Indian Journal of Science and Technology, *Vol 8(28), DOI: 10.17485/ijst/2015/v8i28/83326, October 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Detecting Credit Card Fraud using Data Mining Techniques - Meta-Learning

T. Abdul Razak[1*] and G. Najeeb Ahmed[2]

[1]Research Department of Computer Science, Jamal Mohammed College, Tiruchirapalli - 620 024, Tamil Nadu, India; abdul1964@gmail.com

[2]Research and Development, Bharathiar University, Coimbatore - 641046, Tamil Nadu, India; gnajeeb@rediffmail.com

Abstract

Data mining and machine learning techniques help us to better and deeper understanding of collected data. Meta-learning techniques extend this concept by providing methods for knowledge discovery process automatization. Meta-learning introduces various interesting concepts, including data meta-features, meta-knowledge, algorithm recommendation systems, autonomous process builders, etc. All these techniques aim to improve usually expensive and demanding data mining analysis. This paper focus on general overview of basic data mining, machine learning and meta-learning techniques, while focusing on state-of-the-art, basic formalisms and principles, interesting applications and possible future development in the field of meta-learning.

Keywords: Data Mining, DMA, KDD, Machine Learning, Meta-Learning

1. Introduction

Meta-learning methods are aimed at automatic discovery of interesting models of data. They belong to a branch of Machine Learning that tries to replace human experts involved in the Data Mining process of creating various computational models learning from data. Given new data and description of the goals meta-learning systems should support decision making in classification (Classification), regression (Regression, Statistics), association tasks, and/or provide comprehensible models of data (Rule-Based Methods). The need for meta-learning came with availability of large data mining packages such as Weka that contain hundreds of components (data transformations) that may be connected in millions of ways, making the problem of optimal model selection exceedingly difficult. Meta-learning algorithms that "learn how to learn" and guide model selection have been advanced in statistics, Machine Learning, Computational Intelligence and Artificial Intelligence fields. Learning from data, or understanding data requires many pre-processing steps, selection of relevant information, transformations and classification methods. Meta-learning techniques help to select or create optimal predictive models and reuse previous experience from analysis of other problems, relieving humans from most of the work and realizing the goal of computer programs that improve with experience (Brazil et al. 2009; Jankowski et al. 2011). These methods are designed to automatize decisions required for application of computational learning techniques.

*Author for correspondence

2. Data Mining and Machine Learning

2.1 Process of Data Mining

The term Data Mining is also referred as "Knowledge Discovery from Data (KDD)". This process will provide new Interesting and useful knowledge about collecting data.

KDD is used on very large datasets where there is no such way to get knowledge. Data mining is technically only a small part of the process. The process itself contains these following parts from[1]:

- Data cleaning – removes inconsistencies in the source datasets,
- Data integration – data from different sources have to be combined properly,
- Data selection – task-relevant data are retrieved from the source,
- Data transformation – data have to be transformed to appropriate task specific form,
- Data mining – appropriate algorithms extract data patterns,
- Pattern evaluation – interesting patterns are extracted based on different measures,44
- Knowledge presentation – visualization and knowledge representation to users.

At the end of the whole process, the interesting patterns are sometimes being stored in a system knowledge base in a form of a new knowledge. This technique presents an easy and convenient way for storing and subsequently browsing, comparing and visualizing a groups of related knowledge.

2.2 Data Mining Result

There is a variety of different results can be obtain using data mining and machine learning. In the scope of this report, i won't go into details about different data mining techniques since it is not my main focus; however, it is noted that there are two main types of data mining tasks:

- Descriptive – the result of the task are patterns describing the source data,
- Predictive – the result of the task is an applicable model (or models) with predictive abilities.

To give an example, online retailer has a database with all his customers and their previous orders. With the use of descriptive data mining techniques it is possible to associate which merchandise is selling together (using so-called association analysis). But I could also use certain predictive methods to categorize the retailer's customers into certain groups (e.g. With regards to monthly spending) so I would be able to predict a new customer's behavior (in this case how much he will spend in the store monthly) based on, e.g., his age, location, first few purchases, etc. More on this subject can be found in [2].

