Privacy-Preserving Data Mining based on Integrated Customer Databases from Different Enterprises

Gábor SZŰCS[1], Attila KISS[2]
[1]Inter-University Centre for Telecommunications and Informatics, H-4028 Kassai út 26, Debrecen; Department of Telecommunications and Media Informatics, BME, Hungary
[2] Department of Telecommunications and Media Informatics, BME, Hungary
szucs@tmit.bme.hu

The paper is about data mining projects in real applications, where preserving the users' privacy is important. The aim was to build a secure multiparty computation (SMC) data mining system with SMC data mining algorithm that would be able to solve the task of classification in a horizontally distributed environment with multiple parties trying for a joint data mining project. For solution of this kind of privacy preserving problems we have designed and developed an SMC system with different modules, a client module, a trusted third party and a classification module. We have worked out a new classification method; our k-means based supervised classifier preserves high level anonymity and provides k-anonymity, where k is a user parameter. At the end of the paper a bank example and its results with high accuracy present the efficiency of our system.

Keywords: *Anonymity, Customer Databases, Data Mining, Secure Multiparty Computation*

1 Introduction

Protection against misuse of data from data warehouses is an actual hot problem at the enterprises considering the privacy issues. This is reached by allowing data mining on the data without actually seeing personal information or individual records. The one of the basic approaches solving the issue is the randomization approach that focuses on individual privacy and preserves it by perturbing the data. The main idea of data perturbation is not to provide actual real data to the miner, but instead data that is modified in such a way, that distribution would be like in the original data, so the results of the mining would be still valid. The challenge here is to both perturb the data in such a way that no real information would be left behind, but no mining results would be altered.

At the Secure Multiparty Computation (SMC) the aim is to build a data mining model across multiple databases without revealing the individual records in each database to the other databases 0. In other words, we have a distributed network of data, and we want to make sure that no participant will get any results during the data mining process that cannot be inferred from the participant's input and the general output.

Mining financial information on a large scale would help a lot to financial institutions to understand trends better, to be able to choose their clients more carefully, to shape their product portfolio properly, so summing up: to conduct better business. Privacy concerns of financial information are confidential, we would not like other people to know exactly how much money we have, what we spend it for, where is our money coming from, and so on.

There are multiple methods that can be used to compare anonymizing capability. One type of measure is the simple mathematic probability of being able to estimate the original value of a record 000. Another type of method is measuring the occurrence of quasi-identifier values (these are attributes which identify the respondent with some degree of ambiguity) in the dataset. The goal is to being able to guarantee that there will be at least *k* records – meaning *k* people – who have the same quasi identifiers, so it would be impossible to anyone to narrow down the search for one specific person blow this *k* level. *K*-anonymity means that for any combination of quasi-identifier values at least k records exist in the database sharing the same values.

2 Privacy-Preserving Data Mining for Datasets

If we look at a dataset V, with *n* records and *m* attributes each record is an individual response 0 in the database. Attributes can be classified into four non-disjoint categories:

- *Identifiers:* These are attributes that unambiguously identify the respondent. Examples are the passport number, national insurance number, name-surname, etc.
- *Quasi-identifiers or key attributes:* These are attributes which identify the respondent with some degree of ambiguity. (Nonetheless, a combination of quasi-identifiers may provide unambiguous identification.) Examples are address, gender, age, etc.
- *Confidential outcome attributes:* These are attributes which contain sensitive information on the respondent. Examples are salary, religion, political affiliation, health condition, etc.
- *Non-confidential outcome attributes:* Those attributes which do not fall in any of the categories above.

There is one very simple method to protect personal privacy removing all personal identifiers from the dataset. These attributes can be as follows:

- name, surname, first name
- identification numbers (ID number, driver's license number, passport number, social security number, tax file number, etc.)
- other identification numbers (license plate, bank account number, other account numbers, etc.)
- contact information (full address, phone number, e-mail address, etc.)
- other personal information

Although removing this data is trivial and easily done, it still does not solve the problem in itself. Since even from the remaining data individual identification is more than possible, for example 87% of Americans can be identified based on their birthday gender and zip code 0.

At Secure Multiparty Computation (SMC) method group the goal is to conduct data mining tasks with multiple interested parties in a way that their data would never be published to any other participant or the data miner, and also that there could be no implications present in the end results regarding others original datasets. At SMC the data can be horizontally or vertically partitioned. When the parties have exactly the same attributes but for different data records, we call this case as horizontally partitioned data. At vertically partitioned data the parties have different attributes for same data records.

