J. Parallel Distrib. Comput. 117 (2018) 256–266

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Implications of deep learning for the automation of design patterns organization

Shahid Hussain [a], Jacky Keung [a], Arif Ali Khan [a], Awais Ahmad [b,**], Salvatore Cuomo [c], Francesco Piccialli [c], Gwanggil Jeon [d,*], Adnan Akhunzada [e]

[a] Department of Computer Science, City University of Hong Kong, Hong Kong
[b] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Republic of Korea
[c] University of Naples Federico II, Naples, Italy
[d] Department of Embedded Systems Engineering, Incheon National University, Republic of Korea
[e] Comsats Institute of Information Technology, Islamabad, Pakistan

HIGHLIGHTS

- There is a need to bridge the gap between the semantic relationship between patterns.
- We propose an approach by leveraging a powerful deep learning algorithm named Deep Belief Network (DBN).
- The DBN learns on the semantic representation of documents formulated in the form of feature vectors.
- We performed a case study in the context of a text categorization based automated system.
- The experimental promising results suggest the significance of the proposed approach to construct a more representative feature set.

ARTICLE INFO

Article history:
Received 1 March 2017
Received in revised form 27 April 2017
Accepted 27 June 2017
Available online 20 July 2017

Keywords:
Design patterns
Deep learning
Feature set
Performance
Classifiers

ABSTRACT

Though like other domains such as email filtering, web page classification, sentiment analysis, and author identification, the researchers have employed the text categorization approach to automate organization and selection of design patterns. However, there is a need to bridge the gap between the semantic relationship between design patterns (i.e. Documents) and the features which are used for the organization of design patterns. In this study, we propose an approach by leveraging a powerful deep learning algorithm named Deep Belief Network (DBN) which learns on the semantic representation of documents formulated in the form of feature vectors. We performed a case study in the context of a text categorization based automated system used for the classification and selection of software design patterns. In the case study, we focused on two main research objectives: 1) to empirically investigate the effect of feature sets constructed through the global filter-based feature selection methods besides the proposed approach, and 2) to evaluate the significant improvement in the classification decision (i.e. Pattern organization) of classifiers using the proposed approach. The adjustment of DBN parameters such as a number of hidden layers, nodes and iteration can aid a developer to construct a more illustrative feature set. The experimental promising results suggest the significance of the proposed approach to construct a more representative feature set and improve the classifier's performance in terms of organization of design patterns.

1. Introduction

The text categorization approach has been applied in many domains to introduce an automated system to decrease the cost and computational time across the problems. The aim of automated systems is to classify the texts into appropriate classes with respect to their content, such as spam email filtering [17], web page classification [29], sentiment analysis [23], and author identification [43].

* Corresponding author.
** Corresponding author.
E-mail addresses: Shussain7-c@my.cityu.edu.hk (S. Hussain), jacky.keung@cityu.edu.hk (J. Keung), aliakhan2-c@my.cityu.edu.hk (A.A. Khan), aahmad.marwat@gmail.com (A. Ahmad), salvatore.cuomo@unina.it (S. Cuomo), francesco.piccialli@unina.it (F. Piccialli), gjeon@inu.ac.kr (G. Jeon), a.qureshi@comsats.edu.pk (A. Akhunzada).

