

Election Algorithms Applied to the Global Aggregation in Networks of Comparators

Łukasz Sosnowski
 Dituel Sp. z o.o.
 ul. Ostrobramska 101 lok. 206,
 04-041 Warsaw, Poland
 Systems Research Institute,
 Polish Academy of Sciences
 ul. Newelska 6, 01-447 Warsaw, Poland
 e-mail: l.sosnowski@dituel.pl

Andrzej Pietruszka
 Institute of Mathematics,
 University of Warsaw
 ul. Banacha 2,
 02-097 Warsaw, Poland
 Dituel Sp. z o.o.
 ul. Ostrobramska 101 lok. 206,
 04-041 Warsaw, Poland
 e-mail: a.pietruszka@dituel.pl

Stanisław Łazowy
 Section of Computer Science,
 The Main School of Fire Service
 ul. Słowackiego 52/54,
 01-629 Warsaw, Poland
 e-mail: lazowy@inf.sgsp.edu.pl

Abstract—The paper shows the application of election algorithms in networks of comparators. We have described and adopted six election methods which have been used as an aggregator of partial results. We have performed experiments on the data gathered at the fire ground. All of them have been well described and results have been compared. The paper includes a discussion and interpretation of results obtained. It indicates the algorithm with the greatest potential to adapt and to obtain the best results.

Index Terms—Networks of comparators, election algorithms, aggregation of partial results, similarity based reasoning, compound objects, fire rescue actions

I. INTRODUCTION

SIMILARITY [1] is one of the fundamental aspects of reasoning methods used in AI. There are many techniques used by researchers to implement resemblance in practice. We can find many kinds of neural networks [2] which resolve pattern recognition problems, fuzzy sets [3] which are able to model complicated processes, rough sets [4] to perform knowledge discovery in data and many others. All of them are well-known and explored. There are many extensions of mentioned methods developed, e.g. neural networks with compound signals [5]. All these methods are specialized in resolving one of the defined problems.

In previous researches a common approach to similarity-based reasoning was developed. The same workflow is used to resolve various problems. The basic element of the authors' concept is a dedicated logical component called comparator [6]. It is responsible for examining the resemblance of a given feature between an input object and reference objects [7]. The comparator can be formally described as a function $C_B : A \rightarrow 2^{B \times [0,1]}$, where A is a set of input objects and B is a set of reference objects. Comparator outcome takes a form of weighted subsets of reference objects $C_B(a) =$

The research was supported by the Polish National Centre for Research and Development (NCBiR) - Grant No. O ROB/0010/03/001 in the frame of Defence and Security Programmes and Projects: "Modern engineering tools for decision support for commanders of the State Fire Service of Poland during Fire&Rescue operations in the buildings

$F(\{(b, g(\mu(a, b))) : b \in B\})$, where F is a function responsible for filtering partial results of a single comparator, e.g. *min*, *max*, *top*. Furthermore, $\mu(a, b)$ is a membership function of the fuzzy relation [3], which returns a similarity degree between $a \in A$ and $b \in B$, and $g(x)$ is an activation function which filters out results that are too weak. We put

$$g(x) = \begin{cases} 0 & : x < p, \\ x & : x \geq p \end{cases} \quad (1)$$

where p denotes the lowest acceptable similarity. One may also introduce some constraints which make $\mu(a, b) = 0$ based on the so-called exception rules [6].

The approach is based on a network of such comparators. This concept makes it possible to design a structure-driven solution as well as a flat one. The network consists of layers. They include comparators, aggregators and translators [8].

There are two types of aggregators: local and global. The functionality of a local aggregator comes down to selecting the best results for a given layer based on partial results. The functionality of the second one is focused on the synthesis of results of individual layers in order to calculate the final result.

The translator is an unit expressing the results of the one layer by objects existing in another layer. The general scheme of the type of network in question is shown in Figure 1. Complete information of the construction and operation of a network of comparators is not the subject of this article. It has been described well in the previous publication [8].

This article attempts to explore several different global aggregation methods and compare the results obtained by means of such methods. The authors also attempt to verify the importance of selection of the best results at the final stage of processing in this kind of networks. Quality level of this selection is expressed by the value of efficiency measures.

This research was motivated by authors' previous experiments. It was noticed that the final aggregation method may have a pronounced impact on results achieved. It can improve the efficiency of the model regardless of the way of evaluating

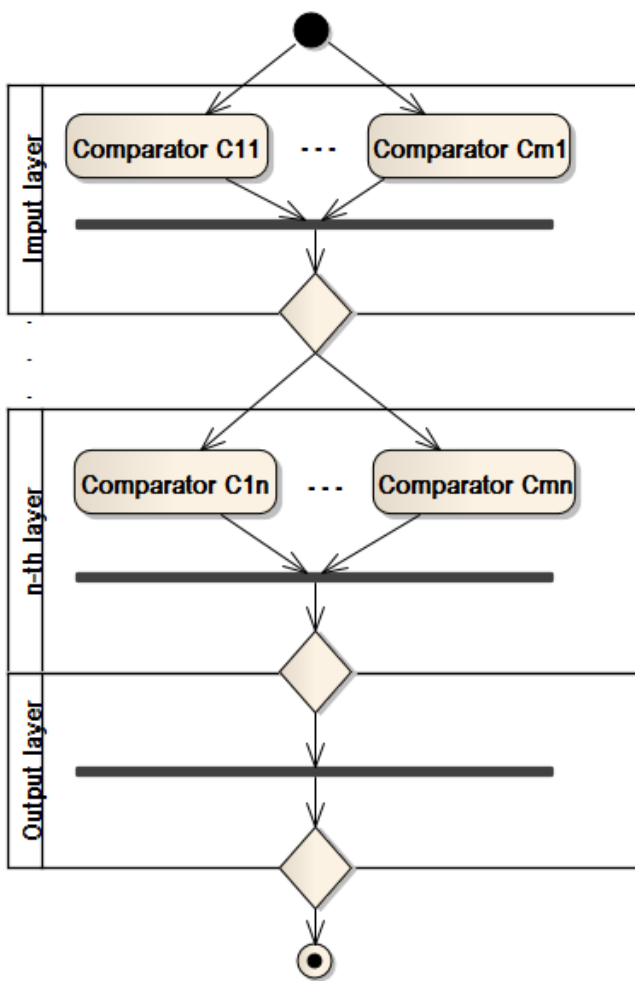


Fig. 1. General scheme of network of comparators. This is not an UML activity diagram. It only uses similar notation: oval boxes represent comparators, black horizontal lines are aggregators, the one in the output layer is a global aggregator. Diamonds between layers are translators.

similarity. Such comparison of voting algorithms has already been performed several times in the field of machine learning [9], [10]. The authors' work focuses strictly on the area of network comparators. Election algorithms were chosen as a group of methods which is well-known and described in literature. They are not the only methods which could be used, but they are easily accessible and implementable. Experiments use results obtained earlier [11] and take into account only the aspect of optimizing the process of global aggregation. This fact means that already calculated similarity results (from a previous research) are used and now attention is paid only to selecting a final set of objects. This is the main subject of this article called *global aggregation*.

