Computing and Informatics, Vol. 39, 2020, 361–384, doi: 10.31577/cai_2020_3_361

EFFECTIVE UTILIZATION OF SUPERVISED LEARNING TECHNIQUES FOR PROCESS MODEL MATCHING

Khurram Shahzad, Arslaan Mazhar

Punjab University College of Information Technology
University of the Punjab, Lahore, Pakistan
e-mail: {khurram, mscsf14m032}@pucit.edu.pk

Ghulam Mustafa

University of the Punjab, Gujranwala Campus, Pakistan
e-mail: gmustafa@pugc.edu.pk

Faisal Aslam

Punjab University College of Information Technology
University of the Punjab, Lahore, Pakistan
e-mail: faisal.aslam@pucit.edu.pk

Abstract. The recent attempts to use supervised learning techniques for process model matching have yielded below par performance. To address this issue, we have transformed the well-known benchmark correspondences to a readily usable format for supervised learning. Furthermore, we have performed several experiments using eight supervised learning techniques to establish that imbalance in the datasets is the key reason for the abysmal performance. Finally, we have used four data balancing techniques to generate balanced training dataset and verify our solution by repeating the experiments for the four datasets, including the three benchmark datasets. The results show that the proposed approach increases the matching performance significantly.

Keywords: Business process management, process model matching, artificial intelligence, supervised learning techniques, machine learning, data balancing

1 INTRODUCTION

Process models are the conceptual models that represent business operations of an enterprise. These models are widely acknowledged as useful artifacts for documenting software development requirements as well as configuring ERP systems. Process Model Matching (PMM) refers to the identification of the corresponding activities between a pair of process models that represent identical or similar behavior [1]. For a further understanding of PMM problem, consider the excerpt process models of two universities presented in Figure 1. Both process models are composed of a start node represented by a circle, an end node represented by a solid circle, four activities represented by a rectangle with rounded edges, control flow between activities represented by arrow signs, and two gateways represented by a diamond sign having "+" signs. In the figure, the four shaded areas (C1, C2, C3 and C4) represent the four corresponding activity pairs that should be identified by a matching technique based on the similarity of activity labels. While the identification of corresponding activities having identical labels is a trivial task, the real challenge lies in the identification of activities having similar business semantics but different formulation of labels.

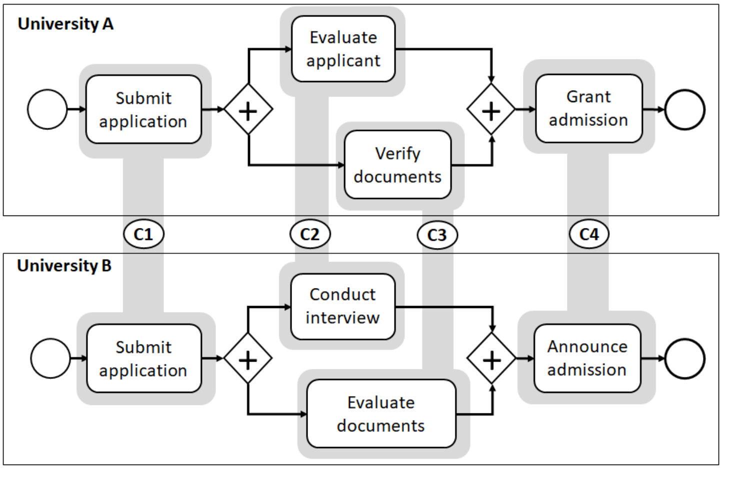


Figure 1. Illustration of the PMM problem

The identification of corresponding activities has several use cases [2]. Firstly, it is a pre-requisite to evaluate if the given process models are similar. Secondly, matching techniques, if embedded to a process model repository, can avoid redundant process models, which may lead to several inconsistencies. Thirdly, it can also play a pivotal role in querying process model repository, as querying involves matching the query model and the models stored in the repository. Finally, harmonizing process model variants is another use of process model matching. Due to these diverse use cases of PMM, a plethora of techniques have been developed [3, 4]. However, a recent study has highlighted that the F1 score of these techniques vary between 0.45 and 0.67 [5]. This lower value of F1 score highlights the need for developing PMM techniques that can identify corresponding activities with a higher accuracy.

Supervised learning techniques have been widely used in the natural language processing for a variety of text processing tasks, such as word sense disambiguation, text matching, and named entity recognition [6, 7]. It is because the supervised learning techniques use training data to elicit knowledge and subsequently utilize it to predict the solution of a given problem. On the contrary, the unsupervised learning techniques use mathematical models or heuristics to generate a solution of the given problem, without using any insights from the existing solutions. From the text processing literature, it is abundantly established that typically supervised learning techniques have outperformed unsupervised techniques for several text processing tasks [8]. However, the potential of supervised learning techniques is yet to be fully exploited in the context of PMM. For instance, a recent attempt [9] to adapt a supervised learning technique on the PMMC '15 datasets has achieved an F1 score of 0.61 that is substantially less than the maximum F1 score of 0.67, achieved by a traditional unsupervised approach [5].

In this study, we have experimentally established the cause of abysmal performance of supervised learning techniques, and we propose a solution to rectify it. Specifically, we make the following main contributions. Firstly, we have employed a systematic protocol to transform the benchmark correspondences into a readily usable form for the supervised learning techniques. Secondly, we have performed several experiments to establish that the presence of imbalance in the datasets impedes the performance of supervised learning techniques. Precisely, we have used ten different feature measures for training, we tweaked the weights of tokens, and we adjusted the size of training datasets. Finally, to rectify the imbalance problem, we have proposed a data balancing based technique and evaluated its effectiveness for process model matching.

The rest of the paper is organized as follows: Section 2 presents the protocol that we have employed to transform the benchmark datasets. Section 3 investigates the causes of below par performance of the supervised learning techniques. Section 4 presents our proposed solution and the experimental setting that we have used to evaluate the proposed technique. The analysis of the results are presented in Section 5. A brief overview of the related work is presented in Section 6. Finally, in Section 7 we draw conclusions.

2 TRANSFORMATION OF BENCHMARK DATASETS

In this section, we present our first contribution, the transformation of benchmark dataset to make it readily available for supervised learning techniques. Below, we introduce the three widely used datasets, and the protocol that we have used to transform the benchmark datasets.

Source datasets. We conducted a comprehensive search of process model matching literature by querying multiple digital repositories, such as `Springerlink` and `ScienceDirect`, using several keywords. The retrieved items were manually screened to obtain 17 research articles that have experimentally evaluated the effectiveness of PMM techniques. Table 1 provides the list of the studies and the datasets used in these studies for the evaluation of the process model matching techniques. It can be observed from the table that all the existing studies have either used PMMC '15 datasets or their earlier versions for the evaluation. This indicates that any findings that stem from the use of the PMMC '15 datasets are acceptable for the community.