http://dx.doi.org/10.1016/j.jpdc.2017.06.022
0743-7315/

The steady increase in the number of design patterns in the literature and online repositories causes certain issues [2] such as (1) deep knowledge to understand the classification scheme used to organize the patterns [3], (2) the lack of classification scheme [11], (3) formal specification of patterns [19,20], (4) the heterogeneous description of design patterns [38], and (5) the selections of right design patterns for inexperienced developer in the existence of complex [3] and spoiled patterns [4]. In order to address these issues, the text categorization approach has been employed to automate the organization and selection of design patterns [15,16]. In the text categorization based automated system, it has been reported that the use of an extreme number of features may have undesirable effects on the classification accuracy and computation time of the automated systems [8,14,37]. The wrapper, embedded, and filter are three common schemes which are used to formulate the feature selection methods. The wrapper and embedded based feature selection methods required classifier interaction, while the filter-based methods do not need such interaction to construct a feature set. Besides, the running time of wrappers and embedded methods may be increased due to the classifiers interaction [36,42]. The wrappers and embedded methods are specific to the learning model as compared to filter-based methods. Consequently, the use of filter-based methods are recommended for the automatic systems implemented via text categorization approach. Though in the context of text categorization based automated systems for the organization and selection of design pattern, the effect of global filter-based feature selection methods have been reported [8,37]. However, the association of semantic relationship between design patterns (i.e. Document) and the constructed features (i.e. Attributes of feature space constructed through feature selection method) still need to be addressed.

Recently, researchers have adopted the use of deep learning algorithms to improve certain research tasks such as to improve fault localization [21], model the programming languages to suggest the code [40], to retrieve program's structural information, and produce features from program source code for the prediction of defects [39,41]. Subsequently, the applicability of unsupervised feature learning and deep learning in certain areas have been reported [1]. In this paper, we look the capacity of deep learning [13] and propose an approach by leveraging a powerful representation learning algorithm Deep Belief Network [12], to learn features from feature vectors extracted from the problem domain of design pattern collection [6,9,12]. The aim of the proposed approach is to fill the gap between the semantic relationship between documents(or Patterns) and the features which are used for the organization of design patterns. The Deep Belief Network (DBN) is a generative graphical model to learn a representation which can aid to reconstruct the training data with a high probability. Before applying DBN to learn features from patterns catalog, we perform some preprocessing tasks such as remove stopwords and words stemming to preserve the features (with their frequencies) in the feature vector form [14]. Subsequently, these feature vectors are used as input to the DBN algorithm. We performed certain experiments to evaluate the effectiveness of the proposed approach. Our main contributions in this paper are.

- We leverage the Deep Belief Network (DBN) algorithm and extract the features which are learned from the semantic relationship between design patterns of a collection in order to improve the performance of classifiers employed for the automation to organize the design patterns.
- We propose an evaluation model to evaluate the performance of classifiers in the context of organization of design patterns.
- We evaluate the capacity of the proposed approach in the construction of more illustrative feature sets as compared to existing filter-based feature selection methods, such as Document Frequency (DF) [44], Improved Gini Index [31], Correlation, Gain Ratio, Information Gain [22].
- We used three well-known classifiers [37] and three widely used design pattern collections [6,9,12] which have been used as a benchmark in different domains and evaluate the effectiveness of the proposed approach.

The rest of the paper is organized into eight sections. In Section 2, we summarized the related work regarding the classification of design pattern, feature selection methods, and the application of deep learning. In Section 3, we present the brief introduction of text categorization approach. In Section 4, we describe the working procedure of the proposed approach. In Section 5, we describe the experimental procedure and present a case study in the domain of classification of design patterns. In Section 6, we discussed the results of the presented case study. Subsequently, in Section 7, we summarized the evaluation results. Finally, in Sections 8 and 9, we discuss some threats to the validity and the conclusion of our work respectively.

2. Related work

We performed a case study to describe the effectiveness of the proposed approach. Since the context of case study includes the empirical investigation of existing filter-based feature selection methods (such as Document Frequency, Improved Gini Index, Correlation, Gain Ratio, Information Gain) and the proposed approach with three widely known design pattern collections. Consequently, we summarized the related work with respect to use of filter-based feature selection methods, organization of design patterns, and deep learning algorithms under certain circumstances.

2.1. Design patterns organizations/classification

The literature depicts that authors have accomplished different classification schemes to categorize the design patterns on the base of their interest and experience in different domains [6,9,27,30], for example, Rising used the application domain as the category for the classification of design patterns [28]. Usually, authors either use the descriptions of problem's intent or the link between solutions of design patterns in order to divide the design patterns into high-level categories. The problem-based classification schemes, including their domain and high-level categories are summarized in Table 1.