One of the largest types of PPDM is the randomization method family that can be further classified to perturbative and non-perturbative methods. In these cases the original datasets are modified with different algorithms so the published dataset for the mining task and the results are not based on actual personal information. For randomization methods the identifiers and quasi-identifiers have been removed from the database or coded in a way that they cannot be used to identify individual records. The goal of randomization is that if we have a dataset V with the original personal information, we instead of that release dataset V' that has been modified in such a way, that it would minimize disclosure risk (potential breach of privacy) and maximizes analysis outcomes (does not change key properties of dataset). The conversion from V to V' can be done by either masking the original data or by generating synthetic data that preserves the key statistical properties of the original dataset.

There are two or more companies that have their own databases and want to indulge in common data mining operations. This is a usual activity because of common business interest, wanting to learn more about the industrial trends, and seeking a better chance at a successful data mining project. The privacy enters into this task as a problem that the co-operative parties need to handle. No party has in their best business interest to share any of their own data with the other, and there are also personal privacy bounds that need to be respected on the customer side. In the next sessions we present our solution for these Secure Multiparty Computation problems.

3 Solution for Secure Multiparty Computation problem

3.1 SMC System Structure

For the solution of Secure Multiparty Computation problems we have designed and developed an SMC system with different modules. Our solution contains the following main components (as can be seen in Fig. 1.): client module, trusted third party and classification module. A client module consists of the subsystems for file handling, randomization processes, communication protocols, message cryptography and the SMC data mining algorithm itself. Every party involved in the joint mining process should run an instance of this component with the local dataset they possess. A trusted third party module is able to communicate with the client modules and able to execute auxiliary tasks for the data mining algorithm. The SMC algorithms need to ensure that any outgoing information is already privacy protected, so communicating this information to the third party would not be a breach even if the third party is compromised. The classification module is responsible for the classification of new cases. The data mining algorithm used in our solution was a k-means based supervised classification. The file handling system reads the database into the client module, and it is also responsible for writing dataset afterwards to a file if needed after modifications. Inputs at every client need to be identical regards to the file structure and attribute order, otherwise faulty operation can be expected. The communications module and the SMC are working very closely since the messages and communications are dependent solely on the data mining algorithm. The cryptography module is an optional part of the system, and actually this is not implemented, because the used SMC is capable without using cryptographic methods. At the randomization module several methods were applied.

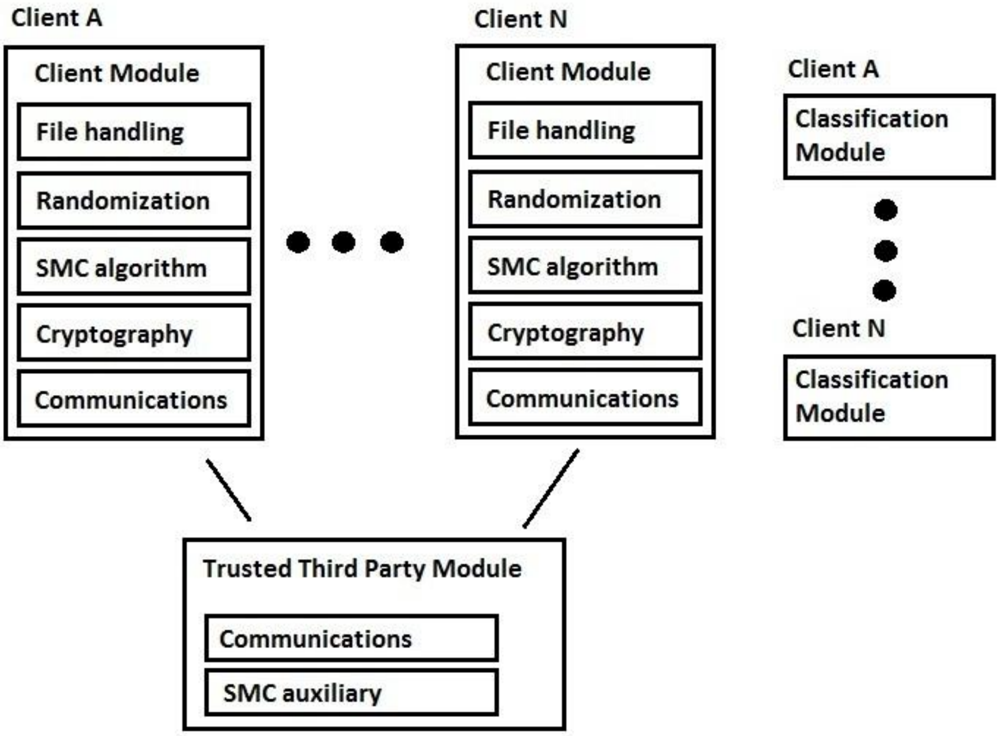

Fig. 1. Solution of SMC system structure

The features of SMC system are as follows.

