics [34], label-specific feature selection [12], and memetic feature selection based on mutual information (MI) [35]. However, if multilabel feature selection methods based on this strategy consider all features and labels simultaneously, the scoring process can be computationally prohibitive or even fail owing to the internal task of finding an appropriate hyperspace using pairwise pattern comparisons [27], a dependency matrix calculation [28], and iterative matrix inverse operations [31].

In our previous work [29], we demonstrated that MI can be decomposed into a sum of dependencies between variable subsets, which is a very useful property for solving multilabel learning problems [12, 15] because unnecessary computations can be determined prior to the actual computation and be rejected [36]. More efficient score functions, specialized into an incremental search strategy [37] and a quadratic programming framework [38], have also been considered. These score functions were employed to improve the

effectiveness of evolutionary searching [35, 39]. However, these MI-based score functions commonly require the calculation of the dependencies between all variable pairs composed of a feature and a label [14]. Thus, they share the same drawback in terms of computational efficiency because labels known to have no influence on the evaluation of feature importance are included in the calculations [15, 40]. In contrast to our previous study, our method proposed in this study discards unimportant labels explicitly prior to any multilabel learning process.

Although the characteristics of multilabel feature selection methods can vary according to the manner in which the importance of features is modeled, conventional methods create a feature subset by scoring the importance of features either to all labels [9, 17, 28] or to all possible combinations drawn from the label set [15, 27, 29]. Thus, these methods inherently suffer from prohibitive computational costs when the dataset is composed of a large number of labels.

3. Proposed Method

In this section, a formal definition of the multilabel classification and feature selection is provided. Based on our definition, the proposed label selection approach is described and a discussion on the influences of label subset selection to the feature selection is presented.

3.1. Problem Definition. Let $\mathscr{W}$ be a set of training examples or patterns where each example $w_i \in \mathscr{W} (1 \le i \le |\mathscr{W}|)$ is described by a set of features $\mathscr{F} = \{f_1, \dots, f_{|\mathscr{F}|}\}$; its association to multiple concepts can be represented using a subset of labels $\lambda_i \subseteq \mathscr{L}$, where $\mathscr{L} = \{l_1, l_2, \dots, l_{|\mathscr{L}|}\}$. In addition, let $\mathscr{T} = \{(t_i, \lambda_i) \mid 1 \le i \le |\mathscr{T}|\}$ be a set of test patterns, where λ_i is a true label set for t_i and is unknown to the multilabel classifier, resulting in $\mathscr{U} = \mathscr{W} \cup \mathscr{T}$ and $\mathscr{W} \cap \mathscr{T} = 0$. The task of multilabel learning is to derive a family of $|\mathscr{L}|$ functions, namely, $h_1, h_2, \dots, h_{|\mathscr{L}|}$ that are induced from the training examples, where each function $h_k : t_i \to \mathbb{R}$ outputs the class membership of t_i to l_k. Thus, relevant labels of t_i based on each function can be denoted as $\widehat{\lambda}_i = \{l_k \mid h_k(t_i) > \phi, 1 \le k \le |\mathscr{L}|\}$, where ϕ is a predefined threshold. For example, in the work of [41], a mapping function h_k for l_k is induced using $\mathscr{W}$. Based on h_k, the class membership value $h_k(t_i)$ for the given test pattern t_i is determined, where $h_k(t_i) \in [0, 1]$. In this work, the threshold ϕ is set to 0.5 according to the maximum a posteriori theorem. Although the algorithm outputs l_k as a relevant label for t_i if the class membership value is larger than 0.5 in their work, the range of class membership value can be different according to the multilabel classification algorithm. Although there are some trials to improve the multilabel learning performance by adapting threshold for each label [42], most conventional studies have employed the same value for all the labels.

One of the problems of multilabel feature selection that distinguishes it from classical single-label feature selection is the computational cost for selecting a subset of features with regard to the given multiple labels. The multilabel feature selection can then be achieved through a ranking process by assessing the importance of $|\mathscr{F}|$ features based on a score function and selecting the top-ranked n features from $\mathscr{F} (n \ll |\mathscr{F}|)$. To perform multilabel feature selection, an algorithm must be able to measure the dependency, i.e., importance score, between each feature and label set. The dependency between a feature $f \in \mathscr{F}$ and label set $\mathscr{L}$ can be measured using MI [43].

$$M(f ; \mathscr{L}) = H(f) - H(f, \mathscr{L}) + H(\mathscr{L}), \tag{1}$$

where $H(\cdot)$ of (1) represents a joint entropy that measures the information content carried by given a set of variables, defined as

$$H(X) = -\sum_{x \in X} P(x) \log_a P(x), \tag{2}$$

where x is a state represented by a variable X and $P(\cdot)$ is a probability mass function. If the base of the log function, a in (2), is two, this is known as Shannon entropy. When $|\mathscr{L}|$ is large, the calculation of $H(f, \mathscr{L})$ and $H(\mathscr{L})$ becomes unreliable because of too many joint states coming from $\mathscr{L}$ with insufficient patterns. For example, to observe all the possible associations between patterns and label subsets, the dataset should contain at least $2^{|\mathscr{L}|}$ patterns. Let X^* be the power set of X and $X_k^* = \{e \mid e \in X^*, |e| = k\}$. Equation (1) can then be rewritten using the work of Lee and Kim [15].

$$M(f ; \mathscr{L}) = \sum_{k=2}^{|\mathscr{L}|+1} (-1)^k V_k(f \times \mathscr{L}_{k-1}^*), \tag{3}$$

where $\times$ denotes the Cartesian product of two sets. Next, $V_k(\cdot)$ is defined as

$$V_k(Y) = \sum_{X \in Y_k^*} I(X), \tag{4}$$

where $I(X)$ is the interaction information for a given variable set X, defined as [44]

$$I(X) = -\sum_{Y \in X^*} (-1)^{|Y|} H(Y). \tag{5}$$

Equation (3) indicates that $M(f ; \mathscr{L})$ can be approximated into interaction information terms involving a feature and all the possible label subsets. With regard to (3), the most efficient approximation of (1) is known as [36]

$$\begin{aligned} \tilde{M}(f ; \mathscr{L}) &= V_2(f \times \mathscr{L}_1^*) \\ &= \sum_{X \in \{f \times \mathscr{L}_1^*\}_2^*} I(X) \\ &= \sum_{l \in \mathscr{L}} I(f, l) \\ &= \sum_{l \in \mathscr{L}} M(f ; l). \end{aligned} \tag{6}$$

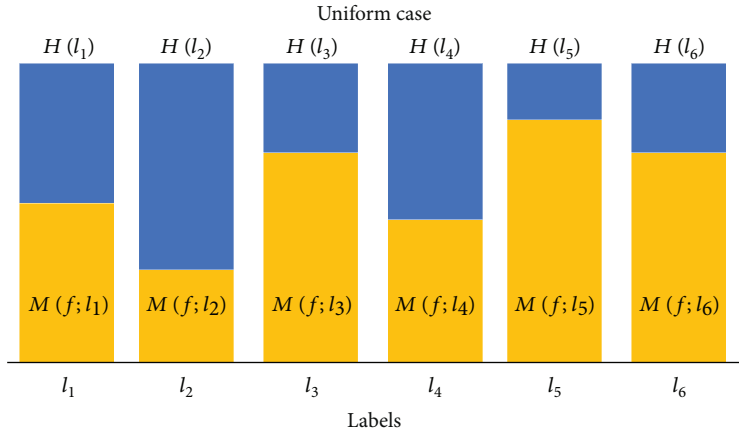


Figure 1: Score value calculation when label entropy values are uniform.

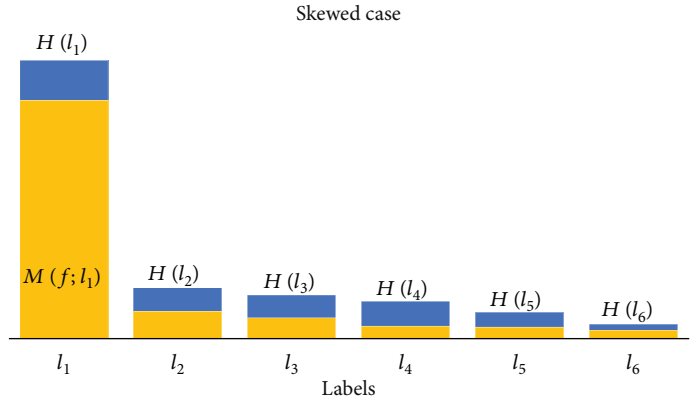


Figure 2: Score value calculation when label entropy values are skewed.

Accordingly, the score function J for evaluating the importance of a given feature f is written as

$$J = \sum_{l \in \mathscr{L}} M(f; l). \tag{7}$$

Equation (7) indicates that the computational cost increases linearly according to $|\mathscr{L}|$. By assuming that the computational cost for calculating a $M(\cdot;\cdot)$ term is a unit cost, the algorithm will consume $|\mathscr{L}|$ unit costs to compute the importance of a feature.

3.2. Label Subset Selection. In our multilabel feature selection problem, the rank of each feature is determined based on importance score using (7). The bound of a MI term is known as

$$0 \le M(f; l) \le \min\left(H(f), H(l)\right). \tag{8}$$

Thus, the bound of (7) is

$$0 \le J \le \sum_{l \in \mathscr{L}} \min(H(f), H(l)). \tag{9}$$

Because $H(f)$ is unknown before actually examining input features and any importance score cannot exceed the sum of entropy value of each label, (9) can be simplified as

$$0 \le J \le \sum_{l \in \mathscr{L}} H(l). \tag{10}$$

Equation (10) indicates that the score value of each feature is influenced by the entropy value of each label, and this fact implies Proposition 1 as follows [40].

Proposition 1 *(upper bound of J). If $\mathscr{L}$ is a given label set, then the upper-bound of J is*

$$\sum_{l \in \mathscr{L}} H(l). \tag{11}$$

Figure 1 represents how the importance score of a feature is determined with regard to Proposition 1; the height of the blue bar indicates the entropy value of the corresponding label, and height of the yellow bar indicates the MI between f and each label. Figures 1 and 2 represent two sample cases wherein each label carries the same amount of information content, and a small subset of label set carries the majority

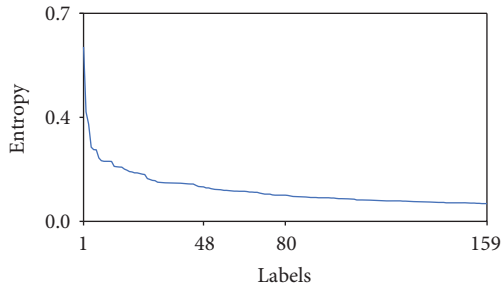


Figure 3: Entropy of each label in BibTeX dataset.

information content, respectively. As shown in Figure 1, the value of $M(f; l_i)$ can be varied according to $l_i \in \mathscr{L}$; however, its value is smaller than the entropy value of each label. When the entropy values of labels are uniformly distributed, all the MI terms between f and each label should be examined because each $M(f; l_i)$ term has same chance of giving significant contribution to the final score J. However, as shown in Figure 2, if there is a set of labels having a small entropy, i.e., if the entropy values of the labels are skewed, there can be MI terms that insignificantly contribute to the extent of J, because all the $M(f; l_j)$ will inherently have a small value, where l_j is a label of small entropy. Although the characteristics of label entropy values can vary between uniform and skewed cases, it is observed from most real-world multilabel datasets that the skewed case occurs more frequently than uniform case [15]. Additionally, as shown in Figure 2, because MI terms between a feature and labels with small entropy will not much contribute to the final score of the feature, they can be excluded for accelerating multilabel feature selection process.

Figure 3 shows the entropy value of each label in a BibTeX dataset [3] composed of 153 labels; please refer to Table 1 for details. The BibTeX dataset is created from the transactions of user activity in a tag recommendation system. For clarity, we represent the tool which is used to describe and process lists of reference as *BibTeX* whereas the name of the corresponding dataset is BibTeX subsequently. In this system, users freely submit *BibTeX* entries and assign relevant tags. The purpose of this system is recommending a relevant tag for the new *BibTeX* entries submitted by users. The system must identify the relation between *BibTeX* entry and relevant tags based on user transactions previously gathered, and hence, it can be regarded as a real-life text categorization system. For clarity, labels are sorted/ordered according to their entropy value. Figure 3 shows that each label gives a different entropy value; however, more importantly, approximately half of the labels give small entropy values, indicating that the MI terms with those labels will contribute weakly to the final score. Therefore, these labels can be discarded to accelerate the multilabel feature selection process.

Suppose that an algorithm selects $\mathscr{Q} \subset \mathscr{L}$ for reducing computational cost for multilabel feature selection. To prevent possible degradation, i.e., a change in the upper bound for J because of label subset selection, it is preferable that $\mathscr{Q}$ implies a similar upper bound compared to J. In other words, a subset of $\mathscr{L}$ that minimizes

$$\arg \min_{\mathscr{C} \subset \mathscr{L}} \sum_{l \in \mathscr{C}} H(l) = \sum_{l \in \mathscr{L}} H(l) - \sum_{l \in \mathscr{Q}} H(l) \tag{12}$$

is preferable, where $\mathscr{C} = \mathscr{L} \setminus \mathscr{Q}$ is a set of discarded labels.

Proposition 2. *The optimal $\mathscr{C}$ is composed of labels with the lowest entropy.*

Proof 1. Our goal is to identify a subset of labels $\mathscr{C}$ that influences the upper bound of J as insignificantly as possible, when $\mathscr{C}$ is discarded from $\mathscr{L}$ for the feature scoring process. Equation (11) indicates that the upper bound of J is the sum of entropy values for each label and the entropy function always gives positive value, therefore the optimal $\mathscr{L}$ should be composed of labels with the lowest entropy.

Proposition 2 indicates that the optimal $\mathscr{C}$ can be obtained by iteratively discarding a label with the smallest entropy until $\mathscr{Q}$ contains a desirable number of labels. After obtaining $\mathscr{Q}$, the approximated score function for evaluating a feature f is written as

$$\tilde{J}(f) = \sum_{l \in \mathscr{Q}} M(f; l). \tag{13}$$

Finally, the difference between J and $\tilde{J}$ can be exactly calculated as

$$J - \tilde{J} = \sum_{l \in \mathscr{C}} \min(H(f), H(l)), \tag{14}$$

where $J - \tilde{J}$ is always positive because $H(X) \geq 0$. Algorithm 1 describes the procedure of the proposed method.

3.3. Number of Remaining Labels. A final issue related to label subset selection has to do with the number of labels that should be discarded. In fact, because the upper bound of (12) gets larger when the number of discarded labels is increased, there is a trade-off between computational efficiency and the accurate score of each feature. However, the actual computational cost can also be easily predicted after examining some features because the computational cost for examining $|\mathscr{F}|$ features based on (7) is easily calculated as $|\mathscr{F}| \cdot |\mathscr{L}|$, and the computational cost based on (13) is $|\mathscr{F}| \cdot |\mathscr{Q}|$. However, if there is no such constraint and a user only wants to determine a reasonable value of $|\mathscr{Q}|$ for a fast analysis, then a simple and efficient way would be helpful.

Suppose that the algorithm attempts to preserve the upper bound of the score function based on $\mathscr{Q}$, then the upper bound should be greater than or equal to the error because of label subset selection; i.e., the inequality (15) should hold.

$$\sum_{l \in \mathscr{Q}} H(l) \geq \sum_{l \in \mathscr{C}} H(l). \tag{15}$$

Table 1: Standard characteristics of multilabel datasets.

Datasets (abbreviation)	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{F}\rvert$	Feature type	$\lvert L\rvert$	Card	Den	Distinct	PDL	Domain	$\lvert\mathcal{S}\rvert$
BibTeX	7395	1836	Nominal	159	2.402	0.015	2856	0.386	Text	86
Emotions	593	72	Numeric	6	1.868	0.311	27	0.046	Music	24
Enron	1702	1001	Nominal	53	3.378	0.064	753	0.442	Text	41
Genbase	662	1185	Nominal	27	1.252	0.046	32	0.048	Biology	26
Language Log (LLog)	1460	1004	Nominal	75	1.180	0.016	304	0.208	Text	38
Medical	978	1494	Nominal	45	1.245	0.028	94	0.096	Text	31
Slashdot	3782	1079	Nominal	22	1.181	0.054	156	0.041	Text	61
TMC2007	28,596	981	Nominal	22	2.158	0.098	1341	0.047	Text	169
Yeast	2417	103	Numeric	14	4.237	0.303	198	0.082	Biology	49
Arts	7484	1157	Numeric	26	1.654	0.064	599	0.080	Text	87
Business	11,214	1096	Numeric	30	1.599	0.053	233	0.021	Text	106
Computers	12,444	1705	Numeric	33	1.507	0.046	428	0.034	Text	112
Education	12,030	1377	Numeric	33	1.463	0.044	511	0.042	Text	110
Entertainment (entertain)	12,730	1600	Numeric	21	1.414	0.067	337	0.026	Text	113
Health	9205	1530	Numeric	32	1.644	0.051	335	0.036	Text	96
Recreation	12,828	1516	Numeric	22	1.429	0.065	530	0.041	Text	113
Reference	8027	1984	Numeric	33	1.174	0.036	275	0.034	Text	90
Science	6428	1859	Numeric	40	1.450	0.036	457	0.071	Text	80
Social	12,111	2618	Numeric	39	1.279	0.033	361	0.030	Text	110
Society	14,512	1590	Numeric	27	1.670	0.062	1054	0.073	Text	120

Input: n, $|\mathcal{Q}|$;
▷ Number of features to be selected, $n \ll d$
▷ Number of labels to be considered, $|\mathcal{Q}| \ll |\mathcal{L}|$
Output: $\mathcal{S}$; ▷ Selected feature subset, $\mathcal{S}$
Initialize $\mathcal{S} \leftarrow 0$
for all $l \in \mathcal{L}$ **do**
    Calculate value of entropy for l;
end for
Create $\mathcal{Q}$ with $|\mathcal{Q}|$ labels of highest entropy from $\mathcal{L}$;
for all $f \in \mathcal{F}$ **do**
    $\tilde{J}(f) \leftarrow$ Assessing importance of f by using Eq. (13);
end for
Sort $\mathcal{F}$ based upon score values $\tilde{J}$ descendingly;
Set $\mathcal{S} \leftarrow$ Top n features of high score in $\mathcal{F}$;

Algorithm 1: Procedure of Proposed Method.

According to the characteristics of the given labels, the number of labels to be discarded can then be identified as Lemmas 1, 2, and 3.

Lemma 1. *Skewed case.*

$$|\mathcal{Q}| = 1. \tag{16}$$

Proof 2. For simplicity, suppose $\mathcal{L}$ is sorted according to the entropy value of each label, such that l_1 has the smallest entropy and $l_{|\mathcal{L}|}$ has the largest entropy. Suppose that the entropy values of the labels are skewed, as shown in Figure 2. If $l_{|\mathcal{L}|}$ is the only one label with a positive entropy and the remaining labels have no entropy, then the algorithm will move $l_{|\mathcal{L}|}$ to $\mathcal{Q}$ and $l_1, \ldots, l_{|\mathcal{L}|-1}$ to $\mathcal{C}$, and then terminate.

So far, we considered the uniform and skewed cases that are the two extremes of the characteristics in the viewpoint of information content carried by each label. Next, we consider an intermediate between the uniform and skewed cases, in which the information content of each label is proportional to their sequence when they are ascendingly sorted according to their entropy values. For this case, about 30% of labels with the largest entropy should be included in $\mathcal{Q}$.

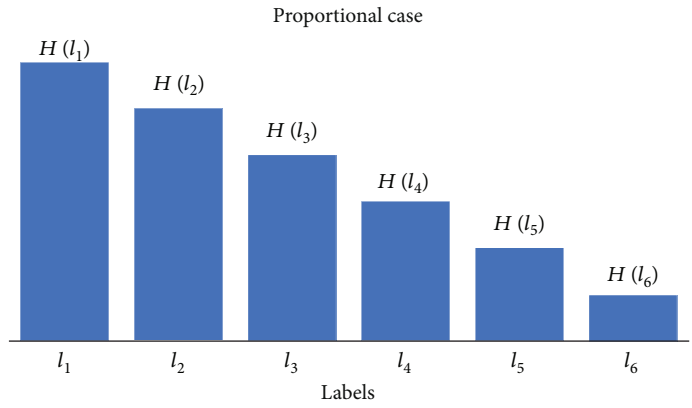


Figure 4: Score value calculation when label entropy values are proportional to their rank.

Lemma 2. *Proportional case.*

$$|\mathcal{Q}| \approx 0.3|\mathcal{L}|. \tag{17}$$

Proof 3. For simplicity, suppose that $\mathcal{L}$ is sorted according to the entropy value of each label, such that l_1 has the smallest entropy value and $l_{|\mathcal{L}|}$ has the largest entropy value. Suppose that the entropy values of the labels are proportional to the sequence number of labels in $\mathcal{L}$ as shown in Figure 4. In this case, an entropy value can be represented as

$$H(l_i) = \alpha \cdot i, \tag{18}$$

where i is the sequence number of label l_i in $\mathcal{L}$. Because the actual entropy value is unnecessary for determining superiority among labels, the term α in (18) can be ignored. Then the entropy value of each label with regard to their sequence can be represented as

$$1, 2, \ldots, |\mathcal{Q}|, \ldots, |\mathcal{L}|. \tag{19}$$

Because the sum of the integers from 1 to i is equal to $i(i+1)/2$, (20) is obtained using (15).

$$\frac{|\mathcal{L}|(|\mathcal{L}|+1)}{2} - \frac{|\mathcal{Q}|(|\mathcal{Q}|+1)}{2} = \frac{|\mathcal{Q}|(|\mathcal{Q}|+1)}{2}. \tag{20}$$

Equation (20) can be simplified as

$$2|\mathcal{Q}|^2 + 2|\mathcal{Q}| - |\mathcal{L}|(|\mathcal{L}|+1) = 0. \tag{21}$$

The solution of (21) is given as

$$\begin{aligned} |\mathcal{Q}| &= \frac{-2 \pm (4 - 4 \cdot 2 \cdot (-|\mathcal{L}|(|\mathcal{L}|+1)))^{1/2}}{4} \\ &= \frac{-2 \pm (4 + 8|\mathcal{L}|(|\mathcal{L}|+1))^{1/2}}{4}. \end{aligned} \tag{22}$$

Because $|\mathcal{Q}|$ is always a positive integer, the negative solution can be ignored. Therefore, we obtain

$$|\mathcal{Q}| = \frac{-2 + (4 + 8|\mathcal{L}|(|\mathcal{L}|+1))^{1/2}}{4}. \tag{23}$$

For clarity, we approximate the solution as

$$\begin{aligned} |\mathcal{Q}| &= \frac{-2 + \left(8|\mathcal{L}|^2 + 8|\mathcal{L}| + 4\right)^{1/2}}{4} \\ &\approx \frac{-2 + \left(2\sqrt{2}|L| + 2\right)^{2\cdot(1/2)}}{4} \\ &= \frac{2\sqrt{2}|L|}{4} \approx 0.7|\mathcal{L}|. \end{aligned} \tag{24}$$

The approximated solution 0.7 $\mathcal{L}$ is slightly greater than the exact solution for $|\mathcal{Q}|$. Therefore, (2) indicates that approximately 70% of labels will be discarded, whereas 30% of labels will remain in $\mathcal{Q}$.

Lemma 3. *Uniform case.*

$$|\mathcal{Q}| = \begin{cases} \dfrac{|\mathcal{L}|}{2}, & \text{if } |\mathcal{L}| \text{ is even,} \\ \dfrac{|\mathcal{L}|}{2+1}, & \text{if } |\mathcal{L}| \text{ is odd.} \end{cases} \tag{25}$$

Proof 4. Suppose that the entropy values of the labels are uniformly distributed as shown in Figure 1. The figure indicates that $|\mathcal{Q}|$ should have corresponding labels with regard to each discarded label. Therefore, for the even case, the number of labels in $\mathcal{Q}$ and $\mathcal{C}$ must be the same for (15) to hold; thus, $|\mathcal{Q}| = |\mathcal{L}|/2$. For the odd case, $\mathcal{Q}$ must have one more label than $\mathcal{C}$; thus, $|\mathcal{Q}| = |\mathcal{L}|/2 + 1$.

The proof indicates that the number of labels to be selected is decreased as the entropy values of labels are skewed. In addition, the proof guarantees that $|\mathcal{Q}|$ must be lesser than $|\mathcal{L}|$ and the computational cost for evaluating

the importance of each feature based on $\mathcal{Q}$ must be smaller than $|L|/2 + 1$. Therefore, Theorem 1 can be obtained.

Theorem 1 *$|\mathcal{Q}|$ is always smaller than $|\mathcal{L}|$.*

Proof 5. Suppose that there are two label sets $\mathcal{Q}$ and $\mathcal{C}$ to be considered and ignored for calculating the importance of each feature, respectively. Because $\mathcal{Q}$ should carry the majority information content than $\mathcal{C}$, $\sum_{l\in\mathcal{Q}} H(l)$ should be larger than $\sum_{l\in\mathcal{C}} H(l)$. As shown in Proposition 2, the algorithm is able to achieve this goal by (1) including a label with the largest entropy in $\mathcal{Q}$ and removing that label from $\mathcal{L}$, (2) including labels with the smallest entropy in $\mathcal{C}$ and removing those labels from $\mathcal{L}$ iteratively until $\sum_{l\in\mathcal{Q}} H(l) > \sum_{l\in\mathcal{C}} H(l)$, and (3) repeating (1) to (2) until $\mathcal{L}$ has no element. If the entropy values of all the labels are the same, i.e., the largest entropy value and the smallest entropy value are the same, one label can be included in $\mathcal{C}$ when a label is included in $\mathcal{Q}$ as Lemma 3. Thus, $\mathcal{C}$ possibly has more labels than $\mathcal{Q}$ in the case when the smallest entropy value is actually smaller than the largest entropy value, indicating that the uniform case is the worst case from the viewpoint of the number of labels in $\mathcal{Q}$. Consequently, the number of labels in $|\mathcal{Q}|$ cannot be larger than $|\mathcal{L}|/2 + 1$.

Because $|\mathcal{Q}|$ is always smaller than $|\mathcal{L}|$ and calculating one MI term is regarded as the unit cost, the computational cost for evaluating each feature using $\tilde{J}$ is constant in the viewpoint of the number of labels.

3.4. Influence to Feature Ranking. The multilabel feature selection is done by ranking each feature according to its importance value. After label subset selection is conducted, the importance score of each feature will be calculated by summing $M(f; l_i)$ terms, where $l_i \in \mathcal{Q}$. However, when the entropy values of labels are skewed, the rank based on J and that based on $\tilde{J}$ are unlikely to change. To demonstrate this aspect, we illustrate how the importance score is calculated under the skewed case in Figure 5. In the figure, there are three labels, namely l_1, l_2, and l_3; l_1 has the highest entropy, whereas l_2 and l_3 have insignificant entropies. The MI between each feature and each label is represented as yellow bars, and the final score of each feature is represented on the right hand side of the figure. The figure indicates that (1) the MI between each feature and each label is bound by the entropy of each label and (2) the MI between each feature and the labels of high entropy mostly determines the final score of each feature. In other words, (3) the influence of MIs between each feature and l_2 and l_3 is insignificant to the final score.

With regard to the process of feature selection, Figure 5 implies three more indications. The first indication is related to the influence of labels with high entropy to the final score. Because the final score is determined by summing MI terms between a feature and all the labels, a feature that is dependent on labels with high entropy is likely to have a high importance score. Therefore, those features will be included to the final feature subset $\mathcal{S}$

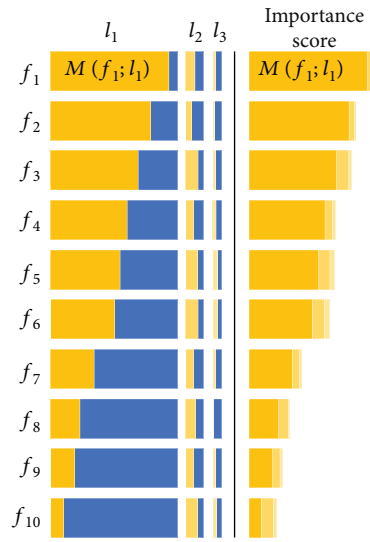


Figure 5: Importance score of each feature in the viewpoint of entropy of each label when entropy values of labels are skewed.

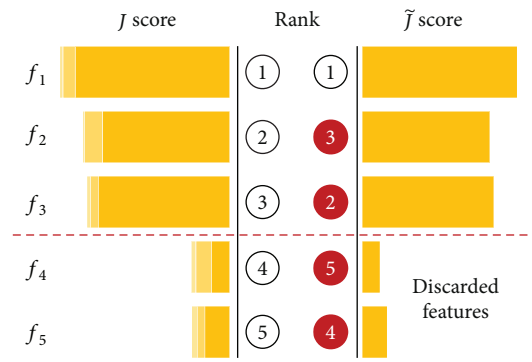


Figure 6: An example of rank change.

because of their higher rank, and they show promise as potential members of $\mathcal{S}$. The second indication is related to the change among similarly ranked features. However, because the goal of feature selection is to select a feature subset that is composed of n features, the specific rank of each feature is unimportant. For example, suppose that the algorithm tries to choose ten features from $\mathcal{S}$ because $\mathcal{F}$ is set to ten by users or there is a limitation on the storage. The label subset selection may change the rank of the second- and the third-ranked features; however, these two features will be included in the final feature subset $\mathcal{S}$ because the algorithm is allowed to select ten features. The final indication is related to the rank among unimportant features. Although there may be a set of features that are dependent on labels with small entropy, these features will have low importance scores and hence will be discarded from $\mathcal{S}$.

Although the example of Figure 6 indicates that the rank of each feature will be unlikely to change or may be changed meaninglessly, empirical experiments should be followed to investigate the availability of label subset selection.

4. Experimental Results

A description of the multilabel datasets, algorithms, statistical tests, and other settings used in the experimental study is provided in this section. Next, the experimental results based on different multilabel learning methods, the datasets, and the analysis are presented subsequently.

4.1. Experimental Settings. Twenty real multilabel datasets were employed in our experiments [12, 25, 35], where the number of relevant and irrelevant features is unknown. Table 1 shows the standard statistics of the multilabel datasets and the meaning of each notation is described as follows:

(i) $|\mathscr{U}|$: number of patterns in the dataset

(ii) $|\mathscr{F}|$: number of features

(iii) Feature type: type of feature

(iv) $|\mathscr{L}|$: number of labels

(v) Card: average number of labels for each instance (label cardinality)

(vi) Den: label cardinality divided by the total number of labels (label density)

(vii) Distinct: number of unique label subsets in $\mathscr{L}$ (distinct label set)

(viii) *PDL*: number of distinct label sets divided by the total number of patterns (portion of distinct labels)

(ix) Domain: applications to which each dataset corresponds

(x) $|\mathscr{S}|$: number of features to be selected ($\sqrt{|\mathscr{W}|}$)

These statistics show that the 20 datasets cover a broad range of cases with diversified multilabel properties. In the case where the feature type is numeric, we discretized the features using the LAIM discretization method [45]. In addition, datasets that are composed of more than 10,000 features are preprocessed to contain the top 2% and 5% features with the highest document frequency [12, 46]. We conducted an 8 : 2 hold-out cross-validation, and each experiment was repeated ten times. The average value was taken to represent the classification performance. A wide variety of multilabel classifiers can be considered to conduct multilabel classification [8]. In this study, we chose the multilabel naive Bayes classifier [41] because the learning process can be conducted quickly, owing to the well-known naive Bayes assumption, without incurring an additional tuning process, and because our primary concern in this study is efficient multilabel learning. Finally, we considered four evaluation measures, which are employed in many multilabel learning studies: execution time for the training and test process, Hamming loss, multilabel accuracy, and subset accuracy [8, 29].