Zimmer considers different aspects of the link between the established solutions of design patterns, for example, how solutions of two patterns are similar and integrated to each other [44]. Kim and Han analyzed the solution structures of design patterns and consider dependencies to group patterns accordingly [18].

2.2. Feature selection methods

In order to decrease the undesirable effect of useless features, feature selection methods are applied. The wrapper, embedded and filter are three common schemes used to describe the feature selection methods. The wrappers and embedded methods are specific to the learning model as compared to filter-based methods [37].

Ogura et al. [24] categorized the filter-based feature selection methods referred as one-sided and two-sided on the basis of their characteristics. The one-sided feature selection methods assign a score (i.e. Score ≥ 0) to a feature of membership class while assigning a score (i.e. Score < 0) for non-membership classes. For

Table 1
Summary of approaches used for the problem-based classification of design patterns.

Author	Domain	High-level categories
Gamma et al. [9]	Object-oriented Design Problems	Creational, Structural and Behavioral
Pree [27]	Object-oriented Design Problems	Pree's classification scheme relies on Recursive Structures and Abstract Coupling
Coad et al. [5]	Object-oriented Design Problems	Transaction, Aggregate, Plan and Interaction
Tichy [34]	Software Design Problems	Decoupling, Variant Management, State Handling, Control, Virtual Machines, Convenience, Compound, Concurrency and Distribution
Rising [28]	Application Type	Accounting, Hypermedia, Trading, C++ Idioms, System Modeling, Air Defense, Interactive, Database, Persistence and Trading
Douglass [6]	Real Time	Architectural Design, Safety and Reliability, Memory, Concurrency Resources and Distribution
Booch [3]	Architectural Design	45 categories

example, the odds ratio is referred as the one-sided feature selection method, and measure the membership and non-membership normalized values via its nominator and denominator [24]. The two-sided feature selection methods assign scores to a feature based on the combination of membership and non-membership to any class. For example, Document Frequency, Information Gain, and improved Gini Index [8,37]. Subsequently, Tasci and Gungor classified filter-based methods into two categories referred as local and global, depend on unique or multi class-based score assign to a feature respectively [33]. In the text classification approaches, the class imbalances should be handled to improve the performance of classifiers. Gunal [10] applied the genetic algorithm and combined several filter methods and investigated the classification performance. Subsequently, the authors reported that combined filter methods outperform the performance of an individual filter method. In a recent study, Uysal [3] introduces an Improved Global Feature Selection Scheme (IGFSS) to improve the classification performance of global feature selection methods by constructing a more illustrative feature set which includes evenly class-wise distributed features. Though, we found numerous approaches from the literature, in the domain for text categorization based automation, the construction of more representative features set is still an ongoing topic for researcher community, which motivates us to introduce a new approach to derive the illustrative features by enabling its association with semantic relationship of design patterns.

2.3. Deep learning in software engineering (SE) domain

Numerous deep learning algorithms have been introduced and adopted by the research community to perform certain tasks in the domain of software engineering. The number of hidden layers and characteristics of datasets are the two potential issues in the training of a conventional deep learning algorithm. However, researchers have started to use distributed processing in order to build more extensive deep learning networks which can enable the research community to manage the large datasets [7]. Due to space limitation, we have summarized the work domain and application of deep learning algorithms in Table 1, which presents its significance. However, we recommend [42] for those readers who have quested to know thoroughly regarding the application of the deep learning approach.

3. Brief introduction of text categorization approach

In machine learning and text mining domains, automatic text categorization is performed either through supervised or unsupervised learning techniques. The former techniques use class labels assigned to documents and latter techniques use attributes of the data and dis (similarity) measures to automate their learning. Text preprocessing is mandatory for the text categorization and should be performed before the learning process [19,20]. The main steps of text preprocessing are:

3.1. Preprocessing

In the preprocessing step, two activities are performed to transform documents into strings. The first activity is to remove stop words that are more frequent words and carry no information, such as Preposition, Conjunction, and Pronouns. The second activity is word stemming, which is performed to make a group of words that have same conceptual meaning. Subsequently, the numbers of words in the documents collection are reduced. Porter's stemmer is a well-recognized stemming algorithm [18], used to transform English words into their stem iteratively through a set of rules.