- Every participant will start running the protocol on their own set of training data.

- At the end of the algorithm (that has a relatively lower iteration number) the updated cluster centers will be shared with the participants (or with the third party) with the following limitations that will guarantee the k-anonymity:
    - Each center has to have at least k members.
    - No cluster member can have the exact same values as the cluster center does.
- Closest cluster members are merged together, refining center values further.
- These steps will be repeated until:
    - The algorithm reaches the iteration number provided by the user.
    - The cluster center points stop moving.
    - The movement is under the threshold limit.

3.2 K-means Based Supervised Classification

The block diagram of our k-means based supervised classification solution can be seen in Fig. 2. This contains many modules, the details of them are described below.

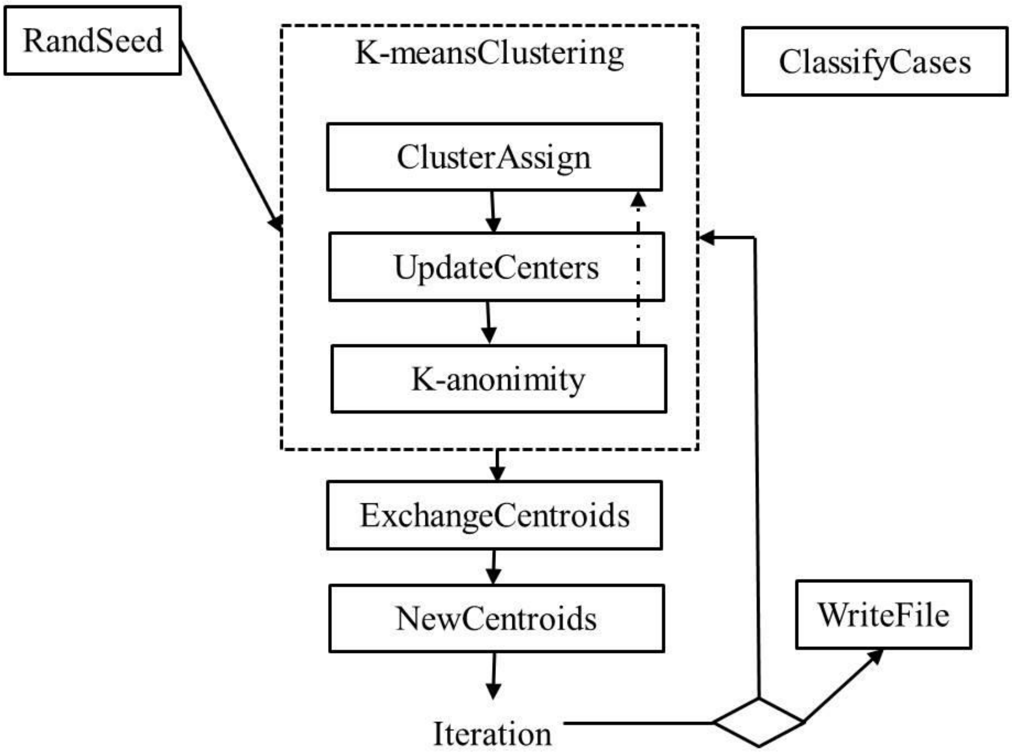

Fig. 2. K-means based supervised classification

RandSeed: as a first step a random seed function produces the *k* starting cluster centers from the original dataset. Selection of random data points is done by choosing random indexes.