The Friedman test was employed to analyze the performance of the multilabel feature selection methods; it is a widely used statistical test for comparing multiple methods over a number of datasets [47]. The null hypothesis of the equal performance of the compared algorithms is rejected in terms of each evaluation measure if the Friedman statistic F_F is greater than the critical value at significance level α. In this case, we need to proceed with certain post hoc tests to analyze the relative performance of the comparison methods. The Bonferroni-Dunn test is employed because we are interested in determining whether the proposed method achieves a performance similar to that of the feature selection process considering all of the labels and to that of the multilabel learning without the feature selection process [48]. For the Bonferroni-Dunn test, the performances of the proposed method and another method are deemed to be statistically similar if their average ranks over all datasets are within one CD. For our experiments, the critical value at the significance level $\alpha = 0.05$ is 2.492, and the CD with $\alpha = 0.05$ is 1.249 because $q_{0.05} = 2.498$ [48].

4.2. Comparative Studies. In this section, we compare the proposed feature selection method based on the label subset selection strategy to the conventional multilabel learning without the feature selection process and the conventional feature selection method without the label subset selection. The detail of each method, besides the proposed method, is described as follows:

(i) No: conventional multilabel learning the without feature selection process. Here, $\mathscr{F}$ is used as the input features for the multilabel classifier

(ii) SL: multilabel learning with the proposed feature selection process. Here, S is used as the input features. In the feature selection process, only one label with the highest entropy is considered to measure the importance of each feature

(iii) 3L: multilabel learning with the proposed feature selection process. Here, S is used as the input features. In the feature selection process, 30% of labels with the highest entropy are chosen by the label selection strategy to compose Q

(iv) 5L: multilabel learning with the proposed feature selection process. Here, S is used as the input features. In the feature selection process, 50% of labels with the highest entropy are chosen by the label selection strategy to compose Q

(v) AL: multilabel learning with the conventional feature selection process. Here, S is used as the input features. The same feature subset can be obtained by setting $Q = L$ for the proposed method

All methods were carefully implemented in a MATLAB 8.2 programming environment and tested on an Intel Core i7-3930 K (3.2 GHz) with 64 GB memory.

Tables 2–5 report the detailed experimental results of each method under comparison on 20 multilabel datasets. For each evaluation measure, ↓ means *the smaller the better* whereas ↑ means *the larger the better*. The best

Table 2: Execution time (↓) for training and testing process of each comparing method (mean ± std. deviation) on 20 multilabel datasets.

Method	BibTeX	Emotions	Enron	Genbase	LLog	Medical	Slashdot
No	141.852 ± 0.386	0.091 ± 0.001	12.819 ± 0.028	4.713 ± 0.032	17.053 ± 0.057	12.996 ± 0.090	7.796 ± 0.020
SL	**9.326 ± 0.322** ·	0.069 ± 0.050	**1.039 ± 0.198** ·	**0.541 ± 0.233** ·	**1.164 ± 0.201** ·	**0.870 ± 0.281** ·	**1.279 ± 0.209** ·
3L	17.820 ± 1.838	0.058 ± 0.018	1.846 ± 0.592	0.980 ± 0.560	2.194 ± 0.719	1.846 ± 0.925	2.241 ± 0.502
5L	20.686 ± 2.355	0.070 ± 0.028	2.118 ± 0.734	1.206 ± 0.740	2.455 ± 0.859	2.176 ± 1.149	2.239 ± 0.503
AL	201.458 ± 41.405	**0.038 ± 0.007** ·	3.112 ± 1.768	1.071 ± 0.903	4.027 ± 2.622	3.353 ± 2.561	3.742 ± 1.166
Method	TMC2007	Yeast	Arts	Business	Computers	Education	Entertain
No	28.134 ± 0.033	0.450 ± 0.004	15.814 ± 0.094	20.921 ± 0.166	38.178 ± 0.369	29.887 ± 0.320	23.983 ± 0.324
SL	**8.962 ± 0.258** ·	0.291 ± 0.056	**2.750 ± 0.182** ·	**4.158 ± 0.291** ·	**6.140 ± 0.318** ·	**5.093 ± 0.350** ·	**5.269 ± 0.280** ·
3L	13.903 ± 0.662	0.314 ± 0.039	4.697 ± 0.508	6.755 ± 0.609	10.701 ± 0.805	8.440 ± 0.910	9.568 ± 0.729
5L	13.908 ± 0.665	0.347 ± 0.054	5.688 ± 0.664	8.047 ± 0.783	12.995 ± 1.066	10.115 ± 1.213	9.572 ± 0.724
AL	84.744 ± 7.584	**0.238 ± 0.012** ·	15.074 ± 2.802	32.428 ± 8.005	97.128 ± 12.655	55.539 ± 10.467	63.178 ± 6.986
Method	Health	Recreation	Reference	Science	Social	Society	Avg. rank
No	28.702 ± 0.206	24.032 ± 0.025	34.620 ± 0.293	34.302 ± 0.050	67.920 ± 0.128	34.704 ± 0.032	4.40
SL	**4.258 ± 0.267** ·	**5.165 ± 0.286** ·	**4.246 ± 0.354** ·	**3.828 ± 0.414** ·	**8.281 ± 0.691** ·	**6.655 ± 0.337** ·	**1.15** ·
3L	7.272 ± 0.689	9.310 ± 0.721	7.495 ± 0.885	6.709 ± 1.133	15.099 ± 1.944	11.544 ± 0.820	2.10
5L	8.780 ± 0.915	9.311 ± 0.720	9.137 ± 1.168	8.154 ± 1.507	18.516 ± 2.596	13.989 ± 1.075	3.10
AL	44.589 ± 6.295	59.392 ± 7.450	54.074 ± 11.295	49.855 ± 12.972	263.023 ± 51.640	107.083 ± 14.488	4.25

Table 3: Hamming loss (↓) performance of each comparing method (mean ± std. deviation) on 20 multilabel datasets.

Method	BibTeX	Emotions	Enron	Genbase	LLog	Medical	Slashdot
No	0.082 ± 0.002	**0.240 ± 0.028** ·	0.214 ± 0.009	**0.007 ± 0.001** ·	0.340 ± 0.024	0.019 ± 0.001	**0.041 ± 0.001** ·
SL	**0.067 ± 0.002** ·	0.268 ± 0.020	0.144 ± 0.005	0.008 ± 0.001	**0.201 ± 0.013** ·	0.032 ± 0.003	0.047 ± 0.002
3L	0.071 ± 0.003	0.266 ± 0.023	**0.139 ± 0.005** ·	0.007 ± 0.001	0.250 ± 0.010	**0.014 ± 0.002** ·	0.044 ± 0.001
5L	0.080 ± 0.002	0.266 ± 0.025	0.140 ± 0.004	0.008 ± 0.002	0.254 ± 0.011	0.015 ± 0.002	0.043 ± 0.002
AL	0.086 ± 0.001	0.265 ± 0.023	0.140 ± 0.003	0.010 ± 0.003	0.253 ± 0.010	0.018 ± 0.002	0.043 ± 0.002
Method	TMC2007	Yeast	Arts	Business	Computers	Education	Entertain
No	0.139 ± 0.001	0.272 ± 0.007	0.109 ± 0.004	0.090 ± 0.002	0.117 ± 0.003	0.079 ± 0.002	0.123 ± 0.004
SL	**0.107 ± 0.001** ·	0.271 ± 0.007	0.072 ± 0.002	**0.050 ± 0.002** ·	0.080 ± 0.003	**0.055 ± 0.002** ·	0.111 ± 0.003
3L	0.125 ± 0.002	**0.270 ± 0.005** ·	0.072 ± 0.002	0.067 ± 0.002	**0.064 ± 0.003** ·	0.058 ± 0.002	**0.078 ± 0.002** ·
5L	0.126 ± 0.001	0.273 ± 0.007	**0.071 ± 0.002** ·	0.069 ± 0.003	0.068 ± 0.003	0.058 ± 0.002	0.081 ± 0.002
AL	0.123 ± 0.001	0.276 ± 0.007	0.072 ± 0.002	0.070 ± 0.003	0.070 ± 0.003	0.059 ± 0.002	0.081 ± 0.002
Method	Health	Recreation	Reference	Science	Social	Society	Avg. rank
No	0.073 ± 0.002	0.129 ± 0.005	0.097 ± 0.003	0.132 ± 0.004	0.077 ± 0.002	0.197 ± 0.003	4.20
SL	0.055 ± 0.002	**0.063 ± 0.001** ·	0.079 ± 0.004	0.056 ± 0.003	**0.040 ± 0.002** ·	0.173 ± 0.005	2.80
3L	0.056 ± 0.001	0.073 ± 0.002	**0.066 ± 0.003** ·	**0.054 ± 0.004** ·	0.045 ± 0.002	0.144 ± 0.007	**2.25** ·
5L	**0.053 ± 0.001** ·	0.071 ± 0.002	0.070 ± 0.004	0.055 ± 0.003	0.051 ± 0.002	0.135 ± 0.007	2.60
AL	0.053 ± 0.002	0.073 ± 0.003	0.071 ± 0.004	0.057 ± 0.003	0.052 ± 0.002	**0.134 ± 0.007** ·	3.15

performance among the five methods under comparison is shown in boldface with a bullet mark. In addition, the average rank of each method under comparison over all the multilabel datasets is presented in the last column of each table. Table 6 represents the Friedman statistics F_F and the corresponding critical values on each evaluation measure. As shown in Table 6, at significance level $\alpha = 0.05$, the null hypothesis of *equal* performance among the methods under comparison is clearly rejected in terms of each evaluation measure.

To show the relative performance of the proposed method and conventional multilabel learning methods,

Table 4: Multilabel accuracy (↑) performance of each comparing method (mean ± std. deviation) on 20 multilabel datasets.

Method	BibTeX	Emotions	Enron	Genbase	LLog	Medical	Slashdot
No	**0.191 ± 0.006** ·	**0.543 ± 0.043** ·	0.196 ± 0.008	0.904 ± 0.019	0.037 ± 0.001	0.335 ± 0.029	**0.445 ± 0.014** ·
SL	0.115 ± 0.006	0.486 ± 0.030	0.229 ± 0.011	0.917 ± 0.018	**0.053 ± 0.004** ·	0.517 ± 0.041	0.265 ± 0.019
3L	0.171 ± 0.008	0.488 ± 0.036	**0.236 ± 0.009** ·	**0.924 ± 0.019** ·	0.044 ± 0.002	**0.705 ± 0.029** ·	0.345 ± 0.012
5L	0.166 ± 0.007	0.490 ± 0.037	0.235 ± 0.009	0.919 ± 0.017	0.043 ± 0.002	0.690 ± 0.030	0.364 ± 0.014
AL	0.162 ± 0.008	0.489 ± 0.036	0.235 ± 0.008	0.919 ± 0.019	0.043 ± 0.002	0.667 ± 0.042	0.362 ± 0.014
Method	TMC2007	Yeast	Arts	Business	Computers	Education	Entertain
No	0.395 ± 0.004	**0.425 ± 0.010** ·	**0.328 ± 0.007** ·	0.627 ± 0.006	0.338 ± 0.006	**0.319 ± 0.008** ·	0.348 ± 0.008
SL	0.410 ± 0.005	0.414 ± 0.011	0.225 ± 0.018	**0.666 ± 0.009** ·	0.399 ± 0.013	0.233 ± 0.008	0.294 ± 0.004
3L	0.417 ± 0.005	0.422 ± 0.010	0.281 ± 0.011	0.649 ± 0.008	**0.434 ± 0.007** ·	0.267 ± 0.008	**0.405 ± 0.004** ·
5L	0.416 ± 0.004	0.419 ± 0.010	0.296 ± 0.009	0.648 ± 0.007	0.434 ± 0.008	0.269 ± 0.009	0.391 ± 0.009
AL	**0.430 ± 0.004** ·	0.416 ± 0.009	0.300 ± 0.011	0.644 ± 0.008	0.431 ± 0.009	0.268 ± 0.007	0.393 ± 0.010
Method	Health	Recreation	Reference	Science	Social	Society	Avg. rank
No	0.476 ± 0.006	0.343 ± 0.011	0.388 ± 0.020	0.215 ± 0.006	0.516 ± 0.009	**0.202 ± 0.004** ·	3.40
SL	0.514 ± 0.004	0.294 ± 0.005	0.410 ± 0.009	0.163 ± 0.016	0.480 ± 0.009	0.168 ± 0.005	4.25
3L	0.516 ± 0.008	0.352 ± 0.018	**0.432 ± 0.006** ·	0.223 ± 0.016	0.542 ± 0.011	0.185 ± 0.007	2.40
5L	**0.518 ± 0.004** ·	**0.369 ± 0.006** ·	0.432 ± 0.008	**0.229 ± 0.010** ·	**0.544 ± 0.009** ·	0.191 ± 0.006	**2.25** ·
AL	0.516 ± 0.003	0.362 ± 0.010	0.431 ± 0.008	0.223 ± 0.014	0.544 ± 0.009	0.192 ± 0.006	2.70

Table 5: Subset accuracy (↑) performance of each comparing method (mean ± std. deviation) on 20 multilabel datasets.

Method	BibTeX	Emotions	Enron	Genbase	LLog	Medical	Slashdot
No	0.063 ± 0.005	**0.242 ± 0.049** ·	0.001 ± 0.001	**0.863 ± 0.027** ·	0.000 ± 0.000	0.301 ± 0.027	**0.357 ± 0.016** ·
SL	0.048 ± 0.006	0.181 ± 0.041	0.003 ± 0.003	0.833 ± 0.032	**0.002 ± 0.002** ·	0.319 ± 0.042	0.233 ± 0.017
3L	0.062 ± 0.006	0.186 ± 0.031	0.004 ± 0.004	0.842 ± 0.034	0.000 ± 0.000	**0.551 ± 0.041** ·	0.298 ± 0.015
5L	0.063 ± 0.006	0.181 ± 0.035	**0.005 ± 0.005** ·	0.835 ± 0.030	0.000 ± 0.000	0.531 ± 0.038	0.311 ± 0.014
AL	**0.064 ± 0.006** ·	0.181 ± 0.037	**0.005 ± 0.005** ·	0.835 ± 0.033	0.000 ± 0.000	0.510 ± 0.052	0.311 ± 0.015
Method	TMC2007	Yeast	Arts	Business	Computers	Education	Entertain
No	0.086 ± 0.005	0.098 ± 0.007	0.164 ± 0.008	0.469 ± 0.014	0.138 ± 0.007	0.179 ± 0.008	0.171 ± 0.008
SL	**0.119 ± 0.003** ·	0.093 ± 0.009	0.146 ± 0.018	**0.504 ± 0.013** ·	0.275 ± 0.019	0.176 ± 0.007	0.150 ± 0.004
3L	0.106 ± 0.005	**0.098 ± 0.010** ·	0.195 ± 0.011	0.490 ± 0.011	**0.335 ± 0.008** ·	0.192 ± 0.007	**0.283 ± 0.012** ·
5L	0.107 ± 0.003	0.096 ± 0.009	0.203 ± 0.010	0.489 ± 0.011	0.332 ± 0.009	**0.193 ± 0.007** ·	0.250 ± 0.020
AL	0.115 ± 0.004	0.093 ± 0.008	**0.206 ± 0.012** ·	0.486 ± 0.012	0.328 ± 0.010	0.191 ± 0.006	0.249 ± 0.020
Method	Health	Recreation	Reference	Science	Social	Society	Avg. rank
No	0.227 ± 0.008	0.140 ± 0.009	0.240 ± 0.035	0.072 ± 0.006	0.402 ± 0.014	**0.069 ± 0.003** ·	3.68
SL	**0.336 ± 0.008** ·	0.223 ± 0.004	0.355 ± 0.009	0.104 ± 0.016	0.389 ± 0.013	0.038 ± 0.006	3.78
3L	0.329 ± 0.009	0.269 ± 0.016	0.375 ± 0.006	0.148 ± 0.009	0.456 ± 0.013	0.055 ± 0.008	2.63
5L	0.336 ± 0.006	**0.285 ± 0.008** ·	**0.376 ± 0.008** ·	**0.158 ± 0.008** ·	**0.460 ± 0.012** ·	0.055 ± 0.008	**2.23** ·
AL	0.333 ± 0.006	0.284 ± 0.010	0.374 ± 0.008	0.151 ± 0.010	0.456 ± 0.011	0.055 ± 0.007	2.70

Figure 7 illustrates the CD diagrams on each evaluation measure, where the average rank of each method is marked along the axis with better ranks placed on the right hand side of each figure [47]. In each figure, any comparison method whose average rank is within one CD to that of the best method is interconnected with a thick line; the length of the thick line indicates the extent of CD on a diagram. Otherwise, any method not connected with the best method is considered to have a significantly different performance from the latter.

Based on the empirical experiments and statistical analysis, the following indications can be observed:

Table 6: Summary of the Friedman statistics F_F ($k = 5$, $N = 20$) and the critical value in terms of each evaluation measure.

Evaluation measure	F_F	Critical value ($\alpha = 0.05$)
Execution time	66.011	2.492
Hamming loss	5.437	
Multilabel accuracy	7.153	
Subset accuracy	4.421	

(1) As Figure 7 shows, the multilabel learning and classification process is significantly accelerated by the feature selection process. In particular, the multilabel classification with SL and 3L is completed significantly faster than No, indicating the superiority of the proposed approach

(2) Focusing on the average rank of AL and No in Figure 7, the advantage of multilabel feature selection from the viewpoint of the execution time is insignificant, indicating that the merit given by feature selection process on the execution time can disappear owing to a large number of labels

(3) As Figure 7 shows, the feature subset selected by AL is able to deliver a statistically similar classification performance to the baseline performance No. This means that the dimensionality of the input space can be reduced to accelerate the multilabel learning process without degrading the predictive performance

(4) The feature subset selected by the proposed methods based on the label subset selection such as 3L and 5L is able to deliver a comparable classification performance to the classifier if a moderate number of labels are considered for evaluating the importance of features

(5) A notable exception can be observed from the experimental results of SL, which considers only one label for the feature scoring process. However, it also gives a statistically better performance than No in the experiments involving Hamming loss and a comparable performance in the experiments involving multilabel accuracy and subset accuracy

(6) Surprisingly, if a moderate number of labels are considered from the feature scoring process like 3L or 5L, the feature subset gives statistically better discriminating power than the baseline performance given by No. For example, in the experiments involving Hamming loss, as shown in Table 3, 3L gives a better Hamming loss performance than No on 85% of multilabel datasets

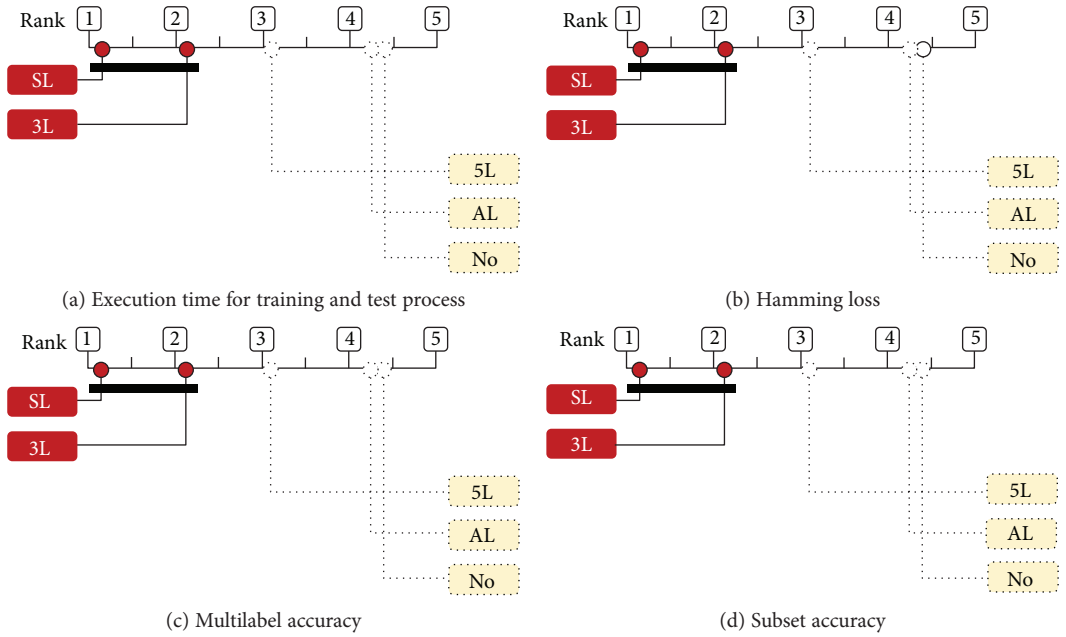
(7) Furthermore, based on the comparison to the multilabel classification performance given by No, the feature subset selected by 3L gives a better Hamming loss performance on 70% of multilabel datasets. This tendency can be observed again from the experiments involving multilabel accuracy based on 5L as it gives a better performance on 80% of datasets

In summary, the experimental results show that the proposed method based on the label subset selection strategy achieves a significantly better execution time than the baseline multilabel setting No and conventional multilabel learning with feature selection AL, indicating that the proposed method is able to accelerate the multilabel learning process. Furthermore, the feature subset selected by the proposed method, such as 3L and 5L, yields a similar classification performance compared to the other methods. Because the proposed method has a lower execution time compared to the other methods, this means that the proposed method is able to quickly identify the important feature subset, without degrading the multilabel classification performance.

Finally, we conducted additional experiments to validate the scalability and efficiency of the proposed method. For this purpose, we employed the Delicious dataset, which is composed of a large number of patterns and labels [3]. Specifically, the Delicious dataset was extracted from the del.icio.us social bookmarking site where textual patterns and associated labels represent web pages and relevant tags. This dataset is composed of 16,105 patterns, 500 features, and 983 labels from 15,806 unique label subsets. To demonstrate the superiority of the proposed method, we employed MLCFS [19] and PPT + RF [24]. In this experiment, we regard 3L as the proposed method because it performs better than SL, 5L, and AL, as shown in Figure 7. Table 7 represents the experimental results of three multilabel feature selection methods, including the proposed method. The experimental results indicate that the proposed method outputs the final feature subset much faster than the compared methods with similar multilabel classification performances in terms of Hamming loss, multilabel accuracy, and subset accuracy.

5. Conclusion

In this study, we proposed an efficient multilabel feature selection method to achieve scalable multilabel learning when the number of labels is large. Because the computational load of the multilabel learning process increases with the increasing number of features in the input data, the proposed method accelerates the multilabel learning process by selecting important features to reduce the dimensionality of features. In addition, with regard to the multiple labels considered for the feature scoring process, we demonstrated that the feature selection process itself can be accelerated for further acceleration of the multilabel learning process. Furthermore, empirical experiments on 20 multilabel datasets showed that the multilabel learning process can be boosted without deteriorating the discriminating power of the multilabel classifier.



(a) Execution time for training and test process

(b) Hamming loss

(c) Multilabel accuracy

(d) Subset accuracy

Figure 7: Bonferroni-Dunn test results of five comparing methods with four evaluation measures. Methods not connected with the best method in the CD diagram are considered to have significantly different performance (significance level $\alpha = 0.05$). This is reproduced from Lee et al. (2017) (under the Creative Commons Attribution License/public domain).

Table 7: Comparison results of proposed method, MLCFS, and PPT + RF on the Delicious dataset.

Methods	Execution time	Hamming loss	Multilabel accuracy	Subset accuracy
Proposed method (3L)	**26.6326 ± 0.9547** ·	0.0201 ± 0.0002	0.0301 ± 0.0002	0.0001 ± 0.0001
MLCFS	144.0414 ± 13.3807	0.0201 ± 0.0002	0.0304 ± 0.0043	0.0001 ± 0.0002
PPT + RF	1556.1397 ± 30.1202	0.0201 ± 0.0002	0.0301 ± 0.0054	0.0002 ± 0.0003

Future research directions include scalability against a large number of training examples. Although this can be achieved by a multilabel classification approach using distributed computing [49], the performance should be tested empirically to validate the potential. In addition, we will investigate the multilabel learning performance with respect to the label selection strategy. Our experiments indicate that the feature subset selected by the proposed method can possibly deliver a better discriminating capability, despite only a part of the labels in a given label set being considered for the feature scoring process. Because this was an unexpected result, as the primary goal of this study was the acceleration of the multilabel learning process, we would like to investigate this issue more thoroughly in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

Both authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science & ICT (MSIT, Korea) (NRF-2016R1C1B1014774).

References

[1] J. Paulin, A. Calinescu, and M. Wooldridge, "Agent-based modeling for complex financial systems," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 74–82, 2018.

[2] G. Le Moal, G. Moraru, P. Véron, P. Rabaté, and M. Douilly, "Feature selection for complex systems monitoring: an application using data fusion," in *CCCA12*, pp. 1–6, Marseilles, France, December 2012.

[3] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, pp. 30–44, Antwerp, Belgium, 2008.

[4] M. Zanin, E. Menasalvas, S. Boccaletti, and P. Sousa, "Feature selection in the reconstruction of complex network representations of spectral data," *PLoS ONE*, vol. 8, no. 8, p. e72045, 2013.

[5] J. Lee, J. Chae, and D.-W. Kim, "Effective music searching approach based on tag combination by exploiting prototypical

acoustic content," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 6065–6077, 2017.

[6] T. Pfingsten, D. J. L. Herrmann, T. Schnitzler, A. Feustel, and B. Scholkopf, "Feature selection for troubleshooting in complex assembly lines," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 465–469, 2007.

[7] T. Rault, A. Bouabdallah, Y. Challal, and F. Marin, "A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications," *Pervasive and Mobile Computing*, vol. 37, pp. 23–44, 2017.

[8] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[9] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang, "Document transformation for multi-label feature selection in text categorization," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 451–456, Omaha, NE, USA, October 2007.

[10] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14*, pp. 681–687, Vancouver, Canada, 2001.

[11] Y. Yu, W. Pedrycz, and D. Miao, "Multi-label classification by exploiting label correlations," *Expert Systems with Applications*, vol. 41, no. 6, pp. 2989–3004, 2014.

[12] M.-L. Zhang and L. Wu, "Lift: multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.

[13] F. Li, D. Miao, and W. Pedrycz, "Granular multi-label feature selection based on mutual information," *Pattern Recognition*, vol. 67, pp. 410–423, 2017.

[14] J. Lee and D.-W. Kim, "SCLS: multi-label feature selection based on scalable criterion for large label set," *Pattern Recognition*, vol. 66, pp. 342–352, 2017.

[15] J. Lee and D.-W. Kim, "Fast multi-label feature selection based on information-theoretic feature ranking," *Pattern Recognition*, vol. 48, no. 9, pp. 2761–2771, 2015.

[16] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[17] N. Spolaôr, M. C. Monard, G. Tsoumakas, and H. D. Lee, "A systematic review of multi-label feature selection and a new method based on label construction," *Neurocomputing*, vol. 180, no. 1, pp. 3–15, 2016.

[18] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "A comparison of multi-label feature selection methods using the problem transformation approach," *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151, 2013.

[19] S. Jungjit, M. Michaelis, A. A. Freitas, and J. Cinatl, "Two extensions to multi-label correlation-based feature selection: a case study in bioinformatics," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1519–1524, Manchester, UK, October 2013.

[20] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with naïve Bayes," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5432–5435, 2009.

[21] J. Read, "A pruned problem transformation method for multi-label classification," in *Proc. 2008 New Zealand Computer Science Research Student Conference*, pp. 143–150, Christchurch, New Zealand, April 2008.

[22] G. Doquire and M. Verleysen, "Mutual information-based feature selection for multilabel classification," *Neurocomputing*, vol. 122, pp. 148–155, 2013.

[23] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of Relief F and RReliefF," *Machine Learning*, vol. 53, no. 1/2, pp. 23–69, 2003.

[24] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 57–78, 2018.

[25] O. Reyes, C. Morell, and S. Ventura, "Scalable extensions of the relieff algorithm for weighting and selecting features on the multi-label learning context," *Neurocomputing*, vol. 161, pp. 168–182, 2015.

[26] Y. Sun, A. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

[27] Q. Gu, Z. Li, and J. Han, "Correlated multi-label feature selection," in *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, pp. 1087–1096, Glasgow, Scotland, UK, October 2011.

[28] X. Kong and P. S. Yu, "gMLC: a multi-label feature selection framework for graph classification," *Knowledge and Information Systems*, vol. 31, no. 2, pp. 281–305, 2012.

[29] J. Lee and D.-W. Kim, "Feature selection for multi-label classification using multivariate mutual information," *Pattern Recognition Letters*, vol. 34, no. 3, pp. 349–357, 2013.

[30] Y. Lin, Q. Hu, J. Liu, and J. Duan, "Multi-label feature selection based on max-dependency and min-redundancy," *Neurocomputing*, vol. 168, pp. 92–103, 2015.

[31] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint ℓ2,1-norms minimization," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1813–1821, 2010.

[32] S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proc. 21th Int. Joint Conf. Artificial Intelligence*, pp. 1077–1082, Pasadena, USA, July 2009.

[33] B. Qian and I. Davidson, "Semi-supervised dimension reduction for multi-label classification," in *Proc. 24th AAAI Conf. Artificial Intelligence*, pp. 569–574, Atlanta, USA, July 2010.

[34] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label reliefF and F-statistic feature selections for image annotation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2352–2359, Providence, RI, USA, June 2012.

[35] J. Lee and D.-W. Kim, "Memetic feature selection algorithm for multi-label classification," *Information Sciences*, vol. 293, pp. 80–96, 2015.

[36] J. Lee and D.-W. Kim, "Mutual information-based multi-label feature selection using interaction information," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2013–2025, 2015.

[37] J. Lee, H. Lim, and D.-W. Kim, "Approximating mutual information for multi-label feature selection," *Electronics Letters*, vol. 48, no. 15, pp. 929-930, 2012.

[38] H. Lim, J. Lee, and D.-W. Kim, "Multi-label learning using mathematical programming," *IEICE Transactions on Information and Systems*, vol. E98.D, no. 1, pp. 197–200, 2015.

[39] J. Lee, W. Seo, and D. W. Kim, "Effective evolutionary multilabel feature selection under a budget constraint," *Complexity*, vol. 2018, Article ID 3241489, 14 pages, 2018.

[40] J. Lee and D.-W. Kim, "Efficient multi-label feature selection using entropy-based label selection," *Entropy*, vol. 18, no. 11, p. 405, 2016.

[41] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive bayes classification," *Information Sciences*, vol. 179, no. 19, pp. 3218–3229, 2009.

[42] I. Pillai, G. Fumera, and F. Roli, "Designing multi-label classifiers that maximize f measures: state of the art," *Pattern Recognition*, vol. 61, pp. 394–404, 2017.

[43] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[44] T. Cover and J. Thomas, *Elements of Information Theory*, vol. 6, Wiley Online Library, New York, 1991.

[45] A. Cano, J. M. Luna, E. L. Gibaja, and S. Ventura, "Laim discretization for multi-label data," *Information Sciences*, vol. 330, pp. 370–384, 2016.

[46] Y. Yang and S. Gopal, "Multilabel classification with meta-level features in a learning-to-rank framework," *Machine Learning*, vol. 88, no. 1-2, pp. 47–68, 2012.

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.

[48] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.

[49] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed nearest neighbor classification for large-scale multi-label data on spark," *Future Generation Computer Systems*, vol. 87, pp. 66–82, 2018.

