

Distributed Data Mining for Data Fusion: Multi-agent Approach

Vladimir Gorodetski
SPIIRAS, St. Petersburg,
Russia
gor@mail.iias.spb.su

Oleg Karsaeyv
SPIIRAS, St. Petersburg,
Russia
ok@mail.iias.spb.su

Vladimir Samoilov
SPIIRAS, St. Petersburg,
Russia
samovl@mail.iias.spb.su

Abstract

The objective of Distributed Data Mining (DDM) is extraction of useful patterns (rules, association rules, decision trees, frequent episodes, etc.) from distributed databases accessible through Intra- or Internet. Data Fusion (DF) is a task aiming at making decisions on the basis of distributed data sources. On the other hand DDM can be considered as a component of Data Fusion (DF) technology that supposes to support for automated design of DF system (distributed) knowledge base. As a matter of fact, the structure of DF depends on many factors, but firstly it depends on the number of data sources and peculiarities of the data of sources. As a rule, these sources contain heterogeneous data that can be represented in various structures (relational, transactional, etc.), can be of different nature (images, signals, numbers, etc.) and accuracy, be measured in different scales (Boolean, categorical, real), can contain uncertainty, etc. An objective of a DF system is to combine data from many different sources to make decision, for instance, classification of an object or an object state, a situation assessment, etc. Within DF specific tasks three issues are of the most significance. The first is meta-model of combining decisions produced on the basis of particular data sources; the second is meta-model of distributed data sources and the third is DDM. Solutions of these tasks, technology accepted for their design and also architecture accepted for DF system design and implementation form together conceptual, algorithmic and architectural basis for software tool aiming at support for a technology of DF system design and implementation. The focus of the paper is multi-agent technology and software tool for DF applied systems design and implementation.

1. Introduction

From general point of view Data fusion (DF) is a task of data processing aiming at making decisions on the basis of distributed data sources specifying an entity (an object or an object state, a situation constituted by a number of autonomous interacting objects, intent of a situation management, etc.). Distributed data used for DF can be of different physical nature (electromagnetic signals, images, sensor data, experts' information, etc.), be of different accuracy and reliability, particular data

may be incomplete and uncertain and be represented in different data structures, contain missed values, etc.

Analysis of the recent publications proved that currently the area of DF is growing and becoming of great interest in many applications. To make clear what kind of applications are considered in the DF scope, let us outline two DF applications.

1. *Detection of intrusions into computer network* ([2]). At present coordinated distributed attacks performed by a team of malefactors from spatially distributed hosts constitute the main threats for computer networks and information. "Traces" of an attack proves in different data perceived or generated by a computer network assurance system. For example, they are displayed in *tcpdump* generated via preprocessing of input traffic, in audit data trail, in sequences of system calls of operating system, in data resulting from monitoring of application servers, queries to databases and directories, in data specifying users' profiles, etc. Such data are generated on different hosts of computer network. The timely detection of a illegitimate user's activity is potentially feasible only in case of fusion data from different available sources. Formally, intrusion detection is a type of classification task which has to be solved on the basis of combining decisions produced on the basis of many data sources. ♦

2. *Analysis and prognosis of natural and man-made disaster development* ([3, 10]). A lot of different kinds of potentially dangerous situations emerge in different regions of many countries. They can emerge due to natural disasters (earthquakes, floods, etc.), man-caused emergencies (chemical, nuclear, etc.) and so on. The specific features of such phenomena are rapid development in time, spread in space, strong dependence on weather conditions, landscape, buildings infrastructure and so on. To assess such a situation in order to predict its development and to prevent its undesirable or catastrophic consequences, it is necessary to use data from different sources. The sources of such data are weather data, information collected by airborne

equipment (photo, TV, radar, infrared, etc), people's messages, simulation data, and so on. ♦

There exists also many other applications in the civil and military areas that need to be stated as DF problems.

Formal methods and models used in DF are borrowed from very diverse areas, for instance, from statistical decision theory, data mining and knowledge discovery, machine learning, classification, image processing, uncertain and fuzzy information processing, Bayes networks, evidence theory, distributed systems, parallel processes, ontology, simulation, and so on, and so forth. Nevertheless the research area of DF is actually challenging and also puts its own specific problems. The most important ones concern to the development of specific protocols (distributed algorithms) needed to coordinate distributed parallel processes of DF systems, meta-models, algorithm and technology for DDM and decision making, and also software technology support.