K-meansClustering: The two major steps are going through all the data points and clustering them, and updating the cluster center once it’s done. The method is regulated by the iteration number which is realized in a cycle.

a. ClusterAssign: this is a clustering function for examining all cases. First step is the measuring the distance between the case and the cluster centers. (There are multiple distance measurements for both numerical and categorical data.) Afterwards the center possessing shortest distance to the actual examined case is assigned to this case.

b. UpdateCenters: after determining a cluster for each case, the update of cluster centers is executed. For numerical values simple average is calculated from the members, whereas for categorical values the mode (attribute that has the highest frequency) is selected.

c. K-anonimity: this is a section of the code in the K-means based supervised classification that is designed to ensure that no center information is shared with anybody if the center has less than k members.

ExchangeCentroids: with this function each participant first writes their own centroids to a file to the designated store with the predetermined format, then in case of using independent center determination they reads all files from other participants, and stores their centroid data in a local storage. If using the Trusted Third Party (TTP) solution then the TTP module will read, calculate and write the ready new centroids to a file and this function will only need to read that.

Iteration: afterwards the iterations of the SMC algorithm start. The first step is calculating new centroids based on the files read in, then the next step is running the new *k*-means, and finally centroids are exchanged again.

a. NewCentroids: calculating new centroids starts with a cycle where distance between own and all other participant's centroids are calculated. After calculating the distances ranking commences that will reside in orders from 1 to *k* between each centroid pair from the own set and any other participant's set. Using this the closest centroids are drawn to *k* clusters with number of participants, and the new centroids are calculated on the model of the UpdateCenters function. This function is only used if there is no TTP module to take over this responsibility.

b. K-meansClustering: as described before.

c. ExchangeCentroids: as presented before.

WriteFile: Finally after all iterations of the SMC algorithm has been executed, thus formed final centroids are stored and also written to a final resulting file, that can be used as for basis of further classification.

ClassifyCases: The ClassifyCases function is responsible for classification of new instances based on the training that was achieved with the help of modules described above. This function is similar to ClusterAssign function, but in this case the cluster centers are considered as class representatives; ClassifyCases searches the closest class representative for the unknown case and the decision will be the label of this representative item.

At the algorithm distances need to be calculated from every centroid. This is done separately for numerical and categorical values, because generally different methods are used. The exact distance measure is selected by the user parameters in options. At the moment two different methods are implemented for each type of data. At numerical type the distance can be *Euclidean* or *Manhattan distance*. At categorical data the distance can be *Overlap distance* (where the values are identical for two cases in question the distance is 0, and where they are different the distance is 1, and the sum of them is divided by the maximum to create a normalized distance between 0 and 1.) and the *Eskin distance* 0 (which is calculated similarly to the Overlap measure, the difference being that for every non-identical dimension the distance only grows by the reciprocal of the number of values that attribute could have taken; thus for dimensions with many possible values the distance will be smaller in case of mismatch than for binomial ones for example).

After calculating the distances the ranking process starts. More methods for ranking were created, one of which also needs to be selected for every run of the program:

- Numerical priority: in this case all centroids are ranked based on the numerical distances and smallest is selected. In case of identical values for numerical distances the smallest categorical distance will be the chosen centroid.
- Categorical priority: similar to the one above, with the difference that categorical ranking is the first aspect.
- Joint with numerical decision: ranks of numerical and categorical distance are added up for each centroid, and the smallest is selected. In case of identical joint distances the smaller numerical distance will be selected.

- Joint with categorical decision: similar to the one above but in case of identical joint distances the categorical distance is decisive.

At the whole system the end user can choose different options. We have implemented 13 options as follows.

1. type of categorical distance measure (Overlap or Eskin distance)
2. type of numerical distance measure (Euclidean or Manhattan distance)
3. type of cluster ranking
4. type of randomization (noise addition, data swapping, rounding, microaggregation)
5. number of desired cluster centers
6. number of desired k-means iterations
7. number of desired SMC iterations
8. participants' ID
9. number of participants
10. common filename
11. classification designation for the current file
12. usage of trusted third party
13. *k* parameter for k-anonymity

The all above elements were integrated into a single system that can be run by anybody on a windows PC if they have the necessary inputs.

4 A Bank Example and its Results

The banks have roughly the same type of data about their customers, and each bank has own set of customers (the horizontally partitioned data). Here they will need to converge their datasets so that they would have a bigger base data for better mining results. The task is to specify and build a system where privacy preserving data mining can be executed on these joint datasets. Classification problems need to be defined on these datasets and an SMC classification technique needs to be developed to execute classification tasks in an SMC manner. After building the algorithm and the model, it needs to be run on the datasets and the results are required to be evaluated.