Hindawi
Complexity
Volume 2018, Article ID 6531203, 11 pages
https://doi.org/10.1155/2018/6531203

Research Article

On the Impact of Labeled Sample Selection in Semisupervised Learning for Complex Visual Recognition Tasks

Eftychios Protopapadakis,[1] Athanasios Voulodimos ,[2] and Anastasios Doulamis[1,3]

[1]*National Technical University of Athens, Zografou, 15780 Athens, Greece*
[2]*Department of Informatics and Computer Engineering, University of West Attica, Egaleo, 12243 Athens, Greece*
[3]*Institute of Communication and Computer Systems (ICCS), Zografou 15773, Athens, Greece*

Correspondence should be addressed to Athanasios Voulodimos; thanosv@mail.ntua.gr

Received 17 March 2018; Accepted 2 August 2018; Published 23 September 2018

Academic Editor: Ireneusz Czarnowski

One of the most important aspects in semisupervised learning is training set creation among a limited amount of labeled data in such a way as to maximize the representational capability and efficacy of the learning framework. In this paper, we scrutinize the effectiveness of different labeled sample selection approaches for training set creation, to be used in semisupervised learning approaches for complex visual pattern recognition problems. We propose and explore a variety of combinatory sampling approaches that are based on sparse representative instances selection (SMRS), OPTICS algorithm, k-means clustering algorithm, and random selection. These approaches are explored in the context of four semisupervised learning techniques, i.e., graph-based approaches (harmonic functions and anchor graph), low-density separation, and smoothness-based multiple regressors, and evaluated in two real-world challenging computer vision applications: image-based concrete defect recognition on tunnel surfaces and video-based activity recognition for industrial workflow monitoring.

1. Introduction

The proliferation of data generated in today's industry and economy raises the expectations for approaching towards the solutions of data-driven problems through state-of-the-art machine learning and data science techniques. One of the obstacles towards this direction, especially apparent in complex real-world applications, is the insufficient availability of ground truth, which is necessary for training and fine-tuning supervised machine learning (including deep learning) models. In this context, semisupervised learning (SSL) appears as an interesting and effective paradigm. Semisupervised learning approaches make use of both labeled and unlabeled data to create a suitable learning model given a specific problem (usually a classification problem) and related constraints. The acquisition of labeled data, for most learning problems, often requires a skilled human agent (e.g., to annotate background in an image, segment, and label video sequences for action recognition) or a physical experiment (e.g., determining the 3D structure of a protein). The cost associated with the labeling process, thus, may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, SSL can be of great practical value.

One major advantage is the easy implementation on existing techniques; SSL can be directly or indirectly incorporated in any machine-learning task. Semisupervised SVMs approaches are a classical example of direct usage of SSL assumptions into the minimization function [1]. Indirect utilization of SSL can be found in multiobjective optimization (MOO) frameworks [2, 3]. In MOO, we have multiple fitness evaluation functions; many of them are based on SSL assumptions. Then, from a large pool of possible solution, we peak those over the Pareto front. Thus, SSL is involved in the best individual selection procedure.

In real life, there are several fields of SSL testing, assuming that there is data availability. The work of [4] evaluates the foundation piles structural condition using graph-based approaches. A scalable graph-based approach was utilized in [5] for the initialization of a maritime surveillance system.

The SSL cluster assumption was used in [6] for the initialization of a fall detection system for elderly people. A self-training approach is adopted in [7] for industrial workflow surveillance purposes in an automobile manufacturer production line. In cultural heritage, SSL has been leveraged in [8] to develop image retrieval schemes suitable to user preferences [9].

Regarding the limitations and requirements pertaining to the selection of labeled data in SSL, there is a set of desirable properties that the utilized data should have: Firstly, representative samples are needed. The labeled samples should be able to describe (or reproduce) the original data set in the best possible way. Secondly, at least one sample per classification category is required, so that model can be able to adjust to the class properties. Finally, the existence of outliers should be considered, given that most data sets contain outliers which could lead to poor performance especially when used as labeled data (all by themselves).

In this paper, we provide a deeper insight on the effectiveness of different data sampling approaches for labeled dataset creation to be used in SSL. The data sampling approaches explored are based on sampling techniques including KenStone algorithm [10], sparse representative modeling selection (SMRS) [11], Ordering Points To Identify the Clustering Structure (OPTICS) algorithm output-based approach [12], and k-means [13] centroids and random selection. Each of the described data selection approaches is scrutinized with respect to different SSL techniques, including low-density separation [14], harmonic functions [15], pseudo-Laplacian graph regularization [16], and semisupervised regressors [17]. Our contribution lies in the investigation of two aspects on the SSL field: how can we interpret the term "few data" and how we select them in an effective manner. A preliminary version of the work presented in this paper appeared in [18]. The present work scrutinizes additional SSL techniques. Furthermore, the experimental evaluation is more thorough and extensive, including a more formal method of cluster determination, additional experiments with a different visual recognition task and dataset, and supplementary comparisons with supervised techniques as well.

The typical data selection approach in several SSL techniques, including the aforementioned ones, is, to our knowledge, the random selection of the training set. Usually, a small portion of the data, i.e., less than 40% is selected (and considered labeled); as the amount of available data increases, the fraction of the required labeled instances decreases [19, 20]. At this point, two problems become apparent: (i) the number of selected instances is subjective to the expert's view and (ii) random selection does not guarantee that the major sources of variance appear in the labeled data set. In this paper, we adopt data-driven approaches for data sampling, trying to identify appropriate sampling selection techniques for SSL models.

The remainder of this paper is structured as follows: In Section 2, we first briefly present four known techniques used in the bibliography for clustering and/or sampling, which we then combine to derive seven data selection approaches. The efficacy of these approaches as labeled data generators for the SSL techniques presented in Section 3 will be evaluated in the context of two complex multiclass visual classification problems, i.e., defect recognition on concrete tunnel surfaces and activity recognition in industrial workflow monitoring. The related experimental results are presented and discussed in Section 4. Finally, Section 5 concludes the paper with a summary of findings.

2. Labeled Sample Selection Approaches for Training Data Set Creation

Given a set of feature values for a data sample, a two-step process is adopted in the analysis conducted in this study. The first step involves data sampling, i.e., the selection of the most descriptive representatives in the available data set. The second step employs popular data mining algorithms; i.e., predictive models are trained over the descriptive subsets of the previous step.

The main purpose of data sampling is the selection of appropriate representative samples to provide a good training set and, thus, improve the classification performance of predictive models. In this section, we present seven (7) data sampling approaches, which are based on the combination or adaptation of four (4) main known sampling techniques [21].

2.1. Main Techniques. The most important factor in data selection is the definition of distance function. For any two given data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{m}}$ let $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denote the distance between them. Let $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ be a symmetric matrix. The distance measure defined as

$$d_{\boldsymbol{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{A} (\boldsymbol{x}_i - \boldsymbol{x}_j)}. \quad (1)$$

Most of the proposed approaches are based on the Euclidean distance (i.e., $\boldsymbol{A} = \boldsymbol{I}$). Sampling algorithms are used over the entire data set $\boldsymbol{\mathcal{X}}$ and create a new set, $\boldsymbol{\mathcal{X}}_r \subset \boldsymbol{\mathcal{X}}$, according to the data relationships, as described by the distance among them. In this study, we need at least one observation from every possible class.

2.1.1. OPTICS Algorithm. Ordering Points to Identify the Clustering Structure (OPTICS) is an algorithm for finding density-based clusters in spatial data [22], i.e., detect meaningful clusters in data of varying density. The points of the database are (linearly) ordered such that points which are spatially closest become neighbors in the ordering.

OPTICS requires two parameters: the maximum distance (radius) to consider (ε) and the number of points required to form a cluster (*MinPts*) MinPts. A point p is a core point if at least MinPts points are found within its ε-neighborhood, $N_\varepsilon(p)$. Once the initial clustering is formed, we may proceed with any sampling approach (e.g., random selection among clusters).

2.1.2. k-Means Algorithm. k-means clustering [13] aims to partition n observations into k clusters, such that each observation is assigned to the cluster it is most similar to (with the cluster centroid serving as a prototype of the cluster). It is a classical approach that can be implemented in many ways

and for various distance metrics. The main drawback is that the number of clusters should be known a priori.

2.1.3. Sparse Modeling for Representative Selection. Sparse modeling representative selection (SMRS) focuses on the identification of representative objects through the solution of the following optimization problem [11]:

$$\begin{aligned} \min \quad & \lambda \|\boldsymbol{C}\|_{1,q} + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 \\ \text{s.t.} \quad & \mathbf{1}^T\boldsymbol{C} = \mathbf{1}^T, \end{aligned} \tag{2}$$

where $\boldsymbol{X}$ and $\boldsymbol{C}$ refer to data points and coefficient matrix, respectively. This optimization problem can also be viewed as a compression scheme, where we want to choose a few representatives that can reconstruct the available data set.

2.1.4. Kennard–Stone Algorithm. Using the classic KenStone algorithm, we can cover the experimental area in a uniform way, since it provides a flat data distribution. The algorithm's main idea is that to select the next sample, it opts for the sample whose distance to those that have been previously chosen (called calibration samples) is the greatest.

Therefore, among all possible points, the algorithm selects the point which is furthest from those already selected and adds it to the set of calibration points. To this end, the distance is calculated between each candidate point $\boldsymbol{x}_0$ to each point $\boldsymbol{x}$ which has already been selected. In the sequel, we determine which one is the smallest, i.e., $\min_i d(\boldsymbol{x}, \boldsymbol{x}_0)$. Among these, we choose the point for which the distance is maximal:

$$d_{\text{selected}} = \max_{i_0} \left(\min_i d(\boldsymbol{x}_i, \boldsymbol{x}_0) \right). \tag{3}$$

2.2. Combinatory Sampling Approaches. The primary goal of sampling approaches is the removal of redundant and uninformative data. Using the algorithms described earlier in Section 2.1 as a basis, we propose six (6) combinatory sampling approaches. A brief description of each one, along with the baseline random selection method, follows:

(i) *OPTICS extrema*: after employing the OPTICS algorithm on the entire data set, the calculated reachability distances are plotted in the same order as data were processed. Over the generated waveform, we locate local maxima and minima. All the identified extrema cases are considered as labeled instances and the rest as unlabeled. This approach results in a very limited training set.

(ii) *Sparse modeling representative selection (SMRS)*: the SMRS technique is employed over the entire data set, resulting in a very limited training set, although larger than the one obtained with OPTICS. In contrast to OPTICS, the selected points are located only on the exterior cell of the available data volume.

(iii) *Combination of k-means and SMRS (k-means SMRS)*: we first divide the set into k subclusters. For each subcluster, we run the SMRS algorithm to get the representative samples among each subcluster. As such, the outcome provides points surrounding each subcluster. The number of clusters, k, was defined using the Silhouette score for all k values, $k \in [2, \mathcal{u} + 4]$, where $\mathcal{u}$ is a heuristic approach estimating the number of clusters, defined as $\mathcal{u} = \lceil \sqrt{n/2} \rceil$, and n denotes the number of available data instances (observations).

(iv) *Combination of OPTICS and SMRS (OPTICS-SMRS)*: SMRS is performed to the subclusters obtained through the OPTICS algorithm. This approach is similar to the work of [19]. A subset is created of representative samples from each subcluster obtained by OPTICS algorithm. The minimum number of data within a cluster, required by OPTICS, was defined as MinPts = min $(\lfloor n/k \rfloor, 8)$.

(v) *Kennard and Stone (KenStone) sampling data points*: after executing the KenStone algorithm, we have data entries spanning uniformly the entire data space.

(vi) *Random selection*: a random selection that picks $p\%$ of the available data as training data, this is the baseline data selection method used in the context of most SSL techniques.

(vii) *Improved random selection*: an alternative approach is the creation of k clusters (using k-means) and a random selection of n_k samples from each cluster (k-means random). It is an improvement of random selection, without involving any advanced techniques. Similar instances are likely to be clustered together. Thus, the few random samples from each cluster are expected to provide adequate information over the data set.

All of the proposed approaches are applied over all available data, labeled or not. As such, it is possible for many of the selected training data to be unlabeled. In that case, an expert would be summoned to annotate the selected data, as would have been the case in any annotation attempt. However, in this case, the annotation effort will be less considerable compared to traditional supervised approaches, which use a significantly higher percentage of the available data for training purposes.

3. Semisupervised Learning Techniques

In this work, four of the most popular types of SSL techniques will be considered: two graph-based approaches, along with low-density separation, and multiple smoothness assumption-related regressors.

3.1. Graph-Based Approaches. Graph-based semisupervised methods define a graph over the entire data set, $\boldsymbol{X} = \boldsymbol{X}_{\text{L}} \cup \boldsymbol{X}_{\text{U}}$, where $\boldsymbol{X}_{\text{L}} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_l, \boldsymbol{y}_l)\}$ is the labeled data set

and $\boldsymbol{X}_{\mathrm{U}} = \{\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+u}\}$ the unlabeled data set. Feature vectors, $\boldsymbol{x}_i \in \mathbb{R}^m$, $i = 1, \ldots, l+u$, are available for all the observations and $\boldsymbol{y}_i \in \mathbb{R}^C$, $i = 1, \ldots, l$ are the corresponding classes of the labeled ones, in a vector form; C denotes the available classes.

The nodes represent the labeled and unlabeled examples in the dataset; edges reflect the similarity among examples. In order to quantify the edges (i.e., assign a similarity value), an adjacency matrix $\boldsymbol{\mathcal{A}}$ is calculated, where

$$\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ close to } \boldsymbol{x}_j \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

Practically, each label is only connected to its k closest labels, so that $\sum_{j=0}^{n} \mathcal{A}_{ij} = k$. The information of the labeled nodes propagates to the unlabeled nodes via paths defined on existing edges provided by $\boldsymbol{\mathcal{A}}$.

Graph methods are nonparametric, discriminative, and transductive in nature. Intuitively speaking, in a graph that various data points are connected, the greater the similarity, the greater the probability of having similar labels. Thus, the information (of labels) propagates from the labeled points to the unlabeled ones. These methods usually assume label smoothness over the graph. That is, if two instances are connected by a strong edge, their labels tend to be the same.

3.1.1. Harmonic Functions. An indicative paradigm of graph-based SSL is the harmonic function approach [23]. This approach estimates a function f on the graph which satisfies two conditions. Firstly, f has the same values as given labels on the labeled data, i.e., $f(\boldsymbol{x}_i) = \boldsymbol{y}_i$, $i = 1, \ldots, l$. Secondly, f satisfies the weighted average property on the unlabeled data:

$$f(\boldsymbol{x}_j) = \frac{\sum_{k=1}^{l+u} w_{jk} f(\boldsymbol{x}_j)}{\sum_{k=1}^{l+u} w_{jk} f}, \quad j = l+1, \ldots, l+u, \tag{5}$$

where w_{ij} denotes the edge weight. Those two conditions lead to the following problem:

$$\begin{aligned} \min_{f: f(\boldsymbol{x}) \in \mathbb{R}} \quad & \sum_{i,j=1}^{l+u} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2 \\ \text{s.t.} \quad & f(\boldsymbol{x}_i) = \boldsymbol{y}_i, \; i = 1, \ldots, l. \end{aligned} \tag{6}$$

The problem has an explicit solution, which allows a soft label estimation for all the edges of the graph, i.e., investigated cases.

3.1.2. Anchor Graph. Anchor graph estimates a labeling prediction function $f : \mathbb{R}^m \to \mathbb{R}$ defined on the samples of $\boldsymbol{X}$; by using a subset $\boldsymbol{\mathcal{U}} = \{\boldsymbol{u}_k\}_k^p \subset \boldsymbol{X}_L$ of the labeled data, the label prediction function can be expressed as a convex combination [16]:

$$f(\boldsymbol{x}_i) = \sum_{k=1}^{p} Z_{ik} \cdot g(\boldsymbol{u}_k), \tag{7}$$

where Z_{ik} denotes sample-adaptive weights, which must satisfy the constraints $\sum_{k=1}^{l} Z_{ik} = 1$ and $Z_{ik} \geq 0$ (convex combination constraints). By defining vectors $\boldsymbol{g}$ and $\boldsymbol{a}$, respectively, as $\boldsymbol{g} = [g(\boldsymbol{f}_1), \ldots, g(\boldsymbol{f}_n)]^{\boldsymbol{T}}$ and $\boldsymbol{a} = [g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_p)]^{\boldsymbol{T}}$, (7) can be rewritten as $\boldsymbol{g} = \boldsymbol{Z}\boldsymbol{\alpha}$ where $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$.

The designing of matrix $\boldsymbol{Z}$, which measures the underlying relationship between the samples of $\boldsymbol{X}_U$ and samples $\boldsymbol{X}_L$, is based on weight optimization; i.e., nonparametric regression. Thus, the reconstruction for any data point is a convex combination of its closest representative samples.

Nevertheless, the creation of matrix $\boldsymbol{Z}$ is not sufficient, as it does not assure a smooth function $\boldsymbol{g}$. There is always the possibility of inconsistencies in segmentation, i.e., different samples with almost identical attributes belong to different classes. In order to deal with such cases, the following SSL framework is employed:

$$\min_{\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_c]} \quad \mathcal{Q}(\boldsymbol{A}) = \frac{1}{2} \|\boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Y}\|_F^2 + \frac{\gamma}{2} \operatorname{trace}\left(\boldsymbol{A}^{\boldsymbol{T}} \widehat{\boldsymbol{L}} \boldsymbol{A}\right), \tag{8}$$

where $\widehat{\boldsymbol{L}} = \boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{Z}$ is a memory-wise and computationally tractable alternative of the Laplacian matrix $\boldsymbol{L}$. Matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_c] \in \mathbb{R}^{p \times c}$ is the soft label matrix for the representative samples, in which each column vector accounts for a class. The matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_c] \in \mathbb{R}^{n \times c}$ is a class indicator matrix on ambiguously labeled samples with $Y_{ij} = 1$ if the label l_i of sample i is equal to j and $Y_{ij} = 0$ otherwise.

The Laplacian matrix $\boldsymbol{L}$ is calculated as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal degree matrix and $\boldsymbol{W}$ is approximated as $\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^T$. Matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{p \times p}$ is defined as $\boldsymbol{\Lambda} = \sum_{i=1}^{n} Z_{ik}$. The solution of (8) has the form

$$\boldsymbol{A}^* = \left(\boldsymbol{Z}^T \boldsymbol{Z} + \gamma \widehat{\boldsymbol{L}}\right) \square^T \boldsymbol{Y}. \tag{9}$$

Each sample label is, then, given by

$$\widehat{l}_i = \arg \max_{j \in \{1, \ldots, c\}} \frac{\boldsymbol{Z}_i a_j}{\lambda_j}, \tag{10}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{1 \times p}$ denotes the i-th row of $\boldsymbol{Z}$, and the normalization factor $\lambda_j = \mathbf{1}^T \boldsymbol{Z}\, \boldsymbol{\alpha}_j$ balances skewed class distributions.

3.2. Low-Density Separation. The low-density separation assumption pushes the decision boundary in regions where there are few data points (labeled or unlabeled). The most common approach to achieving this goal is to use a maximum margin algorithm such as support vector machines. The method of maximizing the margin for unlabeled as well as labeled points is called the transductive SVM (TSVM). However, the corresponding problem is nonconvex and thus difficult to solve [24].

Low-density separation (LDS) is a combination of TSVMs [25], trained using gradient descend, and traditional SVMs using an appropriate kernel defined over a graph using SSL assumptions [14]. Like the SVM approach, the TSVM maximizes the class-separating margin.

The problem can be stated in the following form, which allows for a standard gradient-based approach:

$$\min_{\boldsymbol{w},b}\left[\frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{l} L^2\left(y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)\right) + C^*\sum_{j=l+1}^{l+u} L^*\left(\left|\boldsymbol{w}^T\boldsymbol{x}_j + b\right|\right)\right], \tag{11}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ is the parameter vector that specifies the orientation and scale of the decision boundary and $b \in \mathbb{R}$ is an offset parameter. The above formulation exploits both labeled $\boldsymbol{X}_{\mathrm{L}}$ and unlabeled $\boldsymbol{X}_{\mathrm{U}}$ data. Finally, let us denote as $L(t) = \max(0, 1-t)$ and $L^*(t) = \exp(-3t^2)$.

Such a formulation allows the use of a nonlinear kernel, calculated over a fully connected matrix, $\boldsymbol{W}$, which is formed as $w_{ij} = \exp(\rho - dist(i,j)) - 1$. Dijkstra's algorithm is employed to compute the shortest path lengths, $d_{\mathrm{SP}}(i,j)$ for all pairs of points. The matrix $\mathscr{D}$ of squared ρ-path distances is calculated for all pairs of points as

$$\mathscr{D}_{ij} = \left(\frac{1}{\rho}\log\left(1 + d_{\mathrm{SP}}(i,j)\right)\right)^2. \tag{12}$$

The final step towards the kernel's creation involves multidimensional scaling [23], or MDS, to find a Euclidean embedding of $\mathscr{D}^{\rho}$ (in order to obtain a positive definite kernel). The embedding found by the classical MDS are the eigenvectors corresponding to the positive eigenvalues $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = -\boldsymbol{H}\mathscr{D}^{\rho}\boldsymbol{H}$, where $H_{ij} = \delta_{ij} - 1/(l+u)$. The final representation of $\boldsymbol{x}_i$ is $x_{ik} = U_{ik}\sqrt{\lambda_k}$, $1 \le k \le p$.

3.3. Semisupervised Regression. The safe semisupervised regression (SAFER) approach [17] tries to learn a prediction from several semisupervised regressors. Specifically, let $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_b\}$ be multiple SSR predictions and $\boldsymbol{f}_0$ be the prediction of a direct supervised learner, where $\boldsymbol{f}_i \in \mathbb{R}^U$, $i = 1, \ldots, r$ and r refers to the number of regressors. Supposing there is no knowledge with regard to the reliabilities of learners, SAFER optimizes the performance gain of $g(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_b, \boldsymbol{f}_0)$ against $\boldsymbol{f}_0$, when the weights of SSR learners come from a convex set.

The problem lies in the solution of the following equation:

$$\max_{\boldsymbol{f}\in\mathbb{R}^U}\min_{\substack{\boldsymbol{\alpha}\in\\ \mathscr{M}}}\quad \sum_{i=1}^{r}\alpha_i\left(\|\boldsymbol{f}_0 - \boldsymbol{f}_i\|^2 - \|\boldsymbol{f} - \boldsymbol{f}_i\|^2\right), \tag{13}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_r]$, $\alpha_i \ge 0$, are the weights of individual regressors. Equation (12) is concave to $\boldsymbol{f}$ and convex to $\boldsymbol{\alpha}$. Thus, it is recognized as saddle-point convex-concave optimization [26].

4. Experimental Evaluation

We will hereby examine the applicability and effectiveness of each of the above-described data selection techniques for the SSL approaches presented. SSL is particularly useful in cases where there is limited availability of labeled data and/or the creation of appropriately sized labeled data sets requires a prohibitive amount of resources, as is the case in real-world visual classification problems. Two prominent examples of such applications are (a) automated image-based detection and classification of defects on concrete surfaces in the context of visual inspection of tunnels [27] and (b) human activity recognition from video, e.g., the monitoring of workflow in industrial assembly lines [28, 29].

MATLAB software has been used for the implementation of the proposed approaches. The SSL approaches code, i.e., Harmonic functions, Anchor graph, LDS, and SAFER, were provided by the corresponding authors of [14, 16, 17, 23]. OPTICS, KenStone, and SMRS as well as code implementations were provided by [11, 22, 30], respectively.

4.1. Defect Recognition on Tunnel Concrete Surfaces. The tunnel defect recognition dataset (henceforth referred to in this paper as the *Tunnel dataset*) consists of images acquired by a robot inside a tunnel of Egnatia Motorway, in Greece, in the context of ROBO-SPECT project [27]. Images were used for detecting and recognizing defects on the concrete surfaces. Raw captured tunnel and annotated ground truth images of resolution 600×900 pixels were provided. Figure 1 shows some examples from the Tunnel dataset displaying cracked areas on the concrete surface.

To represent each pixel, we use the same low-level feature extraction techniques as in [27]; in particular, each pixel p_{xy} is described by a feature vector $\boldsymbol{s}_{xy} = [s_{1,xy}, \ldots, s_{k,xy}]^T$, where s are scalars corresponding to the presence and magnitude of the low-level features detected at location (x, y). Figure 2 displays the extracted low-level features. Feature vectors along with the class labels of every pixel are used to form a data set. There are five different classes of defects: (1) crack, (2) staining, (3) spalling, (4) calcium leaching, and (5) unclassified.

We, hereby, briefly describe the features used to form vector $\boldsymbol{s}_{xy}$. First, we take the edges denoted by a pixel-wise multiplication of the Canny and Sobel operators. Secondly, frequency is calculated as $\mathscr{F}_I = \nabla^2 I$. Thirdly, we calculate the entropy in order to separate homogenous regions from textured ones. Texture was described using twelve Gabor filters with orientations 0°, 30°, 60°, and 90° and frequencies 0.0, 0.1, and 0.4. The Histogram of Oriented Gradients (HOG) was also calculated. By combining these features with the raw pixels' intensity, feature vector $\boldsymbol{s}_{xy}$ takes the form of a 1×17 vector containing visual information that characterizes each one of the image pixels.

A typical K-fold validation approach is adopted, resulting in eight (approximately) equal partitions, i.e., disjoint subsets, of the n observations. The training set size is limited at 3% of sample population, when random techniques and KenStone algorithm were applied.

4.2. Activity Recognition from Video for Industrial Workflow Recognition. Action or activity recognition from video is a very popular computer vision application. A significant application domain is automatic video surveillance, e.g., for safety, security, and quality assurance reasons. In this

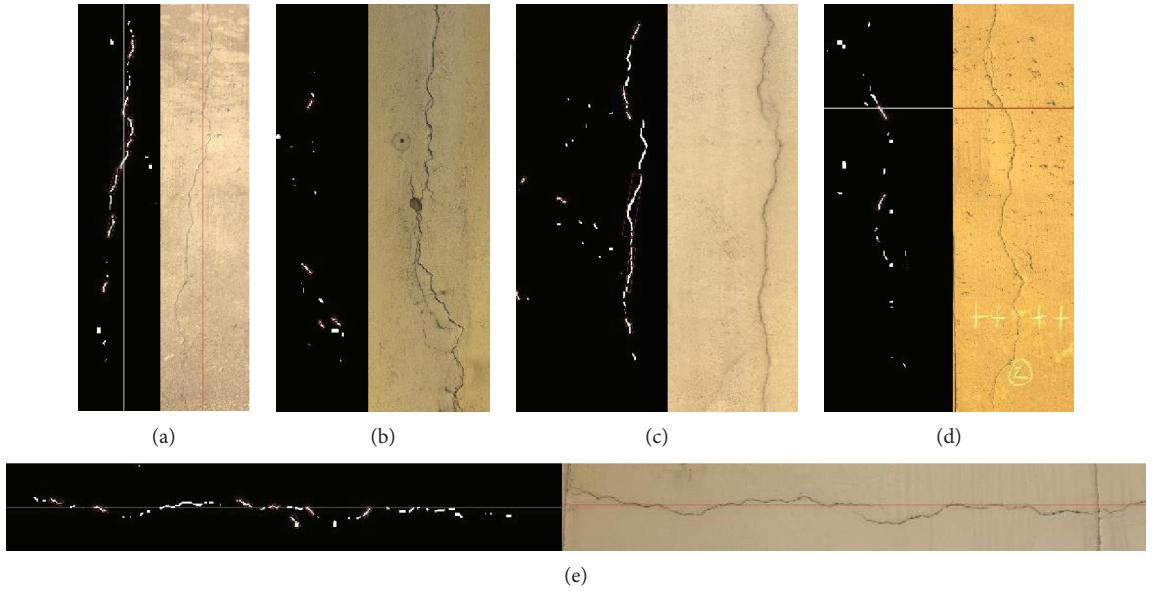


Figure 1: Examples of cracked areas from the Tunnel dataset.

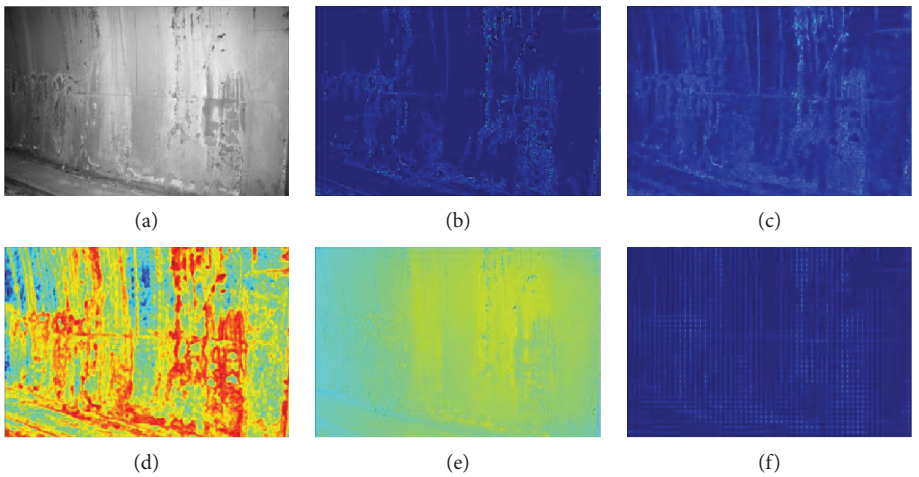


Figure 2: Illustration of the extracted low-level features in the Tunnel dataset: (a) original image, (b) edges, (c) frequency, (d) entropy, (e) texture, and (f) HOG.

experiment, we will make use of real-world video sequences from the surveillance camera of a major automobile manufacturer (NISSAN) [31], captured in the context of the SCOVIS EU project in the publicly available Workflow Recognition (WR) dataset [32].

The production cycle on the industrial line included tasks of picking several parts from racks and placing them on a designated cell some meters away, where welding took place. Each of the above tasks was regarded as a class of behavioral patterns that had to be recognized. The activities (tasks) we were aiming to model in the examined application are briefly the following:

(1) One worker picks part #1 from rack #1 and places it on the welding cell

(2) Two workers pick part #2a from rack #2 and place it on the welding cell

(3) Two workers pick part #2b from rack #3 and place it on the welding cell

(4) One worker picks up parts #3a and #3b from rack #4 and places them on the welding cell

(5) One worker picks up part #4 from rack #1 and places it on the welding cell

(6) Two workers pick up part #5 from rack #5 and place it on the welding cell

(7) Workers were idle or absent (null task)

The WR dataset includes twenty full cycles, each containing occurrences of the above tasks. Figure 3 depicts a typical example of an execution of Task 2. The visual classification problem in this case is to automatically recognize which task is executed at every time instance.

Figure 3: Indicative example of key-frames corresponding to the execution of a task (Task 2).

Table 1: Illustration of the training set data size per sampling approach (averages over all tests).

