

The User Query Based Learning System for Lifetime Prediction of Metallic Components

Esther Ge¹, Richi Nayak¹, Yue Xu² and Yuefeng Li²

¹ School of Information Systems, ² School of Software Engineering and Data Communications
Queensland University of Technology,
Brisbane, Australia
{e.ge, r.nayak, yue.xu, y2.li}@qut.edu.au

Abstract. Real-World Data Mining Applications generally do not end up with the creation of the models. The use of the model is the final purpose especially in prediction tasks. The problem arises when the model is built based on much more information than that the user can provide in using the model. As a result, the performance of model reduces drastically due to many missing attributes values. This paper develops a new learning system framework, called as User Query Based Learning System (UQBLs), for building data mining models best suitable for users use. We demonstrate its deployment in a real-world application of the lifetime prediction of metallic components in buildings.

Keywords: Data Mining, Learning System, predictive model, lifetime prediction, civil engineering

1 Introduction

Since its inception, Data Mining (DM) has been driven by the need to solve practical problems [1]. It is effectively applied to many areas like business, marketing, medical, financial and so on. Civil engineering is one of the areas where a variety of successful real-world data mining applications are reported. For example, Kessler et al. [2] improved prediction of the corrosion behavior of car body steel using a Kohonen self organizing map. Furuta et al. [3] developed a practical decision support system for the structural damage assessment due to corrosion using Neural Network. More recently, Morcous et al. [4] proposed a case-based reasoning system for modelling infrastructure deterioration. Melhem and Cheng [5] first used KNN and Decision Tree for estimating the remaining service life of bridge decks. And later Melhem et al. [6] investigated the use of wrapper methods to improve the prediction accuracy of the decision tree algorithm for the application of bridge decks.

Most of the papers focus on how to build the predictive models and how to improve the prediction accuracy. There are many DM methods such as Naïve Bayes, K-Nearest Neighbors (KNN), Decision Tree (DT), Neural Network (NN), Support Vector Machine (SVM) etc that can be considered to do prediction. Methods such as bagging and boosting can be used [7, 8] to obtain higher accuracy. None of the

existing work, however, deals with how to use these predictive models in real situations when users only have knowledge of limited inputs. The real-world data mining applications generally do not end up with generating the predictive models on the data. The use of the model is the final purpose especially in prediction tasks. When the model is built based on much more information than that user can provide in using the model, the prediction accuracy of model usually decreases drastically in using it [9]. For example, a civil engineering dataset usually contains many professional variables like salt deposition, material mass loss and corrosion rate etc. These variables are very useful in building predictive models. However, users can not provide such kind of information when using the models for predicting the outcome. Therefore, new data (user query) may contain many missing values. If all those professional variables are included in the final model, the predicted result for new data (user query) is much worse than that for test data.

This paper deals with how to make a correct feature selection for building a model suitable for users use and how to get a balance between relative minimums features and relative maximums accuracy. We develop a new learning system framework, called as User Query Based Learning System (UQBLs), for building data mining models best suitable for users use. We demonstrate its deployment in a real-world problem of the lifetime prediction of metallic components in buildings.

2 User Query Based Learning System

We present the User Query Based Learning System (UQBLs) in Figure 1. It includes all phases as explained necessary in the industry standard data mining process model CRISP-DM (Cross Industry Standard Process for Data Mining) [10]. We highlight three procedures that are different from the CRISP-DM. They are critical for the success of our UQBLs model. The feature selection based on a user query is separated from data preprocessing. An external domain knowledge base is involved in data preprocessing and results post-processing phases.