Dataset	**References**
Earlier version of a PMMC '13 dataset	[11, 19, 20]
PMMC '13 datasets	[21, 22]
PMMC '15 datasets	[1, 2, 3, 5, 9, 23, 24, 25, 26, 27, 28, 29]

Table 1. Benchmark datasets used in literature

The PMMC '15 datasets include University Admission (UA), Birth Registration (BR), and Asset Management (AM) collections of 9, 9 and 72 process models, respectively. The models are designed in BPMN, PNML and EPML formats having 289, 238, and 1 993 activities, respectively. Furthermore, each dataset includes 36 process model pairs and benchmark correspondences between activities of each process model pair. The key features of the PMMC '15 datasets that we have used are the following: Firstly, the datasets are publicly available hence the results produced using them are universally verifiable. Secondly, they include real-world process models from three different domains, providing sufficient diversity. Hence, any findings based on these datasets are likely to be applicable to other domains. Lastly, each dataset also includes a collection of gold standard correspondences that can be used as a benchmark for the evaluation of process matching techniques.

Although the collections of process models have sufficient diversity to challenge the capabilities of process matching techniques but the initial screening of the datasets revealed that the benchmark correspondences are not readily usable for the supervised learning techniques. It is due to the following reasons: Firstly, the storage format of each dataset is different. Secondly, the benchmark correspondences are limited to equivalent or optimal equivalent pairs, whereas the sub-optimal equivalent pairs and the pairs in which one label subsumes the other label are not provided. Thirdly, information regarding unequivalent pairs is not explicitly provided.

Transforming the benchmark datasets. In the first transformation step, we wrote a parser that can extract activity labels from BPMN, PNML and EPML formats (the formats in which the three datasets are currently available), and store them in CSV files. Subsequently, we generated a cross-product between all activities of each process model pair which resulted in 36 675, 25 045 and 30 764 activity pairs for the UA, BR and AM datasets, respectively. In the second step, the activity pairs that were declared equivalent in the PMMC '15 gold standard were marked as equivalent pairs in the cross-product, by parsing the gold standard available in the RDF format. In the third step, we engaged two researchers with expertise in process modeling to evaluate the remaining pairs in the cross-product. Specifically, the experts were explicitly told to mark an activity pair as equivalent even if that activity pair is a sub-optimal or a subsumption pair. Finally, the disagreements between the two researchers were resolved with the help of another expert. Accordingly, for each dataset, we generated a CSV file that contains activity labels, as well as human decision about corresponding pairs. Hence, making the datasets readily available for supervised learning techniques by omitting the hassle of parsing multiple data formats. The detailed specifications of each dataset is presented in Table 2.

	UA	**BR**	**AM**
No. of Model Pairs	36	36	36
No. of Pairs in the Dataset	36 675	25 045	30 764
No. of Positive Examples	232	645	222
No. of Negative Examples	36 443	24 400	30 542
Imbalance Ratio	1:157	1:38	1:138

Table 2. Specification of the human benchmark

3 PROCESS MODEL MATCHING USING SUPERVISED LEARNING

In this section, we present our second contribution, experimentation to identify the cause of the below par performance of supervised learning techniques.

3.1 Supervised Learning Techniques

We have selected eight diverse supervised learning techniques for our experiments as each technique has its own strenghts and weaknesses. These techniques are, Naive Bayes, Simple Logistic, IBK, AdaBoostM1, Decision Table, J48, LMT, and Random Forest. We have selected these techniques due to their effectiveness in text processing tasks.

Among these techniques, Naive Bayes is a robust generative classification algorithm that is less sensitive to noisy data and produces stable predictions. It is widely acknowledged as an effective technique for word disambiguation. Whereas, in Simple Logistic, LogitBoost is used as a base weak learner to fit the logistic models. The repetitions of LogitBoost are cross-validated to produce the optimum results. K-Nearest Neighbor (IBK) selects a feature-space based on the nearest neighbors. J48 is an implementation of a decision tree algorithm to predict class labels. AdaBoostM1 algorithm focuses on the hard-to-learn examples using pseudo-loss function. Decision tables are used as hypothesis-space for supervised learning in the Decision Table algorithm. Logistic Model Trees is the supervised learning algorithm that combines logistic regression to decision tree learning algorithm to improve results. These techniques reduce both bias and variance due to its constituent methods. Lastly, Random Forest is an ensemble classification technique which uses bagging and decisions trees to predict the class labels.

3.2 Feature Measures

We have selected ten text matching measures that are widely used to compute similarity between a pair of sentences. The reasons for the choice of such a large number of features are twofold: firstly, to supplement the impact of an individual feature on a learning technique. Secondly, to increase the breadth of the knowledge base of a learning technique and thereby offering more opportunities for eliciting the hidden knowledge. A brief overview of these measures are as follows:

Levenshtein distance. A distance based measure that computes distance between two input strings by computing a normalized score of the minimum number of character edit operations required to convert one label into the other. For two labels l_1 and l_2, the Levenshtein distance is computed as follows:

$$edit_{norm}(l_1, l_2) = 1 - \frac{|edit\ distance(l_1, l_2)|}{\min(|l_1|, |l_2|)}.$$

Cosine similarity. This similarity measure generates a vector representation of both labels. Subsequently, the similarity is computed by the *cosine* of angle between the two vectors.

$$\cos_{sim}(l_1, l_2) = \frac{\overrightarrow{l_1} \bullet \overrightarrow{l_2}}{\left(|\overrightarrow{l_1}||\overrightarrow{l_2}|\right)}.$$

Euclidean distance. Similar to cosine similarity, it first generates a vector representation of both labels. Subsequently, euclidean distance is the square root of the sum of squared differences between the vectors of two labels. Formally, it is defined as follows:

$$EU_{dis}(l_1, l_2) = \left[\left(\overrightarrow{l_1} - \overrightarrow{l_2}\right) \bullet \left(\overrightarrow{l_1} - \overrightarrow{l_2}\right)\right]^{1/2}.$$

Monge-Elkan. A token based approach in which similarity between two labels is computed by measuring the average of the similarity values between pairs of more similar tokens within label l_1 and l_2 [13]. Formally, it is defined as follows:

$$MonEl_{sim}(l_1, l_2) = \frac{1}{|l_1|} \sum_{i=1}^{|l_1|} \max\{sim(l_{1_i}, l_{2_j})\}_{j=1}^{|l_2|}.$$

Block distance. Block distance is depicted in two dimensions with discrete vectors. It calculates the distance between two data points using a grid-like path. The bock distance between two data points is the aggregation of the differences of their corresponding components. Formally, it is defined as follows:

$$Block_{dist}(l_1, l_2) = \sum_{i=1}^{n} |l_{1i} - l_{2i}|.$$

Jaccard similarity. A set theory based measure that treats each label as a collection of tokens. According to this measure, each label is tokenized into words and represented as a set. Jaccard similarity is then the ratio between the number of common words between the two sets and total number of words in both sets.

$$Jac_{sim}(l_1, l_2) = \frac{|S(l_1) \cap S(l_2)|}{|S(l_1) \cup S(l_2)|}.$$

Jaro-Winkler distance. A type of string edit distance that is faster than the Levenshtein distance as it computes the similarity between two labels by counting the number of matching characters in the strings and transpositions. Formally, it is defined as follows:

$$M_w(l_1, l_2) = M_j + (lp\ (1 - M_j)),$$
$$M_j = \frac{1}{3} + \frac{s}{|l_1|} + \frac{s}{|l_2|} + \frac{s-t}{s}$$

where l is the length of the common prefix, p is the constant scaling factor, and s is the number of matching characters between the two labels. Furthermore, M_j represents Jaro distance, s is the number of matching labels, and t is half the number of transpositions.