The focus of the paper is multi-agent software tool for the design and implementation of DF systems. Within this problem we consider meta-model of data fusion (combining particular decisions produced on the basis of local data sources); meta-model of DDM and a number of associated protocols providing consistent cooperation of distributed software components processing data of distributed sources; and also multi-agent architecture of a DF software tool aiming at support of technology for DF system design and implementation. Respectively, the rest of the paper is organized as follows. In *section 2* we consider meta-models of data fusion. In *section 3* we summarize the technology of DF systems developed and its "projection" onto a set of agents supported this technology. *Section 4* outlines conceptually peculiarities of distributed ontology design and a number of associated protocols. Ontology is a key component of both multi-agent DF software tool and any DDM system. *Section 5* describes the approaches used for combining of decisions produced by data source classifiers. This functionality is the key operation in multi-level decision fusion. *Section 6* describes the developed multi-agent architecture of a Data Fusion software tool and *Section 7* gives brief information about two case studies used for debugging and evaluation of the developed DF systems technology. *Conclusion* summarizes the main results and future work.

2. Meta-models of Data Fusion

Although to date several strategies for data fusion are proposed ([12]), the most popular and advantageous one

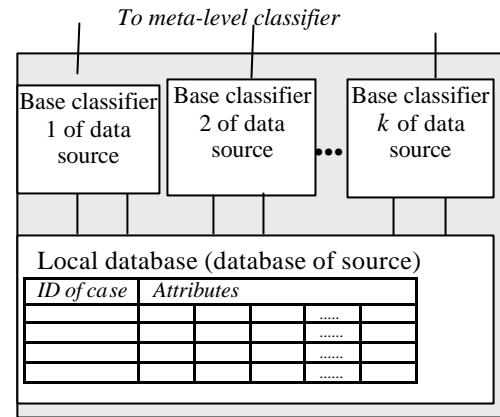


Fig.1. Combining base-level decisions: Variant 1

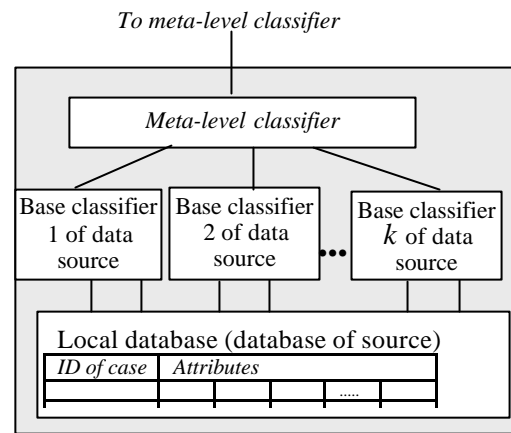


Fig.2. Combining base-level decisions: Variant 2

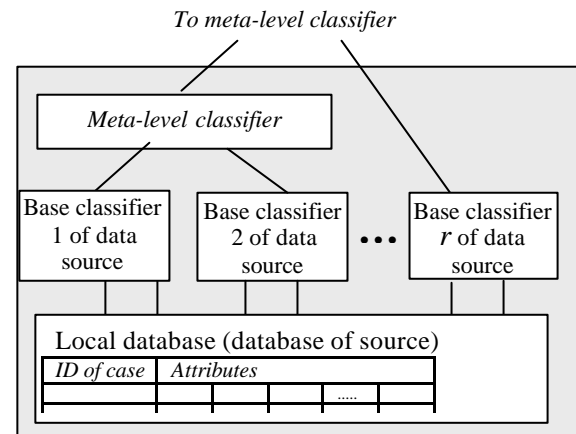


Fig.3. Combining base-level decisions: Variant 3

is the strategy using a multi-level hierarchy of classifiers. In it, the source-based classifiers make decisions on the basis of data of particular sources followed by meta-level decision making based on combining of the source-level decisions. The advantages of such a scheme are (1) decrease of information exchange; (2) simplicity of data source classifier fusion even if they use data of different

of generally-duty, i.e. they are needed for the design and implementation of any applied multi-agent system. We use as such the components of MASDK software tool ([11])¹ developed with participation of the authors of this paper. The components of MASDK are used for *generation* and *deployment* steps of the applied DF system design and implementation.

The components of the second group supplement MASDK by functionalities specifically needed for design and implementation of DF applications. These components (called hereafter *DF-oriented*) are used for the third purpose that is "*specialization*" of DF system. Below we only consider the third component of DF software tool that is DF-oriented.

Fig.4 presents the processes that have to be supported by DF-oriented component of agent-based software tool. These processes are abstracted of particular applications. Fig.4 only presents (in terms of standard IDEF0 diagrams) the high-level view of processes of the DF technology, structure of their interactions, intermediate and final results and also presents allocation of different tasks over DF software tool agents thus conceptually outlining the DF software tool architecture. The basic processes of the technology and their interaction presented in Fig.4 are as follows:

- Distributed application ontology design (A0);
- Design of DF meta-model, i.e. structure for decision combining in particular DF application (A1);

- Distributed data mining (A2)

These three processes are mostly used in design and implementation of DF application and can also be used but not obligatory for its modification if necessary. The following two processes correspond to on-line operation of applied DF system under design:

- Data Fusion (A3);
- Data source monitoring (A4).