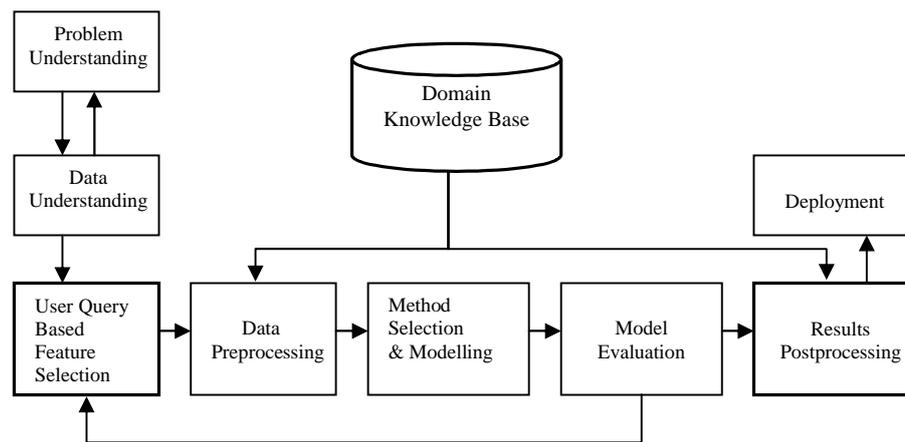


Fig. 1. User Query Based Learning System

2.1 User Query Based Feature Selection

Feature selection is separated from the data preprocessing phase. The usual feature selection in a DM process is for removing redundant and irrelevant attributes. In real-world data mining, it is generally not enough, because the creation of the model is not the end of the Real-World Data Mining Application. When users are using model, if they can not provide all attributes' values as inputs, the performance of model will decrease a lot. In order to solve this problem, the User Query Based Feature Selection includes an extra iterative step for selecting those attributes which can be provided while user is using model. As shown in Figure 1, from feature selection to model evaluation, it is an iterative procedure. The loop continues until a stopping criterion is satisfied [11]. Here we give a detailed description.

Let $A = \{a_1, a_2, \dots, a_k, a_{k+1}, \dots, a_m, a_{m+1}, \dots, a_n\}$ be a set of attributes relevant to target value in a data set. This set of attributes is obtained after pre-processing (removal of redundant and irrelevant features). We divide them into three groups:

- Group 1 ($a_1 - a_k$): Attributes that the user can provide while using the model
- Group 2 ($a_{k+1} - a_m$): Attributes that the user can not provide but can be obtained from the external domain knowledge
- Group 3 ($a_{m+1} - a_n$): Attributes that can not be provided by user or domain knowledge

Group 1 will be included in the final model because they are not only useful in mining but the user can also provide such values while using the model. Group 3 will be rejected because they can not be provided in model use although they have mining value. If we include the attributes of Group 3 in the final model, their values in new data will be missing. As a result, the generalization accuracy will decrease. A decision has to be made for attributes in Group 2, as they can not be provided by user but they can be obtained from external domain knowledge (We will discuss this more in section 2.2). If we include all those attributes in Group 2, the measurements to get some of these values may be too complex and computationally expensive. If we exclude those attributes, the performance of model may not be accepted by user.

Here is our algorithm for selecting features in Group 2:

For $i = m$ to $k+1$

Construct a set of attributes $A_i = \{a_1, \dots, a_i\}$

Build a model using A_i as input attributes

Evaluate the model

If the stopping criterion is reached, select subset A_i as input attributes

Next

The algorithm uses the attributes in both Group 1 and Group 2 to build the model and progressively removes attributes from group 2. The first model usually will have highest prediction accuracy due to inclusion of all attributes. The accuracy of following models after progressively removal of attributes gets worse. The loop is terminated when an acceptable model is built with the minimum set of attributes. The stopping criterion is a threshold according to performance measures. For example, if the Correlation Coefficient (CC) is used to measure the performance, the stopping criterion is that CC is equal to 0.80. When the CC is less than 0.80, we will stop the loop. The stopping criterion can also be decided by consulting with the user since the performance of model should satisfy user. Moreover, several attributes can be

removed in one step in real-world implementation instead of progressively removal as in above algorithm. We demonstrate the efficacy of this algorithm in the lifetime prediction of metallic components.