2.3 Machine Learning

Machine learning is one of many domains data mining derive its techniques. Machine learning focuses on automatic computer learning that is capable of making own decisions based on data. There are several types of machine learning tasks:

- Supervised learning – system is learning from the labeled examples in the training dataset
- Unsupervised learning – system is learning from unlabeled set of training data discovering target classes on the fly
- Semi-supervised learning – system uses both labeled and unlabeled examples learning the model from the labeled data and using unlabeled examples to refine class boundaries,
- Active learning – user is actively participating in the learning process. e.g., labeling unlabeled example on demand.

Term machine learning is sometimes used to address a subset of data mining methods as e.g., classification may be described as supervised learning and clustering as unsupervised learning.

2.4 Data Mining Data Sources

The most common data source for data mining application is a relational database. Another common sources are transactional databases that capture transactions, such as customer's purchases, etc., which are identified by transaction identity number and include a list of transaction items. Besides relational and transaction databases, there are many other forms of databases differing mainly in their semantics. As an example, we can mention temporal databases, spatial and spatio-temporal databases.

Aside from basic database structures, many companies store their big data in so-called data warehouses. Data warehouses are essentially a repositaries of information collected from many different sources under the same schema[3].

A data warehouse is usually presented in a form of a multi-dimensional data cube, where each dimension represents an attribute (or a set of attributes), while the cells themselves store a value of some aggregate measure over chosen dimensions. Data warehouse systems provide tools for Online Analytical Processing (OLAP) for interactive analysis 1of multidimensional data. OLAP enables analysts to view and change a level of abstraction and granularity of displayed measures, as well as arbitrary combine different data dimensions.

Another sources usable for data mining are data streams (infinite continuous streams of data without possibility to rewind or save all records), graph data, hypertext and multimedia data, and the Web. In regards to Web data sources,in recent years, cloud systems are rapidly gaining popularity, while the word cloud became a huge buzzword. The main principle of cloud computing is an idea that everything is stored and performed on external servers that are always accessable over the network. Such services are usually outsourced and provided to users with seemingly unrestricted access to their content. The number of cloud storages and their users grow every year. More information about cloud computing can be found in [4].

To extend on the concept of cloud computing, because of the amount of data that is processed by such systems, it is impossible to store the data in conventional manner. These so-called big data (more on this phenomenon in [5] are often stored in distributed data storages across many storage units. It is obvious, that all operations performed over such data need to be optimized for distributed architecture. To achieve required functionality, Google came up with solution in a form of Map Reduce model. This model automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks[6]. There are many concrete system implementation using this principle, one of the most known and used is Apache Hadoop. Hadoop is basically an open-source framework that supports large cluster applications by using its own distributed file system (HDFS). There are also many Hadoop extensions one of them being Apache Hive, which adds data warehouse infrastructure to the Hadoop system, allowing users to query, summarize, and analyze saved data. Regarding data mining, Apache Mahout is a scalable machine learning library that can work together with Hive and Hadoop to perform some basic data mining tasks. Complete overview of Hadoop and related technologies can be found in[7].

2.5 Data Mining Issues

I can address many issues connected to data mining. One of them may be that data mining as a scientific field and it is very large already and growing every moment. There is countless amount of specific applications and techniques involved, while new kinds of knowledge and algorithms are being discovered constantly. Another issue is related to data mining performance - algorithms should be as efficient and scalable as possible. Due to the existence of many types of data sources, mentioned in section 2.4, it may also be difficult (sometimes even impossible) to transfer newly discovered methods to another data source architectures. Other issues involve directly the data themselves – either regarding their structure (complex data types), or their content (missing values, noise, imbalance). All the main issues are described in[8] in much greater detail. The last issue we will mention in this report is user interaction during the data mining process. User interaction was already mentioned in section 2.3, dividing machine learning tasks into groups based on the type and the amount of interaction. Generally, data mining is highly interactive by its nature. This approach allows users to use their past experience, understanding domain and any type of additional background knowledge. A good domain knowledge, as well as sufficient knowledge of different data mining techniques and methods are necessary requirements for possibly successful process result as shown in (Figure 1). Current machine learning/data mining tools are only as powerful/useful as their users.