TagLink token similarity. It is an adaptive hybrid method of tag-based and link-based similarity in which Tag Commonness (TC) and Link Strength (LS) is dynamically determined. The main idea behind this method is to combine the tag-based and link-based approach to achieve the optimal similarity results. It uses a variation of Jaccard similarity as link-based similarity (δ_{link}) and a variation of tf-idf cosine similarity as a tag-based similarity (δ_{tag}) [33]. Formally, it is defined as follows:

$$TagLink(l_1, l_2) = \frac{TC_{l_1}}{TC_{l_1} + LS_{l_2}} \delta_{tag(l_1,l_2)} + \frac{LS_{l_1}}{TC_{l_1} + LS_{l_2}} \delta_{link(l_1,l_2)}.$$

Matching Coefficient. An elementary vector based approach which counts the number of similar terms on which both vectors are non-zero. This similarity measure is useful if attributes have symmetry in data, i.e., they carry comparable information. It is sensitive to the variable input and does not generalize very well.

$$MC = \frac{number\ of\ matching\ attributes}{total\ number\ of\ attributes}.$$

Soundex. A phonetic based measure that encodes homophones for the same representation so that they can be matched even when there is a minor difference in spellings. In simple words, the words that rely on similarity of pronunciation rather than the vocabulary are declared as similar.

3.3 Datasets

We have used four datasets for experimentation, three PMMC datasets and another Large (LR) dataset that has been recently developed [31], to demonstrate the external validity of our findings. A key feature of the LR dataset is that it is handcrafted to challenge the abilities of PMM techniques. The LR dataset contains 600 process models from different genres and benchmark correspondences between 406 process models pairs. The dataset contains 89 559 activity pairs, including 6 443 equivalent pairs and 83 115 unequivalent pairs with an imbalance ratio of 1:13. All the four datasets used for experimentation were available in CSV format having four columns, an identifier, a pair of activities, and human decisions.

3.4 Conducting the Experiments

For the experiments each label was tokenized, stop words were removed, and each token was stemmed to generate its corresponding stem. Subsequently, the values of all the 10 features were computed for each activity pair which were given as input features to the eight supervised learning techniques. For each experiment, the corresponding dataset was divided into training and testing datasets. To reduce the bias that might occur due to the choice of training and testing dataset, 10-fold cross-validation was performed for each dataset separately. Note that we have used a widely used features selection technique, Information Gain, to identify the optimal set of features, and subsequently used the optimal set of features for experimentation. However, the effectiveness scores were compromised, therefore, the results presented in this study are generated by all the 10 features.

We have used Precision, Recall and F1 scores to evaluate the effectiveness of supervised learning techniques. In the context of process model matching, Precision (P) refers to the proportion of activity pairs that are declared equivalent by a technique and also marked equivalent in the human benchmark. Recall (R) refers to the proportion of activity pairs that are marked equivalent in the human benchmark and also declared equivalent by a technique, whereas, F1 is the harmonic mean of P and R.

3.5 Results

Table 3 shows the average F1 scores of each supervised learning technique. Note, we have synthesized the results to separately evaluate the effectiveness of each technique for equivalent and unequivalent pairs, separately. From the results we have observed the following:

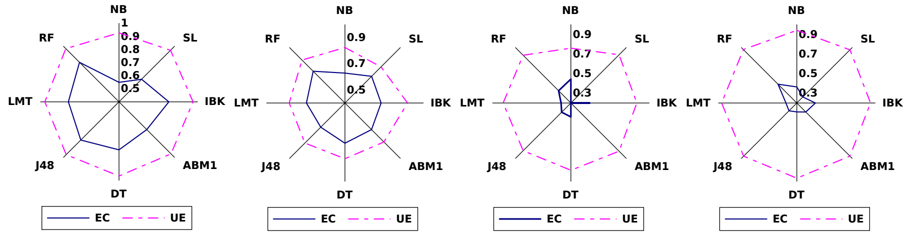
Performance variation between Equivalent (EC) and Unequivalent (UE) pairs. It can be observed from Figure 2 that the F1 scores of EC pairs are significantly less than the corresponding UE pairs. Furthermore, this difference in performance can be observed across all the techniques and for the three PMMC datasets, as well as the additional LR data. Hence, we conclude that the imbalance problem can be generalized as the main cause of the abysmal performance of the supervised learning techniques. We believe that a key reason of this difference in performance is that the unequivalent pairs significantly outnumbered equivalent pairs for all the three datasets. This imbalance in the data limits the learning of each technique and consequently impedes their performance for the equivalent pairs.

Performance variation between techniques. The performance variation between supervised learning techniques is represented by the box plots in Figure 3. It can be observed from the figure that there is a significant variation between the performances of the supervised learning techniques for the equivalent pairs. However, such a variation is not apparent for the unequivalent pairs. These results indicate that the availability of the larger number of unequivalent pairs in the training data provides ample opportunities for *all* the supervised learning techniques to learn and predict the performance of the unequivalent pairs. However, the relatively small number of equivalent pairs in the training data does not provide equal opportunities for all the supervised learning techniques. This is due to two possible reasons, either the available equivalent pairs are not appropriate for an accurate learning, or the available equivalent pairs have contradictions which cannot be resolved due to scarcity of examples. These results confirm our hypothesis that the presence of imbalance in the data impedes the performance of supervised learning techniques.

Performance variation across datasets. It can be observed from Figure 4 that the F1 scores for the unequivalent pairs in all the datasets are comparable. On the contrary, for the equivalent pairs, the F1 scores of AM dataset are significantly less than the other datasets. This indicates that the AM dataset does not have sufficient or appropriate examples to learn and predict the performance of equivalent pairs. It is because the AM dataset contains a large number of process models and activities, whereas the benchmark correspondences merely contain 222 equivalent pairs. Thus, the equivalent pairs of AM dataset do not have sufficient examples to encompass the diversity of their models.

Techniques	UA		BR		AM		LR	
	EC	**UE**	**EC**	**UE**	**EC**	**UE**	**EC**	**UE**
Naive Bayes	0.549	0.927	0.585	0.846	0.444	0.76	0.364	0.942
Simple Logistic	0.644	0.957	0.505	0.854	0	0.896	0.287	0.968
IBK	0.781	0.966	0.693	0.871	0.393	0.874	0.39	0.953
AdaBoostM1	0.7	0.962	0.592	0.861	0	0.896	0.33	0.967
Decision Table	0.767	0.967	0.606	0.867	0.342	0.887	0.291	0.968
J48	0.813	0.971	0.612	0.857	0.333	0.889	0.313	0.968
LMT	0.787	0.967	0.605	0.864	0.3	0.892	0.311	0.968
Random Forest	0.827	0.975	0.694	0.886	0.377	0.89	0.472	0.972

Table 3. F1 scores of supervised learning techniques (EC = Equivalent pairs and UE = Unequivalent pairs)