The paper is organized as follows: the first section contains introductory information about context and a background of the problem presented. The second section presents the prime example used to perform the experiments, certain context required to be known to understand the results. The third

section contains the description of methods used to conduct an experiment. It also contains a detailed description of elections algorithms used and their main properties. The fourth section presents data used in experiments performed and results obtained. The subsequent section discusses results and describes the best and the worst methods. It also introduces criticism of the methods used. The final part contains a brief summary.

II. CONTEXT AND MOTIVATION EXAMPLE

The main example concerns the Fire&Rescue (F&R) actions and emerging risks during the rescue activity. Threats defined in the risk matrix [12] and objects that have vulnerabilities to these threats are taken into account. This matrix is used by fire brigades in certain countries to determine the prevalence of potential risks at the fire ground based on information collected (observation, interview, etc.).

TABLE I
RISK MATRIX USED FOR EVALUATING THE QUALITY OF FIT BETWEEN RESEMBLED ACTIONS. LEGEND: A1 - FEAR, A2 - TOXIC SMOKE, A3 - RADIATION, A4 - FIRE SPREADING, C - CHEMICAL SUBSTANCES, E1 - COLLAPSE, E2 - ELECTRICITY, E3 - DISEASE OR INJURY, E4 - EXPLOSION

Risk/object	A1	A2	A3	A4	C	E1	E2	E3	E4
People (ME)									
Animals (T)									
Environment (U)	-								
Property (S)	-	-				-	-	-	
Rescuers (MA)								-	
Equipment (G)	-	-						-	

The previous research concerned the automation of acquiring potential risks using text descriptions and repository of historical F&R actions. The solution is based on the similarity of actions and assumption that the most resembled ones have a similar list of risks occurring. The resemblance takes into account domain knowledge acquired from experts and injected in a form of measures. In order to perform such reasoning a model of F&R action represented by a set of attributes is necessary. Data from the EWID¹ system are used to create it as well as domain knowledge from experts (e.g. division of F&R action into phases).

On this basis, the following division into stages can be distinguished: notification, disposal, recognition and activities. It is quite a rare division due to the limited number of available attributes.

Notification refers to the act of transfer of information about threat. This phase collects attributes related to time, place and approximate description of event. It contains basic information necessary to make a decision about what forces and resources should be disposed.

Disposals contain quantitative data of already disposed forces and resources. There is information on the number of rescuers, cars and equipment used for the event in question. In addition, there is a number of units of other services (medical, police, etc.) that took part in the F&R action.

¹Polish Incident Data Reporting System used by Polish State Fire Service

Recognition is the stage at which rescuers carry out inspection, search and identification of the situation at the scene after arrival at the place of event. There are attributes describing dimensions of the place of event, building, size of event and also information about the existence of internal hydrants, smoke detectors, etc.

Actions form the actual start of firefighting operations. After collecting the required information at previous stages, firefighters start activities related to the neutralization of threats. They prepare an action strategy, assign tasks for rescuers and decide whether to use the specialized equipment.

These four stages are not independent. They are ordered in a sequence. If the notification materialized the disposal cannot be realized.

The solution mentioned earlier designates subsets of the most similar actions from the perspective of individual parts. The similarity ranking of each part may comprise a different list of preferred objects in order of significance. At this point a method has to be implemented, consisting in the combination of partial results into one coherent answer of the system which is a final result in form of subset of F&R objects with assigned risk labels.

This method is a decision problem of selecting the object that best meets the preferences of individual parts of F&R (based on similarity). In other words, it is analogous to the election case, where support for a candidate is expressed by voting. This analogy was the motivation for carrying out research on the application of elections algorithms to final selection of the results set (global aggregation). It was also an impulse to examining the importance of the choice of method. The authors were interested in the type of impact of the choice of election methods on final results achieved at the fire ground.

III. METHODS

The global aggregation method is a part of a bigger solution mentioned in Section II. Therefore, the full path which must be passed in order to obtain the results described further will be presented here.

The first step of the proposed solution is to design a network of comparators. The network is based on expert knowledge which should be part of the F&R ontology. It consists of concepts and their relations. It describes different aspects of action using evidenced concepts [13].

At the beginning division of an F&R action provided by expert is taken into account and described in Section II. The authors seek a possibility of comparing single parts and having partial results from each of them. This division consists of four parts: notification, disposal, recognition, actions. Each of them is represented by a composite comparator [14]. It means that each of them is independent sub-net. *Notification* consists of nine comparators responsible for examining different features associated with notification of fire. The complete list of these comparators is shown in Table II. The next phase is disposal. The sub-net of *disposal* consists of thirteen comparators. All these features are connected with a group of activities assigned to a disposal part. Other parts are constructed in an

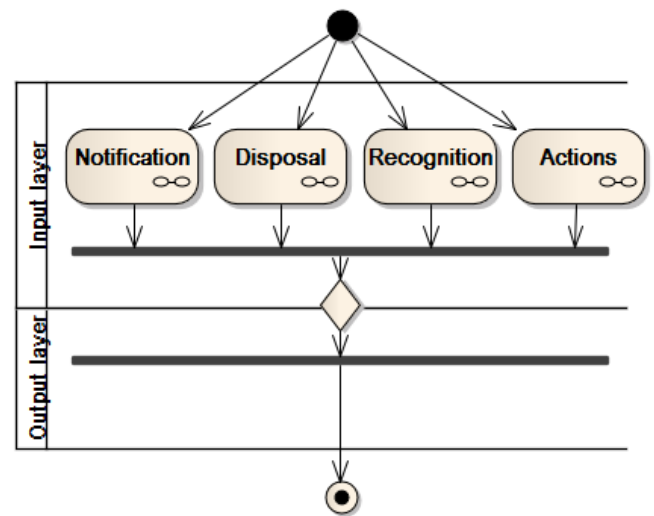


Fig. 2. The scheme of the network of comparators used for the designation of the most similar set of F&R actions. The oval boxes represent composite comparators. The translator in the output layer was skipped because there translation objects are not necessary in this case.

analogous manner. *Recognition* includes eleven comparators and *actions* consists of ten of them. Tables III, IV and V show respectively the detailed list of comparators for particular part of action. These tables also contain similarity measures applied in a particular comparator. This information is required for complete implementation.