3. Meta-Learning

Meta-learning introduces intelligent data mining processes with the ability to learn and adapt based on previously acquired experience. This limits the amount of user input necessary to perform informed data analyzation task, which may be good either for pruning multiple tasks at once without overwhelming the analyst, or for automatic decision making without any need for user

intervention when the user himself may lack the expertise. Moreover, such system can learn from every new task, thus being more experienced and informed over time, providing new levels of adaptation to newly introduced obstacles. This area of research is also referred to as learning to learn. The primary goal of meta-learning is the understanding of the interaction between the mechanism of learning and the concrete contexts in which that mechanism is applicable. Learning at the meta-level is concerned with accumulating experience on the performance of multiple applications of a learning system. The main aim of current research is to develop meta-learning assistant, which are able to deal with the increasing number of models and techniques, and give advice dynamically on such issues as model selection and method combination. More about the basics purpose of meta-learning can be found in [8,9].

3.1 Basic Areas of Meta-learning Application

According to[10], there are several basic applications of meta-learning:

- Selecting and recommending machine learning algorithms,
- Employing meta-learning in KDD,
- employing meta-learning to combine base-level machine learning systems,
- Control of the learning process and bias management,
- Transfer of meta-knowledge across domains

3.2 History of Meta-Learning

As an early precursor of meta-learning, STABB system may be introduced, since it was the first to show that a learner's bias could be dynamically adjusted[11]. Next, VBMS (variable-bias management system) was developed as a relatively simple meta-learning system that learns to select the best among three symbolic learning algorithms as a function of only two dataset characteristics - the number of training instances and the number of features[12]. The first formal attempts at addressing the practice of machine learning by producing rich toolbox consisting of a number of symbolic learning algorithms for classification, datasets, standards and know-how were introduced in[12] in a form of the MLT project. During this project, many important machine learning issues was gained. Based on that, the user guidance system Consultant-2 was developed. Consultant-2 is a kind of expert system for algorithm selection - it provides the user with interactive question-answer sessions that are intended to collect information about the data, the domain and user preferences. Consultant-2, presented in [13], stands out as the first automatic tool that systematically relates application and data characteristics to classification learning algorithms. Later, a Web-based meta-learning system for the automatic selection of classification algorithms, named DMA (Data Mining Advisor), was developed as the main deliverable of the METAL project. This project focused on discovering new and relevant data/task characteristics, and using meta-learning to select best suitable classifiers for a given task. Given a dataset and goals defined by the user in terms of accuracy and training time, the DMA returns a list of algorithms that are ranked according to how well they meet the stated goals.

Other system, called IDA (Intelligent Discovery Assistant), provides a template for building ontology-driven, process-oriented assistant for the KDD process. It includes operations from the three basic steps of KDD - preprocessing, model building and post-processing. The main goal of IDA is to generate a list of ranked DM processes that are congruent with user-defined preferences by combining possible operations accordingly. This approach was presented in[14] and [15]. [16] then extends described concept by using both declarative information (ontology) as well as procedural information (system rules). Finally, in[17] most of the issues surrounding model class selection are addressed as well as a number of methods for the selection itself.

3.3 Employing Meta-learning in KDD

The KDD process can be viewed as a set of simple subsequent operations that can be further decomposed into smaller operations. These sequences can be characterized as partially ordered acyclic graphs and each partial order of operations can be regarded as an executable plan that produces certain effect. Examples can be found in[18].