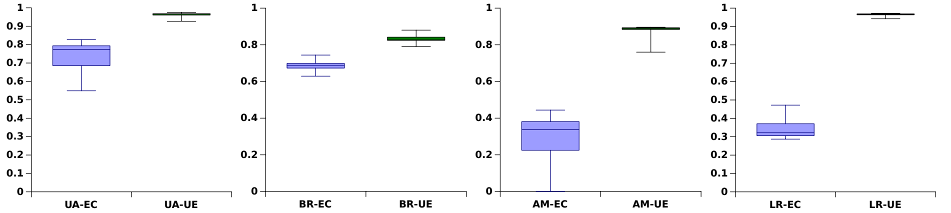


a) UA dataset b) BR dataset c) AM dataset d) LR dataset

Figure 2. Performance variation of F1 scores between equivalent and unequivalent pairs

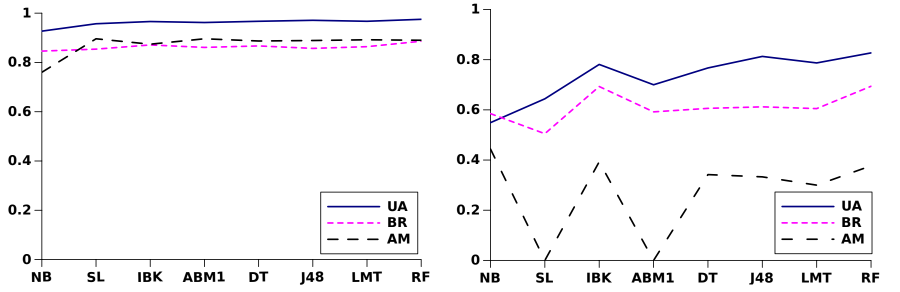
4 THE PROPOSED SOLUTION

In this section, we present our third contribution which is based on the use of data sampling technique to balance the training data and ensure that a supervised learning technique has equal opportunity to elicit the knowledge for equivalent and unequivalent pairs. Listing 1 provides an overview of our proposed solution. It



a) UA dataset b) BR dataset c) AM dataset d) LR dataset

Figure 3. Performance variation of F1 scores between techniques



a) Unequivalent pairs b) Equivalent pairs

Figure 4. Performance variation across datasets of F1 scores between equivalent and unequivalent pairs

includes pseudo-codes for feature extraction, pre-processing of labels, and balancing the training data.

In the listing, Pi is an i^{th} process model, and $L_{Pi,Pj}[\,][\,]$ represents the array of all the activity labels in the two process models, Pi and Pj. Furthermore, $AP_{L_{Pi,Pj}}[\,]$ represents the array of all the activity pairs between the two process models Pi and Pj. Finally, SLT represents a supervised learning technique. In contrast to the existing studies that use term weights as features, we are the first to use a set of similarity scores between activity labels as features. Furthermore, we have also introduced the idea of using sampling techniques to balance the training data in the context of process model matching.

```
float ourAlgo(AP_L{Pi,Pj}[ ]) { // get benchmark decision for each activity
    pair.
   bool bmcDecisions[ ] = getBMC(AP_L{Pi,Pj}[ ]) //10-fold cross-validation
   for (r = 1 to 10) {
      //divide data in two parts, training (TR) and testing (TS) activity
         pairs.
      AP_L{Pi,Pj}[TR][TS] = dataDivider(AP_L{Pi,Pj}[ ], bmcDecisions[ ])
      //check the imbalance ratio a constant.
      if (getImbalanceRatio(bmcDecisions[ ]) < α) { // balance training
         data
         AP_L{Pi,Pj}[TR] = dataBalTech(AP_L{Pi,Pj}[TR])
      }
      //TRAINING
      L_Pi,Pj[ ][ ] = preProcessing(AP_L{Pi,Pj}[TR])
      sim[ ][ ] = feature-extraction(L_Pi,Pj[ ][ ])
      SLT-Trained = tain&learn(STL, sim[ ][ ])
      //TESTING
      L_Pi,Pj[ ][ ] = preProcessing(AP_L{Pi,Pj}[TS])
      sim[ ][ ] = feature-extraction(L_Pi,Pj[ ][ ]) // Returns SLT
```

```
        decisions for each pair
     SLTDecisions[ ][ ] =applySLT(SLT-Trained, sim[ ][ ])
          F1Score[r] = computeAccuracy(SLTDecisions[ ][ ], bmcDecisions[
             ])
     }
   avgF1Score = computeAverage (F1Score[ ])
   return avgF1Score
}
/* * Omits trival variations in labels */
L_{Pi,Pj}[ ][ ] preprocessing(AP_{L_{Pi,Pj}}[ ] ) {
   for (k=1 to length of AP_{L_{Pi,Pj}}[ ]) {
      AP_{L_{Pi,Pj}}[k] = tokenize(AP_{L_{Pi,Pj}}[k])
      AP_{L_{Pi,Pj}}[k] = removeStopWords(AP_{L_{Pi,Pj}}[k]) //coversion to stem
         words
      AP_{L_{Pi,Pj}}[k] = stemming(AP_{L_{Pi,Pj}}[k])
   }
   return L_{Pi,Pj}[ ][ ]
}
/* * Returns similarity scores of each pair */
sim[ ][ ] feature-extraction(L_{Pi,Pj}[ ][ ]) {
   for (each metric m) {
      for (k=1 to length of L_{Pi,Pj}[ ][ ]) {
         sim[k][m]= computeSimilarity(L_{Pi,Pj}[ ][ ], m)
      }
   }
   return sim[ ][ ]
}
```

Listing 1. Our approach to improve performance of supervised learning techniques

4.1 Data Balancing Techniques

A brief overview of the data balancing techniques are as follows:

Distribution-based balancing. In the distribution-based balancing different probability distributions are learned from imbalanced dataset to form a balanced dataset [15]. We have used Gaussian distribution for balancing the training data using Box and Muller method, as it is widely acknowledged as an sampling technique for adequately approximating the models. In our experiments, it models the univariate relation of class labels with the features.

Spread subsample. It uses random subsamples method to select the *spread* between minority and majority classes [16]. In this method, one can use uniform distribution of samples so that the number of majority class samples are reduced to minority class samples to balance the distribution of imbalanced dataset.

Synthetic minority oversampling (SMOTE). SMOTE is a pseudo oversampling technique in which minority class instances are increased by generating new pseudo instances and thus decrease the spread between minority and majority classes [17]. In particular, firstly, the minority class instances are identified. Subsequently, the neighbors of the minority class instances are selected and new minority class instances are added based on the selected neighbors.

Class balancer. In the class balancer technique, the instances in the dataset are reweighed and as a result each class contains the same total weight [18]. The aggregated weight of all instances is kept constant. The weights of only first batch of instances are altered so that it can be employed with the Filtered Classifier.

4.2 Experimental Setup

For the experiments, we have used the same four datasets that were used in the experiments presented in Section 3.4. Furthermore, the same evaluation measures, preprocessing, similarity scores, features, training-testing ratio, 10-fold cross-validation and the same supervised learning algorithms have been used. However, for conducting the experiments, we have employed four different data balancing techniques to identify a training dataset, whereas the testing dataset remained unchanged.