These processes come out from the *specialization* carried out by three previously mentioned processes. Actually, the former are the processes to be designed by executing the latter.

Design of every above process supposes a semi-automated realization and is carried out according to some technology supported by DF-oriented component of DF system software tool. Each of the processes have to be specified formally as a protocol presented at several levels of details in terms of standard IDEF0 diagrams like given in Fig.4 and must be realized by the respective software. In the subsequent section we conceptually outline the sense operations associated with distributed ontology design and with DDM that in many respects are both most important and specific for agent-based DF applied system design.

4. Distributed ontology design

The key DF peculiarities come out of the fact that data sources are *distributed* and *heterogeneous*. As a

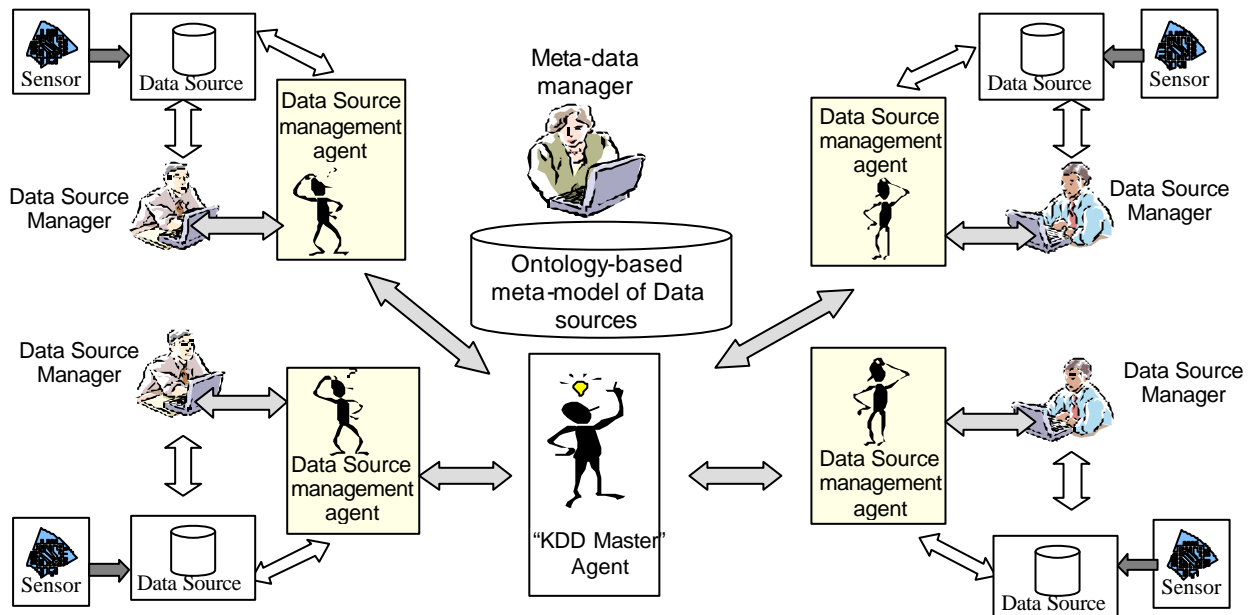


Fig.5. Distributed design of distributed ontology

¹ See also other paper in these Proceedings written with participation of the authors of this paper.

rule, data sources are spatially distributed or, at least, they are represented in different databases and/or located on different hosts. Heterogeneity is entailed by

the diversity of possible data structures, differences in data specification languages, differences in data natures (geographical, statistical, images, etc.) and so on. As a rule, this data are of large scale.

Distribution and heterogeneity of data put new problems which significantly influence on all aspects of DF system design and implementation.

The *first problem* is development of the *shared thesaurus* providing for *monosemantic understanding of the terminology* used in formal specification of domain entities. This problem arises due to the fact that specifications of data belonging to particular sources are being developed by different experts and in most cases these processes are independent (see Fig.5). The experts can denote different domain entities by the same terms and vice versa, they can denote the same entities by different terms thus leading to misunderstanding.

Next class of problems called *non-coherency of data measurement scales* come out of the fact that the same entities can be represented in different sources by various data structures but in DF procedures all of them have to be used equally. That is why it is necessary to provide distributed data *for consistent representation*.

The third problem is so-called "*entity instance identification problem*" ([10]). The data specifying an object is represented in several data sources. Therefore each data source only partially specifies it. Object complete specification is made up of data fragments distributed over the data sources and to form a complete object specification, a mechanism to identify such fragments is needed. It should be noticed that some fragments of data associated with an object can be absent in a number of sources. A graphical explanation of this problem is given in Fig.6.