A particular comparator has a function to evaluate similarity between a given pair of objects. The pairs are created in combination of the input object and the reference object (taken from the F&R repository one by one). Each pair has assigned value of similarity determined by a given comparator. The calculation for a single input object implies a number of local results sets in the form of $\{(b_i, \mu(a, b_i)) : b_i \in B\}$, where B is a set of reference objects.

Our solution consists of two parts. The first one is responsible for building a similarity model and computing partial results of resemblance. The second one is dedicated to optimization and aggregation of results. These two methods are classified as lazy [15] and eager types [16] respectively. The similarity model evaluates a given pair after calculating similarity for each comparator and it is a lazy one. On the other hand, aggregation of partial results requires learning certain parameters (in certain cases) or making certain decisions earlier. This part is classified as eager methods, e.g. learning weights for weighted average aggregator using genetic algorithm.

A *leave one out* [17] method is used to obtain four matrices with partial results. Each object is treated once as input objects against the remaining objects of the reference set. Such processing is performed for every single object one by one.

Our method can evaluate partial similarities and aggregate them to a higher level of resemblance and finally to a global one. The aggregator is a dedicated unit to handle such processes.

TABLE II
COMPARATORS OF FEATURES FOR NOTIFICATION PHASE

Comparator	Description	Comparison function
Time	F&R time. t_{a1} , t_{a2} - action time in compared objects, t_{e1} , t_{e2} - extinguish time in compared objects.	$\frac{t_1+t_2}{2}$, where $t_1 = \begin{cases} 1.0 : & t_{d1} = 0 \\ 1 - \frac{t_{d1}}{\max(t_{a1}, t_{a2})} : & t_{d1} \neq 0 \end{cases}$ $t_2 = \begin{cases} 1.0 : & t_{d2} = 0 \\ 1 - \frac{t_{d2}}{\max(t_{e1}, t_{e2})} : & t_{d2} \neq 0 \end{cases}$ where $t_{d1} = t_{a1} - t_{a2} $, $t_{d2} = t_{e1} - t_{e2} $.
Event place	F&R place. Places divided into a three level hierarchy. The first level is the most general.	$\mu(a, b) = \begin{cases} 1.0 : & a_3 = b_3, \\ 0.8 : & a_2 = b_2, a_3 \neq b_3, \\ 0.55 : & a_1 = b_1, a_2 \neq b_2 \end{cases}$ where $a_n, b_n, n \in 1..3$ - level hierarchy.
Access	Access to F&R place. The Access is divided into a two level hierarchy. The first level is the most general.	$\mu(a, b) = \begin{cases} 1.0 : & a_2 = b_2 \\ 0.55 : & a_1 = b_1, a_2 \neq b_2 \end{cases}$ where $a_n, b_n, n \in 1..2$ - level hierarchy.
Building height	Building height for F&R place. Building heights are divided into clusters.	$\mu(a, b) = ms[C_a][C_b]$ where ms - matrix of similarity between clusters, C_a, C_b - cluster values for building height in compared objects.
Building type	Building type for F&R like a building low, a building medium, a building high.	$\mu(a, b) = \begin{cases} 1.0 : & a = b \\ 0.0 : & a \neq b \end{cases}$ where a, b - building types in compared objects.
Notification channel	Notification channel about fire, e.g. by phone, by radio.	The same as above.
Entity notifier	Entity notifier about fire, e.g. an employees.	The same as above.
ZL category	Building category risk to humans.	$\mu(a, b) = ms[a][b]$ where ms - matrix of similarity, a, b - building category risk to humans.
Distance	Distance between the first units and the F&R.	$\mu(a, b) = \begin{cases} 1.0 : & a = b \\ 1 - \frac{ a-b }{6} : & a \neq b \end{cases}$ where a, b - distance in compared objects.

It has been noted in the research that the elections algorithms theory [18] is one of the approaches available, dealing with methods of selection of the best candidate for a group of voters. In this case, similarity results can be treated as voting results, comparators as voters and reference objects as candidates. The optimization task is to select the best candidate who will become a winner acceptable for the majority of voters.

There are a lot of known and available election algorithms. Six of them have been chosen, those which are the most popular, well-known or fulfil important criteria (e.g. majority, condorcet winner, condorcet loser, etc.) [19].

TABLE III
COMPARATORS OF FEATURES FOR DISPOSAL PHASE

Comparator	Description	Comparison function
Firefighters	How many firefighters were sent to F&R.	$\mu(a, b) = \begin{cases} 1.0 : & a = b \\ 1 - \frac{ a-b }{\max(a, b)} : & a \neq b \end{cases}$ where a, b - number of firefighters in compared objects.
Vehicles	How many vehicles were sent to F&R.	The same as above.
Firefighting cars	How many firefighting cars were sent to F&R.	The same as above.
Special cars	How many special cars were sent to F&R.	The same as above.
Additional vehicles	How many additional vehicles were sent to F&R.	The same as above.
Ambulances	How many ambulances were sent to F&R.	The same as above.
Power emergencies	How many power emergencies were sent to F&R.	The same as above.
Gas emergencies	How many gas emergencies were sent to F&R action.	The same as above.
Forest services	How many forest services were sent to F&R.	The same as above.
Police	How many police services were sent to F&R.	The same as above.
City guards	How many city guards were sent to F&R.	The same as above.
Other services	How many other services were sent to F&R.	The same as above.
Other region vehicles	How many vehicles from other regions were sent to F&R.	The same as above.

A. Plurality voting system

This is a single winner voting system. Each voter votes for one candidate only [20]. The winner is the one who receives the highest number of votes. This voting system is very easy for voters and it is easy to implement. However, it fails to provide information about preferences and support for individual candidates only for the most supported one. This is quite a popular method in real elections.

Implementation performs voting for each input object iteratively. For a given input object the candidate with the highest score is selected. The score is calculated from results of comparators. The winner for given comparator is the first candidate with the maximum similarity value. After this selection there are four votes. The global winner is the one with the highest global score. In the case of a tie, the first candidate is obtained.