2.2 Domain Knowledge for Preprocessing and Post-processing

In real-world data mining, domain knowledge can be involved from the beginning of the problem understanding to the end when the results are presented to the users. It is necessary to understand the project objectives and requirements and then convert them into a data mining problem definition. In our process model, a domain knowledge base is used especially for data preprocessing and results post-processing. As in the previous section we describe that some attributes included in final model may not be provided by users but can be inferred by the domain knowledge base. For instance, annual rainfall is an important factor in determining the service life of building components in civil engineering. However, while using the data mining model to predict the service life of a building component, the user will most likely be inputting the location and material. The user may not be aware of the exact value of rainfall in the area. A domain knowledge base can provide such knowledge. This information can now be treated as input value as well for the model.

Furthermore, the domain knowledge base can be used in reinforcing the outputs made by the predictive model. Since our data mining models are for solving practical problems, the final results are significant to users. For example, an accurate lifetime prediction would save nearly \$5 million maintenance cost for Queensland Department of Public Works (QDPW) [12]. However, mining errors are inevitable even for a perfect model. The domain knowledge base is used to make sure the results are reasonable. For example, it is a domain knowledge that (1) a roof in a severe marine location will not last longer than one in benign environment, and (2) a stainless steel roof should last longer than the one with galvanized steel etc. Such in-built rules will be checked to assure the correctness of the results processed by models.

In general, the external domain knowledge base assists to deal with the vague queries in data preprocessing and to eliminate illogical outcomes in post-processing.

3 Real-World Implementation of UQBS

Our main objective in this research is to develop a building components lifetime prediction tool which will provide economic benefits to our industry partners. The ability to accurately predict the lifetime of building components is crucial to optimizing building design, material selection and scheduling of required maintenance. The material should be selected to match the suitability of the environment. In this section, we deploy the User Query Based Learning System in the lifetime prediction of metallic components in buildings.

3.1 Data Acquisition

The data sets include three different sources of service life information: Delphi Survey, Maintenance database and Holistic Corrosion Model according to various components of building (such as gutter and roof) and various materials (such as Zincalume, Galvanised steel, colorbond). Delphi Survey is expert opinions; Maintenance database is operational while Holistic Model is theoretical. They form three important sources of information for predicting lifetime of metallic components. They are independent but complement each other.

The Delphi Survey includes the estimation of service life for over 30 components and 29 materials. The maintenance database provides a repository of past experiences on component lifetime predictions under specific conditions. The outputs are service life of Zincalume and Galvanized Steel materials for roofs. The holistic Model contains information of corrosion for gutters in Queensland schools according to the theoretical understanding of the basic corrosion processes. The output of this data set is annual mass loss of Zincalume or Galvanized steel. Once the mass loss of material is determined, its service life is measured with appropriate formulas [12]. An independent model for Colorbond is also included in the Holistic model to deal with the particular material Colorbond. The output of Colorbond data set is service life of Colorbond for gutters.

Details of these data sets are presented in Table 1. There is no overlap of predicted service life from Maintenance, Holistic and Colorbond while the predicted result from them can be compared with the result from Delphi.

Table 1. Details of Data Sets

Data Set	Number of cases	Number of attributes	Building Component	Building Material	Target attribute
Delphi Survey	683	10	Roofs, Gutters and others	Galvanized Steel, Zincalume and Colorbond	Mean
Maintenance database	1297	18	Roofs	Galvanized Steel and Zincalume	Zincalume Life Galvanized Life
Holistic Model	9640	11	Gutters	Galvanized Steel and Zincalume	MLannual
Colorbond	4780	20	Gutters	Colorbond	Life of gutter at 600um

3.2 Feature Selection for Each Dataset

The User Query Based Feature Selection is the most important component in our process model. It will influence the performance of the model directly. It removes not only those attributes that are irrelevant to target value, but also those that can not be

provided by user and hard to get from external domain knowledge. This section describes this for each data set.