The above and some other tasks that are very specific for DF system design, in particular, for DDM aiming at forming DF system distributed knowledge base can effectively be resolved on the basis of application ontology shared by all components of DF system. The process "Distributed ontology design" (see Fig.4) that is carried out according to a number of the developed protocols realizes design of such ontology. Let us consider conceptually what kind of procedures is carried out by these protocols.

Thorough and accurate development of the terminology representing the *application ontology* must be carried out at the very initial phase of the DF system conceptual design. The goal of this phase is to determine the names and structures of basic entities constituting application domain and also relationships given over them. The resulting ontology comprises the following components (Fig.7):

- *Problem ontology*, i.e. the part of ontology that must present the top-level component in the specification of any application from problem scope;
- *Shared components of application ontology* that is a part of application ontology common for all components of distributed application (in our case – for all components of the DF system working with the local data sources). This part of ontology is application-specific.
- *Private components of application ontology* owned by distributed application components that specify particular characteristics of data sources.

Within DF software tool, the *entity identification problem* is solved in the following manner. In the application ontology, for each entity the notion of entity identifier ("*ID entity*") is introduced. This entity identifier plays the role of the entity primary key (in analogy with the primary key of a table). For each such identifier, a rule is defined within the application ontology, which can be used to calculate the value of this key. For example, the value of a unique combination of a subset of this entity attributes can be used as the argument of such a rule. A specific rule is defined for each particular data source that uniquely connects the entity identifier and the local primary key in this source. In a special case, it may simply be a list of pairs "*value of entity identifier*" – "*value of local key*". After such rules have been constituted for each particular source a list of all the instances of entities stored in the local sources is formed on the meta-level, and instances specified in several sources are identified.

The next problem called above *non-coherency of data measurement scales* is that the data specifying in different sources the *same entity attribute* can be of different structures. In some cases an attribute presented in different sources may be measured in the same measurement scale but in different units. In our technology this problem is solved as follows. Let us designate the attribute measured differently in different sources as *X*. In the shared part of the application ontology the type and the unit of the attribute *X* measurement are determined. At the next step in all the sources where it is presented expressions are determined for this attribute through which it can further be converted into the same scale in all the sources. This allows using the values of attributes on the meta-level regardless of the data source from which they are originated. At that, the selection of attribute *X* specification and measurement within shared part of application ontology is made by experts via negotiations according to a protocol.

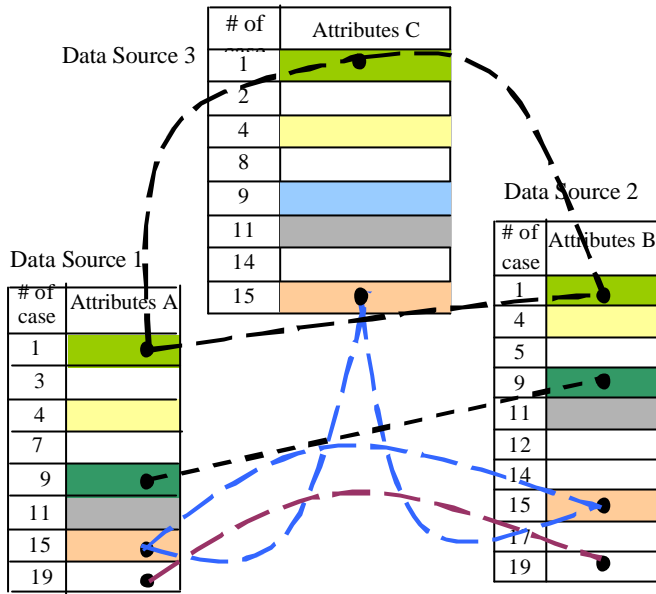


Fig.6. Illustration of the essence of the data identification

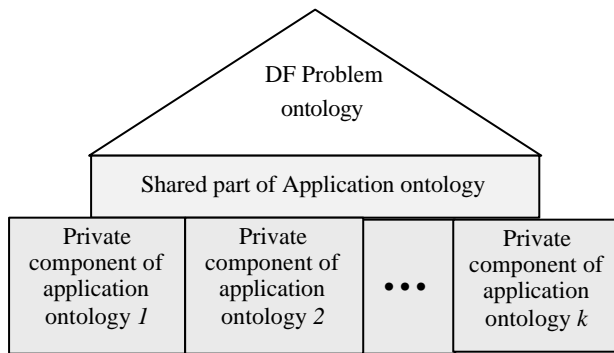


Fig.7. Tower of DF ontology components

In the developed architecture of DF software tool (see section 6) forming of application ontology in a way that provide resolution of the above three problems is the area of responsibility of several agents participating in the iterative and interactive design of private and shared components of the application ontology.

5. Combining decisions of multiple classifiers

It was noted above that in our research and development we follow the strategy of data fusion in which DF is considered as hierarchy of multiple classifiers producing decisions on the basis of particular data sources followed by combining these decisions at the meta- level. In the most DF tasks "decision" is understood as classification of an entity (object, state of

an object, etc.), i.e. assigning the entity a class label from a fixed set.