TABLE IV
COMPARATORS OF FEATURES FOR RECOGNITION PHASE

Comparator	Description	Comparison function
Dust explosion	During F&R there was the dust explosion.	$\mu(a, b) = \begin{cases} 1.0 : a = b \\ 0.0 : a \neq b \end{cases}$ where a, b - the dust explosion in compared objects.
Gas explosion	During F&R there was the gas explosion.	The same as above.
Object dimensions	Object dimensions for F&R place. The object dimension was divided into clusters.	$\mu(a, b) = \begin{cases} 0,8 + (0,2 * f) : diff = 0 \\ 0,5 + (0,3 * f) : diff = 1 \\ 0 : diff > 1 \end{cases}$ where $f = (1 - \frac{ a-b }{\max(a,b)})$, $diff$ - difference between clusters, a, b - dimensions in compared objects.
Event size	Event size for F&R.	$\mu(a, b) = \begin{cases} 1.0 : a = b \\ 1 - \frac{ a-b }{\max(a,b)} : a \neq b \end{cases}$ where a, b - event size of compared objects.
Without PSP	The fire was extinguished without PSP.	$\mu(a, b) = \begin{cases} 1.0 : a = b \\ 0.0 : a \neq b \end{cases}$ where a, b - fire was extinguished without PSP in compared objects.
Fire cause	Fire cause. Fire causes are divided into clusters.	$\mu(a, b) = ms[C_a][C_b]$ where ms - matrix of similarity between clusters, C_a, C_b - cluster values for fire cause in compared objects.
Internal hydrants	Status internal hydrants in a building during F&R.	$\mu(a, b) = ms[a][b]$ where ms - matrix of similarity, a, b - statuses of internal hydrants.
Smoke devices	Status smoke devices in a building during F&R.	The same as above.
Fire extinguishing	Status fire extinguishing in a building during F&R.	The same as above.
Auto transmission	Status auto transmission system in a building during F&R.	The same as above.
Fire detection	Status fire detection system in building during F&R.	The same as above.

This method has a computational complexity estimated at $O(N)$. This provides for good computational properties in terms of time and computational power consumption.

B. Borda count

This is also a single winner voting method where voters produce a ranking of candidates in order of preferences [21]. The candidate receives a number of points connected with a position in ranking. The higher position in the ranking the more points candidate gets. There are various scoring methods: promoting higher place more than the lower, linear or specific one (e.g. only first three places are scored). The

TABLE V
COMPARATORS OF FEATURES FOR ACTIONS PHASE

Comparator	Description	Comparison function
Extinguishing on offensive	Flag indicates than during F&R extinguishing was used in offensive activities.	$\mu(a, b) = \begin{cases} 1.0 : a = b \\ 0.0 : a \neq b \end{cases}$ where a, b - extinguishing in offensive activities in compared objects.
Extinguishing in defense	Flag indicates than during F&R extinguishing was used in defensive activities.	The same as above.
Fire extinguisher	Fire extinguishers used during F&R with values.	$\mu(a, b) = \begin{cases} 1.0 : a = b \\ 1 - \frac{ a-b }{\max(a,b)} : a \neq b \end{cases}$ where a, b - number of fire extinguishers in compared objects.
Extinguishing media	Used extinguishing media during F&R.	The same as above.
Medical	Medical assistance provided during F&R.	The same as above.
Actions taken	Set of activities taken during F&R. Matrix of similarity defined by an expert.	$\mu(A, B) = \forall r \in R, r \in A, r \in B, \sum_{ms[a][b]} n$ where ms - matrix of similarity, A, B - sets of activities in compared objects, R - a reference set, n - size the reference set.
Medical actions	Set of medical activities taken during F&R.	The same as above.
Activities place	Place of activities taken during F&R.	The same as above.
Equipment used	Equipment used during F&R.	The same as above.
Water supply	Water supply methods during F&R.	The same as above.

winner is the candidate with the highest point result.

Implementation treats comparators as four voters. Each voter gives a ranking for all candidates. Rankings are created on the basis of similarity value for a given input object and particular candidates. Candidates are reference objects. Each candidate receives points. Linear scale is used. The candidate of the first place takes maximum number of points (in this case 406). The score function is given by the following formula:

$$score(a) = n(C) + 1 - RankPos(a), \tag{2}$$

where $n(C)$ -quantity of candidates, $RankPos(a)$ - position in ranking of candidate a . After that all points from particular rankings are summed up for each candidate. The last part is selecting a candidate with a maximum number of points. In the case of a tie, the first candidate with the maximum score is obtained.

This method is characterized by a very good computational complexity estimated at $O(N)$. One of the variable parameter

is a method of allocating points. It is an important procedure because it favors higher ranking position.

C. Copeland's method

This is a condorsset [19] method which returns results in form of a ranking. The score for setting a ranking position is calculated from the number of wins in pairs minus the number of defeats [22]. This is a round-robin tournament methods which is easy understandable and easy implementable in software solutions. The winner is a candidate with the highest score. Implementation performs this method for each input object independently. For a given input object it starts with creating the cartesian product of candidates (F&R objects). The winner is calculated for each pair. Every winner is computed with data from four comparators (voters). Each of them specifies which similarity value (between input object and given candidate) is bigger. The local winner gets one point, the local loser gets minus one point. In the case of a tie, both take a zero point. After resolving this problem for all comparators, points for both candidates are summed up. The winner of a single duel is the one who scores more points. This procedure is repeated for each pair of candidates to create full tournament table with score for each candidate. The global winner for the current input object is the candidate with the maximum global score calculated in a following way:

$$score(a) = wins(a) + defeats(a) \quad (3)$$

where $wins(a)$ is a number of points for wins (positive value) for candidate a , $defeats(a)$ is a number of points for defeats (negative value).

This method has a computational complexity estimated at $O(N^2)$. It does not provide any parameters to set.

D. Approval voting

This is a single winner voting method where each voter may approve or disapprove of any candidate from a ballot. It means that the voter has to specify approval or disapproval as regards each candidate. Consequently, the ballot designates the accepted set of candidates for a particular voter. The winner is the candidate who has the highest number of votes of approval [23].

This algorithm has been implemented by means of four comparators as voters and 405 reference objects as candidates. Approval voting is performed for each input object (406 times). This method has been adapted to the requirements in the following way: the approval factor is found, starting from the biggest one (1.0); then it is reduced by 0.01 in the case of failure to obtain majority in voting. Then, the approval factor is the threshold for similarity value of pair of objects. If the resemblance value is greater than or equal to the threshold, the vote is interpreted as approved or as disapproved for a candidate who is the reference object in given pair. In the case of a tie, first candidate with the highest number of *approved* votes is obtained. For each input object the winner is calculated in the same way.

This method is characterized by a very good computational complexity estimated at $O(N)$. This allows for very efficient calculation of the final results. Implementation presented could have various values of the reduction factor. It has impact on the quality of results and the speed of calculation.