3.2.1 Colorbond. The original Colorbond data set has 20 attributes, in which ‘LocID’ is identification information and ‘Building Type’, ‘Position’, ‘Material’, ‘Building Face’ and ‘BuildingFacePos’ only have one value. After removing those irrelevant attributes, the attributes are as follows:

SALannual | Exposure | PositionVsExposure | Gutter Type | rain_annual_mm | cum_MZa_2ndYear | cum_dSTEEL_2ndYear | remCr | normCr | accelerated_corrosion_rate | Time to White Rust of Zinalume | Time to penetration of Zinalume | Time to onset of Red Rust | Life of gutter at 600um

‘Life of gutter at 600um’ is the target attribute.

We found the attributes that user can provide are ‘Exposure’, ‘PositionVsExposure’ and ‘Gutter Type’. They belong to Group 1. Attributes such as ‘Time to White Rust of Zinalume’, ‘Time to penetration of Zinalume’ and ‘Time to onset of Red Rust’ can not be obtained from user or domain knowledge. They belong to Group 3. They are not used as inputs although they may have mining value. Others are in Group 2 which can not be provided by user but can be obtained from domain knowledge. For those attributes in Group 2, we found that ‘SALannual’ and ‘rain_annual_mm’ are easy to get from domain knowledge while others need complex measurements. For comparison purposes, we decided to build three models according to input attributes as follows:

Model A:

SALannual | Exposure | PositionVsExposure | Gutter Type | rain_annual_mm | cum_MZa_2ndYear | cum_dSTEEL_2ndYear | remCr | normCr | accelerated_corrosion_rate

Model B:

SALannual | Exposure | PositionVsExposure | Gutter Type | rain_annual_mm

Model C:

Exposure | PositionVsExposure | Gutter Type

Data modeling and evaluation will be discussed in section 3.3 and 3.4.

3.2.2 Holistic. The original Holistic data set has 11 attributes, in which ‘LocID’ and ‘Location’ are identification information and ‘State’ and ‘Building Type’ only have one value. After removing those irrelevant attributes, the attributes are as follows:

XLong | YLat | SALannual | Material | Gutter Position | Gutter Maintenance | MLannual

As we describe in the Data Acquisition section, the service life is calculated based upon ‘MLannual’. We create a target variable named ‘Service Life’ and remove the false predictor ‘MLannual’. Attributes ‘XLong’, ‘YLat’, ‘Material’, ‘Gutter Position’ and ‘Gutter Maintenance’ belong to Group 1. ‘SALannual’ is easy to get from domain knowledge. Therefore, the final attributes of Model A for Holistic are as follows:

XLong | YLat | SALannual | Material | Gutter Position | Gutter Maintenance | Service Life

3.2.3 Maintenance. Maintenance data set is divided into two parts: one is for ‘Zincalume Life’ named Maintenance_Zi and the other is for ‘Galvanized Life’ named Maintenance_Ga. The attribute ‘Centre Code’ and ‘Centre Name’ are ignored since they are identification information. After that, their attributes are as follows:

Maintenance_Zi:

Longitude | Latitude | Salt Deposition | Zincalume Mass Loss | Marine | N | Zincalume Life

Maintenance_Ga:

Longitude | Latitude | Salt Deposition | Zinc Mass Loss | Steel Mass Loss | Marine | Nzinc | Nsteel | L | M | Zinc Life | Steel Life | Galvanized Life

For Maintenance_Zi, ‘N’ is calculated from ‘Zincalume Mass Loss’ while ‘Zincalume Life’ is calculated from ‘N’. Therefore ‘Zincalume Mass Loss’ and ‘N’ should be rejected. For Maintenance_Ga, ‘Nzinc’, ‘L’ and ‘Zinc Life’ are calculated from ‘Zinc Mass Loss’, ‘Nsteel’, ‘M’ and ‘Steel Life’ are calculated from ‘Steel Mass Loss’ while ‘Galvanized Life’ is calculated from ‘Zinc Life’ and ‘Steel Life’. Therefore all of them should be rejected. After that, their attributes are as follows:

Maintenance_Zi:

Longitude | Latitude | Salt Deposition | Marine | Zincalume Life

Maintenance_Ga:

Longitude | Latitude | Salt Deposition | Marine | Galvanized Life

For the remaining attributes, we found that ‘Longitude’ and ‘Latitude’ can be provided by user. ‘Salt Deposition’ and ‘Marine’ can be obtained from domain knowledge. Therefore they can be the final attributes for building models. We will build Model A for Maintenance_Zi and Maintenance_Ga respectively.

3.2.4 Delphi. The original Delphi data set has 10 attributes. They are ‘Building type’, ‘Component’, ‘Measure’, ‘Environment’, ‘Material’, ‘Maintenance’, ‘Mode’, ‘Mean’, ‘SD’ and ‘Criteria’. The estimated life was stored in two forms: the mode and the mean as well as a standard deviation (SD) for the mean. As we want a real value to be the final predicted result, the attribute ‘mean’ is chosen as the target attribute. Since ‘Mean’ and ‘Mode’ are different measurements for same information, we should remove ‘Mode’. ‘SD’ can not be considered as input because it is a part of output. ‘Criteria’ relates to how good the agreement was in the response from Delphi Survey. It is not useful in mining and should be removed. This data set contains life information of service life, aesthetic life and time to first maintenance. As we are only interested in service life, we remove those instances that ‘Measure’ is not equal to ‘Service Life’. After removing those instances, ‘Measure’ becomes unary and hence should be removed. The other attributes are as follows:

Building type | Component | Environment | Material | Maintenance | Mean

They all could be provided by user. Thus these attributes are kept as inputs to know their influence to the target value. They are the final attributes of Model A for Delphi.

3.3 Method Selection and Modelling

The overflow of prediction model is given in Figure 2. Data mining methods are applied to all three data sets to build three predictors first. After that, three predictors can be used to do prediction for user inputs. The domain knowledge base includes three parts: salt deposition knowledge, rainfall knowledge and generalized rules extracted from three predictors. Salt deposition and rainfall knowledge is used to preprocess user inputs while generalized rules are for post-processing the predicted results.

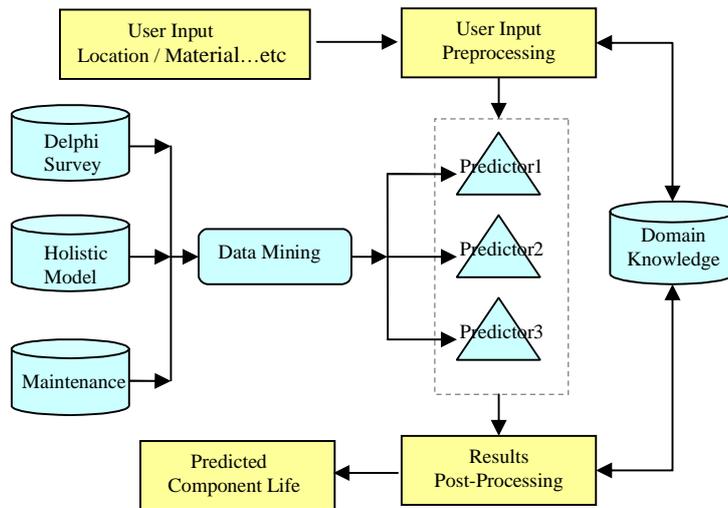


Fig. 2. Overflow of Prediction Model

There are various data mining methods like Naïve Bayes, K Nearest Neighbors (KNN), Decision Tree (DT) and Neural Network (NN) that can be considered to do prediction tasks. The target value in this application is continuous value. We report the experiments to choose the best method for building predictors elsewhere [9]. Our previous experimental results have shown that the DT and naive bayes methods on the discretised output perform poorly [9]. Three best methods, which are M5 [13] for Delphi Survey and Colorbond, KNN for Holistic Model and NN for Maintenance database, were selected.