Row labels	KenStone	kmeansRandom	kmeansSMRS	OPTICS extrema	OPTICS-SMRS	Random	SMRS	Entire set
WR	156	181.25	422.37	289.75	532.39	156	23.62	5199
Tunnel	36.37	38	37.75	55	141.76	36.37	14.12	1200

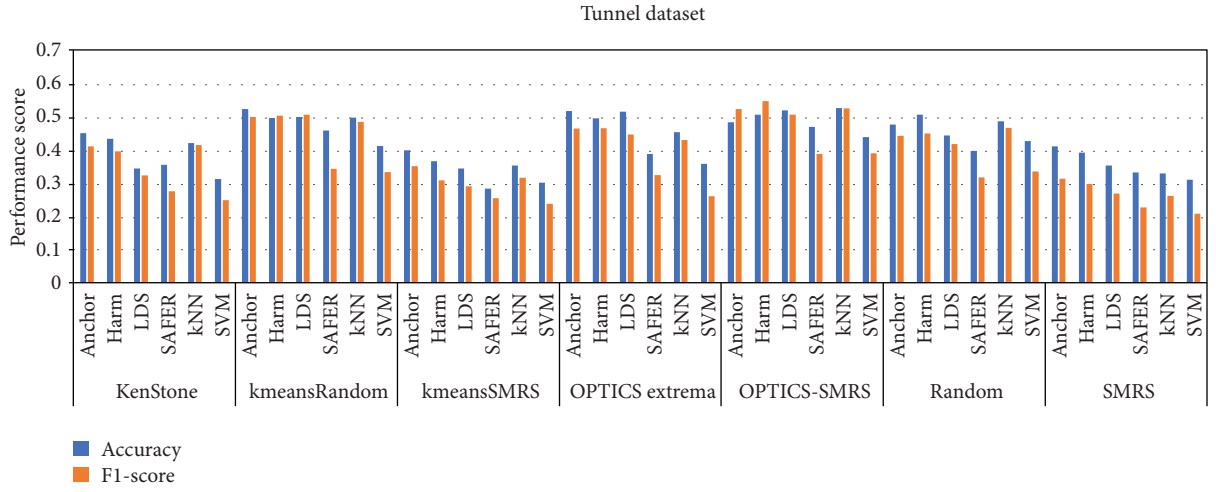


Figure 4: Performance scores for all data selection and SSL combinations (Tunnel dataset).

In all video segments, holistic features such as Pixel Change History (PCH) are used. These features remedy the drawbacks of local features, while also necessitating a far less tedious computational procedure for their extraction [33]. A very positive attribute of such representations is that they can easily capture the history of a task that is being executed. These images can then transform to a vector-based representation using the Zernike moments (up to sixth order, in our case) as applied in [33, 34]. The video features, once exported, had a two-dimensional matrix representation of the form $m \times l$, where m denotes the size of the $1 \times m$ vectors created using Zernike moments and l the number of such vectors.

4.3. Experimental Results. Each of the seven data sampling approaches described in Section 2.2 was paired with each of the four SSL techniques presented in Section 3 as well as two well-known supervised approaches, i.e., SVM and kNN, resulting in 42 combinations in total. Table 1 illustrates the training data set size generated in the case of each data selection approach applied for the two datasets. It is interesting to note here that the OPTICS-SMRS approach provides significantly more data than any other approach.

The classification results in terms of averaged accuracy and F-measure for each combination are depicted in Figure 4 for defect recognition (Tunnel dataset) and Figure 5 for activity recognition (WR dataset). At first look, it appears that among SSL techniques, it is harmonic functions that tend to provide higher accuracy rates, while concerning data sampling approaches, cluster-based selection

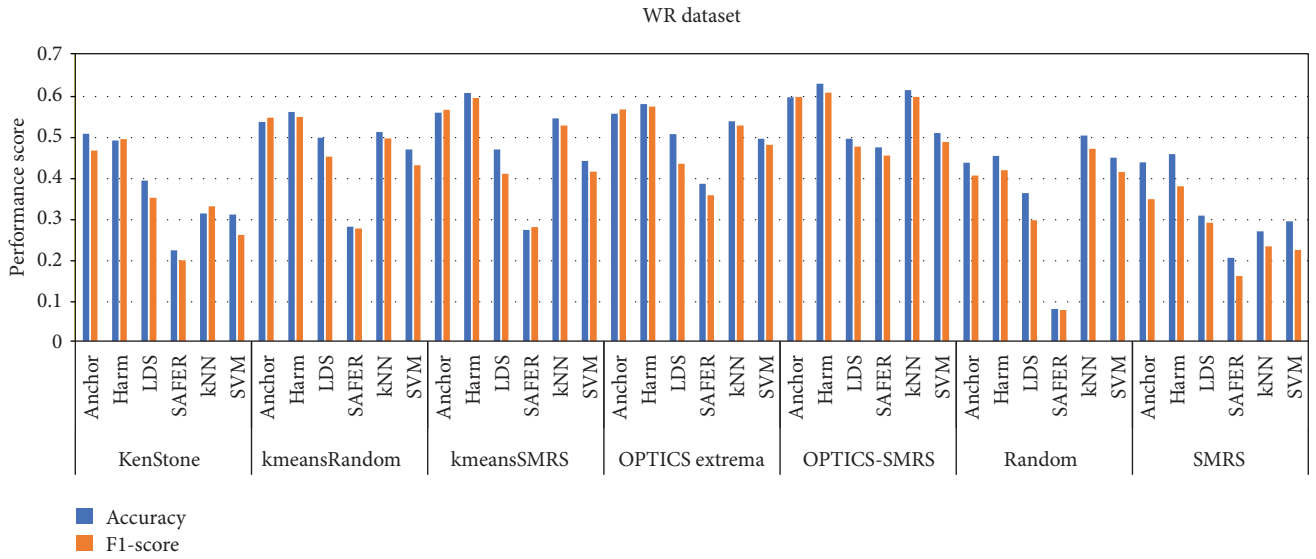


Figure 5: Performance scores for all data selection and SSL combinations (WR dataset).

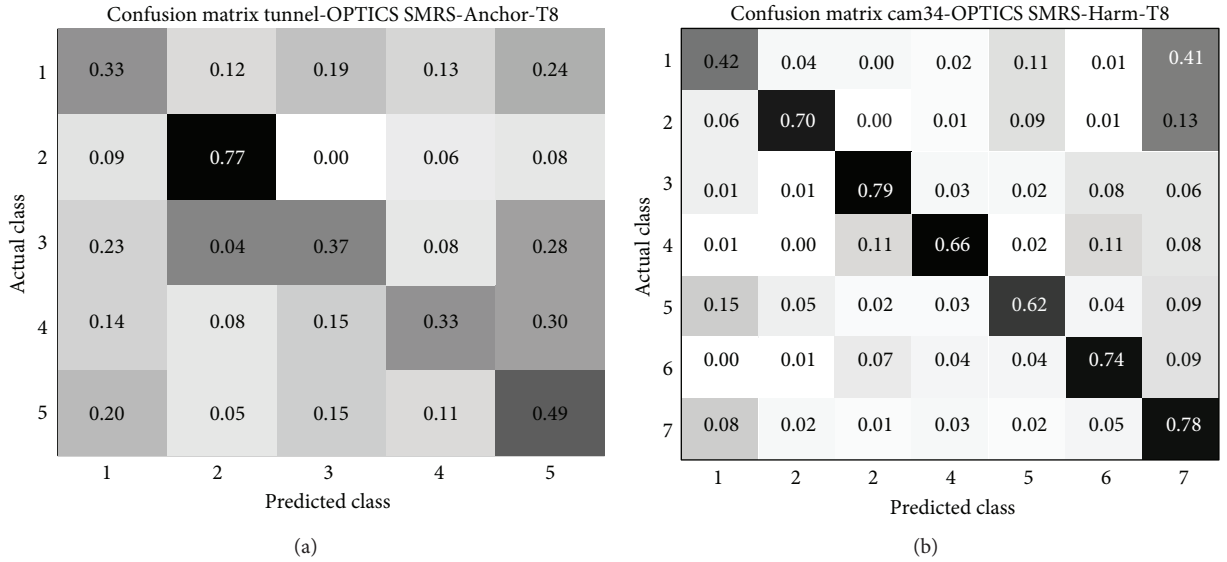


Figure 6: Confusion matrices for OPTICS-SMRS sampling in (a) Tunnel dataset, using anchor graph and (b) WR dataset, using harmonic functions.

(centroid or density-based) appears to give overall better results. Figure 6 provides an example confusion matrix for each visual recognition problem, acquired for OPTICS-SMRS data selection method.

Figure 4 illustrates the performance of the combinatory models in the tunnel surface defect recognition task. Cluster-based selection (OPTICS-SMRS followed by k-means random) appears to be the data selection techniques that lead to the best performance rates. Additionally, graph-based classifiers tend to perform better in most cases. The low performance scores for all the cases can be put down to the extremely challenging nature of the problem, as well as the feature quality; it is very likely for various defect types to have similar feature values when using low-level features [35].

Figure 5 illustrates the performance for the combinatory models in the WR dataset. Again, OPTICS-SMRS sampler appears to lead to the best performance rates, especially when using harmonic functions as SSL technique. It is interesting to note that, when using most of the proposed data selection techniques for training set creation, graph-based SSL techniques (harmonic functions and anchor graph) outperform not only the remaining SSL techniques but also the supervised methods examined, i.e., kNN and SVM. This can be explained by the lower number of training samples used compared to the usual training set sizes in such supervised learning methods.

4.4. Statistical Tests. In order to derive further conclusions regarding the results and the relative performance of the

Table 2: ANOVA results.

Source	Sum sq.	d.f.	Mean sq.	F	p value
Sampling	3.5488	6	0.5915	167.0981	0
Classifier	3.1569	5	0.6314	178.2768	0
Number of classes	0.2687	1	0.2687	75.9157	0
Sampling × classifier	0.3766	30	0.0126	3.5469	0
Sampling × num of classes	0.7855	6	0.1309	36.9865	0
Classifier × num of classes	0.4715	5	0.0943	26.6411	0
Error	2.1769	615	0.0035		
Total	10.7920	668			

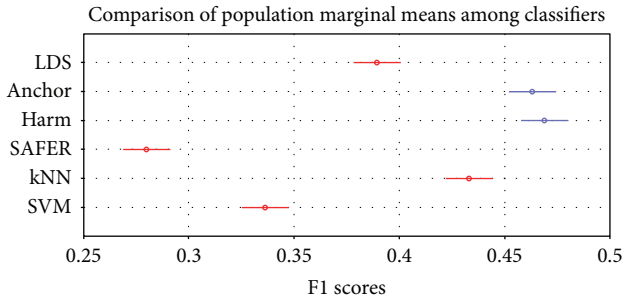


Figure 7: F1 scores by classification method.

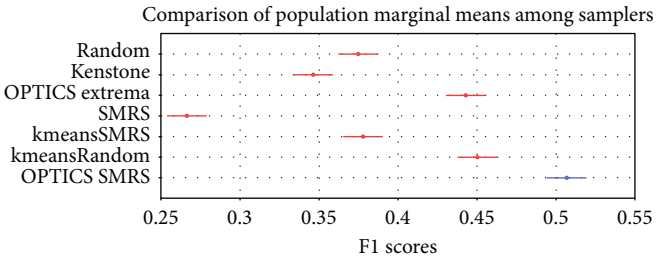


Figure 8: F1 scores by data selection method.

technique combinations explored, we performed an analysis of variance (ANOVA) on the F1 scores for the test samples. ANOVA permits the statistical evaluation of the effects of the two main design factors of this analysis (i.e., the sampling schemes and the SSL techniques). As shown in Table 2, both the sampling scheme and the choice of classifier are strongly significant for explaining variations in F1 scores. The dataset impact is also significant; i.e., performance variations should be expected in other datasets.

Apart from the above basic ANOVA results, we use the Tukey honest significant difference (HSD) post hoc test so as to derive conclusions about the best performing approaches, taking into account the statistical significance of the variations in the values of metrics presented. Figures 7 and 8 illustrate the results for the SSL techniques and the sampling schemes, respectively, for the entirety of experiments conducted.

As far as SSL techniques are concerned, harmonic functions and anchor graph appear to have a statistically significant superiority over all alternatives. The outcome verifies previous analysis outcomes (see Figures 4 and 5) suggesting that graph-based approaches result in better rates compared to the other SSL (or even supervised learning) alternatives (see Figure 7). The low overall performance scores in the comparison of learning techniques can be explained by the challenging nature of both examined problems as well as by the fact that all configurations have been taken into consideration including those yielding very low performance rates.

Finally, as regards data selection techniques, we observe that the OPTICS-based approach combined with SMRS creates training sets that lead to clearly the highest performance rates among all examined techniques, including the traditionally used random sampling. Furthermore, we can see that cluster-based samplers in general yield results that are at least as good as random sampling. On the other hand, SMRS alone provides results significantly worse than all competing schemes.

5. Conclusion

The creation of a training set of labeled data is of great importance for semisupervised learning methods. In this work, we explored the effectiveness of different data sampling approaches for labeled data generation to be used in SSL models in the context of complex real-world computer vision applications. We compared seven sampling approaches, some of which we proposed in this paper, all based on OPTICS, k-means, SMRS, and KenStone algorithm. The proposed data selection approaches were used to create labeled data sets to be used in the context of four SSL techniques, i.e., anchor graph, harmonic functions, low-density separation, and semisupervised regression. Extensive experiments were carried out in two different and very challenging real-world visual recognition scenarios: image-based concrete defect recognition on tunnel surfaces and video-based activity recognition for industrial workflow monitoring. The results indicate that SSL data selection schemes, using density-based clustering prior to sampling, such as a combination of OPTICS and SMRS algorithms, provide better performance results compared to traditional sampling approaches, such as random selection. Finally, as regards the SSL techniques studied, graph-based approaches (harmonic functions and anchor graph) appeared to have a statistically significant superiority for the two visual recognition problems examined.

Data Availability

The WR dataset is publicly available as described in [30]. The Tunnel dataset was created for the research activities of the ROBO-SPECT EU project (http://www.robo-spect.eu) and is not publicly available due to confidentiality restrictions. However, a small number of partially annotated images can be provided by the authors upon request.

Disclosure

Part of the work presented in this paper has been included in the doctoral thesis of Dr. Eftychios Protopapadakis titled "Decision Making via Semi-Supervised Machine Learning Techniques."

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from the European Commission's H2020 Research and Innovation Programme under Grant Agreement no. 740610 (STOP-IT project).

References

[1] W.-J. Chen, Y.-H. Shao, and N. Hong, "Laplacian smooth twin support vector machine for semi-supervised classification," *International Journal of Machine Learning and Cybernetics*, vol. 5, no. 3, pp. 459–468, 2014.

[2] E. Protopapadakis, A. Voulodimos, and N. Doulamis, "An investigation on multi-objective optimization of feedforward neural network topology," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–6, Larnaca, Cyprus, 2017.

[3] A. K. Alok, S. Saha, and A. Ekbal, "Semi-supervised clustering for gene-expression data in multiobjective optimization framework," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 2, pp. 421–439, 2017.

[4] E. Protopapadakis, M. Schauer, E. Pierri et al., "A genetically optimized neural classifier applied to numerical pile integrity tests considering concrete piles," *Computers & Structures*, vol. 162, pp. 68–79, 2016.

[5] K. Makantasis, E. Protopapadakis, A. Doulamis, and N. Matsatsinis, "Semi-supervised vision-based maritime surveillance system using fused visual attention maps," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 15051–15078, 2016.

[6] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and N. Matsatsinis, "3D measures exploitation for a monocular semi-supervised fall detection system," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 15017–15049, 2016.

[7] E. Protopapadakis, A. Doulamis, K. Makantasis, and A. Voulodimos, *A Semi-Supervised Approach for Industrial Workflow Recognition*, INFOCOMP, 2012.

[8] E. Protopapadakis and A. Doulamis, "Semi-supervised image meta-filtering using relevance feedback in cultural heritage applications," *International Journal of Heritage in the Digital Era*, vol. 3, no. 4, pp. 613–627, 2014.

[9] A. S. Voulodimos and C. Z. Patrikakis, "Quantifying privacy in terms of entropy for context aware services," *Identity in the Information Society*, vol. 2, no. 2, pp. 155–169, 2009.

[10] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, no. 1, pp. 137–148, 1969.

[11] R. Vidal, G. Sapiro, and E. Elhamifar, "See all by looking at a few: sparse modeling for finding representative objects," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600–1607, Providence, RI, USA, 2012.

[12] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, D. Dres, and M. Bimpas, "Stacked autoencoders for outlier detection in over-the-horizon radar signals," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 5891417, 11 pages, 2017.

[13] J. Wu, "Cluster analysis and K-means clustering: an introduction," in *Advances in K-means Clustering*, pp. 1–16, Springer, 2012.

[14] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," *AISTATS*, vol. 2005, pp. 57–64, 2005.

[15] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, vol. 3, pp. 58–65, AAAI Press, 2003.

[16] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 679–686, Haifa, Israel, 2010.

[17] Y.-F. Li, H.-W. Zha, and Z.-H. Zhou, "Learning safe prediction for semi-supervised regression," *AAAI*, vol. 2017, pp. 2217–2223, 2017.

[18] E. Protopapadakis, A. Voulodimos, and A. Doulamis, "Data sampling for semi-supervised learning in vision-based concrete defect recognition," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–6, Larnaca, Cyprus, 2017.

[19] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 245–284, 2015.

[20] N. F. F. Da Silva, L. F. S. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *ACM Computing Surveys*, vol. 49, no. 1, pp. 1–26, 2016.

[21] E. Protopapadakis, "Decision making via semi-supervised machine learning techniques," 2016, http://arxiv.org/abs/1606.09022.

[22] M. Daszykowski, B. Walczak, and D. L. Massart, "Looking for natural patterns in analytical data. 2. Tracing local density with OPTICS," *Journal of Chemical Information and Computer Sciences*, vol. 42, no. 3, pp. 500–507, 2002.

[23] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, Washington, DC, USA, 2003.

[24] A. Singla, S. Patra, and L. Bruzzone, "A novel classification technique based on progressive transductive SVM learning," *Pattern Recognition Letters*, vol. 42, pp. 101–106, 2014.

[25] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, 2006.

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87, Springer Science & Business Media, 2013.

[27] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep convolutional neural networks for efficient vision based tunnel inspection," in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 335–342, Cluj-Napoca, Romania, 2015.

[28] A. Voulodimos, D. Kosmopoulos, G. Veres, H. Grabner, L. Van Gool, and T. Varvarigou, "Online classification of visual tasks for industrial workflow monitoring," *Neural Networks*, vol. 24, no. 8, pp. 852–860, 2011.

[29] A. S. Voulodimos, D. I. Kosmopoulos, N. D. Doulamis, and T. A. Varvarigou, "A top-down event-driven approach for concurrent activity recognition," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 293–311, 2014.

[30] M. Daszykowski, B. Walczak, and D. L. Massart, "Representative subset selection," *Analytica Chimica Acta*, vol. 468, no. 1, pp. 91–103, 2002.

[31] C. Lalos, A. Voulodimos, A. Doulamis, and T. Varvarigou, "Efficient tracking using a robust motion estimation technique," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 277–292, 2014.

[32] A. Voulodimos, D. Kosmopoulos, G. Vasileiou et al., "A threefold dataset for activity and workflow recognition in complex industrial environments," *IEEE Multimedia*, vol. 19, no. 3, pp. 42–52, 2012.

[33] D. I. Kosmopoulos, N. D. Doulamis, and A. S. Voulodimos, "Bayesian filter based behavior recognition in workflows allowing for user feedback," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 422–434, 2012.

[34] N. D. Doulamis, A. S. Voulodimos, D. I. Kosmopoulos, and T. A. Varvarigou, "Enhanced human behavior recognition using HMM and evaluative rectification," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams - ARTEMIS '10*, pp. 39–44, Firenze, Italy, 2010.

[35] E. Protopapadakis, K. Makantasis, G. Kopsiaftis, N. Doulamis, and A. Amditis, "Crack identification via user feedback, convolutional neural networks and laser scanners for tunnel infrastructures," in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 725–734, Rome, Italy, 2016.

Hindawi
Complexity
Volume 2018, Article ID 8496187, 21 pages
https://doi.org/10.1155/2018/8496187

Review Article

Complex Power System Status Monitoring and Evaluation Using Big Data Platform and Machine Learning Algorithms: A Review and a Case Study

Yuanjun Guo, Zhile Yang, Shengzhong Feng, and Jinxing Hu

Shenzhen Institute of Advanced Technology Chinese Academy of Sciences, Shenzhen, Guangdong 5108055, China

Correspondence should be addressed to Zhile Yang; zyang07@qub.ac.uk

Received 17 April 2018; Accepted 7 August 2018; Published 20 September 2018

Academic Editor: Ireneusz Czarnowski

Efficient and valuable strategies provided by large amount of available data are urgently needed for a sustainable electricity system that includes smart grid technologies and very complex power system situations. Big Data technologies including Big Data management and utilization based on increasingly collected data from every component of the power grid are crucial for the successful deployment and monitoring of the system. This paper reviews the key technologies of Big Data management and intelligent machine learning methods for complex power systems. Based on a comprehensive study of power system and Big Data, several challenges are summarized to unlock the potential of Big Data technology in the application of smart grid. This paper proposed a modified and optimized structure of the Big Data processing platform according to the power data sources and different structures. Numerous open-sourced Big Data analytical tools and software are integrated as modules of the analytic engine, and self-developed advanced algorithms are also designed. The proposed framework comprises a data interface, a Big Data management, analytic engine as well as the applications, and display module. To fully investigate the proposed structure, three major applications are introduced: development of power grid topology and parallel computing using CIM files, high-efficiency load-shedding calculation, and power system transmission line tripping analysis using 3D visualization. The real-system cases demonstrate the effectiveness and great potential of the Big Data platform; therefore, data resources can achieve their full potential value for strategies and decision-making for smart grid. The proposed platform can provide a technical solution to the multidisciplinary cooperation of Big Data technology and smart grid monitoring.

1. Introduction

Along with the fast installation of computers and communication smart devices, the power industry is also experiencing tremendous changes both in the scale of power grid and in the system complexity. To build up a modern combined energy system of various types of energies including gas, cold, and heat, based on the smart power system, has become a trend of development in the energy industry. As discussed in many literatures [1–3], a modern energy system has several major features: (1) high penetration of new energy resources are supported and utilized effectively; (2) it provides complementation and integration of different types of energies such as electricity, gas, cold, and heat; and (3) an interconnected and relatively open system, distributed resources, and a consumption side are extensively involved. A huge amount of measurement data including production, operation, control, trading, and consumption are continuously collected, communicated, and processed in an amazing speed faster than any period of history [4].

Appropriate and efficient data management and analysis systems are urgently needed to leverage massive volumes of heterogeneous data in unstructured text, audio, and video formats; furthermore, useful information needs to be extracted and shared to meet the fast-growing demands of high-accuracy and real-time performance of modern power and energy systems [5]. Hidden values in power system big data cannot be effectively revealed by means of traditional power system analysis; therefore, Big Data technology and analytics are also in desperate need.

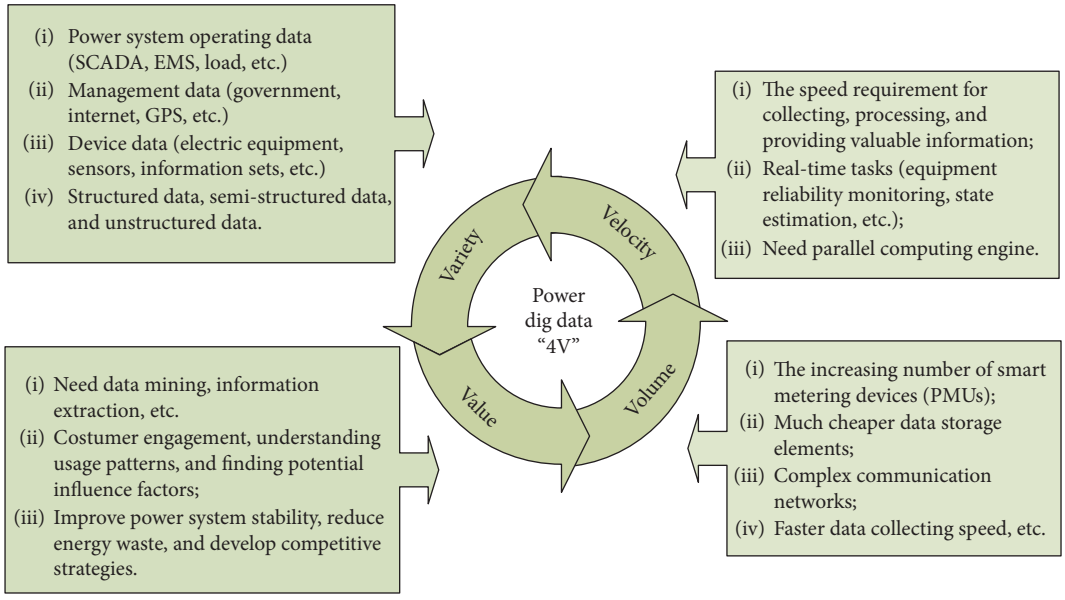


Figure 1: Power system Big Data 4V characteristics.

The Chinese power industry has considerable interests in Big Data analytics associated with power generation and management in order to effectively cope with severe challenges such as limited resources and environmental pollutions, among many others [6]. Actually, Big Data technology has already been successfully applied as a powerful data-driven tool for solving numerous new challenges in power grid, such as price forecasting [7, 8], load forecasting [9], transient stability assessment [10], outlier detection [11], and fault detection and analysis [12], among others [13, 14]. Detailed discussions about Big Data issues and application are reviewed in [15], as well as the insights of Big Data-driven smart energy management in [16]. Major tasks of the architecture for these applications are similar, which focus on two major issues: big power data modeling and big power data analysis.

1.1. Power Grid and Big Data. Supervisory control and data acquisition (SCADA) devices are mainly used in traditional power industries to collect data and to secure grid operations, providing redundant measurements including active and reactive power flows and injections and bus voltage magnitudes [17]. However, the sampling rate of SCADA is slow, and unlike traditional SCADA systems, the phasor measurement unit (PMU) is able to measure the voltage phasor of the installed bus and the current phasors of all the lines connected with that bus. In particular, PMUs are collecting data at a sampling rate of 100 samples per second or higher; therefore, a huge amount of data needs to be collected and managed. To be specific, the Pacific Gas and Electric Company in the USA collects over 3 TB power data from 9 million smart meters across the state grid [18]. The State Grid Corporation of China owns over 2.4 hundred million smart meters, making the total amount of collected data reach 200 TB for a year, while the total number of data in information centers can achieve up to 15 PB. Big Data is also often recognized as challenging in data volume, variety, velocity, and value in many applications [19, 20], and the "4V" characteristics are reflected in the following aspects considering applications in the power system, which is illustrated in Figure 1.

It is possible to get insights from the power system overall Big Data to improve the power efficiency, potentially influence factors of the power system status, understand power consumption patterns, predict the equipment usage condition, and develop competitive marketing strategies. The 4V characteristic can support the whole process of the power system, which is illustrated in Figure 2.

1.2. Challenges. From the above-mentioned research status of Big Data technology and its application in many aspects of the power system, it is easily concluded that Big Data management and analytics are certain development trends of future smart grids. However, there are still challenges that exist in this research area, and strategies and technologies for unlocking the potential of Big Data are still at the early stage of development. First of all, most existing power system utilities are not prepared to handle the growing volume of data, both for data storage and data analytics. On the one hand, traditional machine learning or statistical computing methods are designed for single machines, and an efficient extension of these methods which can be utilized for parallel computing or for large-scale data is urgently needed. On the other hand, most of the analytic methods used in the power system are not suitable to handle Big Data; thus, the gap between Big Data analytics and power system applications still exist, and high-performance computing methods are required. Then, a big hurdle is the lack of an intelligent platform integrating advanced methods for Big Data processing, knowledge extraction and presentation, and support in decision-making. It is believed that the success combination of Big Data technologies and power system analysis will bring

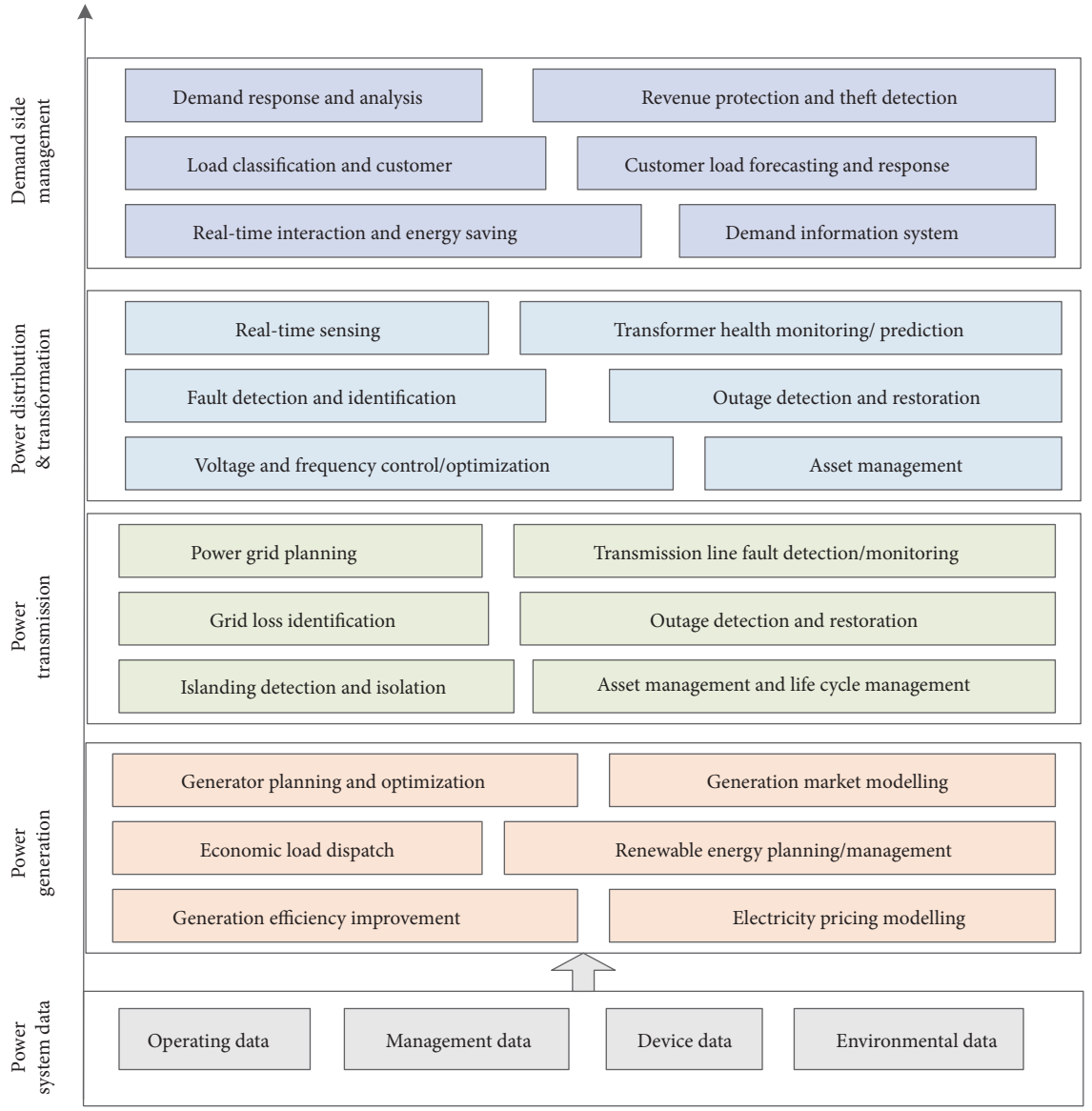


Figure 2: Sketch map of Big Data supporting whole process of the power system.

a number of benefits to the utility grid in the above-mentioned aspects. According to these challenges, this paper will present a novel Big Data platform for complex power system status monitoring and evaluation using machine learning algorithms.