E. Range voting

This is a single winner voting system. The voting is realized by rating ballots. A rate scale is specified, e.g. $[0, 1]$ or $[0, 100]$. Voters rate each candidate with own score matched with a fixed scale [24]. All candidates scores are summarized. The winner is a candidate with the highest sum of points. If certain candidates are not scored, the zero value is assigned. In general, all candidates should be rated.

Implementation takes on the form of calculating the mean value for each pair (input object and reference object). There are four comparators. Each of them provides similarity value for a given pair. The mean value is calculated from these four similarities for each pair.

This method has a computational complexity estimated at $O(N)$. It does not provide any parameters.

F. Weighted voting

This is a voting system which makes it possible to favour certain voters. In real life, one can find this kind of voting on boards of companies, where there are different shares or stocks [25]. The weight connected with a vote makes it stronger or weaker depending on the value of the weight. In this particular case, if all weights are equal, the system is identical to *range voting*.

In this case, voting is implemented in a similar way to *range voting*. This method works only if weights are given. Weights are indicated as a w_i where i is a number of comparator.

The solution has been expanded by adding the sub-optimal learning procedure determining weights for voting. A genetic algorithm [26] is used in order to find weights which give the highest evaluation score for this kind of voting. There are four weights described by the following formulas:

$$w_1 = \frac{n}{n + d + r + a} : (n + d + r + a) \neq 0 \quad (4)$$

$$w_2 = \frac{d}{n + d + r + a} : (n + d + r + a) \neq 0 \quad (5)$$

$$w_3 = \frac{r}{n + d + r + a} : (n + d + r + a) \neq 0 \quad (6)$$

$$w_4 = \frac{a}{n + d + r + a} : (n + d + r + a) \neq 0 \quad (7)$$

where n - factor responsible for *notification*, d - factor responsible for *disposal*, r - factor responsible for *recognition*, a -factor responsible for *actions*. Each of them is in the range of $[0, 63]$, but the sum cannot be zero. The chromosome in this representation contains four parts dedicated to each factor. Each of them is coded in six bits, i.e. there is a twenty four bits chromosome. As genetic operations tournament crossover with probability 0.5 and mutation with probability 0.065 is

TABLE VI
THE PERFORMANCE COMPARISON FOR THE NETWORK OF COMPARATORS
(NOC) WITH DIFFERENT METHODS OF AGGREGATION

Algorithm	Precision	Recall	F1-score
NoC with Approval voting	0.73	0.68	0.65
NoC with Borda count	0.77	0.73	0.69
NoC with Copeland's method	0.77	0.73	0.70
NoC with Plurality voting	0.68	0.63	0.61
NoC with Range voting	0.76	0.73	0.69
NoC with Weighted voting	0.78	0.74	0.70

used. As fitting function the measure in the following form is implemented:

$$f(ch) = \frac{3 * RQ + \text{sum}(F1score)}{4 * ARQ} : ARQ \neq 0. \quad (8)$$

where ch is a chromosome, RQ is a number of identified risks, ARQ is a total quantity of all considered risks and $\text{sum}(F1score)$ is a sum of all F1score values.

Parameters for genetic algorithm were chosen in an experimental way. Population stands at one hundred individuals. Weights have been learnt by means of a training set consisting of 136 out of 406 F&R actions (33%). The procedure was repeated ten times. Every time the population was initialized with random values. Termination condition was reached for a one hundred generation. The final weights are the ones with maximum evaluations. The evaluation function (8) took into account the number of recognized risks, as well as the overall prediction quality.

This method allows to use many different types of cross-over operations, mutations and successions of a population. A number of combinations of particular parts of procedures in question may be considered.

IV. RESULTS

This research is based on data available in the EWID system. The data describe F&R after it had already finished. This is one of the disadvantages of this set of data. There is no information about the point in time when something has happened for a given F&R. The only information is whether it had happened over the duration of the whole action. The full set of data consists of 291 683 F&Rs. A subset of 406 F&Rs is used in these experiments. These actions have been labeled with risks from Table I by experts. The data consist of 506 binary, numeric, multi-value attributes and descriptive attributes. This research does not take into account the descriptive ones. This set has been divided into four subsets according to four stages of action (mentioned in the previous section). Partial results data set is obtained and it contains 164430 pairs of objects after calculating the similarity by composite comparators. The pair consists of two objects representing F&R actions. This is not a complete Cartesian product. Some pairs have been eliminated by activation functions.

Partial results were achieved by means of default parameters of the network. Activation functions parameters have the 0.5 value which limited the results to the ones with similarity value greater than or equal to that value. The aggregations method

TABLE VIII
STATISTICAL MEASURES OF THE RESULTS GROUPED BY RISKS
ABBREVIATIONS: MIN - MINIMUM, MAX - MAXIMUM, AVG - MEAN,
MED - MEDIAN, STD - STANDARD DEVIATION, RAN - RANGE. VALUES
PRESENTED IN *F1-score*

Risk	MIN	MAX	AVG	MED	STD	RAN
A1_MA	0.26	0.38	0.32	0.33	0.05	0.12
A1_ME	0.90	0.92	0.91	0.90	0.01	0.02
A1_T	0.07	0.19	0.11	0.10	0.05	0.12
A2_MA	0.85	0.91	0.87	0.86	0.02	0.06
A2_ME	0.45	0.89	0.77	0.88	0.18	0.44
A2_S	0.07	0.22	0.14	0.13	0.06	0.15
A2_T	0.02	0.20	0.12	0.13	0.06	0.18
A2_U	0.22	0.45	0.33	0.33	0.08	0.23
A4_G	0.11	0.34	0.19	0.16	0.09	0.23
A4_MA	0.16	0.46	0.28	0.27	0.10	0.30
A4_ME	0.16	0.34	0.23	0.22	0.07	0.18
A4_S	0.18	0.38	0.28	0.28	0.07	0.20
A4_T	0.17	0.67	0.41	0.40	0.25	0.50
E1_MA	0.20	0.44	0.31	0.31	0.09	0.24
E1_ME	0.07	0.40	0.25	0.27	0.17	0.33
E2_MA	0.02	0.14	0.07	0.07	0.04	0.12
E2_ME	0.05	0.13	0.09	0.09	0.04	0.08
E2_S	-	-	-	-	-	-
E3_G	-	-	-	-	-	-
E3_MA	0.08	0.17	0.13	0.14	0.03	0.09
E3_ME	0.13	0.36	0.24	0.24	0.10	0.23
E4_MA	0.02	0.22	0.11	0.14	0.09	0.20
E4_ME	-	-	-	-	-	-
E4_S	-	-	-	-	-	-

inside all composite comparators was implemented as a mean function.