4.3 Results

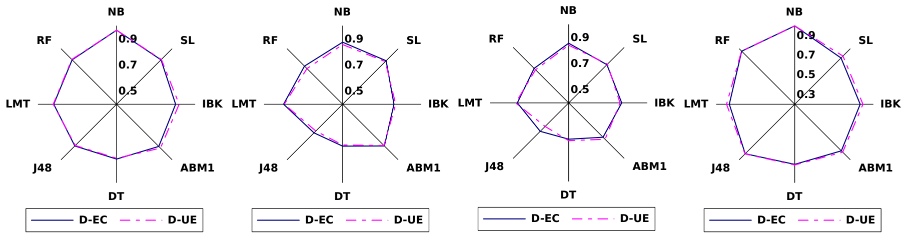
Table 4 shows the average F1 score of 10-fold cross-validation for each supervised learning technique. Similar to the results of the previous experiments, the results of equivalent and unequivalent pairs are separated to highlight the differences between their scores. Note, we generated P, R and F1 scores separately for 12 data subsets, which are produced by applying 4 data balancing techniques on UA, BR and AM datasets. However, for brevity only the F1 scores are presented in the Table 4. From the results, we have observed the following:

Reduced performance variation between equivalent and unequivalent pairs. It can be observed from Figure 5, that after data balancing the F1 scores for equivalent pairs become comparable with the unequivalent pairs. Additionally, from the comparison of Figure 2 and 5, it can be observed that the variation between the F1 scores of equivalent and unequivalent pairs is significantly reduced. This indicates that all the balancing techniques choose appropriate and sufficient examples for equivalent as well as unequivalent pairs. It further indicates that the chosen examples are effective for learning and prediction of equivalent and unequivalent pairs.

Reduced performance variation between techniques. From the comparison of Figure 3 and 6, it can be observed that the sizes of box plot quartiles for

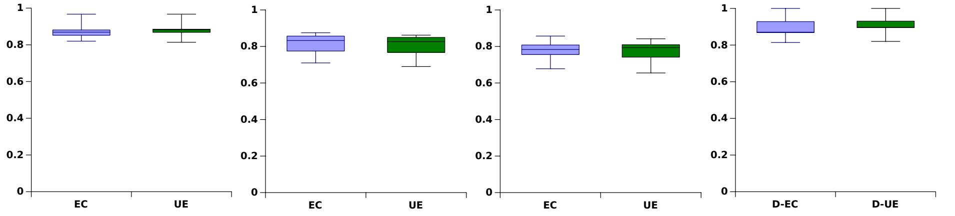
Techniques	**Distribution Base**		**Undersampling**		**SMOTE**		**Class Balancer**	
	EC	**UE**	**EC**	**UE**	**EC**	**UE**	**EC**	**UE**
UA Dataset								
Naive Bayes	0.967	0.967	0.682	0.766	0.689	0.733	0.686	0.771
Simple Logistic	0.881	0.885	0.861	0.861	0.867	0.831	0.833	0.838
IBK	0.852	0.879	0.908	0.909	0.977	0.97	0.903	0.911
AdaBoostM1	0.857	0.875	0.817	0.853	0.841	0.763	0.82	0.857
Decision Table	0.82	0.814	0.841	0.873	0.938	0.922	0.856	0.866
J48	0.852	0.847	0.909	0.908	0.965	0.956	0.912	0.916
LMT	0.881	0.885	0.9	0.897	0.97	0.962	0.902	0.91
Random Forest	0.881	0.885	0.923	0.928	0.98	0.975	0.916	0.922
BR Dataset								
Naive Bayes	0.875	0.857	0.667	0.752	0.653	0.78	0.662	0.75
Simple Logistic	0.871	0.862	0.631	0.718	0.615	0.765	0.64	0.722
IBK	0.793	0.806	0.821	0.812	0.872	0.887	0.799	0.818
AdaBoostM1	0.852	0.847	0.65	0.741	0.646	0.785	0.664	0.757
Decision Table	0.721	0.712	0.677	0.759	0.673	0.785	0.682	0.762
J48	0.71	0.69	0.679	0.745	0.79	0.823	0.779	0.781
LMT	0.852	0.847	0.687	0.695	0.789	0.827	0.767	0.777
Random Forest	0.813	0.786	0.817	0.816	0.849	0.878	0.786	0.813
AM Dataset								
Naive Bayes	0.857	0.842	0.697	0.68	0.673	0.646	0.687	0.639
Simple Logistic	0.814	0.82	0.709	0.687	0.699	0.678	0.689	0.658
IBK	0.806	0.793	0.679	0.658	0.846	0.847	0.675	0.723
AdaBoostM1	0.772	0.794	0.719	0.69	0.717	0.691	0.727	0.689
Decision Table	0.678	0.689	0.73	0.683	0.748	0.72	0.741	0.689
J48	0.708	0.655	0.768	0.709	0.77	0.708	0.743	0.676
LMT	0.793	0.806	0.746	0.7	0.784	0.749	0.736	0.684
Random Forest	0.774	0.759	0.664	0.686	0.837	0.851	0.717	0.748
LR Dataset								
Naive Bayes	1	1	0.587	0.744	0.564	0.812	0.59	0.743
Simple Logistic	0.868	0.896	0.638	0.729	0.683	0.921	0.64	0.733
IBK	0.868	0.896	0.655	0.652	0.821	0.932	0.539	0.743
AdaBoostM1	0.873	0.892	0.595	0.74	0.565	0.813	0.572	0.740
Decision Table	0.814	0.82	0.64	0.732	0.654	0.882	0.629	0.73
J48	0.915	0.918	0.633	0.732	0.898	0.94	0.612	0.743
LMT	0.868	0.896	0.638	0.73	0.73	0.812	0.612	0.752
Random Forest	0.966	0.968	0.69	0.735	0.843	0.883	0.673	0.732

Table 4. Result of supervised learning techniques using balanced datasets



a) UA dataset b) BR dataset c) AM dataset d) LR dataset

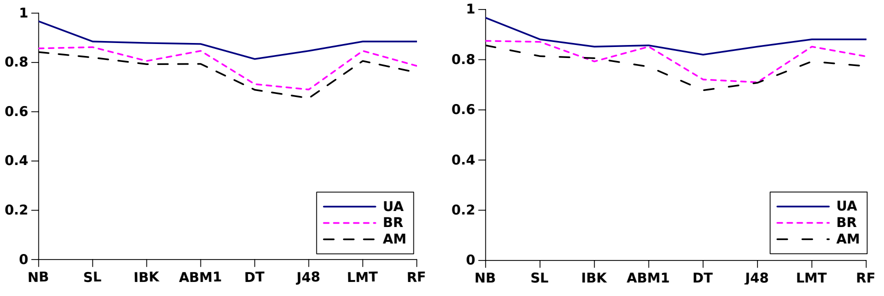
Figure 5. Performance variation of F1 scores between equivalent and unequivalent pairs after data balancing



a) UA dataset b) BR dataset c) AM dataset d) LR dataset

Figure 6. Performance variation of F1 scores between techniques after data balancing

equivalent pairs are decreased for UA and AM datasets, representing that the variation between the performance of supervised learning techniques is reduced for these datasets. It implies that the data balancing has provided equal opportunity for all the supervised learning techniques for UA and AM datasets. On the contrary, the size of the quartiles for BR dataset has increased slightly



a) Unequivalent pairs b) Equivalent pairs

Figure 7. Performance variation across datasets of equivalent and unequivalent pairs after applying data balancing

after data balancing, indicating that the variation between performance of the supervised learning techniques has increased.