To date several approaches to decisions combining are developed. They can be grouped as follows:

1. *Voting* algorithms;
2. *Probability*-based or fuzzy algorithms;
3. Meta-learning algorithms based on *stacked generalization* idea;
4. Meta-learning algorithms based on *classifiers' competence evaluation*.

Voting methods were developed about twenty years ago and were historically the first ([21]). Due to their simplicity and satisfactory accuracy in many applications these methods are to date in use. The most simple of them is called "*majority voting*" ([6], [7]). The "*weighted voting*" approach ([19], [15], [4], [1], [16], etc.), which is a little more sophisticated, was implemented in many particular forms and is currently in broad use.

The methods of the second group are based on probabilistic models like Bayesian model of a posteriori probability assessment, Bayesian networks, Dempster–Shafer theory of evidence, and also on fuzzy set-based models ([3]). These models are in some sense "classical", they have long history and do not need any comments or references. Unfortunately, analysis proved that the area of practical application of such methods in DF scope is not very broad. In any case, the question "Where probabilities come from?" has here to be answered. To our opinion probabilistic and similar models are applicable if the dimensionality of the classification task is small and there is enough information for empirical assessment of needed probabilistic characteristics with satisfactory accuracy and reliability In data fusion scope such methods can reasonably be used at higher levels of decision combining.

However, at present the Knowledge Discovery from Databases (KDD) R&D community pays the most attention to the methods of combining decisions that use some knowledge about properties of base-level classifiers ([23]). General idea of this group of approaches that was proposed in [34] is called "*stacked generalization*". Although the idea of *stacked generalization* itself is very simple, it turned out very effective and gave birth to several particular methods of combining decisions. The most promising variations of *stacked generalization* is proposed in [24], [5], [20] ("*meta-classification*"), [9] ("*cascade generalization*"), and in some others.

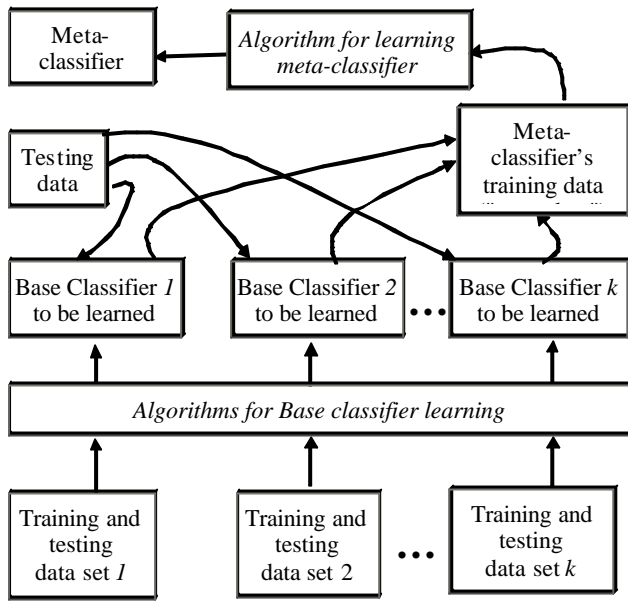


Fig.8. Meta-classification scheme

Generalized structure of decision combining based of *stacked generalization* (for its "meta-classification" variant) is explained in Fig.8. The experience proved that base classifiers, which decisions are supposed to be combined, are to be "diverse", i.e. they must be at least either based on different learning algorithms or to be learned on the basis of different training and testing data.

In general, *stacked generalization*-based methods of decisions combining are popular and still being actively researched. A drawback of this group of methods is their inability to preserve already existing set of classifiers unchanged if a new classifier inserted in the

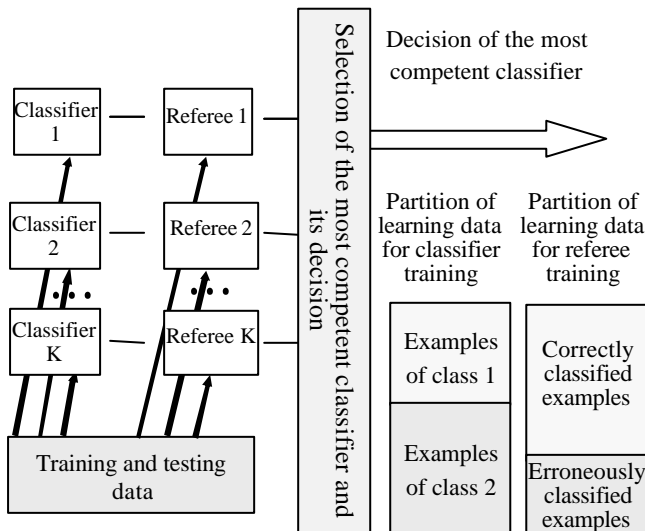


Fig.9. Explanation of the idea of competence-based approach to combining decisions of multiple

classification system ([27]). In contrast, the methods of the fourth group discussed below are free of this drawback.