2. Big Data Technologies for Complex Power System Monitoring

With the increasing varieties of data recording devices, much more unstructured power Big Data are being recorded continuously. Some particular data need to be collected or analyzed under different scales or projected to another dimension to describe the data. Therefore, some conflicts between data structure or semanteme need to be solved when projecting or transforming heterogeneous data into a unified form; the uncertainty and dynamics should also be taken into consideration for data fusion. Based on these concerns, the Big Data platform is designed to consist a generalized management model according to the complex logical relations between data objects, representing the data by normalization and extraction of the principal information. Challenges exist in how to design a flexible data management system architecture that accommodates multimode power data. This section introduces the state of the art of Big Data management technologies and data stream and value management.

2.1. State of the Art of Big Data Management Technology. In terms of distributed structure for Big Data management, the most popular designs are Hadoop [21] and Spark [22]. Hadoop was established in 2005, by Apache Software Foundation, with the key technologies of Map/Reduce [23], Google File System (GFS) [24] developed by Google Lab, and unrelational and high-volume data structure Bigtable [25], which have formed a novel computing distribution model. Base on the techniques above, Hadoop and open-

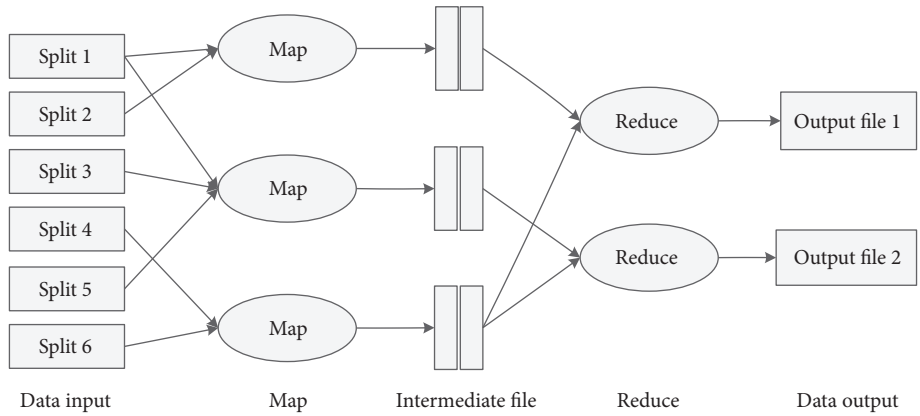


Figure 3: Hadoop MapReduce flowchart.

source projects like Hive and Pig have constituted the entire Hadoop ecosystem [26].

Hadoop, based on the distributed structure idea, enjoys many advantages such as high extensibility and high fault tolerance, and it is able to process heterogeneous massive data at high efficiency and low cost. In the Haddop ecosystem, files stored in HDFS (Hadoop Distributed File System) uses the subordinate structure, which are divided into several blocks; each of them has one or more duplicates distributed on different datanodes, thus the redundancy can prevent data from any loss caused by hardware failures. MapReduce is a programming model and an associated implementation for processing and generating large datasets. The computation can be specified by a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines; the flowchart is given in Figure 3. With the high concurrent processing way, several computing processes are organized simultaneously, thus the data handling capacity can be increased from terabyte level to petabyte level.

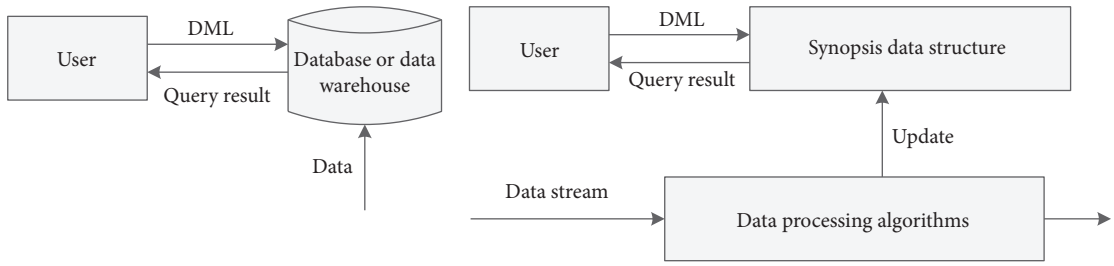
It can be seen that Hadoop technology is able to provide a reliable storage and processing approach; however, there are still limitations due to the Map and Reduce process. For a complex computation process, MapReduce needs a massive amount of Jobs to finish, and the relationships between these Jobs are managed by developers. Moreover, MapReduce is less supportive for interactive data and real-time data processing.

Similar to the computing frame of Hadoop MapReduce, another open-source tool Spark, developed by University of California Berkeley AMP lab, has the same advantages of MapReduce. Further, Spark can keep the intermediate results in RAM rather than write them in HDFS; thus, Spark can be better suitable for recursive algorithms such as data mining and machine learning applications. As a result, Spark is usually applied as a complement to Hadoop.

The key technology to Spark is the Resilient Distributed Dataset (RDD) [27], which is an abstraction to resolving the issue of slower MapReduce frameworks by sharing the data in memory rather than in disks, saving a large amount of I/O operations performed to query the data from disks. Therefore, RDD can greatly improve the recursive operation of machine learning algorithms and the interactive data mining methods.

Recently, a number of Big Data management systems have been developed to handle Big Data issues. For example, four representatives, MongoDB [28], Hive [29], AsterixDB [30], and a commercial parallel shared-nothing relational database system, have been evaluated in [31], with the purpose of studying and comparing Big Data systems using a self-developed microbenchmark and exploring the trade-offs between the performance of a system for different operations versus the richness of the set of features it provides. In terms of Big Data platform and tools that are suitable for power system and smart grid utilities, main contributions are made by leading IT companies like IBM [32], HP [33], and Oracle [34]. A number of IBM cases are done in order to improve the energy efficiency. For example, Vestas increases wind turbine energy production using a Big Data solution to more accurately predict weather patterns and pinpoint wind turbine placement [35]. CenterPoint Energy applies analytics to millions of streaming messages from intelligent grid devices enabling it to improve electric power reliability [36]. In the meantime, some newly established small technology companies, like C3 IoT [37], Opower [38] which has been acquired by Oracle in June 2016, Solargis [39], and AutoGrid [40], are doing Big Data analytics research and development according to the electricity market demand.

The large Internet companies in China, namely, Baidu [41], Aliyun [42], and Tencent [43], are all developing Big Data platform, tools, and applications according to their own business. For example, Baidu has been first in the world to open its Big Data engine to the public, which consists of key technologies of Big OpenCloud, Data Factory, Baidu Brain, and others. In this way, Baidu has won the prior opportunities to cooperate with the government, organizations, manufacturing companies, medical services, finance, retail, and education fields. Other companies like Inspur [44], Huawei [45], and Lenovo [46] also provide hardware from computer servers and storages to the Big Data analytic software, which have laid a good foundation for the development of the Big Data platform.



(a) Traditional data processing model (b) Data stream processing model

Figure 4: Comparison between the traditional data processing model and the data stream processing model.

2.2. Data Stream and Value Management. One of the most important ways to form Big Data is real-time data streaming, which is recorded continuously with time series. The data stream can be limitless, bringing a critical challenge for the data management system to store and process the streaming data. A definition was first proposed by Guha and Mcgregor in [47] that streaming data is considered to be an ordered sequence which can only be read one or a few times. Therefore, data stream management technology is the key issue to handling Big Data storage and processing.

Figure 4 shows the comparison between the traditional data processing model and the data stream processing model. For the traditional database, data storage is statical and not queried or updated often. Users send data manipulation language (DML) statements as queries, and the system will return the results after searching in the database. Therefore, there are inevitable I/O exchanges generated which will slow down the searching efficiency. For real-time processing of large amounts of streaming data, the traditional approach cannot meet the requirement. On the contrary, only synopsis data structure is stored instead of storing the entire dataset, and the data volume is much less and simpler to query compared to the traditional model.

The early research and design of the Big Data stream management system was only for single task applications. In order to handle streaming data with multiple tasks, the continuous query language was first proposed by Terry in Tapestry [48] in 1992, mainly used for filtering E-mails and the bulletin board system. Then it was followed by Mark Sullivan of Bell Labs in 1996, who designed a real-time monitoring tool named Tribeca [49] for the application of network surveillance. Tribeca was able to provided a limited number of continuous query languages and query operations. NiagaraCQ [50] was cooperatively developed by the Oregon Graduate Institute and University of Wisconsin, which support continuous query language and monitoring of durable and stable datasets in the entire wide-area network. In addition, Viglas and Naughton from the same project proposed a rate-based optimization on the issues of data streaming query speed [51]. In order to meet the requirements of data stream applications, a general data stream management is needed, and the official concept of a data stream management system was proposed in [52].

Nowadays, the most popular general data stream management system can be summarized as follows: Aurora [53], which was developed by the Massachusetts Institute of Technology, University of Brown, and Brandeis University, has a simple but special frame and can be used especially for data streaming surveillance based on a key technology of trigger networks. Aurora has a good balance on accuracy, response time, resource utilization, and practicability, but with a drawback of a simple query approach using the load shedding technique. TelegraphCQ [54], developed by the University of California Berkeley, is mainly used for sensor networks, which comprise a front end, a sharing storage, and a back end. The data stream in a constantly changing and unpredictable environment can be adaptively referred in any query. However, the approximate query mechanism will be neglected when the resource is insufficient. STREAM [55], developed by Stanford University, is the prototype system based on relational database. Under the circumstances of limited resources, STREAM can extend the searching language and execute the queries with high efficiency; thus, STREAM has a better performance on the continuous query. Other very famous data stream management systems are also released to cope with data stream challenges, such as Storm by Twitter [56], Data Freeway by Facebook [57], Samza by LinkedIn [58], TimeStream by Microsoft [59], and Gigascope by AT&T [60].

Data value in power systems can provide guidance towards data acquisition, data processing, and data application. Data valuation can be determined by several factors, including data correlation, data fidelity, and data freshness [61]. To be specific, data correlation can be considered from two aspects: one is how it is related with power dispatch, fault evaluation, and risk assessment; the other one is the correlation within the data itself, where the data value will be higher when the correlation is higher. Data fidelity refers to the conformance of the collected data to the real data situation. Defects of collected data always exist due to the sampling rate, noise, and data acquisition equipment from different devices across the entire grid; thus, the real data situation may not be revealed. At last, data freshness is also an important factor which determines the data value, especially in power systems where most data is streaming data, which is recorded without interrupt.

3. Analytical Tools and Methods for Power System Big Data

3.1. Big Data Analytical Open-Source Tools. Data analysis approaches such as machine learning play an important role

Table 1: Open-source/free software of Big Data machine learning method brief descriptions.

Name	Date	Developer	Brief descriptions
Octave	1993	James Rawlings, University of Wisconsin-Madison; John Ekerdt	A high-level language for numerical computations; suitable for solving linear and nonlinear problems; mostly compatible with Matlab, batch-oriented language [64].
Weka	1994	University of Waikato	Can be applied directly or called from a self-developed Java code and well-suited for developing new machine learning schemes [65].
R	1996	Ross Ihaka, Robert Gentleman	A language and environment for statistical computing and graphics; provides more than 70 packages of statistical learning algorithm; highly extensible [66].
Shogun	1999	Soeren Sonnenburg and Gunnar Raetsch	It provides a wide range of unified machine learning methods; easily combines multiple data representations, algorithm classes, and general purpose tools; rapid prototyping of data pipelines and extensibility of new algorithms [67].
http://AForge.net	2008	Andrew Kirillov, Fabio Caversan	It is an open-source C# framework in the fields of Computer Vision and Artificial Intelligence; image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc. [68].
Mahout	2009	Grant Ingersoll, Apache Software Foundation	It is an environment for quickly creating scalable machine learning applications; a framework to build scalable algorithms; has mature Hadoop MapReduce algorithms; suitable for Scala + Apache Spark, H2O, and Apache Flink [69].
MLlib	2009	UC Berkeley AMPLab, The Apache Software Foundation.	It is the Spark implementation of machine learning algorithms; easy to write parallel programs; and has potential to build new algorithms [70].
scikit-learn	2010	David Cournapeau, Matthieu Brucher, etc.	It is built on NumPy, SciPy, and matplotlib in Python environment; accessible, reusable in various contexts, and with simple and efficient tools [71].
Orange	2010	Bioinformatics Lab, University of Ljubljana, Slovenia	It is a data visualization and data analysis software; has interactive workflows with a large toolbox and a visualized process design based on Qt graphical interface [72].
CUDA-Convnet	2012	Alex Krizhevsky	It is a machine learning library with a built-in GPU acceleration; has been written by C++; with the CUDA GPU processing technology by NVidia [73].
ConvNetJS	2012	Andrej Karpathy, Stanford University	It is a JavaScript library for training deep learning models in the browser; is able to specify and train convolutional networks; comprises an experimental reinforcement learning module [74].
Cloudera Oryx	2013	Sean Owen, Cloudera Hadoop Distribution	It provides simple real-time large-scale machine learning and predictive analytics infrastructure; is able to continuously build/update models from large-scale data streams and query models in real time [75].

in power systems as algorithms can be trained using historical data collected over time, providing useful information for system operators. As historical data is collecting at an increasing speed with large volume, effective machine learning approaches are urgently needed in discovering valuable information and providing to power system operators. Big Data is stored in a distributed way on multiple computers; thus, it is not appropriate for all machine learning methods to process. Moreover, if data analytics needs to be finished on a single computer, it may be too large to fit into the main memory. Most traditional libraries/tools, such as R [62], Weka [63], and Octave [64], implemented machine learning algorithms in a single-threaded fashion by design and are not able to analyze large volumes of distributed data. More recently, advanced modern Big Data processing platforms are designed and implemented with parallel machine learning algorithms in order to achieve high efficiency. First of all, this section gives a comprehensive literature survey of state-of-the-art machine learning libraries and tools for Big Data analytics in Table 1.

From Table 1, it can be seen that along with the rapid development of the computer technology, a hot favorite of

developing machine learning library/tools started in the early 1990s. In almost a decade, the research trend moved forward to distributed and large volumes of data from the traditional single machine algorithm design. Octave is the earliest developed machine learning package, performing numerical experiments using a language that is mostly compatible with Matlab. Similarly, Weka was also developed by universities, which makes this free software suitable for academic use by integrating general purpose machine learning packages. In particular, R has been widely used in both academia and industry due to the comprehensive statistical computing and graphics software environment. As mentioned above, Octave, Weka, and R are designed for single-threaded computing and thus are not able to handle large volumes of power system data.

In recent years, the R community has developed many packages for Big Data processing. For example, the *biglm* package [76] is able to perform linear regression for large data, and the *bigrf* [77] package provides a Random Forest algorithm in which trees can be grown concurrently on a single machine, and multiple forests can be built in parallel on multiple machines then merged into one. Another group of R packages, such as *hive* [78], focus on providing interfaces between R and Hadoop, so that developers can access HDFS and run R scripts in the MapReduce paradigm.

Among the oldest, most venerable of machine learning libraries, Shogun was created in 1999 and written in C++, but is not limited to working in C++. In terms of supported language, Shogun can be used transparently in such languages and environments: as Java, Python, C#, Ruby, R, Lua, Octave, and Matlab, thanks to the SWIG library [79]. Another machine learning project designed for Hadoop, Oryx comes courtesy of the creators of the Cloudera Hadoop distribution. Oryx is designed to allow machine learning models to be deployed on real-time streamed data, enabling projects like real-time spam filters or recommendation engines.

3.2. Machine Learning and Statistical Processing Methods

3.2.1. Machine Learning Algorithms. Besides the powerful open-source algorithms or tools mentioned above, machine learning and statistical processing methods are also applied to support handling various issues of the power data. Basic machine learning algorithms are embedded in different open-source libraries/tools. Table 2 gives a comprehensive study and comparison.

There are many benefits for the modern power system since machine learning algorithms have been applied in many aspects of power systems successfully. Firstly, system stability and reliability have been remarkably increased. Many literatures have reported impressive experimental results of various machine learning algorithms with applications in oscillation detection, voltage stability, fault or transient detection and restoration, islanding detection and restoration, postevent analysis, etc. [80–83]. With the emergence of the Big Data analytics and smart grid technology, the above-mentioned monitoring and detection methods have been greatly improved, and an increasing number of novel approaches are being studied. For instance, real-time identification of dynamic events using PMUs is proposed in [84]; based on data-driven and physics models, security of power system protection and anomaly detection are greatly improved, thanks to the rich synchrophasor data.

Secondly, power equipment utilization and efficiency are greatly increased. In the power industry, the issues of waste of equipment resources are difficult to handle, and data resource is independent, thus it is impossible to evaluate the exact status of each asset. Big Data analytics can provide better validation and calibration of the models, eliminate the independence of data resources, and help operators understand the operating characteristics and life cycles of the equipments. For example, a data-driven approach for determining the maintenance priority of circuit breakers is introduced in [85]; the proposed method can consider both equipment-level condition monitoring parameters and system-level reliability impacting indices; thus, the maintenance priority list can be generated.

Thirdly, Big Data visualization can help operators improve situation awareness and assist decision-making. Machine learning and data analytics only produce numerical results or two-dimensional charts and diagrams, which need operators with professional skills or experience to give accurate and timely decision. A Big Data platform with 3D visualization in [86] manages massive power Big Data with multimode heterogeneous characters, showing the tripping lines and affected areas based on a 3D environment. Thus, the operators can make quicker and more reliable decisions and take possible preventive actions under the circumstance of thunder and lightning weather.

3.2.2. Statistical Processing Control Methods. Statistical processing control methods originally are applied in industrial quality control, employing statistical methods to monitor and control a process based on historical and online data. In our early work, some data-driven methods based on linear principal component analysis (PCA) [87] were applied in power system data analysis [88], setting up a distributed adaptive learning framework for wide-area monitoring, capable of integrating machine learning and intelligent algorithms in [89]. In order to handle power system dynamic data and nonlinear variables, dynamic PCA [90] and recursive PCA [91] were also developed to improve the model accuracy. It is worth mentioning that linear PCA is unable to handle all process variables due to the normal Gaussian distribution assumption imposed on them, and many extensions using neural networks have been developed [92, 93]. To address the challenges of handling the redundant input variables, obtaining higher model accuracy, and utilizing non-Gaussian distributed variables, an improved radial basis function neural network model-based intelligent method is also proposed in the early work [94]. The neural input selection is based on a fast recursive algorithm (FRA) [95, 96], which was proposed for the identification of nonlinear dynamic systems using linear-in-the-parameter models. It is possible to utilize optimization methods in order to get more accurate models by tuning algorithm-specific parameters, such as particle swarm pptimization (PSO), genetic algorithm (GA), differential evolution (DE), artificial bee colony (ABC), and ant

Table 2: Comparisons of open-source machine learning tools/algorithms for Big Data.

Category	Algorithm	Open source/free software						
		Weka	R	Shogun	Mahout	MLib	Orange	Oryx
Classification	Logistic regression	√		√	√	√	√	
	(Complementary) naive Bayes	√		√	√	√	√	
	Decision tree	√				√	√	
	Neural networks	√		√				
	SVM	√		√		√	√	
	Random forest	√	√				√	√
	Hidden Markov models			√	√			
Regression	Linear regression	√	√		√	√	√	
	Generalized linear models		√			√		
	Lasso/ridge regression		√		√		√	
	Decision tree regression	√				√		
Clustering	k-means	√		√	√	√	√	√
	Fuzzy k-means				√	√		
	Gaussian mixture model (GMM)					√		
	Streaming k-means					√		
Collaborative filtering	Alternating least squares (ALS)		√		√	√		√
	Matrix factorization-based				√			
Dimensionality Reduction	Singular value decomposition (SVD)			√	√	√		
	Principal component analysis	√	√	√		√		
Optimization primitive	Stochastic gradient descent)					√	√	
	Limited-memory BFGS (L-BFGS)			√		√		
Feature extraction	TF-IDF					√		
	Word2Vec					√		
Frequent pattern mining	FP growth	√				√		
	Association rules	√				√	√	

colony optimization (ACO), among other heuristic methods. The proper tuning of the algorithm-specific parameters is a very crucial factor which affects the performance of the above-mentioned algorithms. The improper tuning of algorithm-specific parameters either increases the computational effort or yields the local optimal solution. In our early work [97–99], teaching-learning-based optimization (TLBO) has been utilized for training an RBF neural network battery model. The TLBO method does not have any algorithm-specific parameters and significantly reduces the load of tuning work.

These methods mentioned above can be programmed and integrated as part of the analysing engine to support the processing of the power Big Data. Therefore, the data processing engine can support overall system operation and control by building a dynamic, global, and abstract power data model, based on which consequences are inferred and decisions are made. A detailed method comparison can be found in Table 3.

The fundamental assumption for many standard data-driven methods such as PCA, PLS, and LDA is that the measurement signals follow multivariate Gaussian distributions. As introduced in Table 3, PCA and PLS have similar principals to extract latent variables, but they perform in different ways. PCA tries to extract the biggest variance from the covariance matrix of the process variables, while PLS attempts to find factors or latent variables (LVs) to describe the relationship of output and input variables. PCA and LDA are also closely related in finding linear combinations of variables to explain data. However, LDA deals with the discrimination between classes, while PCA deals with the entire data samples without considering the class structure of the data. Similar to PLS, SIMs require both the input process data and the output data to form input-output relations. A brief comparison among the above-discussed basic data-driven methods is given in Table 4.

The issues of Gaussian distribution assumption on data, requirement of input-output relationships, the number of principal components or latent variables, and the computational complexity for these methods are compared in this table. In addition, LDA is comparable with PCA and the datasets should be well documented in order to

Table 3: An overview of state-of-the-art intelligent processing methods.

Category	Method	Descriptions	Applications
Standard	Principal component analysis (PCA)	PCA summarizes the variation in a correlated multiattribute data to a set of uncorrelated components, a linear combination of the original variables.	Pattern recognition [100], dimension reduction [101], feature extraction [102], process monitoring [103].
	Partial least squares (PLS)	PLS can find the fundamental relations between two data matrices, and latent variables are needed to model the covariance structure in these spaces.	Power load forecasting [104], performance evaluation of power companies [105], etc.
	Linear discriminant analysis (LDA)	LDA finds a linear combination of features that characterizes or separates two or more classes of objects or events.	Face recognition [106], feature selection for power system security assessment [107].
	Subspace identification methods (SIM)	SIMs are powerful tools for identifying the state space process model directly from data.	Power oscillatory state space model [108], power system stability analysis [109], etc.
Time-varying	Recursive PCA	RPCA is a generalization of PCA to time series; the eigenvector and eigenvalue matrices are updated with every new data sample.	Voltage stability moitoring [110], power system fault location detections [111].
	Dynamic PCA	DPCA includes dynamic behavior in the PCA model by applying a time lag shift method while retaining the simplicity of model construction.	Industrial monitoring [112, 113], dynamic economic evaluation of electrical vehicles [114].
Nonlinear	Kernel PCA/PLS	KPCA is first to map the input space into a feature space via nonlinear mapping and then to compute the PCs in that feature space.	Power equipment assessment [115], real-time fault diagnosis [116], power system monitoring [117], etc.
	Neural networks	Neural networks are computational models that can be used to estimate or approximate unknown nonlinear functions.	Dimension reduction [118, 119], voltage stability assessment [120], fault location detection [121], etc.
Non-Gaussian	Independent component analysis (ICA)	ICA decomposes multivariate signals into additive subcomponents which are independent non-Gaussian signals.	Fault detection [122], power quality monitoring [123], and estimation [124].
	Gaussian mixture models (GMM)	GMM describe an industrial process by local linear models using finite GMM and Bayesian inference strategy.	Power flow modeling [125], power load modeling [126].
	Support vector data description (SVDD)	SVDD defines a boundary around normal samples with a small number of support vectors.	Classification, process monitoring [127], oscillation modes detection [128], etc.

Table 4: A brief comparison among basic data-driven methods.

	PCA	PLS	LDA	SIM
Gaussian distribution	√	√	√	
Input-output relationship		√		√
Number of principal components	√			
Number of latent variables		√		
Computational complexity	Low	Medium	Medium	Medium

offer detailed information about the normal operating condition and faulty cases. SIM does not impose any special assumption on the process data since it only investigates the input-output relationship, and different threshold computation methods are available for Gaussian and non-Gaussian distributed data. The number of PCs and LVs are important design parameters in PCA and PLS methods, which can affect modeling performance. The main computation burden comes from performing SVD on the covariance matrix of different dimensions; thus, the standard PCA has lower computational cost over other basic methods.

Table 5: Comparisons of the non-Gaussian data methods.

Method	Data assumption	Parameters	Disadvantages
ICA	Can be described as a linear combination of non-Gaussian variables	Number of ICs	(1) High computational cost (2) Hard to determine the control limit
GMM	Can be described by local linear models	Multiple parameters in the model	(1) Complicated to train the models (2) Hard to determine the number of local models
SVDD	No strict assumption of data distribution	Kernel parameters in the model	(1) Hard to tune the kernel parameters (2) Trade-off between accurate boundary and low false alarm control limit

For time-varying process methods, recursive and adaptive methods are able to track slow-varying processes with a stable model structure. However, the model updating may be carried out randomly if no appropriate updating scheme is available. Meanwhile, dynamic process monitoring methods are easy to implement in practice, but the number of dynamic steps significantly affects the monitoring results and the window size is difficult to be determined.

Compared to linear monitoring methods, nonlinear approaches can be used in much wider applications due to the flexibility of nonlinear functions, which can model nonlinear relationships between variables. Especially for the kernel methods, various nonlinearities can be modelled by introducing different kernel functions. Similarly, neural networks are also capable of modeling any kind of nonlinearity theoretically. However, there are still some drawbacks; for example, the structure of the neural networks is difficult to determine and the training of the network parameters is also computationally demanding. A similar issue exists to kernel-based methods and an appropriate kernel parameter tuning method is needed, and the selection of a kernel function is not a trivial issue. A new approach to tackle the issues of representing nonlinear behavior as well as the non-Gaussian distributed variables is urgently needed.

For non-Gaussian distributed data, the basic methods cannot perform well due to the Gaussian distribution assumption. Alternatively, ICA, GMM, and SVDD are three most widely used and promising methods for non-Gaussian process monitoring. Although these methods were developed separately, they are actually highly related to each other. Sometimes, these methods can even be combined, and they are also capable of handling more than only one data characteristic. For example, ICA is used to describe the measurement signals as a linear combination of non-Gaussian variables, while GMM has a similar assumption that the process dataset can be described by several local linear models. Moreover, the calculation of control limits for ICA-based non-Gaussian process monitoring involves kernel density estimation, which is commonly used for SVDD. Detailed comparative advantages and disadvantages of these methods are listed in Table 5.

4. A Real-System Case

In this paper, a Big Data platform integrated with data management and analytical engine is proposed as a real-system case study. This platform was designed to meet the special condition of power grid in South China, such as large-scale, complex geographical and weather conditions and AC/DC mixed operation over long distances. Big Data technologies are applied to this power network to assist with condition monitoring and state estimation of the transmission and distribution systems, collecting multiplatform power data and realizing high-efficiency processes and analysis of data from the power grid at different levels.

4.1. The Framework of Electric Power Big Data Platform. The framework of the electric power Big Data platform consists of database, data interface, Big Data Management system, analytic engine with various machine learning tools and algorithms, and application and 3D visualization modules; a detailed structure is given in Figure 5. The first challenge is to set up an efficient database for the large volume of multisource heterogeneous power data which are collected through different sources. A traditional power system database is designed to store structured data files using tables; thus, the size of storage is limited and the data operation efficiency is low. For Big Data platforms, various data are collected, for example, operational data collected from the production management system and energy management system, real-time data recorded from an online monitoring system and equipment monitoring system, and other forms of heterogeneous data of weather files, geography information, images, and video data. In terms of data status, historical data, real-time data, and data streaming are all needed for Big Data processing and analysis. This platform integrates several data storages according to each data structure, so that the platform can provide useful and timely information to assist decision-making by processing large amounts and different data structures with high efficiency. All the information and knowledge can be integrated to provide strategies for system operation and evaluation, system inspection, and status estimation for power equipments and the entire power grid.

In order to efficiently manage and store the multisource Big Data, this paper proposes a special data acquisition structure. For various databases, SQOOP is a tool designed for efficiently transferring bulk data between HDFS and structured datastores such as relational databases (MySQL, Oracle). For messages between databases and the platform, MQTT (message queuing telemetry transport) is chosen as part of the data interface. MQTT is well known as an "Internet of Things" connectivity protocol, and it was designed as an extremely lightweight published/subscribed

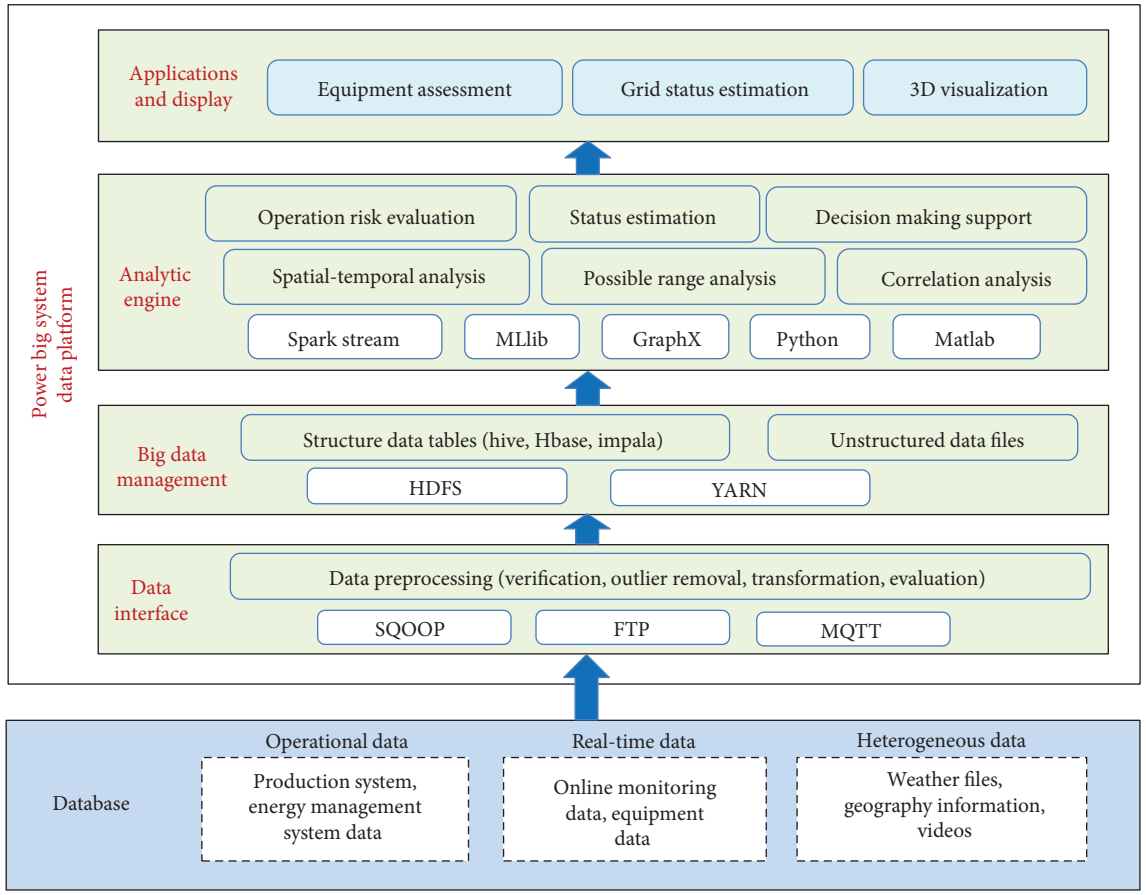


Figure 5: Big Data processing and analysing platform for electric power system condition monitoring.

messaging transport. For files such as documents and working logs of each equipment, transmission lines, and substations, FTP (file transfer protocol) is the common tool to transfer through the Internet to the platform.