Reduced performance variation across datasets. It can be observed from Figure 7 that for the equivalent as well as the unequivalent pairs there is no significant gap between the performances of the supervised learning techniques across the three datasets. This indicates that data balancing has provided better opportunities to all the techniques across the three datasets.

5 ANALYSIS OF THE RESULTS

To evaluate the significance of performance gain that was achieved due to the proposed approach, we have applied Friedman ANOVA test between balanced and imbalanced datasets. The test is applied to the F1 scores of the 10-folds by setting significance level to 0.05. The results of Distribution based balancing technique are presented in Table 5. From the results presented in Table 5 we have observed the following:

Significant performance gain: The increase in the performance of the supervised learning techniques is statistically significant for the equivalent pairs. This observation is valid for all the supervised learning techniques and across the three datasets.

Insignificant performance reduction for unequivalent pairs: For unequivalent pairs, the reduction in the performance is statistically insignificant in majority of the cases. Furthermore, in a few cases the performance even improved significantly.

Based on the above observations, we conclude that the use of data balancing in supervised learning techniques enhances the efficiency of process model matching, whenever the imbalance is large. However, the balance techniques may not be equally effective when the imbalance ratio is small.

6 RELATED WORK

Process model matching was initially considered a rudimentary problem for computing similarity between process models, querying process model repositories, harmonization of process models, detection of process clones [32, 34], etc. However, recent studies [2, 3, 4] have recognized the importance of process model matching beyond its traditional usage. Consequently, a plethora of process model matching techniques have been developed. We have conducted a comprehensive survey of process model matching techniques by employing a snowballing approach. More specifically, we started with a process model matching survey [5] and performed forward and backward tracing to identify the studies that focus on identifying corresponding activities between a pair of process models. Accordingly, we identified

Technique	UA Dataset		BR Dataset		AM Dataset	
	EC	**UE**	**EC**	**UE**	**EC**	**UE**
Naive Bayes	S+	S+	S+	S+	S+	S+
Simple Logistic	S+	NS	S+	NS	S+	NS
IBK	S+	NS	S+	NS	S+	NS
AdaBoostM1	S+	S-	S+	S-	S+	S-
Decision Table	NS	NS	S+	NS	S+	NS
J48	S+	S-	S+	S-	S+	S-
LMT	S+	NS	S+	NS	S+	NS
Random Forest	S+	S-	S+	S-	S+	S-

Table 5. Friedman-ANOVA test between imbalanced and distribution based balanced datasets. S represents significant, NS insignificant, where + and − represent increase and decrease in performance, respectively.

domains specific, as well as generic studies. However, for brevity, we have only discussed generic studies. Our comprehensive examination of techniques revealed that the matching techniques can be subdivided into two broad categories, supervised and unsupervised techniques.

Unsupervised Techniques. These techniques use text matching measures to identify corresponding activities between a pair of process models. Typically, these techniques are composed of two phases [9]. The first phase computes similarity score between activities of process models, whereas the second phase converts the similarity score into a binary decision of corresponding activities or not [5]. These techniques are further divided into syntactic and semantic measures [5]. A brief overview of these techniques are as follows:

Syntactic measures: The measures in this category merely rely on the surface form of the words that constitute the labels of the participating activities. That is, these measures compute the similarity between a given pair of activities by tokenizing labels into words, and subsequently comparing the words by using string comparison operations [35]. Typically, distance-based measures, such as Edit-distance, Levenshtein distance, Hamming distance, and Jensen-Shannon distance, have been used for the comparison. These measures compute the similarity between two words by counting the minimum number of string edit operations (insertion, deletion, and update) required to convert one word into the other word. A lower value of edit distance represents higher similarity between the labels and vice versa.

In addition to the distance-based measures, Dice similarity and Cosine similarity are also used for syntactic comparison. Dice similarity is the ratio between the number of shared words between two activities and the total number of words in the two activities [36]. On the contrary, Cosine similarity transforms each word or a label into a vector and computes the similarity as cosine of angle between the activity vectors [19]. Furthermore, other syntactic similarity measures, such as Jaccard similarity and Longest Common Subsequence, have been used for process matching.

The syntactic measures yield high accuracy in case the vocabulary of participating activity labels is comparable, however there are at least two cases in which these measures do not yield high accuracy:

1. the considered activities are composed of unrelated words having similar spellings, such as "give contract" and "live contact",
2. the activities are composed of words with same meaning but different vocabulary, such as "evaluate applicant" and "assess candidate".

Semantic measures. These measures address the limitations of the syntactic measures as they take into account the meanings or relatedness of the considered words. To that end, this technique relies on a large lexical English database, called WordNet [37]. Specifically, the participating labels are tokenized into words and pre-processed to generate root form of each word. Subsequently, the similarity between words is computed by using the *similarity* or *relatedness* of words.

The similarity based measures rely on the similarity between two words by considering their synonyms. The examples of the similarity based techniques are Static weighted word comparison [38], Dice with synonyms, and intersection of synonyms. These techniques yield higher matching accuracy between the labels in which a concept is represented by different words.

In contrast to the similarity based measures, a relatedness based measure takes into consideration the co-relatedness of words, represented by `is-a` relationship between the concepts in the WordNet topology. That is, the words having shorter path between them are considered more similar that the ones having longer path. Lin, Lesk, Wu & Palmer, Leacock, and Jiang similarity are the relatedness based semantic measures [39].

Supervised Techniques. Recent studies have attempted to increase the accuracy of process model matching using supervised learning techniques [9, 3, 40]. The F1 scores achieved by the supervised learning technique for the three PMMC '15 datasets are 0.54, 0.38 and 0.61, which are slightly higher than the F1 scores achieved by the matchers participating in the latest edition of the PMM contest in 2015. Furthermore, a state-of-the-art approach has proposed to combine the strengths of individual matchers using an ensemble of multiple matchers [3] to improve the accuracy of PMM. The most recent study [40] has proposed a word-embeddings based approach to increase the accuracy of process matching to achieve an F1 score of 0.84, 0.72 and 0.91. A key limitation of the proposed approach is that its analysis is limited to a unified F1 score, without distinguishing between equivalent and unequivalent pairs.

7 CONCLUSION

Due to the growing interest in process model matching, a plethora of unsupervised learning techniques have been developed. Recent attempts have been made to introduce supervised learning for process model matching, without achieving a significantly higher accuracy than the traditional unsupervised techniques. This is an anomaly, because the supervised learning techniques are proven to significantly outperform unsupervised techniques for a variety of text processing tasks. Therefore, the aim of this paper is to investigate the cause behind the abysmal performance of the supervised learning techniques in process model matching and suggest a solution to improve their performance.