These methods are based on the *evaluations of classifiers' competences* with regard to each particular record of input data, specifying an object. The main idea of competence-based methods of decision combining is to assign to each particular classifier the region in attribute space where it is most competent as compared with the other base-level classifiers. Firstly this idea was proposed in the papers [23] and [17]. The core of this idea is that a special procedure called "referee" (see Fig.10) is associated with each particular classifier. A responsibility of referee is to assess the competence of the respective classifier with regard to the particular input data ([18]). To be able to solve this task the referee has to be learnt. Referee learning is reduced to the routine learning task, which can be solved on the basis of the same training and testing data that has been used for training and testing of classifiers themselves. A specific of the last task is that in it other partition of learning data is used. For referee training, the same training and testing data are partitioned into two subsets of positive and negative examples like shown in Fig.10.

In competence-based methods, decision combining consists of two steps: (1) detection of the most competent classifier and (2) selection of the classification produced by the most competent one.

Further important development of this method was proposed in the papers [22], [25] and [26], at that the last two papers proposed the definitely significant improvement of the basic method.

In general, competence-based methods are very promising ([17], [18], [23], [26]). Its advantages are higher accuracy (as compared with the voting and stacked generalization-based methods) and also its capability to preserve already existing set of classifiers unchanged if a new classifier are inserted in classification system.

Two types of methods discussed in this section, i.e. meta-classification and competence-based methods are included into the multi-agent DF software tool as DDM algorithms. It is noteworthy to mention that in DF applications both above methods cannot be used in "straightforward" manner because of peculiarities of data and needs to be adapted and the necessity to use distributed learning.

6. Multi-agent architecture of data fusion and data fusion learning components

Any DF system is distributed in its nature. Actually it deals with distributed data sources and realizes

- (1) Components responsible for the design of source based parts of the designed DF system, and
- (2) Component supporting iterative and interactive design of the meta-level part of the designed DF system.

- ontology and its consistency maintenance with regard to shared component of ontology. In addition, it plays the role of the gateway to the source database.

3. *Local classification agents of DF system.* These agents constitute a part of the designed DF system producing source-based decisions, i.e. they are learned classification agents operating with the source data.

This component also includes local database and user interface providing interactive mode of its operation.

1. *Meta-Learning agent (“KDD Master”)*. Its functions are as follows:

- Design of meta-model of decision making.
2. *Meta-level KDD agent*. It aims at solving the tasks of training and testing of meta-level classification agents (meta-classifiers or /and referees);

4. *Data Fusion management agent*. It is responsible for coordination of performance of components of *Agent-classifier of meta-level* and *Meta-level KDD agent* both in learning and decision fusion.

5. *Server (library) of KDD methods.* It stores KDD methods, metrics for evaluation of the learning quality and other functionalities needed for operation of *Meta-level KDD agent*.

Two case studies were used for design and implementation of multi-agent DF system prototypes. They were developed with use of Multi-Agent System Development Kit [11]



Architectures of these components are depicted in Fig.10 and 11. Each data source-based component comprises the following parts (Fig.10):

The diagram illustrates the architecture of the KDD system. It features a central "KDD Master" box at the top, which is connected via black double-headed arrows to a "User interface" on the left and an "Agent-classifier of meta-level" box on the right. The "Agent-classifier" contains a stack of boxes labeled "Meta", "Meta", "Meta-classifier", and "Referee", with an "Inference engine" box below them. Below the "KDD Master" is a "Meta-level KDD agent" box, which is connected via black double-headed arrows to the "User interface" and the "Agent-classifier". At the bottom is a "Server of learning methods" box, connected via black double-headed arrows to the "Meta-level KDD agent" and the "Inference engine". On the far right is a "Data Fusion management agent" box, connected via black double-headed arrows to the "Inference engine". Pink arrows indicate data flow: from the "User interface" to the "Meta-level KDD agent", from the "Meta-level KDD agent" to the "KDD Master", from the "KDD Master" to the "Meta-level KDD agent", from the "Meta-level KDD agent" to the "Server of learning methods", from the "Server of learning methods" to the "Inference engine", from the "Inference engine" to the "Data Fusion management agent", and from the "Data Fusion management agent" to the "Inference engine". Labels at the bottom indicate data flow to various agents: "To the Data source managing agent", "To the source-based KDD agents", "To the source-based classification agents", and "To the Data source managing agent".

Fig.11. Architecture of meta-level component of DF software tool

developed by authors of this paper. The prototypes implemented case studies outlined below.