The procedure of global aggregation using the Weighted Voting algorithm was performed for the following weights: $\frac{2}{70}, \frac{48}{70}, \frac{1}{70}, \frac{19}{70}$ for notification, disposals, recognition and actions comparators respectively. These values have been achieved in the learning process described above. The described partial results set is a data source for experiment in this particular paper.

The experiment consists of selecting the best reference object for each input object from leave-one-out method in such a way as to ensure the greatest similarity within each part of the F&R action (according to the adopted division). Here, six algorithms in question were applied (one by one) and efficiency of the overall solution was measured by the quality of final results.

In the experiment, measures dedicated for classifiers such as: *precision*, *recall* and *F1-score* are used. Two types of measurements were performed for each of the algorithms. The first was on the assessment of individual pairs as classification results while the other evaluated the efficacy from the perspective of individual risks.

In the first case, the relevant set is a collection of risks labels associated with the input object, and the retrieved set is the one assigned to the reference object identified as the most similar. *Precision*, *Recall* and *F1-score* are calculated for each best pair. Lastly, all these three factors are averaged. Final results of this measurement are presented in Table VI.

In the second case, classification effectiveness measures are

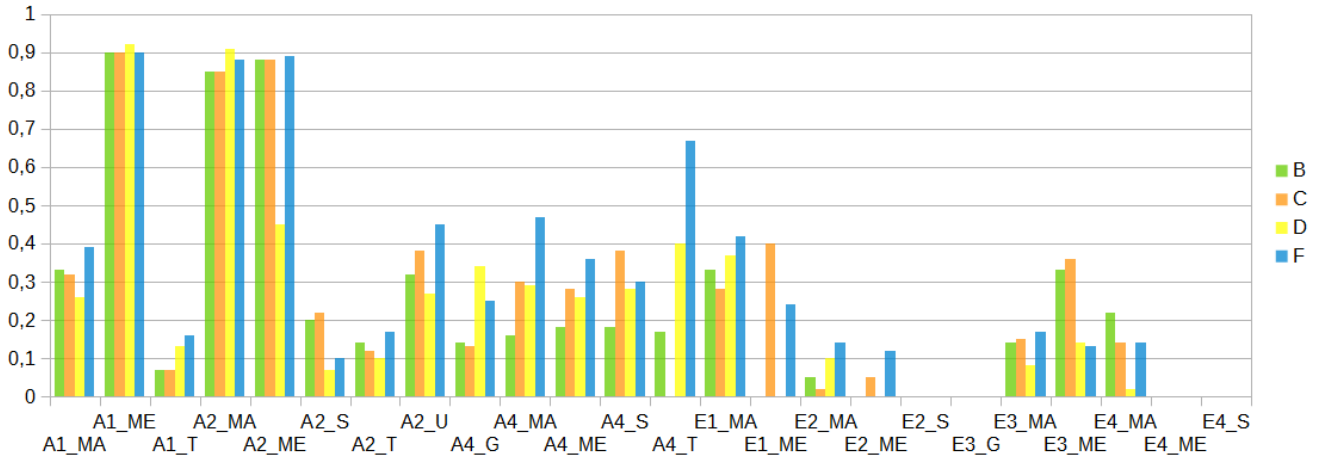


Fig. 3. The best algorithms grouped by risks evaluated by the *score* function. Abbreviations: B- Borda count, C-Copeland’s method, D-Approval, F-Weighted voting. The left axis shows the *F1-score* value, the bottom one selected risks from the risk matrix.

TABLE VII

EXPERIMENTAL RESULTS FOR ALL AGGREGATION ALGORITHMS IN QUESTION EXPRESSED IN F1-SCORE, PRECISION AND RECALL VALUES. ABBREVIATIONS: A - PLURALITY, B - BORDA COUNT, C - COPELAND’S METHOD, D - APPROVAL, E - RANGE VOTING, F - WEIGHTED VOTING

F1-score							Precision						Recall					
Risk	A	B	C	D	E	F	A	B	C	D	E	F	A	B	C	D	E	F
A1_MA	0.27	0.33	0.32	0.26	0.36	0.38	0.37	0.37	0.36	0.46	0.40	0.41	0.21	0.30	0.28	0.18	0.33	0.36
A1_ME	0.90	0.90	0.90	0.92	0.91	0.90	0.87	0.91	0.91	0.87	0.91	0.91	0.93	0.89	0.89	0.97	0.91	0.89
A1_T	0.07	0.07	0.07	0.13	0.12	0.19	0.08	0.08	0.10	0.21	0.11	0.20	0.06	0.06	0.06	0.09	0.13	0.19
A2_MA	0.87	0.85	0.85	0.91	0.85	0.86	0.86	0.87	0.87	0.86	0.86	0.87	0.89	0.83	0.84	0.96	0.84	0.85
A2_ME	0.66	0.88	0.88	0.45	0.89	0.88	0.88	0.89	0.89	0.91	0.89	0.89	0.53	0.87	0.87	0.30	0.89	0.87
A2_S	0.07	0.20	0.22	0.07	0.17	0.09	0.11	0.33	0.50	0.07	0.22	0.14	0.07	0.14	0.14	0.07	0.14	0.07
A2_T	0.02	0.14	0.12	0.10	0.13	0.20	0.03	0.14	0.13	0.20	0.12	0.22	0.02	0.14	0.11	0.07	0.14	0.18
A2_U	0.22	0.32	0.38	0.27	0.34	0.45	0.35	0.34	0.38	0.52	0.37	0.49	0.16	0.31	0.39	0.18	0.32	0.42
A4_G	0.18	0.14	0.13	0.34	0.11	0.25	0.25	0.14	0.13	0.40	0.09	0.22	0.14	0.14	0.14	0.29	0.14	0.29
A4_MA	0.22	0.16	0.30	0.29	0.24	0.46	0.29	0.20	0.35	0.32	0.25	0.42	0.18	0.14	0.27	0.27	0.23	0.50
A4_ME	0.18	0.18	0.28	0.26	0.16	0.34	0.21	0.21	0.30	0.33	0.18	0.32	0.15	0.15	0.26	0.22	0.15	0.37
A4_S	0.28	0.18	0.38	0.28	0.23	0.31	0.35	0.24	0.46	0.41	0.25	0.32	0.24	0.15	0.33	0.21	0.21	0.30
A4_T	-	0.17	-	0.40	-	0.67	-	0.14	-	0.33	-	0.50	-	0.22	-	0.50	-	1.00
E1_MA	0.20	0.33	0.28	0.37	0.22	0.44	0.18	0.50	0.25	0.43	0.22	0.44	0.22	0.25	0.33	0.33	0.22	0.44
E1_ME	0.07	-	0.40	-	-	0.27	0.11	-	1.00	-	-	0.28	0.05	-	0.25	-	-	0.26
E2_MA	0.06	0.05	0.02	0.10	0.07	0.14	0.09	0.05	0.03	0.21	0.07	0.13	0.05	0.05	0.02	0.07	0.07	0.15
E2_ME	-	-	0.05	-	0.09	0.13	-	-	0.06	-	0.09	0.17	-	-	0.05	-	0.10	0.10
E2_S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E3_G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E3_MA	0.17	0.14	0.15	0.08	0.13	0.12	0.21	0.29	0.40	0.17	0.25	0.18	0.14	0.09	0.09	0.05	0.09	0.09
E3_ME	0.19	0.33	0.36	0.14	0.28	0.13	0.17	0.67	1.00	0.20	0.40	0.17	0.22	0.22	0.22	0.11	0.22	0.11
E4_MA	0.14	0.22	0.02	0.02	-	0.15	0.01	0.50	0.14	0.01	-	0.17	0.29	0.14	0.14	0.29	-	0.14
E4_ME	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E4_S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Score	0.62	0.62	0.66	0.61	0.59	0.70	0.62	0.63	0.68	0.63	0.59	0.70	0.61	0.62	0.66	0.62	0.58	0.70
Mean	0.19	0.23	0.26	0.22	0.22	0.31	0.23	0.29	0.34	0.29	0.24	0.31	0.19	0.21	0.24	0.22	0.21	0.32