The main goal, under this framework, is to automatically compose suitable executable plan with regard to the source data and previous system experience. The problem of generating a plan may be formulated as identifying a partial order of operations to satisfy certain criteria or maximize certain evaluation measures[19]. Naturally, the difficulty of this optimization process raises with the rising number of possible operations.

Generally, there can be two ways to approach the generation of the new plan:

- The system begins with an empty plan and gradually extends it with the composition of operators. This approach was presented in [20].
- The system starts with a suitable previously used plan and adapts it further to the exact needs of current task. More on this approach can be found in [21].

Although the idea of completely automatic generation of KDD process might be very appealing, it is important to note that this approach is inherently difficult. There needs to be many possibilities considered, some of them with high computational complexity. In the context of meta-learning, meta-knowledge can be used to facilitate this task. Past plans may be enriched with additional meta-information and can serve as procedural meta-knowledge. Other meta-knowledge may be captured about the applicability of existing plans to support reuse and on how they can be adapted to new circumstances. More information about this topic can be found in [22].

3.4 Combining Base-Level ML Systems

The approach of model combination is quite common nowadays, although it's not usually associated with the term meta-learning. However, its principles correspond with the meta-learning philosophy. Model combination consists of creating a single learning system from a collection of learning algorithms[23]. There are two basic approaches to this concept:

- System consists of multiple copies of a single algorithm that are applied to different subsets of the source data.
- System consists of a set of different mining algorithms that are trained over the same data.

The primary motivation for the model combination is usually to increase the accuracy of the final model; however, because it draws information about base-level learning (e.g., data characterization, algorithm characteristics . . .) methods for model combinations are often considered to be part of meta-learning. Perhaps the most known techniques for exploiting variation in data are bagging and boosting. They combine multiple models built from a single learning algorithm by systematically varying the training data[24]. Bagging, introduced in [25], produces replicate training sets by sampling with replacement from the set of training instances. This training set is the same size as the original data, but some tuples may not appear in it while others appear more than once (hence "with replacement"). Boosting (from[26]), on the other hand, maintains a weight for each training data instance – the higher the weight, the more the instance influences the classifier. At each trial, the vector of weights is adjusted to reflect the performance of the corresponding classifier in such way that the weight of misclassified instances is increased[27]. Bagging and boosting are easily applicable to various base-level learners and are proven to successfully increase the classification accuracy of created result models. Bagging and boosting exploit variation in the source data, thus they are methods belonging to the first mentioned model combination concept. Stacking and cascade generalization are then methods belonging to the second mentioned concept – they combine multiple learners to create a new learning method. Stacking creates a new learner that builds a meta-model mapping the precitions of the base-level learners to target classes. This method was presented in [28]. Cascade generalization, described in [29], also builds a meta-learner but rather then building it based on parallel results from base-level learners, it builds it subsequently – results of every base-level learner are enriched of meta-information and given on to the next base-level learner creating a chain-like structure. More proposed methods for model combination meta-learning, including cascading, delegating, arbitrating and meta-decision trees, are described in [30].

3.5 Meta-knowledge Transfer Across Domains

Accumulating meta-knowledge is one of the main targets of meta-learning. The amount of acquired meta-knowledge has a direct impact on the learning process itself – the methods prosper directly from greater amount of meta-knowledge (concrete quantifications of such benefits can be found in [31]). Because of this principle, it would be convenient to be able to transfer acquired meta-knowledge across different domains, potentially across different meta-learning systems. This problem is also known as inductive transfer. Methods have been proposed for transporting meta-knowledge across domains while preserving the original data mining/machine learning algorithm – there are methods for inductive transfer across neural networks, kernel methods and parametric Bayesian models (for more details on each method refer

to [32]). There are also other methods of transfer that are not directly connected to concrete models, such as probabilistic transfer, transfer by feature mapping and transfer by clustering. However, the issue of knowledge transfer is quite complicated and to be able to create a method for unlimited inductive transfer one would have to create a standardized high level meta-knowledge description language and corresponding ontology. For the complete overview and deeper description of inductive transfer methods and connected issues refer to [33].