For the banking dataset a Portuguese retail bank's dataset was used that was the basis for a data mining project on a marketing campaign 0. The dataset contains 45000 cases that are all individual customers of the bank, and all of these cases have 17 attributes. This dataset was split into three parts (as 3 banks), in a horizontal manner, meaning that different parts possess the same attributes and respectively 40%-40%-20% of the cases.

The dataset contains information about the customer's financial situation, and demographics. The attributes in the database are as follows: (*identity of attribute – name of attribute: what is it about (type: possible values)*)

- 1 - age (numeric)
- 2 - job : type of job (categorical: "admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services")
- 3 - marital : marital status (categorical: "married", "divorced", "single")
- 4 - education (categorical: "unknown", "secondary", "primary", "tertiary")
- 5 - default: has credit in default? (binary: "yes","no")
- 6 - balance: average yearly balance, in euros (numeric)
- 7 - housing: has housing loan? (binary: "yes","no")
- 8 - loan: has personal loan? (binary: "yes","no")
- 9 - contact: contact communication type (categorical: "unknown", "telephone", "cellular")
- 10 - day: last contact day of the month (numeric)
- 11 - month: last contact month of year (categorical: "Jan", "Feb", "Mar", ..., "Nov", "Dec")
- 12 - duration: last contact duration, in seconds (numeric)
- 13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- 14 - pdays: number of days that passed by after the client was last contacted from a previous campaign

(numeric, -1 means client was not previously contacted)

- 15 - previous: number of contacts performed before this campaign and for this client (numeric)
- 16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown", "other", "failure", "success")
- 17 - y: has the client subscribed a term deposit? (binary: "yes","no")

The task is to classify instances whether they would subscribe to a term deposit, so in this case, the label is the already given attribute 17, and all other attributes are inputs.

After getting the system up and running it we have tested whether every component works as it should be, and to see how the anonymity and efficiency values will be. First we have started with the whole test dataset without splitting it for multiple users. For latter comparison some classification methods were used on this dataset, and the results could be used as a reference point. Obviously for best possible results the parameters of the functions should have been altered, and the best result of each method can be seen in Table 1.

Table 1. Accuracy and recall results of original data

	J48 (C4.5)	**Naïve Bayes**	**k-NN**
Accuracy	89.77%	87.75%	88.18%
Recall	47.13%	52.74%	24.64%

After running the reference results, the parameter setup for the SMC system could start. Different algorithms were run by parameters: 70% of full dataset for training in two steps, k-means iterations, SMC iterations, number of clusters, k-anonymity parameter. At our example there were 3 banks, thus the data were split to 40-40-20 stratified (meaning that the ratio of classes was the same in files) part. (The three partitions were further split to 70-30 parts for the training and test sets as mentioned before.) The parameters for each case were as follows:

Table 2. Parameters of test runs

Run	**# clusters**	**#k-m. iter.**	**#SMC it.**	**k-anonymity**	**distance and ranking**
No.1	10	25	5	20	E, O, 3
No.2	15	20	15	10	E, E, 3
No.3	10	25	10	15	M, O, 4
No.4	20	20	15	20	M, E, 4
No.5	10	25	10	10	E, O, 3
No.6	15	20	15	15	E, O, 3

At Table 2. the first column has the name for the run, the second has the number of cluster centroids requested, the third is the number of k-means iterations, the fourth is the SMC iteration number, the next is the k-anonymity. The last column shows the distance and ranking, where the first letter E means Euclidian, M means Manhattan; the second letter (for categorical type) O means Overlap, and E means Eskin distance; the third number is for the ranking method where 1 is numerical, 2 is categorical, 3 and 4 are joint, with 3 having numerical focus, 4 having categorical focus.

Efficiency values for the different versions run, as results of our test can be seen in the next table (Table 3.):

Table 3. Results of multiparty runs of the SMC system

Run	**No.1**	**No.2**	**No.3**	**No.4**	**No.5**	**No.6**
Accuracy	70.8%	82.05%	72.34%	68.2%	75.6%	74.27%

Recall	54.32%	65.0%	56.8%	53.2%	57.9%	54.8%

Efficiency values show that the recall is much higher than at original one (besides the accuracy is a little bit lower). The results of both the SMC system and the SMC algorithm are more than satisfactory. The system works well, and it delivers adequate efficiency and gives anonymity possibility.