Based on this data interface, power system data collected by smart devices can be managed in real time. Data preprocessing, including data verification, outlier removal, transformation, and evaluation process, can be realized to provide a solid and practical database for the analysis procedure. Moreover, other relative unstructured data such as weather condition, lighting and storms, geography information, and human activities (local population, age distribution, professionals, behavior and active pattern, internet sentiment, and so on) can be connected to a certain extent with the power load, power generation, consumptions, electricity market, and so on. These data sources mentioned above are impossible to be processed and analyzed simultaneously through the traditional way; only this novel approach using Big Data to deal with the challenges can establish a more comprehensive knowledge model of the city power grid.

4.2. High-Performance Analytical Engine. To effectively manage the Big Data is only the first step; the key issue is to set up an analytic engine with high efficiency. Based on the functional modules and the need for power system applications, this particular analytic engine can provide with several practical functions, such as operation risk evaluation, status estimation, and decision-making support. The detailed structure of the Big Data computational engine is given in Figure 6.

This analytical engine integrates a number of open-source basic algorithm packs and self-developed algorithms. The open-source algorithm packs mentioned in Section 3 have been developed and tested by researchers and companies for many years. For example, Apache Spark, a fast and general engine for large-scale data processing, can be used interactively with Scala, Python, and R shells. Many powerful computing libraries are integrated in Apache Spark, such as numerical computing tool NumPy, science computing tool SciPy, data analysing library Pandas, scalable machine learning library MLlib [70], API for graphs and graph-parallel computation GraphX [129], and so on. In addition, this platform has combined an interactive developing and operating environment IPython and Jupyter [130]. Effective power grid decision-making depends critically on anlytic methods in the platform. Therefore, effective methods for the real-time exploitation of large volumes of power data are needed urgently. Robust data analytics, high-performance computation, efficient data network management, and cloud computing techniques are critical towards the optimized operation of power systems.

For self-developed algorithms, spatial-temporal correlation analysis is able to mine both the strong and weak connections among the numerous variables in a power grid, by

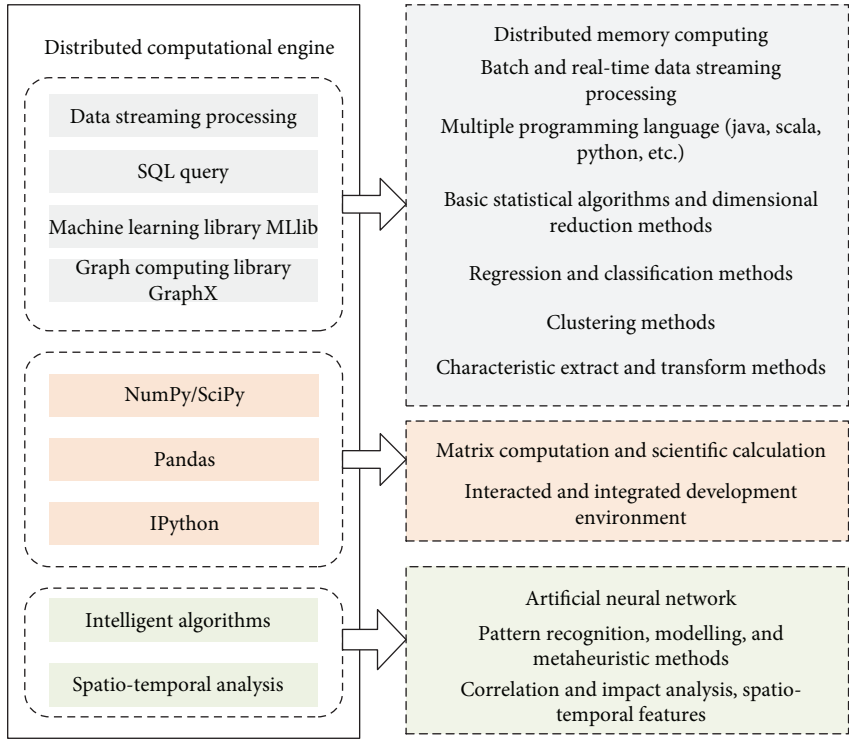


Figure 6: Structure of Big Data platform computational engine.

setting up a power system spatial-temporal model and a data-driven model based on the process history database. Modelling methods are provided, including artificial neural networks, linear and nonlinear analysis methods, Gaussian-based kernel methods, regression and classification methods, and clustering methods. Pattern recognition methods for spatial-temporal correlations are provided, and the spatial proximity weights, time delay, and correlation effect are calculated and quantized [131]. This idea is suitable for analysing the consumption behaviors of citizens in different locations and time, as well as the effect on power transmission lines by the power grid surroundings including geographic information, weather variations, human activities, and road vehicles and traffic situations [132]. A knowledge base of interconnected factors within the entire city grid can be set up for analysis and predictions.

This proposed distributed computational engine is the key element of the entire Big Data platform; many functional modules can be developed based on these open-source tools. It is believed that this novel approach will gradually change the traditional way of power system analysis and operation, which is also the only efficient way to realize future smart grids with high level of automation and intelligence.

4.3. 3D Visualization. The geographic information system (GIS) has been widely used in electric power systems [133, 134], which is vital for improving the operation efficiency of the electric power system. It can maintain, manage, and analyze power data and integrate power network models, maps, and related data in a solution for desktops, webs, and mobile devices. Most power GIS systems mainly adopt a two-dimensional map as the visualization model. However, 2D GIS has significant limitations in terms of presentation and analysis of geospatial and power data, and it is difficult to display panorama information of power running status. The proposed Big Data platform adopts a web-based visualization method based on Cesium and 3D City Database (3DCityDB) [135] to construct a three-dimensional panorama electric power visualization system, which is given as in Figure 7.

The 3D models of electric tower, line, equipment, and geographical entity (buildings, roads, etc.) will be visualized in Cesium scene and managed by a Cesium manager. In the server side, Java Servlet and JavaServer Pages for power-related data processing functions reside in Tomcat which directly communicate with web client and process client requests. The two-dimensional map requests will be submitted to the Geoserver, while three-dimensional map requests will be processed by a 3DCityDB web feature service. 3DCityDB is a free open-source package consisting of a database scheme and a set of software tools to import, manage, analyze, visualize, and export virtual 3D city models according to the CityGML standard. In this architecture, 3DCityDB has two important tasks: one is to convert a two-dimensional electric map model to a three-dimensional model and save into the PostgreSQL database, and the second is used to provide a three-dimensional web feature service for a power system client based on Cesium.

Based on the model calculation and Big Data analytical engine, the visualization of spatial information and power system applications can be realized in the way of providing services. Thus, the power system equipments and power grid can be merged together with GIS and revealed on the map, as well as the environmental factors. Therefore, many demands

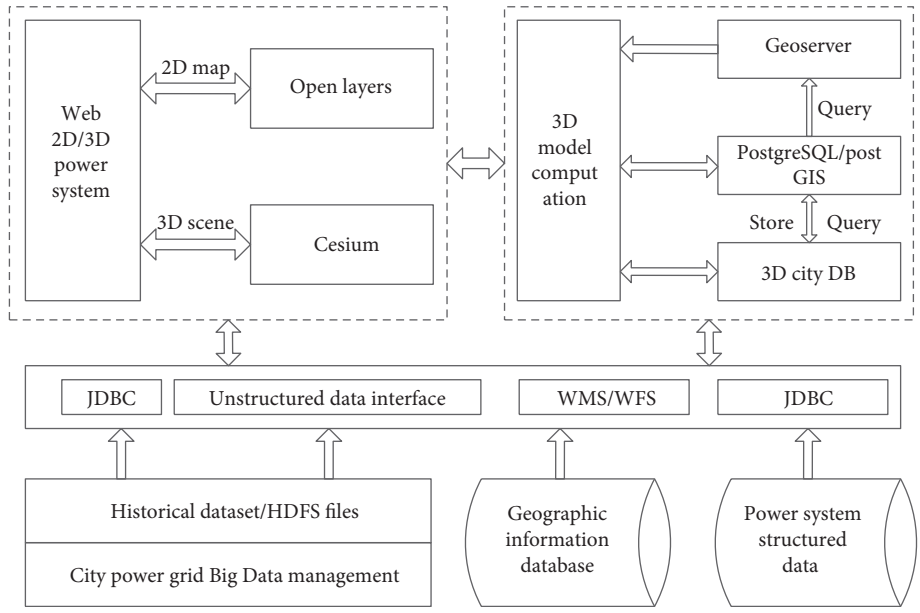


Figure 7: Framework of the 3D display system.

of power grid visualization can be reached, including real-time monitoring, analysis, and decision-making, among others. The development of the 3D visualization system can provide an optimal way of presenting the huge amount of information and improve the situation awareness of system operators as well as the novel explanation of newly appeared information; thus, the accuracy of decision-making for the entire power system can be greatly increased.

5. Application Study

5.1. Development of Power Grid Topology and Parallel Computing Using CIM Files. Power element data, connections, and their status are stores as common information model (CIM) files in the power system, which are significant for power system analytics. The first step is to extract the connectivity between each electric point as data to be stored in the relational database. For most of the analytic methods, the above-mentioned CIM file extraction is applied to fit in the relational database. However, a topology analysis needs plenty of correlation analyses between multiple and complex tables; it is hard to meet the demand of real-time and fast-speed processing requirement. The proposed platform in this paper develops a fast-processing scheme for the power grid topology setup; thus, the analysis can be realized with high efficiency. The diagram is given in Figure 8.

The proposed platform detects any update of CIM files which were transmitted into the FTP end, load new data into memory, and correlate with other structured data using Spark SQL, generating a preprocessed data table. After that, a fast search according to "physical-electrical-physical" rules in the power grid is applied to set up a topology of the grid. The whole process is realized based on the Spark SQL database and parallel computation; thus, the analysis efficiency is greatly improved, thanks to the fast and parallel correlation analysis. Under this framework, many tasks can be done easily including analysis result extraction, power grid topology setup, power system branch model calculation, and "bus-branch" model analysis and other functions. Therefore, this platform is able to provide a database and analytical engine for power grid large-scale parallel computation, real-time status analysis for smart grids, and other useful applications.

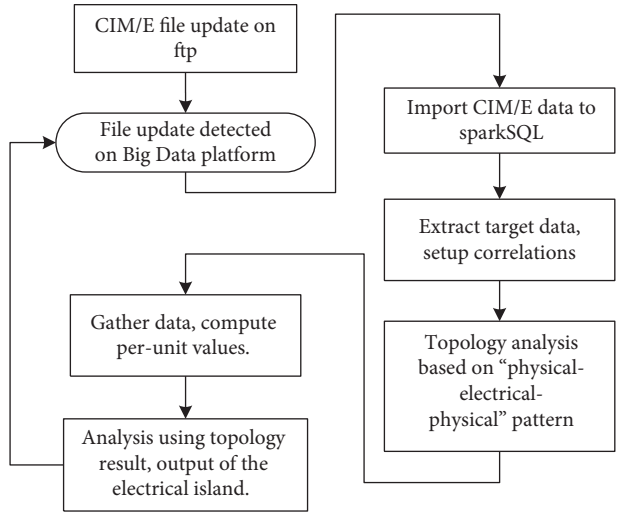


Figure 8: Fast-speed analysis flowchart for CIM files.

5.2. High-Efficiency Load-Shedding Calculation. The calculation of load-shedding in the power system can quantify how much loss the real system is undergoing after equipment failure in an objective way; thus, it can measure the operation risks and provide significant information for decision-making of equipment reconditioning or replacement. The actual reduction of load-shedding for different types on each electrical point is needed for the calculation; thus, it is very

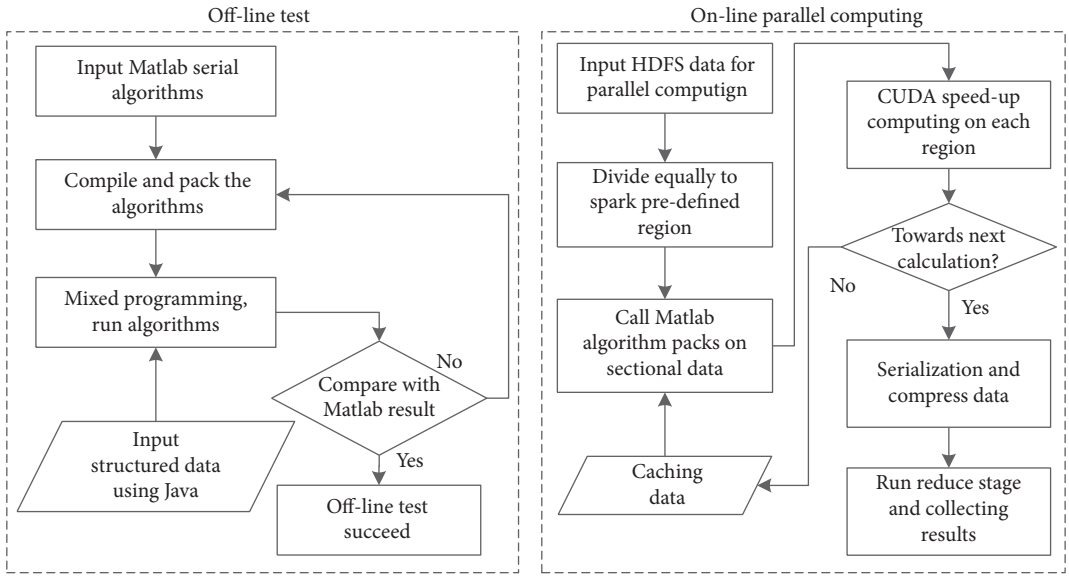


Figure 9: Fast-speed analysis flowchart for CIM files.

Table 6: The comparisons of parallel computing with single machine results.

Machines	Model	Memory	Cores	Executor	Time
Machines	YARN	60G	30	30	11 min
Single	Local	200G	20	20	2.5 hours

time-consuming to calculate power grid risk evaluation with plenty of predefined fault scenarios. In the proposed platform, a calculation scheme based on Spark and Compute Unified Device Architecture (CUDA) is applied, as shown in Figure 9.

The complete load-shedding scheme contains two stages: offline test stage and online parallel computing stage. The computation tasks are firstly divided into different working regions on Spark, then Matlab algorithms are packed and called, and further processing of each computation task is transmitted to working threads on every division, where parallel computing is realized. After that, results at each step are collected progressively; thus, the risk evaluation tasks for multiple scenarios can be finished. For real-system cases, a total number of 6000 scenario files with 1.2 GB size are calculated according to the flowchart given in Figure 9, and the comparison with calculation time on a single machine is given in Table 6.

It can be easily seen that parallel computing is able to solve the problem of low efficiency when risk evaluation in multiscenarios is taken in the power system. The load level of each electrical point can be monitored dynamically, and the topology change of power grid due to any system maintaining or drop out of multiple power system units can also be calculated with high efficiency, therefore, the computation time is greatly shortened.

5.3. Power System Transmission Line Tripping Analysis Using 3D Visualization. With the support of the Big Data platform, transmission line trip records, power quality data, weather data, and other related data can be collected, in order to monitor and analyze the transients. In addition, a three-dimensional visualization system is developed to merge together all the analysis results with geographic, landforms, and even weather conditions, then display in a very intuitive way. Therefore, situation awareness of system operators is greatly enhanced. Two main tasks are introduced in this section: firstly, the correlation between line trips and power transients is analyzed by employing statistical methods, especially the distribution patterns of line tripping and power quality voltage dips against the lasting time. Secondly, the interconnection rules among line tripping, weather condition, voltage dips, and voltage swells and other disturbances are exploited.

In order to analyze the correlation between transmission line trips and voltage dips, multisource data is needed, consisting of (1) transmission line tripping data, recording tripping time, fault description, fault type, and so on, and (2) voltage disturbance data, including monitoring location, disturbance type, happening time, lasting time, and magnitude. The first step to analyze the transients is data fusion, combining two sets of data according to the unified time tags, and the preprocess diagram is given in Figure 10.

For analysis of voltage dips at different voltage levels of 110 kV and 10 kV, the voltage dip recordings are divided into four kinds, including voltage dips caused by line trips at 110 kV and 10 kV, not by line trips at 110 kV and 10 kV. Taking 10 kV voltage level for example, the scatter plot is generated and shown in Figure 11.

In this figure, each symbol represents a transient event, with duration as the x-axis and magnitude as the y-axis. In order to separate the transient events by their causing reasons, the blue dot represents the voltage dip caused by line trips while the red x shows that the occurring voltage dip was not due to line trips, both at the 10 kV voltage level. The x-axis has taken the logarithm for the purpose of

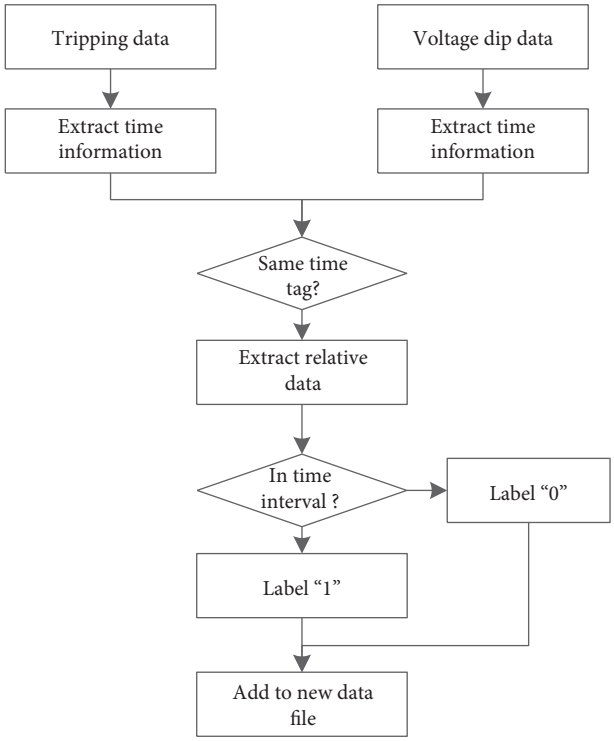


Figure 10: Diagram of data files fusion preprocess.

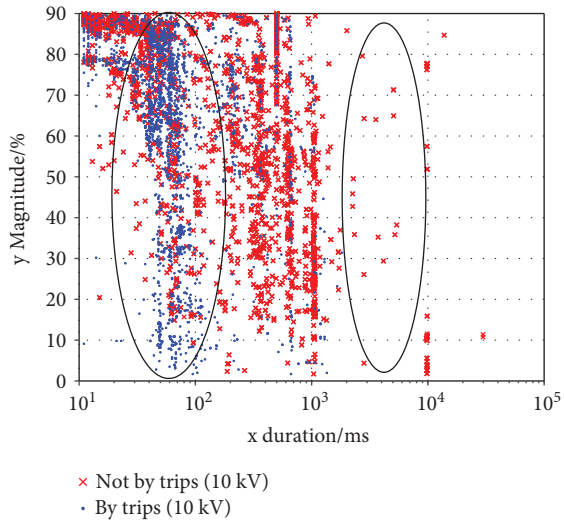


Figure 11: Scatter plot of voltage dips and breakdowns against duration under 10 kV voltage level (half logarithmic axis).

showing the distribution more clearly. Generally speaking, it can be seen from Figure 11 that at the 10 kV level, the lasting time of voltage dips caused by line trips is less than that caused by other reasons, as shown in the left ellipse, with duration around 100 ms. And the voltage dips caused by other reasons last for a longer time, as enclosed in the right ellipse.

The scattered points only show the distribution of durations against magnitudes of the voltage dips. It is necessary to combine substation coordinates, maps, and other geographic information with these transients; thus, the transmission line status and the affected substation can be shown in terms of voltage dip magnitudes and durations. Therefore, the possible influence of transmission line trippings to the substations can be visualized to system operators. The Big Data platform employs a 3D simulation display system, using data from the management layer as well as the model output directly from the computing engine, including 3D models of power line and electric equipment, 3D building models, geospatial data, and power attribute data. Geospatial data as a 3D virtual environment can show geographic objects (e.g., roads, bridges, and rivers) around the electricity network. The generated 3D virtual environment with power transmission line situation is given in Figure 12.

In Figure 12, the green line represents the normal operational transmission line, while the red lines are with the appearance of the line trips. In order to show the voltage transient status, a cylinder with blue color shows the voltage dip magnitude, and the pink cylinder is the duration, and the name of the affected transmission lines is shown in the floating red tags above the cylinders. Therefore, the affected area can be directly visualized through the 3D virtual environment, and the dynamic change of the power grid operational status is easier to control for the system operators. If any transient happened, actions can be taken in time to prevent any enlargement of the accident.

6. Discussion and Conclusion

This paper reviewed both the issues of Big Data technologies for power systems and employed a Big Data platform for power system monitoring and evaluation analysis. Based on the review of Big Data management technology and analytical tools and machine learning methods, a case study of the proposed novel Big Data platform for a power system is given with three application cases introduced. The framework of the power system Big Data platform consists of database collecting power data from all different parts across the grid, data interface, Big Data management system integrating different management technologies, analytic engine with various machine learning tools and algorithms, applications, and 3D visualization modules for further optimizing the strategy and decision-making assistance.

Based on the various power data sources, the proposed platform has integrated different data interfaces and distributed data storage according to the data structure; thus, the platform is able to handle traditional structured data, semi-structured data, and unstructured data simultaneously. For the analytical engine, both open-source tools and self-developed models are integrated as modules. In our early work, intelligent processing methods have been proved to be able to handle linear, dynamic, nonlinear, and non-Gaussian distributed variables by setting up accurate and efficient models. This has enabled the decision-making subsystem to focus on generating an optimized equipment maintenance strategy and providing a global view for situation awareness and information integration.

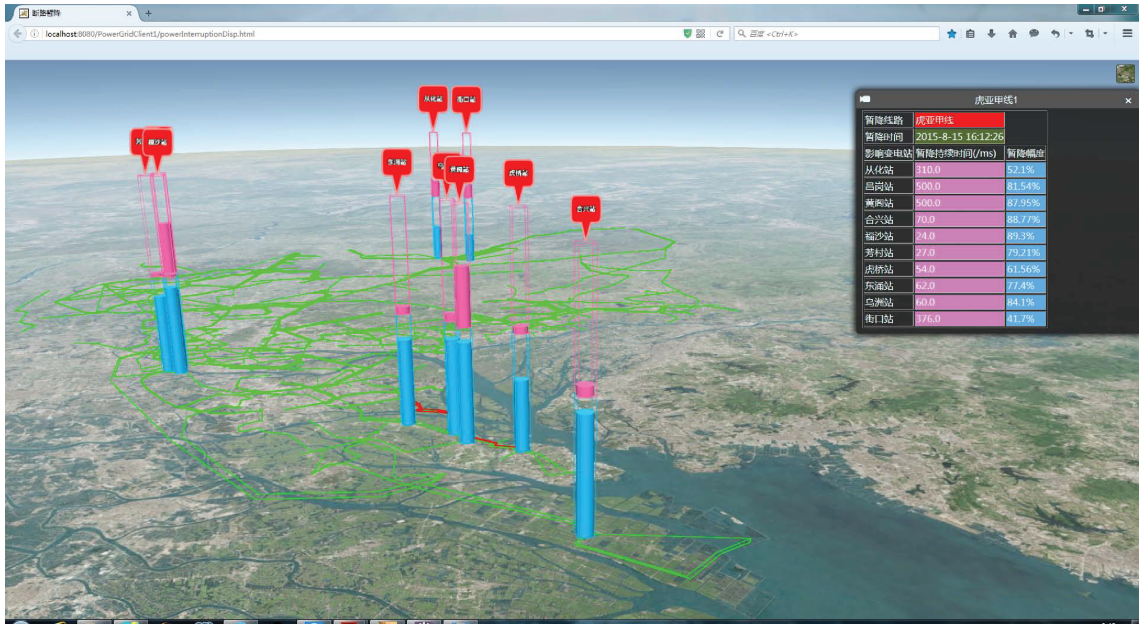


Figure 12: 3D display of voltage dips and breakdown transmission lines with geographic information.

In order to demonstrate the effectiveness of the proposed platform, three real-system cases are introduced including development of power grid topology and parallel computing using CIM files, high-efficiency load-shedding calculation, and power system transmission line tripping analysis using 3D visualization. These cases are all realized based on the proposed Big Data platform; the key issue in case one is to extract the connectivity between each electric point from different databases. It is suitable to process high-volume and multimode heterogeneous data using multiple data storage methods in the proposed Big Data platform with very high efficiency. In case two, with a proposed parallel computing scheme based on Spark and CUDA, load-shedding calculation in power systems under different scenarios can be realized in a very fast-speed way, and a comparison between single-machine and multiple-machine parallel computing is given, which demonstrated the high efficiency of the scheme. A highlight in the third case is the utilization of a Big Data platform with the 3D visualization system. With the help of the Big Data virtual environment, the affected transmission lines and areas can be directly detected, with detailed dynamic information of line tripping time, location, duration, and causes. With the help of the 3D visualization system, digital results become more valuable and situation awareness of system operators is greatly enhanced, which is a reliable way to improve the safety and reliability of the entire power grid.

As mentioned in this survey, the development of future smart grid will towards a huge and complex energy system, which is deeply integrated with traditional power and renewable energies, as well as the powerful information and communication systems. The energy system also represents three levels or perspectives of the entire objective existence: physical energy level, industrial information level, and human society level. Under this big picture, more researchers are focusing on novel dimensions. For newly developed machine learning and data mining tools, deep learning, transfer learning, and multidata fusion methods are receiving extensive attention in recent years. Deep learning integrates supervised and unsupervised learning, with multiple hidden layer artificial neural network structures, and is capable of extracting abstract conceptions from data. While transfer learning makes a break through fundamental assumptions of the statistical learning theory, it can improve learning accuracy by utilizing the correlated data with different distributions. Multidata fusion technique is capable of analysing heterogeneous datasets collecting from different data sources; thus, it can extract more useful information.

By applying the above-mentioned new methods and technologies, more research directions and topics gradually appear. Firstly, the load prediction and modeling problem is the earliest application of data mining and analytics. Along with the fast installations of smart meters, much more precisely load modeling can be achieved by utilizing the equipment data and electrical measurements at both transient and steady states. More machine learning methods are available for load prediction and modeling, including feed-forward artificial neural network, SVMs, recurrent artificial neural networks, and regression trees, among others. Secondly, the fusion and merging analysis of the power system and transportation system can be done along with the increasing number of electrical vehicles. Considering the load data from charging stations, traffic flow and transportation network, on-board GPS tracks of electrical vehicles, and other data related to the driving and charing behaviors, a research on the driving and charging behavior characteristics is achievable. Closely related to that, the electricity market prediction and simulation is another possible hot topic, which can also be applied in many aspects such as evaluation of market shares for the individual power company,

investment income for power generation, and decision-making for power market mechanism design.

In conclusion, this paper has demonstrated a glance of the crossover and merging of the latest Big Data technology and smart grid technology. There are still many researchworks to do in the future. From all the application aspects, Big Data technology for human behavior in panorama mode has a great and long-term potential in real-time future smart grid and energy system, even the city planning, pollution abatement, transportation planning, and other useful applications.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (51607177, 61433012, and U1435215), Shenzhen Science and Technology Innovation Commission application demonstration project (No. KJYY20160608154421217), and China Postdoctoral Science Foundation (2018M631005).

References

[1] Y. XUE, "Energy internet or comprehensive energy network?," *Journal of Modern Power Systems and Clean Energy*, vol. 3, no. 3, pp. 297–301, 2015.

[2] Y. Xue and Y. Lai, "Integration of macro energy thinking and big data thinking part one big data and power big data," *Automation of Electric Power Systems*, vol. 40, no. 1, pp. 1–8, 2016.

[3] Y. Xue and Y. Lai, "Integration of macro energy thinking and big data thinking: part two applications and exploration," *Automation of Electric Power Systems*, vol. 40, no. 8, pp. 1–13, 2016.

[4] A. Gandomi and M. Haider, "Beyond the hype: big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[5] X. He, Q. Ai, R. C. Qiu, W. Huang, L. Piao, and H. Liu, "A big data architecture design for smart grids based on random matrix theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, p. 1, 2017.

[6] Chinese Society of Electrical Engineering, "Chinese white paper on the development of large power data," pp. 1–10, 2013.

[7] A. A. Munshi and Y. A.-R. I. Mohamed, "Big data framework for analytics in smart grids," *Electric Power Systems Research*, vol. 151, pp. 369–380, 2017.

[8] K. Wang, C. Xu, Y. Zhang, S. Guo, and A. Zomaya, "Robust big data analytics for electricity price forecasting in the smart grid," *IEEE Transactions on Big Data*, p. 1, 2017.

[9] D. Wang and Z. Sun, "Big data analysis and parallel load forecasting of electric power user side," *Proceedings of the Csee*, vol. 35, no. 3, pp. 527–537, 2015.

[10] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, "Power system transient stability assessment based on big data and the core vector machine," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2561–2570, 2016.

[11] W. Alves, D. Martins, U. Bezerra, and A. Klautau, "A hybrid approach for big data outlier detection from electric power scada system," *IEEE Latin America Transactions*, vol. 15, no. 1, pp. 57–64, 2017.

[12] Y. Zhao, P. Liu, Z. Wang, L. Zhang, and J. Hong, "Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods," *Applied Energy*, vol. 207, pp. 354–362, 2017.

[13] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 233–244, 2015.

[14] S. J. Plathottam, H. Salehfar, and P. Ranganathan, "Convolutional neural networks (cnns) for power system big data analysis," in *2017 North American Power Symposium (NAPS)*, pp. 1–6, Morgantown, WV, USA, September 2017.

[15] S. Sagiroglu, R. Terzi, Y. Canbay, and I. Colak, "Big data issues in smart grid systems," in *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1007–1012, Birmingham, UK, November 2016.

[16] K. Zhou, C. Fu, and S. Yang, "Big data driven smart energy management: from big data to big insights," *Renewable & Sustainable Energy Reviews*, vol. 56, pp. 215–225, 2016.

[17] G. N. Korres and N. M. Manousakis, "State estimation and bad data processing for systems including pmu and scada measurements," *Electric Power Systems Research*, vol. 81, no. 7, pp. 1514–1524, 2011.

[18] Pacific Gas and Electric Company, "Pacific gas and electric," 2013, https://www.pge.com/.

[19] C. Tu, X. He, Z. Shuai, and F. Jiang, "Big data issues in smart grid – a review," *Renewable & Sustainable Energy Reviews*, vol. 79, pp. 1099–1107, 2017.

[20] J. Zhu, E. Zhuang, J. Fu, J. Baranowski, A. Ford, and J. Shen, "A framework-based approach to utility big data analytics," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2455–2462, 2016.

[21] The Apache Software Foundation, "The apache hadoop," 2005, http://hadoop.apache.org/index.html.

[22] The Apache Software Foundation, "The apache spark," 2000, http://spark.apache.org/.

[23] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[24] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, p. 29, 2003.

[25] F. Chang, J. Dean, S. Ghemawat et al., "Bigtable: a distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, pp. 1–4, 2008.

[26] J. Yates, J. D. Mcgregor, J. E. Ingram, and J. Yates, "Hadoop and its evolving ecosystem, in: International Workshop on Software," in *5th International Workshop on Software Ecosystems (IWSECO)*, pp. 57–68, Potsdam, Germany, June 2013.

[27] R. B. Ray, M. Kumar, and S. K. Rath, *Fast Computing of Microarray Data Using Resilient Distributed Dataset of Apache Spark*, Springer International Publishing, 2016.