4. Meta-Learning Systems

Meta-Learning in practice focuses on offering support for data mining. The meta-knowledge induced by meta-learning provides the means to inform decisions about the precise conditions under which a given algorithm, or sequence of algorithms, is better than others for a given task. In this chapter, which is based on the information from[34] and [35], we describe some of the most significant attempts at integrating meta-knowledge in DM decision support systems. While usual data mining software packages (e.g., Rapid-Miner,Weka) provide user-friendly access to wide collections of algorithms and DM process building, they generally offer no real decision support for non-expert users. It is also obvious, that not all phases of the KDD process can/should be automatized. Usually, the early stages (problem formulation, domain understanding) and the late stages (interpretation and evaluation) require significant human input as they depend heavily on business knowledge. Most of systems from this chapter has been already briefly mentioned in section 3.3; however, in following paragraphs we are going to describe the most interesting ones in greater detail.

4.1 Mining Mart and Preprocessing

Mining Mart, presented in[36] and [37], is a result of another large European research project focused on algorithm selection for data pre-processing. As mentioned in section 2.1, preprocessing is generally very time consuming (according to[38] almost 80% of the overall KDD process time) and it consists of nontrivial sequences of operations or data transformations. Because of that, the advantages of automatic user guidance are greatly appreciated. Mining Mart provides a case-based reuse of successful preprocessing phases across applications. It uses a metadata model, to capture information about data and operator chains through a user-friendly interface. Mining Mart has its own case base and every new mining task leads to its search through while looking for the most appropriate case for the task at hand. After that the system generates preprocessing steps that can be executed automatically for the current task. Similar efforts are also described in [39].

4.2 Data Mining Advisor and Ranking Classification Algorithms

The Data Mining Advisor (DMA) serves as a meta-learning system for the automatic selection of model building classification algorithms. The user provides the system with a source dataset, specific goals in terms of result model accuracy and process training time; subsequently, DMA returns a list of algorithms that are ranked according to user-defined goals (currently, there are 10 different classification algorithms). The DMA guides the user through a step-by-step process wizard defining the source dataset, computing dataset characteristics, and setting up the ranking method via defining selection criteria and selecting the ranking mechanism.

4.3 METALA and Agent-Based Mining

METALA is an agent-based architecture for distributed data mining, supported by meta-learning. It can be viewed as a natural extension of the DMA, mentioned in previous section. METALA provides the architectural mechanisms necessary to scale DMA up to any number of learners and tasks. Each learning algorithm is embedded in an agent that provides clients with a uniform interface so the system is able to autonomously and systematically perform experiments with each task and each learner to induce a meta model for algorithm selection. When a new algorithm or new task are added to the system, it performs corresponding experiments and the meta model is updated. More information about METALA can be found in [40] and [41].

5. Conclusions

In this paper I have discussed a generic architecture of a meta-learning system and showed how different com-