To this end, in this paper, we have made three main contributions. Firstly, we have transformed the existing benchmark correspondences into a readily usable form for the supervised learning techniques. Secondly, we have conducted a series of experiments using eight state-of-the-art supervised learning techniques and synthesized the results to establish that the presence of imbalance in the datasets adversely affects the matching results. Thirdly, we applied four different data balancing techniques to achieve groundbreaking accuracy in the process model matching. That is, our proposed solution achieved a maximum F1 score of 0.98, whereas the plethora of existing techniques for process model matching (including both supervised and unsupervised techniques) were able to achieve a maximum F1 score of merely 0.67. Furthermore, even the average F1 score of 0.79 achieved by our solution is higher than the maximum F1 score of 0.67 achieved by all the existing techniques. In the future, we plan to use the state-of-the-art deep learning techniques for process model matching.

REFERENCES

[1] Kuss, E.—Leopold, H.—van der Aa, H.—Stuckenschmidt, H.—Reijers, H. A.: Probabilistic Evaluation of Process Model Matching Techniques. In: Comyn-Wattiau, I., Tanaka, K., Song, I. Y., Yamamoto, S., Saeki, M. (Eds.): Conceptual Modeling (ER 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9974, 2016, pp. 279–292, doi: 10.1007/978-3-319-46397-1_22.

[2] Antunes, G.—Bakhshandeh, M.—Borbinha, J.—Cardoso, J.—Dadashnia, S. et al.: The Process Model Matching Contest 2015. In: Kolb, J. et al. (Eds.): Enterprise Modelling and Information Systems Architectures. Gesellschaft für Informatik, Bonn, Lecture Notes in Informatics, Vol. 248, 2015, pp. 127–155.

[3] Meilicke, C.—Leopold, H.—Kuss, E.—Stuckenschmidt, H.—Reijers, H. A.: Overcoming Individual Process Model Matcher Weaknesses Using Ensemble Matching. Decision Support Systems, Vol. 100, 2017, pp. 15–26, doi: 10.1016/j.dss.2017.02.013.

[4] Kuss, E.—Leopold, H.—Meilicke, C.—Stuckenschmidt, H.: Ranking-Based Evaluation of Process Model Matching. In: Panetto, H. et al. (Eds.): On the Move to

Meaningful Internet Systems (OTM 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10573, 2017, pp. 298–305, doi: 10.1007/978-3-319-69462-7_19.

[5] Jabeen, F.—Leopold, H.—Reijers, H. A.: How to Make Process Model Matching Work Better? An Analysis of Current Similarity Measures. In: Abramowicz, W. (Ed.): Business Information Systems (BIS 2017). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 288, 2017, pp. 181–193, doi: 10.1007/978-3-319-59336-4_13.

[6] Malik, M. K.: Urdu Named Entity Recognition and Classification System Using Artificial Neural Network. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Vol. 17, 2017, No. 1, Art. No. 2, 13 pp., doi: 10.1145/3129290.

[7] Yahaya, M. F.—Rahman, N. A.—Bakar, Z. A.—Hasmy, H.: Evaluation on Knowledge Extraction and Machine Learning in Resolving Malay Word Ambiguity. Journal of Fundamental and Applied Sciences, Vol. 9, 2017, No. 5S, pp. 115–130, doi: 10.4314/jfas.v9i5s.10.

[8] Pang, B.—Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. Vol. 2, 2008, No. 1–2, pp. 1–135, doi: 10.1561/1500000011.

[9] Sonntag, A.—Hake, P.—Fettke, P.—Loos, P.: An Approach for Semantic Business Process Model Matching Using Supervised Machine Learning. 24th European Conference on Information Systems (ECIS), Istanbul, Turkey, 2016, Research-in-Progress Papers, Art. No. 47.

[10] Cardoso, J.—Bakhshandeh, M.—Faria, D.—Pesquita, C.—Borbinha, J.: Ontology-Based Approach for Heterogeneity Analysis of EA Models. In: Dumas, M., Fantinato, M. (Eds.): Business Process Management Workshops (BPM 2016). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 281, 2016, pp. 131–142, doi: 10.1007/978-3-319-58457-7_10.

[11] Klinkmüller, C.—Weber, I.—Mendling, J.—Leopold, H.—Ludwig, A.: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In: Daniel, F., Wang, J., Weber, B. (Eds.): Business Process Management. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8094, 2013, pp. 211–218, doi: 10.1007/978-3-642-40176-3_17.

[12] Allweyer, T.: BPMN 2.0: Introduction to the Standard for Business Process Modeling. BoD – Books on Demand, 2016.

[13] del Pilar Angeles, M.—Espino-Gamez, A.: Comparison of Methods Hamming Distance, Jaro, and Monge-Elkan. The Seventh International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2015), Rome, Italy, 2015, pp. 63–69.

[14] Büttcher, S.—Clarke, C. L. A.—Cormack, G. V.: Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press, 2016.

[15] Bermejo, P.—Gámez, J. A.—Puerta, J. M.: Improving the Performance of Naive Bayes Multinomial in E-Mail Foldering by Introducing Distribution-Based Balance of Datasets. Expert Systems with Applications, Vol. 38, 2011, No. 3, pp. 2072–2080, doi: 10.1016/j.eswa.2010.07.146.

[16] Tan, M.—Tan, L.—Dara, S.—Mayeux, C.: Online Defect Prediction for Imbalanced Data. Proceedings of the 37th International Conference on Software Engineering, Vol. 2, IEEE Press, 2015, pp. 99–108, doi: 10.1109/ICSE.2015.139.

[17] Chawla, N.—Bowyer, K. W.—Hall, L. O.—Kegelmeyer, W. P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research, Vol. 16, 2002, pp. 321–357, doi: 10.1613/jair.953.

[18] Alinejad-Rokny, H.—Sadroddiny, E.—Scaria, V.: Machine Learning and Data Mining Techniques for Medical Complex Data Analysis (Editorial). Neurocomputing, Vol. 276, 2017, p. 1, doi: 10.1016/j.neucom.2017.09.027.

[19] Weidlich, M.—Dijkman, R.—Mendling, J.: The ICoP Framework: Identification of Correspondences Between Process Models. In: Pernici, B. (Ed.): Advanced Information Systems Engineering (CAiSE 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6051, 2010, pp. 483–498, doi: 10.1007/978-3-642-13094-6_37.

[20] Leopold, H.—Niepert, M.—Weidlich, M.—Mendling, J.—Dijkman, R.—Stuckenschmidt, H.: Probabilistic Optimization of Semantic Process Model Matching. In: Barros, A., Gal, A., Kindler, E. (Eds.): Business Process Management (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7481, 2012, pp. 319–334, doi: 10.1007/978-3-642-32885-5_25.

[21] Cayoglu, U.—Dijkman, R.—Dumas, M.—Fettke, P.—Garcia-Banuelos, L. et al.: Report: The Process Model Matching Contest 2013. In: Lohmann, N., Song, M., Wohed, P. (Eds.): Business Process Management Workshops (BPM 2013). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 171, 2013, pp. 442–463, doi: 10.1007/978-3-319-06257-0_35.

[22] Klinkmüller, C.—Leopold, H.—Weber, I.—Mendling, J.—Ludwig, A.: Listen to Me: Improving Process Model Matching Through User Feedback. In: Sadiq, S., Soffer, P., Völzer, H. (Eds.): Business Process Management (BPM 2014). Springer, Cham, Lecture Notes in Computer Science, Vol. 8659, 2014, pp. 84–100, doi: 10.1007/978-3-319-10172-9_6.