As shown in Figure 2, a domain knowledge base is built to be used for user inputs preprocessing and results post-processing. This knowledge base includes a set of rules which are extracted from three predictors built already. It identifies the range of service life for various components using different materials in different locations. Therefore, the comprehensibility of predictors is important as well. The M5 model tree [13] not only learns efficiently for this task but also is easy to understand. Also the model trees generated are generally much smaller than regression trees.

We also combine the instance-based and model-based learning [14] to improve the performance of M5. This method first uses the instance-based approach to find a set

of cases similar to the target case. Then the class values of similar cases are adjusted using the value predicted by the model tree before they are combined in some way.

Detailed algorithm is given as follow:

Let T be Training Set
P be a set of cases assembled by the instance-based method
M be a predictive model constructed by the model-based method
For a case C, V(C) is the class value of C
M(C) is the value predicted for C by M
Suppose we are to predict the class value for an unseen case U, say V(U)
1) Get M(U) by using the predictive model M
2) Get a subset {P₁, P₂, ... , P_k} of cases similar to U by using the instance-based method
3) {V(P₁), V(P₂), ... , V(P_k)} is a subset of class values for these similar cases
4) For i =1 to k
 diff(i) = M(P_i) – M(U)
 V(P_i) = V(P_i) – diff(i)
5) V(U) = (V(P₁) + V(P₂) + ... + V(P_k)) / k

3.4 Evaluation

A public data mining tool, Weka, was chosen to build the M5 model trees for each dataset. The integrated model combining the instance-based and model-based methods uses the KNN (K=3) and M5. Tenfold cross validation (10-CV) was used throughout the experiments described in this paper.

The average correlation coefficient on test set over the 10-CV results is reported in Table 2. Each row is associated with one data set, each column with one model built using different input attributes. The detail for input attributes of corresponding Models for each data set is described in Section 3.2. For Delphi, Holistic and Maintenance, only one model was built. We do not have to define selection criteria (stopping criteria). However, there are three models for Colorbond. We define our selection criteria is that correlation coefficient is not less than 0.85. Therefore, Model B is selected as the final Model. From the result of Colorbond, we can confirm that the more attributes included in model, the better performance it has. However, as the model is ultimately used by users, we have to consider the input attributes carefully. Only those attributes that user can provide or obtained from domain knowledge can be included in final models.

Table 2. Average Correlation Coefficient on Test Set

Data set	Correlation Coefficient (CC)		
	Model A	Model B	Model C
Delphi Survey	0.9198	N/A	N/A
Holistic Model	0.979	N/A	N/A
Colorbond	0.9991	0.9103	0.4321
Maintenance for Galvanized	0.9421	N/A	N/A
Maintenance for Zinalume	0.8692	N/A	N/A

The performance of the models is improved by combining KNN and M5 learning methods as defined in previous section. The results are presented graphically in Figure 3, where y-axis is Correlation coefficient and x-axis is data set.

D: Delphi H: Holistic C: Colorbond M_G: Maintenance for Galvanized
M_Z: Maintenance for Zincalume

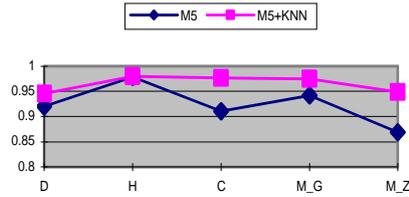


Fig. 3. Correlation Coefficient of M5 and M5 + KNN on Test Set

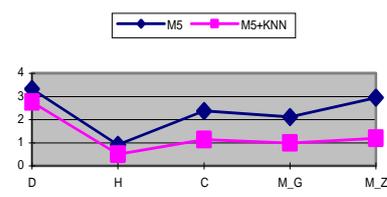


Fig. 4. Mean Absolute Error of M5 and M5 + KNN on Test Set

The Figure 3 shows the better correlation coefficients can be obtained by combining M5 and KNN learning methods. The method seems to provide robust improvement for relatively weaker models like Maintenance for Zincalume and Colorbond. While for the near-perfect models like Holistic, the improvement is not so obvious.