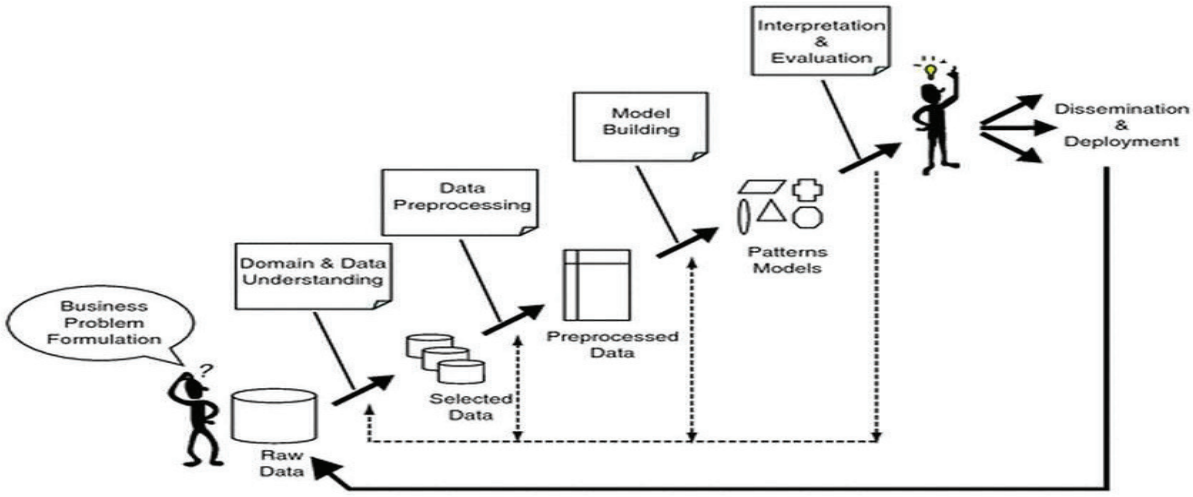

Figure 1. Data Mining Process – Sources.

ponents interact. I have provided a survey of relevant research in the field, together with a description of available tools and applications. One important research direction in meta-learning consists of searching for alternative meta-features in the characterization of datasets (Section 3.1). A proper characterization of datasets can interact between the learning mechanism and the task under analysis. While data mining and machine learning provide sufficient tools for deep data analysis, a lack of experience or other resources may prolong the pursuit of desired data knowledge. Meta-learning presents various techniques within different kinds of application to make the data mining process more autonomous, based on collected meta-knowledge. It presents some new concepts, e.g., meta-knowledge base, data meta-features, their extraction, base-learner combinations and even continuous data stream data mining. Meta-learning is a very variable field and its applications may severely differ. Many systems have been developed to include different types of meta-learning features; however, there is still much room for improvement as well as the development of new ideas. In [42] the authors claim that the focus should be on trying to determine not so much when certain algorithms work or fail.

6. References

1. Han J. Data mining: Concepts and techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2011.
2. Zhang S, Zhu C, Sin JKO, Mok PKT. A novel ultrathin elevated channel low-temperature poly-Si TFT. IEEE Electron Device Lett. 1999 Nov; 20:569–71.
3. Metev SM, Veiko VP. Laser Assisted Microtechnology. 2nd ed. In: Osgood RM Jr, editor. Berlin, Germany: Springer-Verlag; 1998.
4. Sorace RE, Reinhardt VS, Vaughn SA. High-speed digital-to-RF converter. United States Patent 19971605668842. 1997 Sep 16.
5. Breckling J. The analysis of directional time series: Applications to wind speed and direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer; 1989.
6. Shell M. IEEEtran homepage on CTAN. 2002. Available from:http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
7. Thrun S. Learning to learn: Introduction. In Learning To Learn. Berlin, Heidelberg: Kluwer Academic Publishers; 1996.