KDDCup-99 –based Case study of DF system

Application corresponding to the KDDCup-99 data set [29] deals with Intrusion Detection Learning Task. Inherently this data are not distributed but they were split artificially to model multiple sources and then to use the resulting task as a case study. The KDDCup-99 data set is specified by 36 attributes of different types (numerical–28, categorical–4, Boolean–4), and the total size of training and testing data records used in case study is equal to 33460 at that $TT=7100$ of them were used for training and testing of base-level classifiers and meta-classifier and the rest, i.e. $FT=26360$, were used for final evaluation of the accuracy and performance of the developed multi-agent DF system. The TT data set was artificially split into two data sources DS1 and DS2 (they have 1 common Boolean attribute). In turn, the DS1 and DS2 data sets were also split into 3 and into 4 subsets respectively. The last splitting aimed to form training and testing data for particular base-level classifiers and meta-classifier at that the total number of base-level classifiers was chosen equal to 7 (3 of them were used in DS1 and 4-in DS2). The scheme of interaction of base-level classifiers and meta-level classifier is demonstrated in Fig.12. The base-level classifiers differ in attribute sets and also in training and testing data sets and some of them also differ in learning algorithms used. Two basic algorithms of learning were used in this case study: "Visual analytical Mining" ([12], [13]) intended to mine numerical data, and $GK2$ algorithm ([14]) intended to

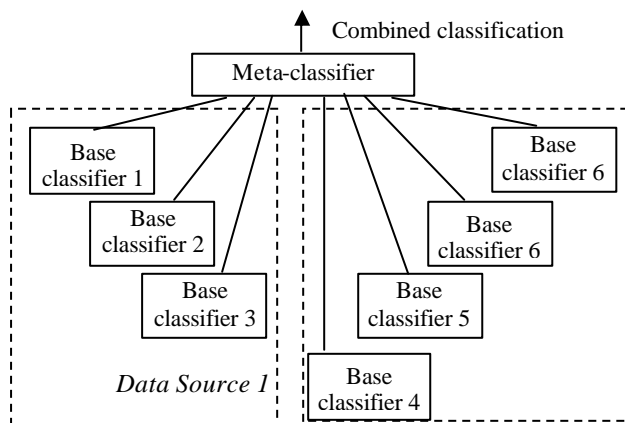


Fig.12. Classification structure in KDDCup case

mine discrete data. Both algorithms were developed by the paper authors.

This case study was intended to be used for evaluation of correctness, advantages and drawbacks of the partially developed DF software tool and also multi-

agent approach to the design and implementation of DF systems and justify its feasibility as well as quality of the resulting DF system operation.

Multi-Spectral Image classification

The second case study which objective is multi-spectral image classification has also been developed. It uses Landsat Scanner image dataset of UCI repository [30].

8. Conclusion

The paper is devoted to the design and implementation issues of Multi-agent Data Fusion software tool. Development of the latter puts several new non-specific tasks and challenges that are in the focus of the paper.

The key problems of the design and implementation of DF software tool come out of the fact that data sources are physically distributed and heterogeneous. As a rule, data sources are spatially distributed or, at least, they are represented in different databases and/or located on different hosts. Heterogeneity is entailed by the diversity of possible data structures, differences in data specification languages, differences in data natures (geographical, statistical, etc.), etc. These and some other specific properties of data to be processed in DF applications constitute basic challenges analyzed in the paper. Within them, three issues are of the most significance. The *first* is meta-model of combining decisions produced on the basis of particular data sources; the *second* is meta-model of distributed data sources and the *third* is DDM. The basic results of the paper are analysis of these tasks, their solutions proposed, development technology and also architecture of DF software tool.

Although the main paper subject is DF technology and software tool, its results provide also the basis for Multi-agent Distributed Data Mining technology and respective software tool.

Future research will be focused on the further development of technological aspects of Data Fusion software tool, its software implementation and its use for the development of a number of particular applications in this area to accumulate experience and to make the software tool in question more industrial-oriented.

References

- [1] K.Ali and M.Pazzani. "Error reduction through learning multiple descriptions". *Machine Learning*, 24(3), 173-202, 1996.
- [2] T.Bass." Intrusion Detection System and Multisensor Data Fusion: Creating Cyberspace Situational