Another performance measure investigated is Mean Absolute Error on test set. The results are presented graphically in Figure 4 where y-axis is Mean Absolute Error and x-axis is data set. As shown in Figure 4, Mean Absolute Error of M5 + KNN for each data set are reduced. This confirms the integrated method can improve the prediction accuracy significantly. Both Figure 3 and Figure 4 indicate for all datasets, the M5 + KNN works better than M5.

Finally, we compare our proposed UQBLs method with the method when we select all attributes. In other words, the method (named models D) uses all attributes in groups 1, 2 & 3 to build the models. We test these models with the same test cases as used in models A, B & C (as discussed in Table 2). Such test set reflects predictive performance of the models D on unseen cases. The Correlation Coefficient and Mean Absolute Error of M5 on the training set and test set of the models D are summarized in Table 3.

Table 3. Correlation Coefficient of M5 on Training set and Test Set

Data set	Correlation coefficient (CC)		Mean Absolute Error (MAE)	
	Training	Test	Training	Test
Delphi Survey	0.9333	0.9156	3.1412	3.8315
Holistic Model	0.9892	0.8265	0.9113	11.2955
Colorbond	1	0.4518	0.065	7.2102
Maintenance for Galvanized	0.9883	0.6982	0.664	6.1305
Maintenance for Zincalume	0.9971	0.6054	0.3119	5.7801

Table 3 shows that the performance on training set is nearly perfect. That is because all useful information is included in building the model. However, the performance on test set reduces significantly due to many missing attributes values. The more missing attributes values, the more performance it reduces. Therefore, the

predictive accuracy of the models D on training set is high, the generalization accuracy on test set relatively lower. This infers that such models are not for the purpose of using them in practice. Comparison result between the models D and our proposed method (UQBLs) is presented in Figures 5 and 6, where y-axis is Correlation coefficient and Mean Absolute Error respectively and x-axis is data set.

The Figures 5 and 6 shows the CC of UQBLs is always higher than ones of Models D while the MAE of UQBLs is always lower than ones of Models D. This proves that our proposed method (UQBLs) outperforms the method which selecting all attributes (Models D).

D: Delphi H: Holistic C: Colorbond M_G: Maintenance for Galvanized
M_Z: Maintenance for Zinalume

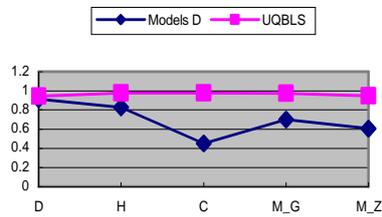


Fig. 5. Correlation Coefficient of Models D and UQBLs on Test Set

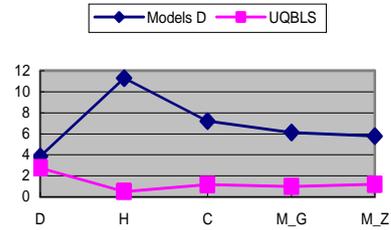


Fig. 6. Mean Absolute Error of Models D and UQBLs on Test Set

4 Conclusions

The real-world data mining applications generally have some specific use. The lifetime prediction system will be a big aid for saving maintenance cost in civil engineering field. This paper develops a new learning system framework, called as User Query Based Learning System (UQBLs) for solving reduced performance in using the predictive models in practice where all input information are not available for querying to the system.