calculated for each individual risk from the threat matrix. The calculation uses a set of actions containing a given risk as a

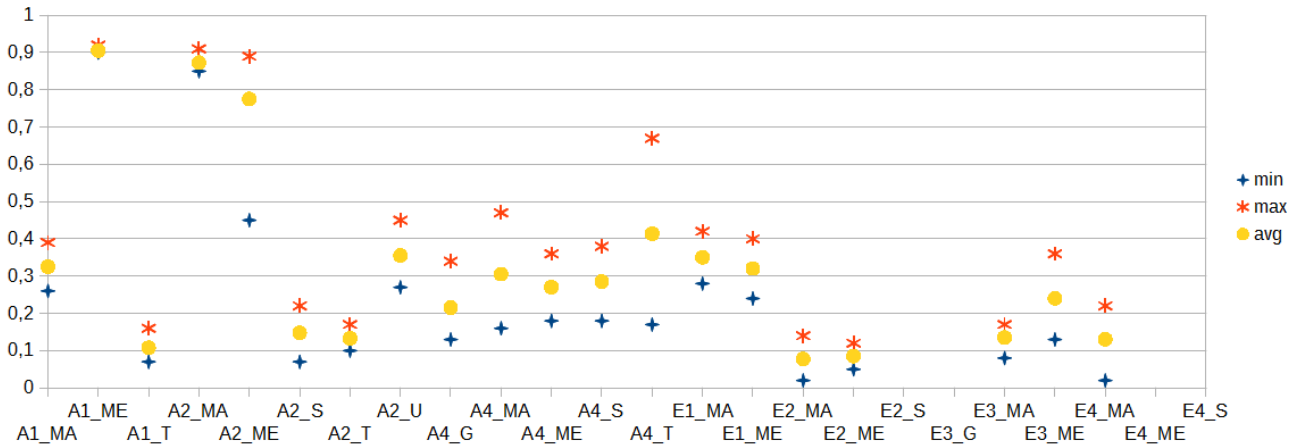


Fig. 4. Chart with the minimum, maximum and average results value for particular risks. It shows the range of values between the minimum and maximum followed by the mean value for all algorithms.

relevant set (on the basis of assignment of labels by an expert). The retrieved set consists of actions (reference objects) found by means of our solution. Table VII contains the performance comparison of all classification methods expressed by *F1-score*, *Precision* and *Recall*. Additionally, there are the *score* and *mean* rows at the bottom. They show the values of global scoring of algorithms. The first one is a normalized value of the number of recognized risks boosted tree times and summed with a total sum of *F1-score*. The second one is a mean value but calculated for all risks from the domain. It means that the zero value is taken for those risks which have not been identified. The *score* function is used to evaluate global efficiency of a particular algorithm in application in this experiment.

Table VIII contains statistical factors of the achieved results. They have been calculated on the basis of rows of Table VII limited to the *F1-score* measure. These are the most frequently used factors such as: min, max, mean, etc. The important one is the range factor which shows the difference between the best and the worst result for a given risk.

Versatility in identifying risks is a desired feature that should characterize the best solutions. Therefore, the function of evaluation algorithms (score) highly rewards those algorithms that identify the broadest spectrum of risks. Evaluation of a particular algorithm using only a number of identified risks is presented in Figure 5.

Results presented are extensions of results presented in the previous publication of the authors [11]. Three other methods are described in the publication: Naive Bayes, ESA, kNN Canberra. Authors' experiments are based on the same set of data. The results in relation to the previous ones show progress in terms of growth performance measures.

V. DISCUSSION

It can be noted in Tables VI and VII that the best results for the whole described solution have been achieved by Weighted

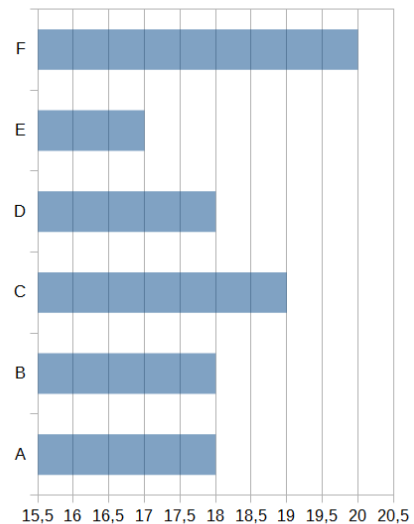


Fig. 5. The comparison of the number of identified risks by particular method. X-axis represents the number of different risks which were at least once properly assigned to an F&R action. The entire space consists of 24 risks. Abbreviations: A - Approval voting, B - Borda count, C - Copeland's method, D - Plurality voting system, E - Range voting, F - Weighted voting.

Voting algorithm. They have been obtained in terms of the largest number of identified risks as well as the highest values of effectiveness measures. Network of Comparators (NoC WA) with global aggregation implemented by means of this algorithm has identified 20 risks among all 24 existing in the threat matrix. All other algorithms obtained worse results. The second in this ranking is the result of the Copeland's method with a value of one less.