5 Conclusion

Data mining services require accurate input data for their results, but privacy concerns may influence users to provide spurious information. To preserve customers' privacy in the data mining process, a variety of techniques can be used based on randomization in order to avoid users' doubt.

After a thorough examination of the randomization methods (both perturbative and non-perturbative) most currently used algorithms and methods are included into our solution. These methods try to protect anonymity with data manipulation and query restrictions. The main focus of this paper was the secure multiparty computation (SMC) algorithm. The SMC methods are concerned with distributed data mining on horizontal data in different settings and with different data problems.

Our task was to build an SMC data mining system with an SMC algorithm that would be able to solve the task of classification in a horizontally distributed environment with multiple parties trying for a joint data mining project. We have planned and developed a system with secure multi-party computation technique for classification of customers in an integrated database from different enterprises. Our method is a new approach to the problem: using K-means as managed learning algorithm for classification all with preserving high level anonymity and providing k-anonymity (*k* being a user parameter). At the end of the paper a bank example was shown, our classification method was tested, the privacy-preserving data mining system was evaluated and the high accurate results present the efficiency of our system.

Acknowledgements

The publication was supported by the TÁMOP 4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

References

[1] D. Agrawal and C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms", In *Proceedings of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Santa Barbara, CA, 2001, May 21–23, pp. 247–255.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining", In *Proceedings of the ACM SIGMOD Conference on Management of Data*, ACM Press, 2000, pp. 439–450.

[3] B. Bergstein, "Research explores data mining, privacy", USA Today, 18 June, 2006.http://www.usatoday.com/tech/news/surveillance/2006-06-18-data-mining-privacy_x.htm

[4] E. Bertino, D. Lin, and W. Jiang, "A survey of quantification of privacy preserving data mining algorithms". In *Privacy-preserving data mining*. Springer US, 2008, pp. 183-205.

[5] J.Domingo-Ferrer, "A survey of inference control methods for privacy-preserving data mining". In *Privacy-preserving data mining.* Springer US, 2008. pp. 53-80.

[6] E. Eskin, A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, "A geometric framework for unsupervised anomaly detection", In *Applications of data mining in computer security,* Springer US, 2002. pp. 77-101.

[7] S. Moro, R. Laureano and P. Cortez, "Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology", In P. Novais et al. (Eds.), *Proceedings of the European Simulation and Modelling Conference - ESM'2011*, Guimarães, Portugal, October, 2011. EUROSIS. pp. 117-121.

[8] G. Szűcs, "Decision Trees and Random Forest for Privacy-Preserving Data Mining". In K. Tarnay, S. Imre, & L. Xu (Eds.), *Research and Development in E-Business through Service-Oriented Solutions* (Chapter 4), 2013, pp. 71-90. Hershey, PA, USA, ISBN: 978-1-4666-4181-5, doi:10.4018/978-1-4666-4181-5.ch004

Gábor SZŰCS was born in 1970 in Hungary. He has received MSc in Electrical Engineering from Budapest University of Technology and Economics (BME) in 1994. He is experienced in modeling and simulation, railway systems, traffic systems; he has received PhD degree in this field from BME in 2002. His further and currently research areas are data mining in business areas, multimedia mining, content based image retrieval, semantic search. He is associate professor at Department of Telecommunications and Media Informatics of BME. The number of his publications is more than 80. He is vice president of the Hungarian Simulation Society (EUROSIM), deputy director of the McLeod Institute of Simulation Sciences Hungarian Center. He has earned János Bolyai Research Scholarship of the Hungarian Academy of Science in 2008.

Attila KISS was born in Hungary, in 1990. He received his B.Sc. degree in computer engineering from the Budapest University of Technology and Economics, Hungary in 2012, with a substitute degree in IT entrepreneurship from Aquincum Institution of Technology, Budapest, Hungary in same year. He is currently concluding M.Sc. studies in business information systems at the Budapest University of Technology and Economics, Hungary, with a term abroad at University of Technology Sydney, Australia in 2012. In 2011 he was a trainee at SAP Labs Hungary, working on a new wave SAAS development project. He was a board member at Simonyi College for Advanced Studies with the role of CFO in 2010, and President in 2011. His main research interests are revolving around customer segmentation and churn management. Mr. Kiss awards and honors include the Life Technologies Scholarship (Life Technologies Inc, 2011) and the UTS Business Faculty Scholarship (University of Technology Sydney, 2012).