8. Thrun S. Lifelong learning algorithms. In: Thrun S, Pratt L, editors. Learning to Learn. US: Springer US; 1998. p. 181–209.
9. Giraud-Carrier C. Metalearning-a tutorial. Tutorial at the 2008 International Conference on Machine Learning and Applications, ICMLA; La Jolla, California. 2008 Dec 11–13.
10. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran.
11. Utgoff PE. Shift of bias for inductive concept learning. Machine learning: An artificial intelligence approach 2. Burlington: Morgan Kaufmann; 1986. p. 107–48.
12. Rendell L, Seshu R, Tcheng D. Layered concept-learning and dynamically variable bias management. Proceedings of IJCAI-87; Milan, Italy. Burlington: Morgan Kaufmann; 1987. p. 308–14.
13. Craw S, Sleeman D, Graner N, Rissakis M, Sharma S. Consultant: Providing advice for the machine learning toolbox. Proceedings of the Research and Development in Expert Systems IX; Cambridge: Cambridge University Press; 1993 Jul. p. 5–23.
14. Bernstein A, Provost F. An intelligent assistant for the knowledge discovery process. Information Systems Working Papers Series. 2001.
15. Bernstein A, Provost F, Hill S. Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. Knowledge and Data Engineering. IEEE Transactions. 2005; 17(4):503–18.
16. Charest M, Delisle S. Ontology-guided intelligent data mining assistance: Combining declarative and procedural knowledge. In: Angel Pasqual del Pobil, editor. Artificial Intelligence and Soft Computing; Palma De Mallorca, Spain. Calgary, Canada: Acta Press; 2006. p. 9–14.
17. van Someren M. Model class selection and construction: Beyond the procrustean approach to machine learning applications, Machine Learning and Its Applications. Berlin, Heidelberg: Springer; 2001. p. 196–217.
18. Bernstein A, Provost F, Hill S. Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. Knowledge and Data Engineering, IEEE Transactions. 2005; 17(4):503–18.
19. Shell M. IEEEtran homepage on CTAN. 2002. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
20. Bernstein A, Provost F, Hill S: Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification. Knowledge and Data Engineering, IEEE Transactions. 2005; 17(4):503–18.
21. Morik K, Scholz M. The miningmart approach to knowledge discovery in databases, Intelligent Technologies for information Analysis. Berlin, Heidelberg: Springer; 2004. p. 47–65.
22. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
23. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
24. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
25. Breiman L. Bagging predictors. Machine Learning. 1996; 26(2):123–40.
26. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. 1997.
27. Quinlan JR. Bagging, boosting, and c4.5. Proceedings of the Thirteenth National Conference on Artificial Intelligence; Portland, Oregon. California: AAAI Press; 1996. p. 725–730.
28. Wolpert DH. Stacked generalization. Neural Networks. 1992; 5:241–59.
29. Gama J, Brazdil P. Cascade generalization. Machine Learning. 2000; 41(3):315–43.
30. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran
31. Baxter J. A model of inductive bias learning. Journal of Artificial Intelligence Research. 2000; 12:149–98.
32. Pratt L, Jennings B. A survey of connectionist network reuse through transfer. In: Thrun S, Pratt L, editors. Learning to Learn. US: Springer US; 1998. p. 19–43.
33. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran.
34. Shell M. IEEEtran homepage on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran.
35. Giraud-Carrier C. Metalearning-a tutorial. In: Tutorial at the 2008 International Conference on Machine Learning and Applications; 2008 Dec 11-13; San Diego, CA. US; ICMLA; 2008.
36. Morik K, Scholz M. The miningmart approach to knowledge discovery in databases. Intelligent Technologies for Information Analysis. Berlin, Heidelberg: Springer; 2004. p. 47–65.
37. Kietz JU, Vaduva A, Zucker R. Miningmart: Metadata-driven preprocessing. Proceedings of the ECML/PKDD Workshop on Database Support for KDD; Freiburg. Berlin, Heidelberg: Springer; 2001. p. 15–65.
38. Shell M. IEEE on CTAN. 2002. Available from: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran

39. Weiss S, Indurkhya N, Zhang T, Damerau F. TextMining: Predictive Methods for Analyzing Unstructured Information. Berlin, Heidelberg: Springer Verlag; 2004.
40. Chan PK, Wei F, Prodromidis AL, Stolfo SJ. Distributed Data Mining in Credit Card Fraud. IEEE Intelligent Systems. 1999 Nov-Dec; 14(6):67–74.
41. Ogwueleka FN. Credit card fraud detection using data mining techniques [PhD Dissertation]. Awka, Nigeria: Department of Computer Science, Nnamdi Azikiwe University; 2008.
42. de Souto MCP, Prudencio RBC, Soares RGF. Ranking and Selecting Clustering Algorithms Using a Meta-Learning Approach. 2008.