- Awareness". *Communication of the ACM*, vol.43, No. 4, 99-105 (2000).
- [3] E.Blash. "Tutorial TD2: Fundamentals of Information Fusion and Applications". *Information Fusion -2002*, USA, July 2002.
 - [4] W.L.Buntine. "A theory of learning classification rules". *Ph.D thesis*, University of Technology, School of Computing Science, Sydney, Australia, 1990.
 - [5] P.K.Chan and S.J.Stolfo. "A comparative evaluation of voting and meta-learning on partitioned data". In *Proceedings of Twelfth 4th International Conference on machine Learning*, Tahoe City, CA, 90-98, 1995.
 - [6] T.Dietterich. "Machine Learning Research: Four Current Directions". *AI magazine*. 18(4), 97-136, 1997.
 - [7] T.Dietterich. "Ensemble Methods in Machine Learning". In *M.Arbib (Ed.) Handbook of Brain Theory and Neural Networks*, 2nd Edition, MIT Press, 2001.
 - [8] Y.Freund and R.Shapire. "Experiments with a new boosting algorithm". In L.Saitta (Ed.) *Machine Learning. Proceedings of the 13th International Conference*. Morgan Kaufmann, 1996.
 - [9] J.Gama and P.Brazdil. "Cascade generalization". *Machine Learning*, 41(3), 315-342, 2000.
 - [10] I.Goodman, R.Mahler, and H.Nguen." Mathematics of Data Fusion". Kluwer Academic Publishers, 1997.
 - [11] V.Gorodetski, O.Karsaev, I.Kotenko, and A.Khabalov. "Software Development Kit for Multi-agent Systems Design and Implementation". In B.Dunin-Keplicz and E.Nawareski (Eds.) "From Theory to Practice in Multi-agent Systems". *Lecture Notes In Artificial Intelligence*, vol. 2296, 121-130, Springer Verlag, 2002.
 - [12] V.Gorodetski, V.Skormin, and L.Popyack. "Data Mining Technology for Failure Prognostics of Avionics", *IEEE Transactions on Aerospace and Electronic Systems*, accepted for publication in 2002.
 - [13] V.Gorodetski, V.Skormin, L.Popyack, and O.Karsayev. "Distributed Learning in a Data Fusion System". In *Proceedings of the Conference of the World Computer Congress (WCC-2000) "Intelligent Information Processing" (IIP2000)*, , 147-154, 2000.
 - [14] V.Gorodetski and O.Karsayev. "Algorithm of Rule Extraction from Learning Data". In *Proceedings of the 8th International Conference "Expert Systems & Artificial Intelligence" (EXPERSYS-96)*, 133-138, 1996.
 - [15] S.Hashem. "Optimal linear combination of neural networks". *Ph.D. thesis*, Purdue University, , Lafayette, IN, 1997.
 - [16] M.Jordan and R.Jacobs. "Hierarchical mixtures of experts and the EM algorithm". *Neural Computations*, 6(2), 181-214, 1994.
 - [17] C.J.Merz. "Combining classifiers using correspondence analysis". In *Advances in Neural Information Processing*, 1997.
 - [18] J.Ortega, M.Coppel, and S.Argamon. "Arbitrating Among Competing Classifiers Using Learned Referees". *Knowledge and Information Systems* 3 (2001) 4, 470-490, 2001.
 - [19] M.Perrone and L.Cooper. "When networks disagree: Ensemble methods for hybrid neural networks". In R.J.Mammone (Ed.), *Neural Networks for Speech and Image Processing*, Chapman and Hall, 1993.
 - [20] A. Prodomidis, P. Chan, and S. Stolfo. "Meta-learning in distributed data mining systems: Issues and approaches". In P.Chan and H.Kargupta (Eds.) *Advances in Distributed Data Mining*, AAAI Press, 1999. Also available at <http://www.cs.columbia.edu/~sal/hpapers/DDMBOOK.ps.gz>.
 - [21] L.Rastrigin and R.Erenstein. "Methods of collective pattern recognition". Moscow, Energoizdat Publishers, 1981. (In Russian.)
 - [22] A.K.Seewald and J.Fuernkranz. "An evaluation of grading classifiers". In *Proceedings of 4th International Conference "Intelligent data Analysis"*, LNCS 2189, 115-124, 2001.
 - [23] K.Ting. "The characterization of predictive accuracy and decision combination". In *Proceedings of 13th International Conference on Machine Learning*, Morgan Kaufman, 498-506, 1996.
 - [24] K.Ting and I.Witten. "Issues in stacked generalization". *Journal of Artificial Intelligence Research*, 10, 271-289, 1999.
 - [25] L.Todorovski and S.Dzeroski. "Combining classifiers with meta decision trees". In D.A.Zighen, J.Komarowski and J.Zitkov (Eds.) *Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-2000)*, Lyon, France, 54-64, Springer Verlag, 2000.
 - [26] L.Todorovski and S.Dzeroski. "Combining multiple models classifiers with meta decision trees". In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, 54-64. Berlin, Springer Verlag, 2000.
 - [27] R.Vilalta and Y.Drissi. "A perspective view and survey of meta-learning". Submitted to the *Journal of Artificial Intelligence Review*, <http://www.research.ibm.com/people/v/vilalta/papers/jaireview01.ps>
 - [28] D.Wolpert. "Stacked generalization". *Neural Network*, 5(2), 241-260, 1992.
 - [29] <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
 - [30] <http://ics.uci.edu:pub/machine-learning-databases>.