[23] Niesen, T.—Dadashnia, S.—Fettke, P.—Loos, P.: A Vector Space Approach to Process Model Matching Using Insights from Natural Language Processing. Multikonferenz Wirtschaftsinformatik (MKWI), Illmenau, Germany, 2016, pp. 93–104.

[24] Schoknecht, A.—Fischer, N.—Oberweis, A.: Process Model Search Using Latent Semantic Analysis. In: Dumas, M., Fantinato, M. (Eds.): Business Process Management Workshops (BPM 2016). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 281, 2016, pp. 283–295, doi: 10.1007/978-3-319-58457-7_21.

[25] Leopold, H.—van der Aa, H.—Pittke, F.—Raffel, M.—Mendling, J.—Reijers, H. A.: Searching Textual and Model-Based Process Descriptions Based on a Unified Data Format. Software and Systems Modeling, Vol. 18, 2019, pp. 1179–1194, doi: 10.1007/s10270-017-0649-y.

[26] Klinkmüller, C.: Adaptive Process Model Matching: Improving the Effectiveness of Label-Based Matching Through Automated Configuration and Expert Feedback. Ph.D. Dissertation, Macquarie University, Sydney, Australia, 2017.

[27] Kuss, E.—Stuckenschmidt, H.: Automatic Classification to Matching Patterns for Process Model Matching Evaluation. In: Cabanillas, C., España, S., Farshidi, S. (Eds.): Proceedings of the ER Forum 2017 and the ER 2017 Demo Track. 36th International Conference on Conceptual Modeling (ER 2017), Valencia, Spain, 2017. CEUR Workshop Proceedings, Vol. 1979, 2017, pp. 306–319.

[28] Ali, M.—Shahzad, K.: Enhanced Benchmark Datasets for a Comprehensive Evaluation of Process Model Matching Techniques. In: Pergl, R., Babkin, E., Lock, R., Malyzhenkov, P., Merunka, V. (Eds.): Enterprise and Organizational Modeling and Simulation (EOMAS 2018). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 332, 2018, pp. 107–122, doi: 10.1007/978-3-030-00787-4_8.

[29] Kuss, E.—Leopold, H.—van der Aa, H.—Stuckenschmidt, H.—Reijers, H. A.: A Probabilistic Evaluation Procedure for Process Model Matching Techniques. Data and Knowledge Engineering, Vol. 117, 2018, pp. 393–406, doi: 10.1016/j.datak.2018.04.008.

[30] Rana, M.—Shahzad, K.—Nawab, R. M. A.—Leopold, H.—Babar, U.: A Textual Description Based Approach to Process Matching. In: Horkoff, J., Jeusfeld, M., Persson, A. (Eds.): The Practice of Enterprise Modeling (PoEM 2016). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 267, 2016, pp. 194–208, doi: 10.1007/978-3-319-48393-1_14.

[31] Shahzad, K.—Nawab, R. M. A.—Abid, A.—Sharif, K.—Ali, F.—Aslam, F.—Mazhar, A.: A Process Model Collection and Gold Standard Correspondences for Process Model Matching. IEEE Access, Vol. 7, 2019, pp. 30708–30723, doi: 10.1109/ACCESS.2019.2900174.

[32] Becker, M.—Laue, R.: A Comparative Survey of Business Process Similarity Measures. Computers in Industry, Vol. 63, 2012, No. 2, pp. 148–167, doi: 10.1016/j.compind.2011.11.003.

[33] Phan, N. H.—Hoang, V. D. T.—Shin, H.: Adaptive Combination of Tag and Link-Based User Similarity in Flickr. Proceedings of the 18th ACM International Conference on Multimedia (MM'10), ACM, 2010, pp. 675–678, doi: 10.1145/1873951.1874049.

[34] Ekanayake, C. C.—Dumas, M.—García-Bañuelos, L.—La Rosa, M.—ter Hofstede, A. H.: Approximate Clone Detection in Repositories of Business Process Models. In: Barros, A., Gal, A., Kindler, E. (Eds.): Business Process Management (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7481, 2012, pp. 302–318, doi: 10.1007/978-3-642-32885-5_24.

[35] Dijkman, R.—Dumas, M.—García-Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H. A. (Eds.): Business Process Management (BPM 2009). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5701, 2009, pp. 48–63, doi: 10.1007/978-3-642-03848-8_5.

[36] Jin, T.—Wang, J.—La Rosa, M.—ter Hofstede, A.—Wen, L.: Efficient Querying of Large Process Model Repositories. Computers in Industry, Vol. 64, 2013, No. 1, pp. 41–49, doi: 10.1016/j.compind.2012.09.008.

[37] Miller, G. A.: WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41, doi: 10.1145/219717.219748.

[38] Dijkman, R.—Dumas, M.—van Dongen, B.—Käärik, R.—Mendling, J.: Similarity of Business Process Models: Metrics and Evaluation. Information Systems, Vol. 36, 2011, No. 2, pp. 498–516, doi: 10.1016/j.is.2010.09.006.

[39] Shahzad, K.—Pervaz, I.—Nawab, A.: WordNet-Based Semantic Similarity Measures for Process Model Matching. In: Zdravkovic, J., Grabis, J., Nurcan, S., Stirna, J. (Eds.): Joint Proceedings of the BIR 2018 Short Papers, Workshops and Doctoral Consortium, 17$^{\text{th}}$ International Conference Perspectives in Business Informatics Research (BIR 2018), Stockholm, Sweden, 2018. CEUR Workshops Proceedings, Vol. 2218, 2018, pp. 33–44.

[40] Shahzad, K.—Kanwal, S.—Malik, K.—Aslam, F.—Ali, M.: A Word-Embedding-Based Approach for Accurate Identification of Corresponding Activities. Computers and Electrical Engineering, Vol. 78, 2019, No. 1, pp. 218–229, doi: 10.1016/j.compeleceng.2019.07.011.

Khurram Shahzad is Tenure Track Assistant Professor at the Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore. He obtained his Masters and Ph.D. from KTH – Royal Institute of Technology, Stockholm. He has been associated with Information Systems Groups at Technical University Eindhoven, Eindhoven, and University of Fribourg, Fribourg. He has published more than 35 papers in international conferences and journals.

Arslaan Mazhar received his Bachelor's degree in software engineering from the Mirpur University of Science and Technology, Mirpur, Pakistan, and his M.Phil. degree in computer science from Punjab University College of Information Technology, University of the Punjab, Lahore.

Ghulam Mustafa received his M.Sc. degree in computer science from COMSATS Institute of Information Technology (CIIT), Lahore, Pakistan, in 2007 and his Ph.D. degree in computer science and technology from Beijing Institute of Technology (BIT), Beijing, China in 2015. He is currently Assistant Professor at the Department of Information Technology, The University of the Punjab, Gujranwala Campus, Pakistan. His research interests include artificial intelligence, recommender systems and machine learning.

Faisal Aslam received his Ph.D. from University of Freiburg, Germany and Post Doctorate from TU Delft, The Netherlands. He was also Research Fellow at Lund University, Sweden. Currently, he is Assistant Professor at the University of the Punjab. He has published in reputed journals and conferences.