[28] K. Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media, Inc., 2013.

[29] A. Thusoo, J. S. Sarma, and N. Jain, "Hive - a petabyte scale data warehouse using hadoop," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pp. 996–1005, Long Beach, CA, USA, March 2010.

[30] S. Alsubaiee, K. Faraaz, E. Gabrielova et al., "Asterixdb: a scalable, open source bdms," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 1905–1916, 2014.

[31] P. Pirzadeh, M. J. Carey, and T. Westmann, "Bigfun: a performance study of big data management system functionality," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 507–514, Santa Clara, CA, USA, November 2015.

[32] International Business Machines, "Ibm energy and utilities," 2015, April 2018, http://www-935.ibm.com/industries/energy/case_studies.html.

[33] A. Joiner, "Big data changes everything," 2014, April 2017, http://h20435.www2.hp.com/t5/HP-Software/Big-Data-is-changing-everything/ba-p/100623#.V3JBok9Z_hV.

[34] ORACLE, "Leverage big data and analytics," 2014, April 2018, https://www.oracle.com/industries/utilities/electricity/index.html.

[35] International Business Machines, "Ibm energy and utilities," 2017, August 2018, https://www.ibm.com/industries/uk-en/energy/.

[36] International Business Machines, "Ibm energy and utilities, centerpoint energy," 2017, August 2018, https://www.ibm.com/industries/uk-en/energy/case-studies.html.

[37] C3IoT, "C3 iot platform," 2009, April 2018, http://c3iot.com/products/#energy_grid.

[38] Opower, "Elevate your customer experience," 2007, April 2018, http://www.opower.com/.

[39] Solargis, "Accurate and efficient solar energy assessment," 2010, April 2018, http://solargis.info/.

[40] AutoGrid, "Turning data into power," 2011, December 2017, http://www.auto-grid.com/.

[41] Baidu, "Baidu Big Data," 2011, December http://bdp.baidu.com/.

[42] Aliyun, "Aliyun data ide," 2011, December 2017, https://data.aliyun.com/product/ide?spm=a2c0j.7906235.header.11.ntdqdP.

[43] Tencent, "Tencent big data," 2009, December 2017, http://bigdata.qq.com/.

[44] Inspur, "Inspur," 2009, December 2017, http://www.inspur.com/.

[45] Huawei, "Fusioninsight," 2015, December 2017, http://e.huawei.com/cn/products/cloud-computing-dc/cloud-computing/bigdata/fusioninsight.

[46] Lenovo, "Lenovo thinkclouds," 2016, December 2017, http://appserver.lenovo.com.cn/Lenovo_Series_List.aspx?CategoryCode=A30B03.

[47] S. Guha and A. Mcgregor, "Stream order and order statistics: quantile estimation in random-order streams," *SIAM Journal on Computing*, vol. 38, no. 5, pp. 2044–2059, 2009.

[48] D. Terry, D. Goldberg, D. Nichols, and B. Oki, "Continuous queries over append-only databases," *ACM SIGMOD Record*, vol. 21, no. 2, pp. 321–330, 1992.

[49] M. Sullivan, "Tribeca: A Stream Database Manager for Network Traffic Analysis," in *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, Mumbai (Bombay), India, September 1996.

[50] J. Chen, D. J. Dewitt, F. Tian, and Y. Wang, *NiagaraCQ: a Scalable Continuous Query System for Internet Databases*, ACM, 2000.

[51] S. D. Viglas and J. F. Naughton, "Rate-based query optimization for streaming information sources," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data - SIGMOD '02*, pp. 37–48, Madison, Wisconsin, June 2002.

[52] A. Arasu, S. Babu, and J. Widom, "The cql continuous query language: semantic foundations and query execution," *VLDB Journal*, vol. 15, no. 2, pp. 121–142, 2006.

[53] D. Carney, U. Çetintemel, M. Cherniack et al., "Monitoring streams — a new class of data management applications," in *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 215–226, Hong Kong SAR, China, August 2002.

[54] J. M. Hellerstein, M. J. Franklin, S. Chandrasekaran et al., "Adaptive query processing: technology in evolution," *IEEE Data Engineering Bulletin*, vol. 23, pp. 7–18, 2000.

[55] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data streams systems," in *PODS '02 Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1–16, Madison, Wisconsin, June 2002.

[56] A. Toshniwal and D. Taneja, "Storm @ twitter," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ACM*, pp. 147–156, Snowbird, Utah, USA, June 2014.

[57] Z. Shao, "Real-time analytics at facebook," 2015, http://www-conf.slac.stanford.edu/xldb2011/talks/xldb2011_tue_0940_facebookrealtimeanalytics.pdf.

[58] C. Riccominig, "How linkedin uses apache samza," 2014, http://www.infoq.com/articles/linkedin-samza.

[59] Z. Qian, Y. He, C. Su et al., "Timestream: reliable stream computation in the cloud," in *Proceedings of the 8th ACM European Conference on Computer Systems - EuroSys '13*, pp. 1–14, Prague, Czech Republic, April 2013.

[60] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk, "Gigascope: a stream database for network applications," in *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data - SIGMOD '03*, pp. 647–651, San Diego, California, June 2003.

[61] R. Meier, E. Cotilla-Sanchez, B. Mccamish, and D. Chiu, "Power system data management and analysis using synchrophasor data," in *2014 IEEE Conference on Technologies for Sustainability (SusTech)*, pp. 225–231, Portland, OR, USA, July 2014.

[62] R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of Computational & Graphical Statistics*, vol. 5, no. 5, pp. 299–314, 1996.

[63] G. Holmes, A. Donkin, and I. H. Witten, "Weka: a machine learning workbench," in *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference*, pp. 357–361, Brisbane, Queensland, Australia, December 1994.

[64] J. W. Eaton, "gnu octave," 2014, January 2018, http://www.gnu.org/software/octave/.

[65] R. R. Bouckaert, E. Frank, M. A. Hall et al., "WEKAâˆ'Experiences with a Java Open-Source Project," *Journal of Machine Learning Research*, vol. 11, no. 5, pp. 2533–2541, 2010.

[66] F. Morandat, B. Hill, L. Osvald, and J. Vitek, "Evaluating the design of the r language - objects and functions for data analysis," in *Proceedings of the 26th European Conference on Object-Oriented Programming*, pp. 104–131, Beijing, China, June 2012.

[67] S. Sonnenburg, G. Tsch, S. Henschel et al., "The shogun machine learning toolbox," *Journal of Machine Learning Research*, vol. 11, pp. 1799–1802, 2010.

[68] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow, *AForge.NET*, Betascript Publishing, 2010.

[69] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*, Manning Publications Co., 2011.

[70] X. Meng, J. Bradley, B. Yavuz et al., "Mllib: machine learning in apache spark," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2015.

[71] F. Pedregosa, G. Varoquaux, and E. Duchesnay, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[72] J. Demšar, T. Curk, A. Erjavec et al., "Orange: data mining toolbox in python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.

[73] A. Krizhevskey, "Cuda-convnet," 2012, April 2018, http://code.google.com/p/cuda-convnet/.

[74] A. Karpathy, "Convnetjs:deep Learning in your browser," http://cs.stanford.edu/people/karpathy/convnetjs/.

[75] S. Owen, "Cloudera oryx: Simple real-time large-scale machine learning infrastructure," 2014, https://github.com/cloudera/oryx.

[76] T. Lumley, "biglm: bounded memory linear and generalized linear models," 2014, http://cran.r-project.org/web/packages/biglm/index.html.

[77] L. B. A. Lim and A. Cutler, "bigrf: Big random forests: classification and regression forests for large data sets," 2014, http://cran.rproject.org/web/packages/bigrf/index.html.

[78] S. T. I. Feinerer, "hive: Hadoop interactive," 2014, http://cran.rproject.org/web/packages/hive/index.html.

[79] D. M. Beazley, "Swig: an easy to use tool for integrating scripting languages with c and c++," in *4th Annual Tcl/Tk Workshop*, Monterey, CA, July 1996.

[80] G. Marchesan, M. R. Muraro, G. Cardoso, L. Mariotto, and A. P. de Morais, "Passive method for distributed-generation island detection based on oscillation frequency," *IEEE Transactions on Power Delivery*, vol. 31, no. 1, pp. 138–146, 2016.

[81] X. Xu, Z. Yan, M. Shahidehpour, H. Wang, and S. Chen, "Power system voltage stability evaluation considering renewable energy with correlated variabilities," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3236–3245, 2018.

[82] J. Liu, N. Tai, and C. Fan, "Transient-voltage-based protection scheme for DC line faults in the multiterminal VSC-HVDC system," *IEEE Transactions on Power Delivery*, vol. 32, no. 3, pp. 1483–1494, 2017.

[83] M. Sahraei-Ardakani, X. Li, P. Balasubramanian, K. W. Hedman, and M. Abdi-Khorsand, "Real-time contingency analysis with transmission switching on real power system data," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2501-2502, 2016.

[84] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-time identification of dynamic events in power systems using PMU data, and potential applications—models, promises, and challenges," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.

[85] J. Zhong, W. Li, C. Wang, and J. Yu, "A rankboost based data-driven method to determine maintenance priority of circuit breakers," *IEEE Transactions on Power Delivery*, vol. 33, no. 3, pp. 1044–1053, 2018.

[86] Y. Liu, Y. Guo, Z. Yang, J. Hu, G. Lu, and Y. Wang, "Power system transmission line tripping analysis using a big data platform with 3d visualization," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Honolulu, HI, USA, November 2017.

[87] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[88] Y. Guo, K. Li, and D. Laverty, "A statistical process control approach for automatic anti-islanding detection using synchrophasors," in *2013 IEEE Power & Energy Society General Meeting*, pp. 1–5, Vancouver, BC, Canada, July 2013.

[89] K. Li, Y. Guo, D. Laverty, H. He, and M. Fei, "Distributed adaptive learning framework for wide area monitoring of power systems integrated with distributed generations," *Energy and Power Engineering*, vol. 5, no. 4, pp. 962–969, 2013.

[90] Y. Guo, K. Li, and D. M. Laverty, "Loss-of-main monitoring and detection for distributed generations using dynamic principal component analysis," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 423–431, 2014.

[91] Y. Guo, K. Li, D. M. Laverty, and Y. Xue, "Synchrophasor-based islanding detection for distributed generation systems using systematic principal component analysis approaches," *IEEE Transactions on Power Delivery*, vol. 30, no. 6, pp. 2544–2552, 2015.

[92] A. Kheirkhah, A. Azadeh, M. Saberi, A. Azaron, and H. Shakouri, "Improved estimation of electricity demand function by using of artificial neural network, principal component analysis and data envelopment analysis," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 425–441, 2013.

[93] A. Onwuachumba and M. Musavi, "New reduced model approach for power system state estimation using artificial neural networks and principal component analysis," in *2014 IEEE Electrical Power and Energy Conference*, pp. 15–20, Calgary, AB, Canada, November 2014.

[94] Y. Guo, K. Li, Z. Yang, J. Deng, and D. M. Laverty, "A novel radial basis function neural network principal component analysis scheme for pmu-based wide-area power system monitoring," *Electric Power Systems Research*, vol. 127, pp. 197–205, 2015.

[95] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.

[96] K. Li, J.-X. Peng, and E.-W. Bai, "Two-stage mixed discrete-continuous identification of radial basis function (rbf) neural models for nonlinear systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 630–643, 2009.

[97] Z. Yang, K. Li, Q. Niu, Y. Xue, and A. Foley, "A self-learning tlbo based dynamic economic/environmental dispatch considering multiple plug-in electric vehicle loads," *Journal of Modern Power Systems and Clean Energy*, vol. 2, no. 4, pp. 298–307, 2014.

[98] Z. Yang, K. Li, Q. Niu, and Y. Xue, "A comprehensive study of economic unit commitment of power systems integrating various renewable generations and plug-in electric vehicles," *Energy Conversion and Management*, vol. 132, pp. 460–481, 2017.

[99] Z. Yang, K. Li, Q. Niu, and Y. Xue, "A novel parallel-series hybrid meta-heuristic method for solving a hybrid unit

commitment problem," *Knowledge-Based Systems*, vol. 134, pp. 13–30, 2017.

[100] Q. Shao and C. J. Feng, "Pattern recognition of chatter gestation based on hybrid pca-svm," *Applied Mechanics and Materials*, vol. 120, pp. 190–194, 2011.

[101] M. D. Farrell and R. M. Mersereau, "On the impact of pca dimension reduction for hyperspectral detection of difficult targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, 2005.

[102] L. I. Kuncheva and W. J. Faithfull, "Pca feature extraction for change detection in multidimensional unlabeled data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 69–80, 2014.

[103] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 377–386, 2016.

[104] R. Zhang, W. Cai, L. Ni, and G. Lebby, "Power system load forecasting using partial least square method," in *2008 40th Southeastern Symposium on System Theory (SSST)*, pp. 169–173, New Orleans, LA, USA, March 2008.

[105] W. Zheng and H. Wang, "Organizational performance evaluation of power supply with partial least-squares regression," in *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management*, pp. 161–163, Changchun, China, September 2011.

[106] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data — with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[107] C. A. Jensen, M. A. El-Sharkawi, and R. J. Marks, "Power system security assessment using neural networks: feature selection using fisher discrimination," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 757–763, 2001.

[108] R. Eriksson and L. Soder, "Wide-area measurement system-based subspace identification for obtaining linear models to centrally coordinate controllable devices," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 988–997, 2011.

[109] C. Luo and V. Ajjarapu, "Invariant subspace based eigenvalue tracing for power system small-signal stability analysis," in *2009 IEEE Power & Energy Society General Meeting*, pp. 1–9, Calgary, AB, Canada, July 2009.

[110] J. Yang, W. Li, T. Chen, W. Xu, and M. Wu, "Online estimation and application of power grid impedance matrices based on synchronised phasor measurements," *IET Generation, Transmission & Distribution*, vol. 4, no. 9, p. 1052, 2010.

[111] A. H. Al-Mohammed and M. A. Abido, "A fully adaptive PMU-based fault location algorithm for series-compensated lines," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2129–2137, 2014.

[112] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 81–93, 2000.

[113] R. Srinivasan, C. Wang, W. K. Ho, and K. W. Lim, "Dynamic principal component analysis based methodology for clustering process states in agile chemical plants," *Industrial & Engineering Chemistry Research*, vol. 43, no. 9, pp. 2123–2139, 2004.

[114] M. Chen and L. X. Guo, "The synthetic evaluation method of the dynamic performance and economic performance of battery electric vehicle based on principal component analysis," *Applied Mechanics and Materials*, vol. 215-216, pp. 1259–1262, 2012.

[115] W. Sun and G. Ma, "Condition assessment of power supply equipment based on kernel principal component analysis and multi-class support vector machine," in *2009 Fifth International Conference on Natural Computation*, pp. 485–488, Tianjin, China, August 2009.

[116] J. Ni, C. Zhang, and S. X. Yang, "An adaptive approach based on KPCA and SVM for real-time fault diagnosis of HVCBs," *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 1960–1971, 2011.

[117] Z. Weiqing, S. Fengqi, X. Zhigao, Q. Zongliang, and Z. Jianxin, "An investigation on system anomaly source diagnosis using kpca-fpsdg," in *2012 Asia-Pacific Power and Energy Engineering Conference*, pp. 1–4, Shanghai, China, March 2012.

[118] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[119] S. Theodoridis and K. Koutroumbas, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[120] D. Q. Zhou, U. D. Annakkage, and A. D. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1566–1574, 2010.

[121] M.-R. Mosavi and A. Tabatabaei, "Traveling-wave fault location techniques in power system based on wavelet analysis and neural network using gps timing," *Wireless Personal Communications*, vol. 86, no. 2, pp. 835–850, 2016.

[122] Y. Zhang and S. J. Qin, "Fault detection of nonlinear processes using multiway kernel independent component analysis," *Industrial & Engineering Chemistry Research*, vol. 46, no. 23, pp. 7780–7787, 2007.

[123] M. Ruiz-Llata, G. Guarnizo, and C. Boya, "Embedded power quality monitoring system based on independent component analysis and svms," in *The 2011 International Joint Conference on Neural Networks*, pp. 2229–2234, San Jose, CA, USA, July 2011.

[124] C. Uzunoglu, M. Ugur, F. Turan, and S. Cekli, "Amplitude and frequency estimation of power system signals using independent component analysis," in *2013 21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, Haspolat, Turkey, April 2013.

[125] G. Valverde, A. T. Saric, and V. Terzija, "Stochastic monitoring of distribution networks including correlated input variables," *IEEE Transactions on Power Delivery*, vol. 28, no. 1, pp. 246–255, 2013.

[126] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 29–37, 2010.

[127] X. Liu, L. Xie, U. Kruger, T. Littler, and S. Wang, "Statistical-based monitoring of multivariate non-gaussian systems," *AICHE Journal*, vol. 54, no. 9, pp. 2379–2391, 2008.

[128] X. Liu, D. McSwiggan, T. B. Littler, and J. Kennedy, "Measurement-based method for wind farm power system oscillations monitoring," *IET Renewable Power Generation*, vol. 4, no. 2, p. 198, 2010.

[129] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "Graphx: a resilient distributed graph system on spark," in *First International Workshop on Graph Data Management*

Experiences and Systems - GRADES '13, pp. 1–6, New York, June 2013.

[130] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 21–29, 2007.

[131] L. Yin and S.-L. Shaw, "Exploring space–time paths in physical and social closeness spaces: a space–time gis approach," *International Journal of Geographical Information Science*, vol. 29, no. 5, pp. 742–761, 2015.

[132] Z. Yang, K. Li, and A. Foley, "Computational scheduling methods for integrating plug-in electric vehicles with power systems: a review," *Renewable & Sustainable Energy Reviews*, vol. 51, pp. 396–416, 2015.

[133] M. Jahangiri, R. Ghaderi, A. Haghani, and O. Nematollahi, "Finding the best locations for establishment of solar-wind power stations in middle-east using gis: a review," *Renewable & Sustainable Energy Reviews*, vol. 66, pp. 38–52, 2016.

[134] M. A. Anwarzai and K. Nagasaka, "Utility-scale implementable potential of wind and solar energies for Afghanistan using gis multi-criteria decision analysis," *Renewable & Sustainable Energy Reviews*, vol. 71, pp. 150–160, 2017.

[135] B. He, W. X. Mo, J. X. Hu, G. Yang, G. J. Lu, and Y. Q. Liu, "Development of power grid web3d gis based on cesium," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 2465–2469, Xi'an, China, October 2016.

Hindawi
Complexity
Volume 2018, Article ID 1935938, 10 pages
https://doi.org/10.1155/2018/1935938

Research Article

Similarity-Based Summarization of Music Files for Support Vector Machines

Jan Jakubik and Halina Kwaśnicka

Department of Computational Intelligence, Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland

Correspondence should be addressed to Jan Jakubik; jan.jakubik@pwr.edu.pl

Received 19 April 2018; Accepted 4 July 2018; Published 1 August 2018

Academic Editor: Piotr Jędrzejowicz

Automatic retrieval of music information is an active area of research in which problems such as automatically assigning genres or descriptors of emotional content to music emerge. Recent advancements in the area rely on the use of deep learning, which allows researchers to operate on a low-level description of the music. Deep neural network architectures can learn to build feature representations that summarize music files from data itself, rather than expert knowledge. In this paper, a novel approach to applying feature learning in combination with support vector machines to musical data is presented. A spectrogram of the music file, which is too complex to be processed by SVM, is first reduced to a compact representation by a recurrent neural network. An adjustment to loss function of the network is proposed so that the network learns to build a representation space that replicates a certain notion of similarity between annotations, rather than to explicitly make predictions. We evaluate the approach on five datasets, focusing on emotion recognition and complementing it with genre classification. In experiments, the proposed loss function adjustment is shown to improve results in classification and regression tasks, but only when the learned similarity notion corresponds to a kernel function employed within the SVM. These results suggest that adjusting deep learning methods to build data representations that target a specific classifier or regressor can open up new perspectives for the use of standard machine learning methods in music domain.

1. Introduction

Recently, in our digital world, there are huge resources of data, images, video, and music. Advanced methods of automatic processing of music resources remain in the sphere of interest of many researchers. The goal is to facilitate music information retrieval (MIR) in a personalized way for the needs of an individual user. Despite the involvement of researchers and use of state-of-the-art methods, such as deep learning, there is a lack of advanced search engines, especially able to take into account users' personal preferences. Observed quick increase in the size of music collections on the Internet resulted in the emergence of two challenges. First is the need for automatic organizing of music collections, and the second is how to automatically recommend new songs to a particular user, taking into account the user's listening habit [1]. To recommend a song according to user's expectations, it is beneficial to automatically recognize the emotions that a song induces to the user and the genre to which a song belongs.

Music, similarly to a picture, is very emotionally expressive. In developing system for music indexing and recommendation, it is necessary to consider emotional characteristics of music [2]. Identifying the emotion induced by music automatically is not yet solved and the problem remains a significant challenge. The relationship between some basic features as timbre, harmony, or lyrics and emotions they can induce is complex [3]. Another problem is a high degree of subjectivity of emotions induced by music. Even if we take into account the same listener, then the emotions induced by a given piece of music may depend on their mood, fatigue, and other factors. All of the above makes the automatic recognition of emotions (by classification or regression) a difficult task.

In emotion recognition, there are categorical [4] and continuous space [5] models of emotion; both are research topics [6, 7]. The most popular model is two-dimensional continuous valence-arousal scale. Positive and negative emotions are indicated on one coordinate axis, and arousal separates low activation from high on the second. This model of emotions is derived from research concerning emotions in general. Authors of [8] consider emotion recognition as a regression separately for arousal and valence. Other types of emotions are considered in Geneva Emotional Music Scale (GEMS) [9]. Categories defined in GEMS are domain-specific. They are the result of surveys in which participants were asked to describe emotion induced by listened music. Emotions in GEMS are organized in three levels: the higher level contains generic emotion groups; the middle level consists of nine categories: wonder, transcendence, tenderness, nostalgia, calmness, power, joy, tension, and sadness; and the lowest contains specific nouns.

Another research topic in MIR area is the problem of automatic classification of music pieces taking into account genre [10]. In music analysis, genre represents a music style. Members of the same style (genre) share similar characteristics such as tempo, rhythm patterns, and types of instruments and thus can be distinguished from other types of music.

As music data is extremely complex, the key issue when handling it in machine learning systems becomes summarization of them in a form that a classifier can process. While research datasets typically employed in MIR studies are not large in terms of file count, the complexity and variety within each individual file are significant. For both genre and emotion recognition, the use of machine learning methods is largely reliant on the appropriate selection of features that describe the music samples. In general, automatic music analysis such as music classification (or regression when we deal with emotion recognition) encompasses two steps: feature extraction and the classification (regression). Both are difficult and strongly influence the final result. Early works used manually defined set of features based on expert domain knowledge. Many researchers have studied the relationship between emotion and different features that describe music [5]. In [11], the authors added harmonic features to a set of popular music features to the predicting community consensus task with GEMS categories. They show that adding harmonic features improves the accuracy of recognition.

The authors of [12] proposed the use of feature learning on low-level representations instead of defining a proper set of features manually. Codebook methods have been shown to learn useful features even in shallow architectures [13–15]. The use of simple autoencoder neural network to learn features on a spectrogram for predicting community consensus task with GEMS categories gives comparable results as traditional machine learning with the use of a manually well-chosen set of features [16]. Deep learning improves these results further, resulting in state-of-the-art performance. Convolutional recurrent neural networks, working on a low-level representation of sounds, have been used for learning features that would be useful in classification task [17, 18]. While deep learning in itself performs very well, it creates new opportunities for the use of older machine learning methods. The features can be taken from the selected level of deep network and used as an input to a support vector machine (SVM), or a regression method such as SVR, or any other classifier [19].

In our research, we are interested in the possibility of improving the usefulness of traditional machine learning methods, in particular, SVM, when combined with deep learning as a feature extractor. For training a deep neural network for classification, typically the softmax activation function for prediction and minimization of cross entropy loss is employed. Effectively, the network is trained to maximize the performance of its final layer, which works as a linear classifier on features from the previous layers. However, one of the biggest advantages of SVM among standard machine learning methods is its performance on nonlinear problems. It is largely reliant on the so-called kernel trick—replacing the inner product in the solved optimization problem with kernel functions, which can be understood as similarity measures. Given that a neural network can be trained to minimize any loss differentiable with respect to the network's weight matrices, it may be possible to adjust it so that it produces features specifically fit for a kernel SVM, rather than a linear classifier. Knowing the basic principle of kernel trick, we attempt to train the network to replicate certain notion of similarity between annotations that describe genres or emotions of the music pieces, within representation space that is the output of a neural network. The goal of this study is to test whether the proposed change in the approach to training the feature extracting network will yield performance improvements over simply using an NN for both feature learning and classification or regression, as well as SVM deployed on features extracted from a NN learned with a standard loss function.

Our approach is similar to the one presented in [20], where the author replaces the softmax layer of an NN by linear SVM. However, the approach presented by Tang is concerned with the integration of linear SVM within the network. In contrast, we treat SVM as a classifier separate from the feature learning process, assuming the feature learning takes place first, and then the classifier is trained on features extracted by the network. This is in line with the growing trend of transfer learning, which seeks to reuse the complex architectures trained on large datasets, for multiple problems. A feature extracting network could be easily reused on other similar tasks while only retraining the classifier SVM, similarly to [21].

We consider tasks of classification and regression on five different datasets. Focusing on emotion, we use three music mood recognition datasets, one for classification and two for regression. We complement these with two classification datasets, one for genre recognition and one for dance style recognition. The paper is organized into three sections: "Introduction," "Materials and Methods," and "Results and Discussion." The second section contains all theoretical background, dataset descriptions, and other information required to replicate the study, while in the third, we present and discuss the obtained results.

2. Materials and Methods

The goal of our research is to evaluate the possibility of using recurrent neural networks as a feature learner while changing its loss function to one based on pairwise similarity rather than one explicitly predicting annotations within the network. We hypothesize this approach will better fit an SVM-based classifier or regressor. This section contains a description of neural network architectures employed in the study and the datasets on which we performed our experiments. Conditions of the experiments, such as hyperparameters of the algorithms, are also described. We refrain from explaining SVM in detail, as our contribution does not develop the method itself.

2.1. Gated Recurrent Neural Networks. Recurrent neural networks (RNN) are useful for modelling time series [22]. A basic recurrent layer is defined by

$$h_t = \sigma(\mathbf{W}\mathrm{x}_t + \mathbf{U}h_{t-1} + \mathbf{b}), \tag{1}$$

where σ is an activation function, which can be logistic sigmoid function (σ_{sig}) or hyperbolic tangent activation ($\sigma_{\tanh}$); $\mathbf{W}$ and $\mathbf{U}$ are matrices of weight; and $\mathbf{b}$ is the bias vector. x_t is a current input, in a series of l input vectors, $(x_1, x_2, \dots, x_l)$. Matrices $\mathbf{W}$ and $\mathbf{U}$ and the bias vector $\mathbf{b}$ are learned using backpropagation algorithm.

As the more complex models, with the use of gating mechanisms, have been applied to natural language processing with success, they became a common research subject within the deep learning area. In these, a special "unit" replaces a recurrent layer. It consists of multiple interconnected layers. Outputs can be multiplied or added element-wise. When element-wise multiplication of any output with an output of a log-sigmoid layer is applied, a "gating" mechanism is created. The log-sigmoid layer is a kind of gate that decides if the output passes (multiplication by 1) or not (multiplication by 0). Long short-term memory (LSTM) [23] network is the most popular model that uses gating. LSTM is defined by

$$\begin{aligned}
r_t &= \sigma_{\mathrm{sig}}(\mathbf{W}_r x_t + \mathbf{U}_r h_{t-1} + \mathbf{b}_r),\\
i_t &= \sigma_{\mathrm{sig}}(\mathbf{W}_i x_t + \mathbf{U}_i h_{t-1} + \mathbf{b}_i),\\
o_t &= \sigma_{\mathrm{sig}}(\mathbf{W}_o x_t + \mathbf{U}_o h_{t-1} + \mathbf{b}_o),\\
c_t &= r_t \circ c_{t-1} + i_t \circ \sigma_{\tanh}(\mathbf{W}_o x_t + \mathbf{U}_o h_{t-1} + \mathbf{b}_o),\\
h_t &= o_t \circ \sigma_{\tanh}(c_t),
\end{aligned} \tag{2}$$

where r_t, i_t, and o_t are the outputs of gates (standard log-sigmoid recurrent layers); $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_i$, $\mathbf{U}_i$, $\mathbf{W}_o$, and $\mathbf{U}_o$ are weight matrices; $\mathbf{b}_r$, $\mathbf{b}_i$, and $\mathbf{b}_o$ are bias vectors; and $\circ$ denotes element-wise multiplication. c_t is a cell memory stat; it is calculated using the two weight matrices $\mathbf{W}_c$ and $\mathbf{U}_c$ and a bias vector $\mathbf{b}_c$.

The authors of [24] present a simplified version of gated model that gives results similar to LSTM. Gated recurrent unit (GRU) reduces the internal complexity of a unit; it is defined by

$$\begin{aligned}
z_t &= \sigma_{\mathrm{sig}}(\mathbf{W}_z x_t + \mathbf{U}_z h_{t-1} + b_z),\\
r_t &= \sigma_{\mathrm{sig}}(\mathbf{W}_r x_t + \mathbf{U}_r h_{t-1} + \mathbf{b}_r),\\
c_t &= r_t \circ h_{t-1},\\
h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_{\tanh}(\mathbf{W}_h x_t + \mathbf{U}_h c_t + b_h).
\end{aligned} \tag{3}$$

In GRU, the memory state is not separated from the output. The output depends only on the current input and the value of the previous output. GRU uses two gates z_t and r_t. As c_t represents the previous output after gating, there is no need to store it between timesteps. The numbers of weight matrices and bias vectors are reduced in GRU to six matrices ($\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_h$, and $\mathbf{U}_h$) and three bias vectors ($\mathbf{b}_r$, $\mathbf{b}_i$, and $\mathbf{b}_o$). Chung et al. compared GRU and LSTM in [25]. Both networks perform similarly and better than standard recurrent neural networks. The advantage of GRU lies in its simplicity, comparing to LSTM; therefore, we prefer GRU networks in our studies.

2.2. Similarity-Based Loss for a Neural Network. A GRU network produces a sequence of feature vectors in its final layer. For a sequence of n output vectors $(h_1, h_2, \dots, h_n)$, that is, the result of input $(x_1, x_2, \dots, x_n)$, we can calculate the average to obtain a feature vector f describing the whole music piece:

$$f(x_1, x_2, \dots, x_n) = \frac{1}{n}\sum_{t=1}^{n} h_t, \tag{4}$$

where the sequence $(h_1, h_2, \dots, h_n)$ is calculated according to (3). The standard approach for training recurrent networks for sequence classification is to use this vector as an input to a final nonrecurrent layer. A loss function is then calculated using mean square error or (after applying softmax function over outputs) cross entropy. We seek an adjustment to loss function that would take into account the properties of SVM as nonlinear classifiers and the fact we can simply ignore the need of a nonrecurrent layer if we use the network as a dedicated feature learner.