A number of lessons are learned from deploying the UQBLs in lifetime prediction of metallic components.

- Our data sets include three sources of life information, which contains completely different features. However, they all can be grouped into three types: 1) that can be provided by users 2) that can be obtained from domain knowledge 3) neither 1) nor 2)
- The User Query Based feature selection is the most important procedure in UQBLs. It indicates that feature selection is not only for removing irrelevant features. If a feature, which belongs to type 3, is included in the final model, the performance of model reduces significantly in using model. Therefore, such features, even if they are useful in mining, should be rejected.
- The User Query Based feature selection may result in too many very useful features being rejected. This reduces the performance of the predictive models. We show that the integrated method combining M5 (model based) and KNN (instance based) is successfully applied in such cases for improving performance.

- The User Query Based Learning System is compared with usual method on prediction accuracy. The results prove that our methodology performs better especially when the datasets contains many attributes that user can not provide in using the system.
- External domain knowledge is used for dealing with incomplete and vague queries and post-processing the predicted service life from different predictors. This novel use of domain knowledge improves the prediction accuracy when users can not provide all inputs.

In summary, UQBLs has been proven to effectively solve a special issue of real-world data mining application.

References

- [1] G. Melli, O. R. Zaiane, and B. Kitts, "Introduction to the special issue on successful real-world data mining applications," SIGKDD Explor. Newsl., vol. 8, pp. 1-2, 2006.
- [2] W. Kessler, R. W. Kessler, M. Kraus, R. Kubler, and K. Weinberger, "Improved prediction of the corrosion behaviour of car body steel using a Kohonen self organising map," presented at Advances in Neural Networks for Control and Systems, IEE Colloquium on, 1994.
- [3] H. Furuta, T. Deguchi, and M. Kushida, "Neural network analysis of structural damage due to corrosion," presented at Proceedings of ISUMA - NAFIPS '95 The Third International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society, 1995.
- [4] G. Morcous, H. Rivard, and A. M. Hanna, "Case-Based Reasoning System for Modeling Infrastructure Deterioration," Journal of Computing in Civil Engineering, vol. 16, pp. 104-114, 2002.
- [5] H. G. Melhem and Y. Cheng, "Prediction of remaining service life of bridge decks using machine learning," Journal of Computing in Civil Engineering, vol. 17, pp. 1-9, 2003.
- [6] H. G. Melhem, Y. Cheng, D. Kossler, and D. Scherschligt, "Wrapper Methods for Inductive Learning: Example Application to Bridge Decks," Journal of Computing in Civil Engineering, vol. 17, pp. 46-57, 2003.
- [7] L. Breiman, "Bagging Predictors," Machine Learning, vol. 24, pp. 123-140, 1996.
- [8] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," presented at Proceedings of the Thirteenth International Conference on Machine Learning, 1996.
- [9] E. Ge, R. Nayak, Y. Xu, and Y. Li, "Data Mining for Lifetime Prediction of Metallic Components (AusDM 2006)," accepted at The Australasian Data Mining Conference, Sydney, 2006.
- [10] CRISP-DM, "Cross Industry Standard Process for Data Mining."
- [11] M. D. a. H. Liu, "Feature selection for classification," presented at Intelligent Data Analysis, 1997.
- [12] I. Cole, M. Ball, A. Carse, W. Y. Chan, P. Corrigan, W. Ganther, T. Muster, D. Paterson, G. Tinidad, M. L. Maher, and P. Liew, "Case-Based Reasoning in Construction and Infrastructure Projects - Final Report," 2002-059-B, March 2005 March 2005.
- [13] J. R. Quinlan, "Learning with Continuous Classes," presented at 5th Australian Joint Conference on Artificial Intelligence, 1992.
- [14] J. R. Quinlan, "Combining instance-based and model-based learning," presented at Proceedings of the Tenth International Conference on Machine Learning, Amherst, Massachusetts, 1993.