In terms of classification effectiveness from the risk point

of view it has achieved the highest value of score function calculated for all the three considered measures (0.7). Also, the mean value is the highest for both *F1-score* and *Recall* case and a little bit lower in the case of *Precision*. Similarly to the averaged measure for each best pair. The NoC WA has achieved 0.78, 0.74 and 0.70 for *Precision*, *Recall* and *F1-score* respectively.

Results presented prove that the problem of selecting the global aggregation method in a network of comparators is important and may have big impact on achieved results. They also show that election algorithms are a very good solution for optimization problems described. From the point of view of the arithmetic mean (range voting), all the methods used have obtained a higher value of the *score* function. Table VIII shows statistical factors for results generated from the risk perspective. Figure 4 demonstrates we can see exactly what the extreme values achieved by algorithms for individual risks are. This shows that in this range results can vary depending on the selection of an algorithm. It is clear that in certain cases it is the value of 0.5 which is a 50% of the scale. Additionally, it demonstrates how the average value fits the range.

Weak points of experiments are the tie solving methods. In experiments, the authors have used the method of getting the first candidate as a winner. This is quite a random method not showing real preferences.

VI. CONCLUSIONS

This paper shows similarity-based solution for recognition and identification purposes. This article in particular raises the problem of selecting the optimal results set. It presents theory and working solution on the example of recognition risks at the fire ground.

In order to investigate the problem of selecting the aggregation method for the network of comparators, six methods have been tested. The theory part for each of them is presented in detail as well as adaptation for our concrete problem. One of the strongest conclusions is that the election algorithms are a very powerful methods for implementation in such cases.

The results achieved are very promising. They are very good in comparison with different attempts to resolve risk recognition problem using the same set of data [12]. Based on these results we recommend a Weighted Average election algorithm to use as a global aggregator in the network of comparators. This method makes it possible to adapt to a specific problem by learning weights. For the purposes of the NoC solution, we have developed a learning method using a genetic algorithm.

The NoC is a very useful solution in the field of AI applications. It allows to build complex networks of comparators that can be used as ensemble classifiers [27]. In this way, very complex decision support problems can be solved as well as classification, recognition and identification problems.

The future work should concentrate on propagating the optimization method on the level of a single comparator and particular layer of the network (not only for output layer). Our research has showed that this might be a large field of improvement of performance for the whole solution.

REFERENCES

- [1] A. Tversky and E. Shafir, *Preference, Belief, and Similarity: Selected Writings*, ser. Bradford books. MIT Press, 2004.
- [2] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.
- [3] J. Kacprzyk, *Multistage Fuzzy Control: A Model-based Approach to Fuzzy Control and Decision Making*. John Wiley & Sons, Limited, 2012.
- [4] S. O. Kuznetsov and D. Slezak, "Data mining and soft computing," *Int. J. General Systems*, vol. 42, no. 6, pp. 543–545, 2013.
- [5] M. S. Szczuka and D. Slezak, "Feedforward neural networks for compound signals," *Theor. Comput. Sci.*, vol. 412, no. 42, pp. 5960–5973, 2011.
- [6] D. Slezak and Ł. Sosnowski, "SQL-based Compound Object Comparators: A Case Study of Images Stored in ICE," in *Proc. of FGIT-ASEA 2010*, ser. Communications in Computer and Information Science, vol. 117, 2010, pp. 303–316.
- [7] Ł. Sosnowski and D. Slezak, "How to design a network of comparators," in *Brain and Health Informatics*, 2013, pp. 389–398.
- [8] —, "Networks of compound object comparators," in *FUZZ-IEEE*, 2013, pp. 1–8.
- [9] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles," *Knowl. Inf. Syst.*, vol. 38, no. 2, pp. 259–275, 2014.
- [10] M. Wozniak and K. Jackowski, "Some remarks on chosen methods of classifier fusion based on weighted voting," in *HAIS*, ser. Lecture Notes in Computer Science, E. Corchado, X. Wu, E. Oja, Á. Herrero, and B. Baruaque, Eds., vol. 5572. Springer, 2009, pp. 541–548.
- [11] Ł. Sosnowski, A. Pietruszka, A. Krasuski, and A. Janusz, "A resemblance based approach for recognition of risks at a fire ground," in *Active Media Technology - 10th International Conference, AMT 2014, Warsaw, Poland, August 11-14, 2014. Proceedings*, 2014, pp. 559–570.
- [12] A. Krasuski and A. Janusz, "Semantic tagging of heterogeneous data: Labeling fire & rescue incidents with threats," in *FedCSIS*, 2013, pp. 77–82.
- [13] S. Staab and A. Maedche, "Knowledge Portals: Ontologies at Work," *AI Magazine*, vol. 22, no. 2, pp. 63–75, 2001.
- [14] Ł. Sosnowski, "Applications of comparators in data processing systems," *Technical Transactions, Automatic Control*, 2014, to appear.
- [15] D. W. Aha, Ed., *Lazy Learning*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.
- [16] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.
- [17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." Morgan Kaufmann, 1995, pp. 1137–1143.
- [18] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, "Using complexity to protect elections." *Commun. ACM*, vol. 53, no. 11, pp. 74–82, 2010.
- [19] E. Baharad and Z. Neeman, "The asymptotic strategy proofness of scoring and condorcet consistent rules," *Review of economic design : RED.*, 2002.
- [20] X. Lin, S. Yacoub, J. Burns, and S. Simske, "Performance analysis of pattern classifier combination by plurality voting," *Pattern Recogn. Lett.*, vol. 24, no. 12, pp. 1959–1969, Aug. 2003.
- [21] M. Barbie, C. Puppe, and A. Tasnadi, *Non-manipulable domains for the borda count*, ser. Bonn econ discussion papers, 2003, no. 13.
- [22] D. G. Saari and V. R. Merlin, "The copeland method i; relationships and the dictionary," *Economic Theory*, vol. 8, 1996.
- [23] S. J. Brams and P. C. Fishburn, "Going from theory to practice: the mixed success of approval voting." *Social Choice and Welfare*, vol. 25, no. 2-3, pp. 457–474, 2005.
- [24] M. Balinski and R. Laraki, "A theory of measuring, electing, and ranking," *Proceedings of the National Academy of Sciences*, vol. 104, no. 21, pp. 8720–8725, May 2007.
- [25] G. Levitin and A. Lisnianski, "Reliability optimization for weighted voting system." *Rel. Eng. & Sys. Safety*, vol. 71, no. 2, pp. 131–138, 2001.
- [26] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, Jun. 1994.
- [27] M. Wozniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, pp. 3–17, Mar. 2014.