A particularly well-known way to improve the performance of SVMs is to use the so-called kernel trick. Assume an optimization problem that is posed in such a way that it does not require access to a full data matrix $\mathbf{D}$, but rather, a product of the matrix and its transpose $\mathbf{D}\mathbf{D}^{\mathrm{T}}$. Linear SVM is an example of such problem. Then, we can replace $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ with a matrix $\mathbf{K}(\mathbf{D}, \mathbf{D})$, built using a real-valued kernel function:

$$K(X, Y)_{ab} = k(x_a, y_b), \tag{5}$$

whereby $K_{a,b}$ denote an element of matrix $\mathbf{K}$ in ath row, bth column, while x_a denotes the ath row of matrix $\mathbf{X}$. In other words, the kernel function k replaces the inner product during optimization. If there is a mapping ϕ such that

$$k(x, y) = \phi(x)^T \phi(y), \tag{6}$$

we can say that the problem is instead being solved in an implicit feature space, where the coordinates of the classified samples x and y are $\phi(x)$ and $\phi(y)$, respectively. In this space,

certain classification problems which were not linearly separable in the original feature space may become linearly separable. Similarly, for regression problems in which linear regression produced a bad fit, regression in implicit feature space often improves results. The advantage of kernel trick is that it allows operating within the implicit feature space without actually calculating $\phi(x)$ and $\phi(y)$.

Kernel functions typically employed in SVM training can be understood as measures of similarity. Our intuition for the feature learning NN is, therefore, to attempt to replicate certain similarity relations between the annotations in the learned feature space. We can stack feature vectors calculated according to (4) for different files in the dataset as rows of a feature vector matrix **F**. Similarly, known annotation vectors for these files form an annotation matrix **A**. For regression problem in a multidimensional space, these annotations consist of all regressed values. For example, for a music piece annotated with two values regarding its position on valence-arousal plane, a vector of two real values is the annotation. For classification, we can consider one-hot encoding of classes. We can define a similarity-based loss function as

$$L(X) = \|\mathbf{K}(\mathrm{F}, \mathbf{F}) - \mathbf{K}(\mathbf{A}, \mathbf{A})\|, \tag{7}$$

where K can be built using an arbitrary notion of similarity $k(x, y)$, by analogy to kernel SVM, as in (5), and $\| \dots \|$ denotes Frobenius norm. For batch learning, which is currently the standard procedure for learning neural networks due to performance considerations, the matrices $\mathbf{K}(\mathbf{F}, \mathbf{F})$ and $\mathbf{K}(\mathbf{A}, \mathbf{A})$ can be calculated over batches instead of full dataset. We described this approach in less general terms in [19] as semantic embedding, borrowing the idea from the domain of text processing [26]. Semantic embedding in texts seeks to learn similarity between documents using pairs of similar and dissimilar files and could be considered a special case of the described idea (with cosine similarity as the k function and $\mathbf{K}(\mathbf{A}, \mathbf{A})$ being built as a matrix of ones and zeroes from known relation of similarity, rather than calculated from annotations).

2.3. Measures of Similarity between Vectors. To define a similarity-based loss, we need to define a similarity function that will be used. For the purpose of this study, we focus on three similarity measures:

(i) *Cosine*: the similarity notion that we used in the earlier paper [19], where we first tackled learning similarity. It was previously used in the approach to learning similarity between documents called semantic embedding.

(ii) *Radial basis function (RBF) kernel*: one of the popular kernels often employed in support vector machines and the one we use in the SVM classifier or regressor deployed on learned features.

(iii) *Polynomial kernel*: the other popular kernel employed in support vector machines which we use for comparison. We need this comparison to establish whether the performance gains are related to fitting specific similarity notion to the kernel employed in SVM or simply rewriting loss function to use similarity yields benefits over a loss function that tries to predict labels directly

Cosine similarity is a simple measure that normalizes both compared vectors, therefore ignoring their norm and only focusing on the direction (i.e., for a vector **x**, $\cos(x, 2x) = 1$). The function is defined as

$$k_{\cos}(x, y) = \frac{x^T y}{\|x\| \|y\|}. \tag{8}$$

Cosine similarity is bound between 0 and 1 regardless of space dimensionality, which may be a useful property for our purposes as annotation space and learned feature space could have largely varying dimensionalities. Radial basis function kernel is defined as

$$k_{\mathrm{rbf}}(x, y) = e^{-\gamma \|x-y\|^2}. \tag{9}$$

The exponent guarantees that the value is in the $(0, 1]$ interval and the similarity between two vectors never equals 0. In practice, the lower bound of this measure will be affected by the maximal distance between vectors which will exist in real datasets. For example, for annotation space of n dimensions, if we assume all labels can range from 0 to 1 (as in the Emotify dataset), the distance between two annotations can be at the most square root of n. The lower bound for similarity is therefore $e^{-\gamma n}$.

Polynomial kernel is defined as

$$k_{\mathrm{pol}}(x, y) = \left(x^T y + b\right)^p. \tag{10}$$

The polynomial kernel is not bound to a particular interval (although for even p result is always nonnegative), and the result is greater when comparing vectors with larger norms. Polynomial kernel properties are not theoretically fit for our task since dimensionality would largely affect the similarity score between vectors. However, in preliminary studies, we found it performed surprisingly well in classification tasks even despite the fact that SVM was using an RBF kernel. Therefore, we include it in the study as a possible RBF kernel alternative.

2.4. Datasets. We performed our experiments on five datasets, two for regression and three for classification. These datasets represent three distinctive label types, with focus on emotion recognition. Links to all datasets are provided at the end of the article, in the "Data Availability" statement. A short summary is presented in Table 1.

We chose both datasets that were previously tested in [19] and three complementary datasets. Complementary data represents an important form of emotion regression (predicting the values of valence and arousal) and two music classification tasks not concerning emotion. We found it important to extend our research to V-A emotion recognition, as it is the most common form of annotating emotion in the existing literature.

Table 1: Summary of the datasets.

Dataset	Label type	Task	Labels	Files
Lastfm	Emotion	Classification	4	2906
Emotify	Emotion	Regression	9	400
Songs	Emotion	Regression	2	744
GTZAN	Genre	Classification	10	1000
Ballroom	Dance style	Classification	9	754

The Lastfm dataset [27] is the largest one we test. It contains more than 2000 files annotated with labels inferred from user-generated tags on the music-centric social network https://www.last.fm/. There are four classes, representing four quadrants of a valence-arousal plane: happy, sad, angry, and relaxed. The labels are unreliable and the classification task very hard, with previous research showing 54% classification accuracy as the top result. Despite that, we believe it represents a realistic scenario of musical data acquisition and the problems one may face when attempting to infer emotional content from unregulated tagging by a large community. Songs in the dataset are 30-second long excerpts.

Emotify game dataset [28], similarly to Lastfm, is based on crowd-sourced annotations, although the gathering process was much more controlled. Nine emotional labels represent nine middle emotions of GEMS, and the predicted values represent the percentage of users agreeing that particular emotion is induced by a particular music piece. It is important to note the explicit distinction between induced emotion versus perceived emotion. The dataset focuses on the former, and as a result, disagreement between annotators is very common. This disagreement is in part a result of variables that cannot be accounted for by music alone, including individual mood during annotation and individual connotations regarding specific words used to describe emotions. Because of that, predictions one can obtain through music audio analysis are poor on average: regarding Pearson's R coefficient, the correlation between predicted and actual values achieved in the first paper on this dataset was 0.47, averaged between all emotions (i.e., less than 25% of variance explained). However, there is a significant variance in figures of merit between specific emotions. For example, the emotion of amazement is almost unpredictable (below 0.3 correlation), while for joyful activation, Pearson's R above 0.7 is possible to achieve. Excerpts in the dataset are of varying lengths, 30- to 60-second long.

MediaEval 2013 data, also known as "1000 songs" dataset [29], is a set created for machine learning benchmarking. It consists of 744 (after duplicate removal) unique song excerpts. The dataset was annotated by crowd workers on Amazon Mechanical Turk platform, separately in valence and arousal dimensions, with each song receiving ten annotations. The publicly available data contains both mean and standard deviation of these evaluations. Songs in the dataset are 45-second excerpts.

GTZAN [30] is a famous dataset concerning genre recognition, and one of the most cited of all music information retrieval datasets. While it has been criticized for faults in its construction [31], as our research does not concern genre recognition specifically, we find it acceptable to use GTZAN for the sake of comparison between presented methods. GTZAN contains 30-second excerpts and is annotated as a classification dataset with ten genre labels.

Ballroom data [32] was originally meant for tempo estimation tasks. However, as the dataset offers clear split between classes corresponding to different dancing styles, we use these labels as a basis for a classification problem. It is interesting, as the distinction between dance styles is significantly more dependent on tempo and rhythm than the usual MIR tasks. Eight dance styles represented in the dataset are chacha, rumba, samba, quickstep, tango, slow waltz, and Viennese waltz.

2.5. Dataset Enhancement. As training of neural network models is strongly dependent on the quantity of available data, research datasets currently available for MIR tasks may pose the problem of insufficient files. We approach this issue using dataset enhancement, artificially expanding the number of possible training samples.

We use the following dataset enhancement scheme: during training, in each epoch instead of full feeding sequences corresponding to all music files in the dataset to the network, we choose a short excerpt of each file. This excerpt contains frames from t to $t + 100$ for a t randomly drawn from a uniform distribution.

Regarding dataset size, this approach hugely increases the number of potentially different samples a neural network will see during training. Consider a dataset of 1000 files, which are represented by sequences of 1200 vectors, 40 elements in a vector. These numbers correspond to spectrograms of 30-second files with extraction parameters employed in this article. Dimensions of the dataset are therefore $1000 \times 1200 \times 40$. However, with the enhancement, every 1200-frame long sequence has $1200 - 100 = 1100$ possible shorter excerpts of length 100. We are therefore effectively sampling from 1,100,000 possible excerpts that are sequences of 100 vectors, that is $1{,}100{,}000 \times 100 \times 40$.

It should be noted that the samples largely overlap and the network is likely to finish training before seeing every possible one. Additionally, this approach equates to learning on 2.5-second-long excerpts, thus ignoring any dependencies between frames separated by a time interval longer than 2.5 seconds within a music file. Nevertheless, the enhancement scheme allows us to test feature learning methods in a very efficient manner. We have previously shown [33] that this approach improves both convergence speed and the final result of the learning process for multiple music classification tasks when compared to training neural networks (both GRU and LSTM) on complete music files.

2.6. Common Conditions of the Experiments. For each of the performed experiments, we kept a similar core structure of the neural network and parameters for the said network. The network consists of two GRU recurrent layers, respectively, 100 and 50 units, and an additional nonrecurrent output layer in case of the baseline approach. The network is trained with Adadelta adaptive gradient descent method

[34]. Implementation of neural network logic and gradient operations uses the Theano library [35].

For SVM, we use an existing implementation from the scikit-learn python library. The hyperparameters of SVM were fit on the smallest dataset, Emotify, and reused in other experiments. Parameter C was set to 1, and the RBF kernel was employed with $\gamma = 0.5$. We ensured that on other datasets, a change in SVM parameters does not alter the results drastically, but we did not attempt further tuning the parameters for each dataset, as the resulting boost in performance was small, at the expense of creating unrealistic experimental conditions.

As an input to the feature extracting NN, we use a mel-frequency spectrogram with logarithmic power scale. The inputs are normalized to zero mean and standard deviation equal to 1 over each frequency bin, for each dataset independently. Frames of spectrogram are 50 ms long with 25 ms overlap, and we use 40 mel-frequency bins (the parameters were derived from defaults in MIRToolbox [36], a popular MATLAB toolbox for MIR feature extraction).

The input sequence to a recurrent network consists of 80-element vectors. These vectors contain the values of 40 spectrogram bins and the approximate of the first derivative (change from previous value) for each bin.

3. Results and Discussion

In the experiments, we seek to establish whether the proposed approach of learning a feature extracting neural network and supplying learned features to another classifier or regressor can improve results. As the main proposition of this paper is to adjust the learning goal (i.e., loss function) for a feature learning NN to one based on a notion of similarity as well as use a specific classifier on the learned features, we need two baselines for comparison. First one is a result of a full neural network-based approach, where the GRU network directly predicts classes or regressed variables. The second one is a result of an SVM deployed on bottleneck features (representation in a penultimate layer) from the aforementioned neural network. Altogether, we will compare five approaches:

(i) Baseline neural network approach (*NN*): GRU neural network approach in which the network is trained to classify or predict continuous values with standard sum square error loss.

(ii) SVM with baseline feature learning approach (*FL*): features are taken from the penultimate layer of the GRU neural network trained to classify or predict, then an SVM is trained on them.

(iii) SVM with NN learning RBF similarity (*RBF-SL*): feature extracting GRU neural network is learned with similarity-based loss using RBF kernel ($\gamma = 0.5$) for similarity, then SVM is trained on output features of the network.

(iv) SVM with NN learning cosine similarity (*Cos-SL*): feature extracting GRU neural network is learned with similarity-based loss using cosine similarity, then SVM is trained on output features of the network.

(v) SVM with NN learning polynomial similarity (*Poly-SL*): feature extracting GRU neural network is learned with similarity-based loss using polynomial kernel ($p = 2$, $b = 0$) for similarity, then SVM is trained on output features of the network.

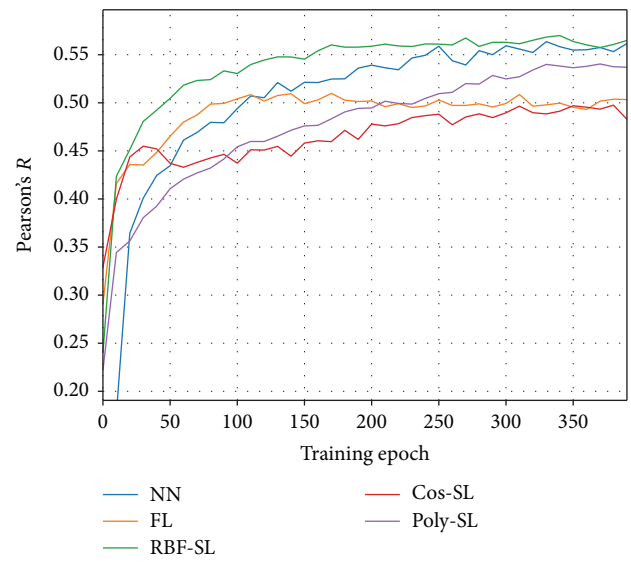


Figure 1: Emotify dataset, average prediction quality over all 9 GEMS emotions.

To demonstrate how the performance changes over the training process, we save the output of a feature extracting NN at every tenth epoch of its training. SVM is fully trained from zero at every one of these points to obtain a task-dependent measure of performance (accuracy for classification, Pearson's R for regression). We chose this way of presenting the results since, for the given dataset size, the time it takes to train SVM on learned features is a fraction of the time required to fully train a recurrent NN.

All presented results were obtained in 10-fold cross-validation experiments, in which the training-test split was applied to the learning of both the feature extracting NN and the SVM deployed on learned features afterwards.

3.1. Results on Emotion Regression Data. Results on emotion regression datasets are shown in Figures 1–3. On the plots of performance over the duration of training, we can see that the proposed approach with RBF kernel as a similarity measure achieves the best results and fastest convergence. This is consistent with our expectations, as RBF kernel was also used in the SVM model of regression. Compared to an SVM deployed on bottleneck features from a standard neural network, the loss function adjusted to learning similarity leads to improvements on all datasets. Compared to a purely NN approach, we can see either improvement or comparable performance. While cosine similarity measure consistently appears to perform the worst on the regression problems, it is hard to draw a comparison between polynomial kernel for similarity and SVM deployed on bottleneck features from

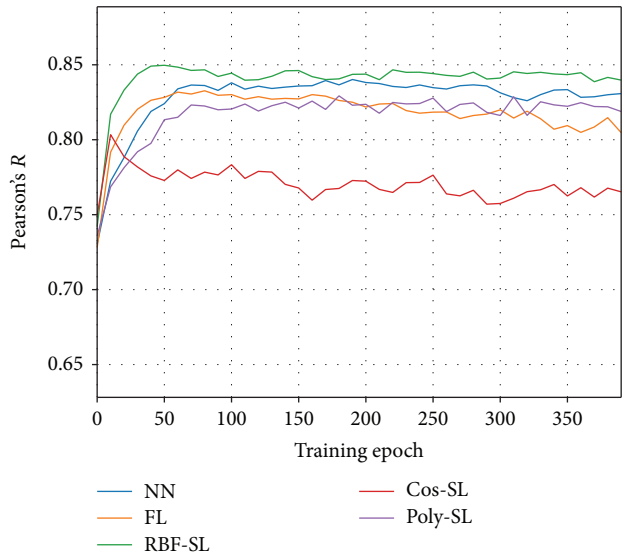


Figure 2: Performance on MediaEval2013 dataset, arousal prediction.

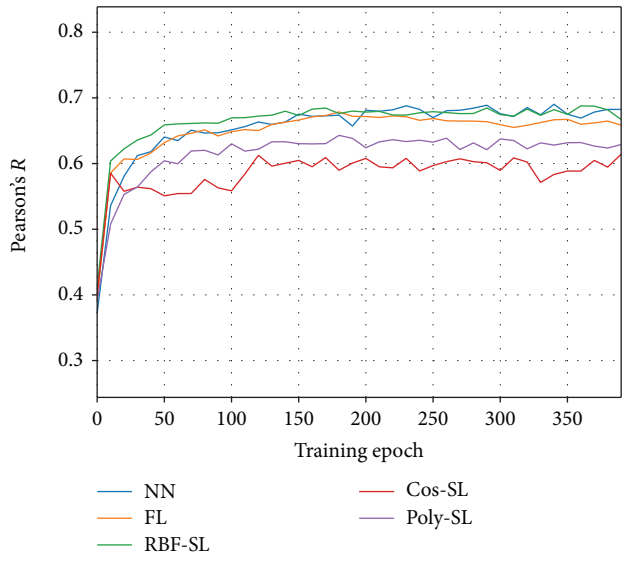


Figure 3: Performance on MediaEval2013 dataset, valence prediction.

regular NN. We can note that on the Emotify dataset, where using polynomial similarity yields improvement, the learning process is much slower than in other cases. Cosine similarity performs very badly on the MediaEval 2013 data, which can be explained by ignoring the norm of compared vectors. In V-A space, an emotional content labeled as 0.5 valence and 0.5 arousal will be vastly different than one labeled as 1 valence and 1 arousal.

In Table 2, we present the detailed results for recognition of specific emotions on the Emotify game dataset. The approach of learning RBF similarity within the feature extracting networks performs best for 7 out of 9 emotions. While results indicate a low quality of predictions, it can be noted that the proposed approach improves the worst results.

Table 2: Pearson's R on the Emotify dataset. Asterisk denotes SVM use.

Emotion	NN	FL*	RBF-SL*	Cos-SL*	Poly-SL*
Amazement	0.38	0.25	**0.40**	0.28	0.25
Solemnity	**0.56**	0.48	0.51	0.47	0.46
Tenderness	0.59	0.58	**0.62**	0.59	0.59
Nostalgia	0.58	0.56	**0.61**	0.53	0.56
Calmness	0.59	0.56	**0.60**	0.53	0.58
Power	0.54	0.49	**0.56**	0.52	0.52
Joyful act	0.69	0.67	**0.72**	0.65	0.68
Tension	**0.58**	0.56	0.57	0.51	0.56
Sadness	0.43	0.29	**0.48**	0.37	0.36

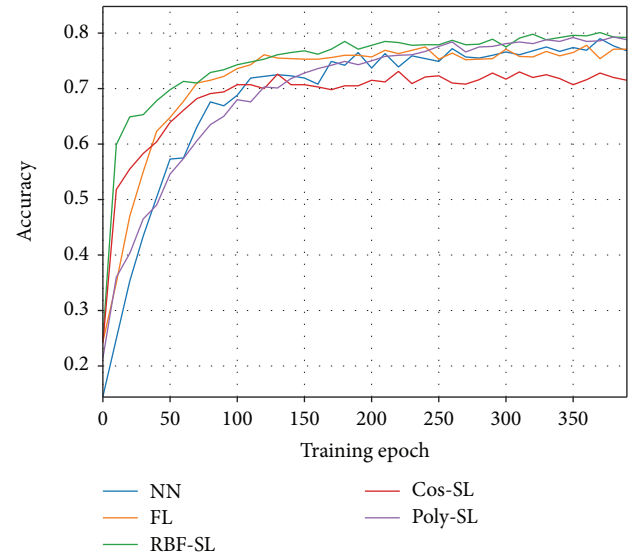


Figure 4: Classification performance on GTZAN dataset.

Notably, prediction of amazement reaches $R = 0.4$. This equates to the coefficient of determination $R^2 = 0.16$, that is, 16% of the variance in ground truth variable explained by the model. As previous results on the dataset indicated near-complete unpredictability of amazement category [11], the fact this one is above what is typically considered correlation by chance can be seen as relevant. For less subjective emotions, where making a prediction is more feasible, improvements can also be seen. In particular, the proposed approach is the only one where the model explains more than 50% of the variance for joyful activation ($R = 0.72$, $R^2 = 0.51$), but only if the similarity notion is chosen properly.

3.2. Results on Classification Tasks. Results on classification datasets GTZAN, ballroom, and Lastfm are shown in Figures 4–6. On the first two, we can see the proposed approach achieves the best performance with RBF kernel as a similarity measure.

In the proposed approach and SVM deployed on bottleneck features, the final result on the LastFM dataset is similar. However, the detailed look at the training process shows this is a result of overfitting on the part of the feature extracting

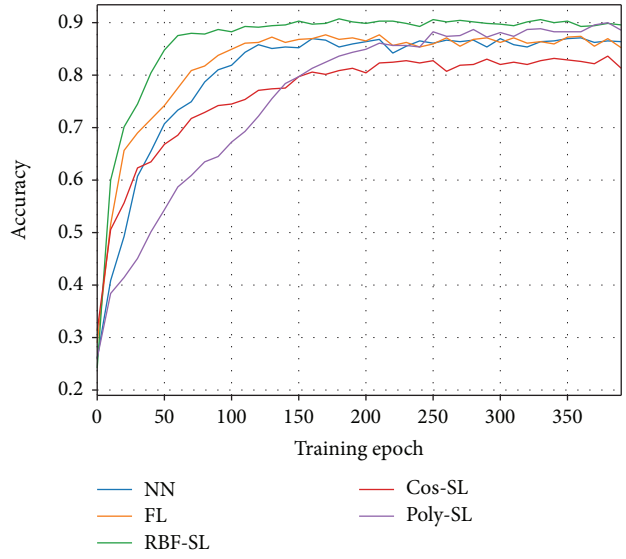


Figure 5: Classification performance on ballroom dataset.

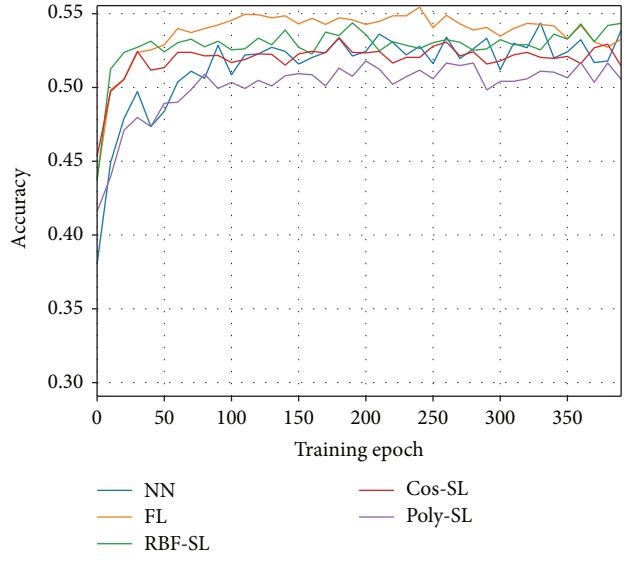


Figure 6: Classification performance on LastFM dataset.

NN, as in epochs 50–350 it achieves superior performance and only later drops to lower levels. It should be noted that LastFM data has the poorest quality of annotations among the examined datasets, which causes all of the tested approaches to be harder to evaluate.

Interestingly, polynomial kernel as a similarity measure appears to work significantly better on two of the tested classification datasets, and although it converges slowly, eventually it achieves the same results as RBF kernel on both GTZAN and ballroom data.

3.3. Statistical Significance of the Results. In cross-validation experiments, we observed that the proposed approach with appropriately selected similarity function either improves the baseline results or performs as well as the baselines on four of the examined datasets, while the results on LastFM data are inconclusive. To further confirm our conclusions rergarding the four datasets, we performed additional tests of best performing similarity function (RBF-SL) in comparison with regular NN and FL baseline approaches. These were repeated experiments on purely random 9 : 1 training-test split, without cross-validation, intended to gather a bigger sample size for testing the statistical significance of results. We repeated the random split experiment 100 times and tested the obtained results for statistical significance using Welch's t-test for unpaired samples with unequal variance. At the standard threshold ($p < 0.05$), we confirmed improvements after 400 epochs of training in comparison with NN baseline on datasets GTZAN and ballroom. In comparison with FL baseline, the improvements were confirmed on Emotify, as well as MediaEval2013. The RBF-SL approach did not perform worse than either of the two baselines in a statistically significant way on any of the datasets.

3.4. Conclusions and Future Work. From the presented results, we can conclude that the proposed approach of adjusting a loss function within the feature learning neural network to a similarity-based one can indeed improve the performance of an SVM later deployed on learned features. On all datasets, the proposed adjustment either outperforms purely NN-based approach or performs at least as well, when the learned notion of similarity is RBF kernel. This corresponds to the kernel used in the classifying SVM. When the learned notion of similarity is different, the performance can be vastly worse (cosine similarity), or comparable on some datasets, but worse on others (polynomial kernel). The modified loss function also shortens the learning of neural network feature extractor which, due to the complex nature of recurrent networks, is the most performance-demanding part of the learning process.

Our results are promising for the perspectives of future use for traditional machine learning approaches on musical data. While recent trends in machine learning focus on replacing older techniques with deep learning, in our experiments, best results are obtained when combining deep networks with a standard SVM approach. However, to achieve these results, the network has to be trained in a way that is adjusted to the specific classifier. A perspective for future research opens for creating similar adjustments targeting other standard machine learning approaches. One could also extend the possible future research to other types of data, where using deep learning on low-level representations is preferable to the extraction of features, for example, images and videos.

Data Availability

This study is based on previously reported data [27–30, 32]. As of writing the article (April 2018), music files and annotations for all of the examined datasets are available online at the following URLs: (i) Lastfm https://code.soundsoftware.ac.uk/projects/emotion-recognition, (ii) Emotify http://www.projects.science.uu.nl/memotion/emotifydata, (iii) MediaEval2013 http://cvml.unige.ch/databases/emoMusic, (iv) Ballroom http://mtg.upf.edu/ismir2004/contest/tempoContest/

node5.html, and (v) GTZAN http://marsyasweb.appspot.com/download/data_sets.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the statutory funds of the Department of Computational Intelligence, Faculty of Computer Science and Management, Wrocław University of Science and Technology.

References

[1] M. Defferrard, S. P. Mohanty, S. F. Carroll, and M. Salathe, "Learning to recognize musical genre from audio," 2018, https://arxiv.org/abs/1803.05337v1.

[2] C. C. Pratt, "Music as the language of emotion," *Bulletin of the American Musicological Society*, vol. 11, pp. 67-68, 1948.

[3] K. R. Scherer and M. Zentner, "Emotional effects of music: production rules," in *Music and Emotion: Theory and Research*, pp. 361–392, Oxford University Press, 2001.

[4] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[5] Y. E. Kim, E. M. Schmidt, R. Migneco et al., "Music emotion recognition: a state of the art review," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 255–266, Utrecht, Netherlands, 2010.

[6] J. Skowronek, M. McKinney, and S. van de Par, "A demonstrator for automatic music mood estimation," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pp. 345-346, Vienna, Austria, 2007.

[7] C. Laurier, O. Lartillot, T. Eerola, and P. Toiviainen, "Exploring relationships between audio features and emotion in music," in *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009)*, pp. 260–264, Jyväskylä, Finland, 2009.

[8] Y. H. Yang, Y. C. Lin, Y. F. Su, and H. H. Chen, "A regression approach to music emotion recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 448–457, 2008.

[9] M. Zentner, D. Grandjean, and K. R. Scherer, "Emotions evoked by the sound of music: characterization, classification, and measurement," *Emotion*, vol. 8, no. 4, pp. 494–521, 2008.

[10] A. Nasridinov and Y. H. Park, "A study on music genre recognition and classification techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 4, pp. 31–42, 2014.

[11] A. Aljanaki, F. Wiering, and R. Veltkamp, "Computational modeling of induced emotion using gems," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pp. 373–378, Taipei, Taiwan, 2014.

[12] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: new directions for music informatics," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 461–481, 2013.

[13] U. Schimmack and R. Rainer, "Experiencing activation: energetic arousal and tense arousal are not mixtures of valence and activation," *Emotion*, vol. 2, no. 4, pp. 412–417, 2002.

[14] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 681–686, Miami, FL, USA, 2011.

[15] Y. Vaizman, B. McFee, and G. Lanckriet, "Codebook-based audio feature representation for music information retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1483–1493, 2014.

[16] J. Jakubik and H. Kwaśnicka, "Sparse coding methods for music induced emotion recognition," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, pp. 53–60, Gdańsk, Poland, 2016.

[17] Y. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2392–2396, New Orleans, LA, USA, 2017.

[18] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[19] J. Jakubik and H. Kwaśnicka, "Music emotion analysis using semantic embedding recurrent neural networks," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 271–276, Gdynia, Poland, 2017, IEEE.

[20] Y. Tang, "Deep learning using linear support vector machines," in *International Conference on Machine Learning 2013: Challenges in Representation Learning Workshop*, Atlanta, GA, USA, 2013.

[21] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, pp. 141–149, Suzhou, China, 2017.

[22] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, pp. 347–352, Washington, DC, USA, 1996.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, 2015.

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[26] H. Wu, M. R. Min, and B. Bai, "Deep semantic embedding," in *Proceedings of Workshop on Semantic Matching in Information Retrieval co-located with the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SMIR@SIGIR 2014)*, pp. 46–52, Gold Coast, QLD, Australia, 2014.

[27] Y. Song, S. Dixon, and M. Pearce, "Evaluation of musical features for emotion classification," in *Proceedings of the 13th*

International Society for Music Information Retrieval Conference (ISMIR 2012), pp. 523–528, Porto, Portugal, 2012.

[28] A. Aljanaki, F. Wiering, and R. C. Veltkamp, "Studying emotion induced by music through a crowdsourcing game," *Information Processing & Management*, vol. 52, no. 1, pp. 115–128, 2016.

[29] M. Soleymani, M. N. Caro, E. M. Schmidt, C. Y. Sha, and Y. H. Yang, "1000 songs for emotional analysis of music," in *Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia - CrowdMM '13*, Barcelona, Spain, 2012.

[30] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[31] B. L. Sturm, "The GTZAN dataset: its contents, its faults, their effects on evaluation, and its future use," 2013, https://arxiv.org/abs/1306.1461.

[32] K. Seyerlehner, G. Widmer, and D. Schnitzer, "From rhythm patterns to perceived tempo," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pp. 519–524, Vienna, Austria, 2007.

[33] J. Jakubik, "Evaluation of gated recurrent neural networks in music classification tasks," in *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture And Technology, ISAT 2017* of *Advances in Intelligent Systems and Computing*, pp. 27–37, Szklarska Poręba, Poland, 2018.

[34] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," 2012, https://arxiv.org/abs/1212.5701.

[35] Theano Development Team, "Theano: a python framework for fast computation of mathematical expressions," 2016, https://arxiv.org/abs/1605.02688.

[36] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," in *International Conference on Digital Audio Effects (DAFX 2018)*, pp. 237–244, Aveiro, Portugal, 2007.