3.2. Indexing

In the indexing step, Vector Space Model (VSM) is used as well-known indexing method to describe the documents. The term-document category of VSM is widely used as compared to word-context and pair-pattern matrices categories [35]. In VSM, each document is described by a vector of words. Commonly, a document collection is described through a word-by-document matrix D, where each entry refers to the word's occurrence in the document.

$$D = W_{wd}. \tag{1}$$

In Eq. (1), W_{wd} is the weight of word w in document d. There are many ways to determine the weight W_{wd} of word w in document d. For example, Binary, Term Frequency (TF), Term Frequency Inverse Document Frequency (TFIDF), Term Frequency Collection (TFC), Length Term Collection (LTC), and Entropy are commonly used weighting methods in text categorization [14]. The term weighting schemes are used to improve the performance and remove the noise from the documents.

3.3. Feature selection

In feature selection step, different techniques are used to remove the features, however, computational time and classification accuracy are the main issues related to the discriminative power of each technique [37]. The wrappers, filters, and embedded methods are the three main groups used to categorize the feature selection techniques. The wrapper and embedded techniques are not recommended for text categorization since these techniques require a frequent interaction of classifiers in their flow, which may increase the running time. Due to its simplicity, efficiency and low running cost, filter-based techniques such as Document Frequency (DF), Information Gain (IG), Mutual Information (MI), Correlation

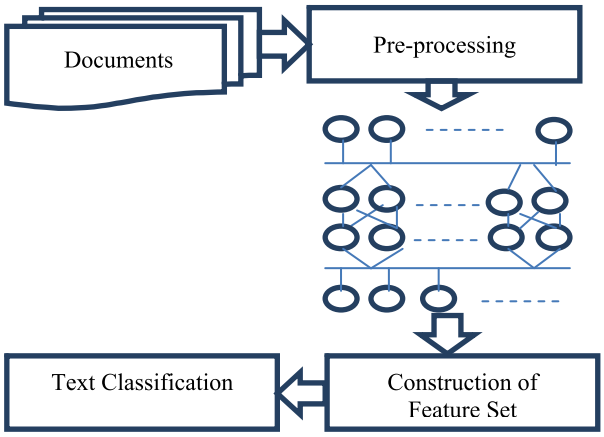


Fig. 1. Overview of the proposed approach.

Coefficient (CC) and Chi-Square (CHI) are commonly used in the text categorization [8,14] approach.

4. Proposed approach

We consider the design pattern organization as an information retrieval problem [15,16]. We look at the capacity of deep learning algorithm to address the gap between the semantic relationship between design patterns and the features which are used for the organization of design patterns. The aim of the proposed approach is to construct a more illustrative feature set to improve the performance of classifiers employed in the text categorization based automated system. The proposed approach is applied in three phases named Pre-processing, Construction of Feature Set and text classification. (See Fig. 1.)

4.1. Preprocessing

In the preprocessing phase, four activities (shown in Fig. 2) are performed to transform documents into strings and formulate them into feature vector form. The first activity is to remove stop words that are more frequent words and carry no information, such as Preposition, Conjunction, and Pronouns. The second activity is word stemming, which is performed to make a group of words that have same conceptual meaning. Subsequently, the numbers of words in the documents collection are decreased. Porter's stemmer is a well-recognized stemming algorithm [26], used to transform English words into their stem iteratively through a set of rules. The third activity is related to present the feature vectors into integer vector as prerequisite for DBN algorithm [12]. Though, in the text classification numerous weighting methods such as Term Frequency (TF), Binary, Term Frequency Inverse Document Frequency (TFIDF), Term Frequency Collection (TFC), Length Term Collection (LTC), and Entropy are recommended. However, in this study, we use TF for representation of an integer vector. The size of feature vectors is same as prerequisite inputs to DBN algorithm. Finally, the feature selection methods besides the proposed approach are applied to construct the feature sets for training of classifiers.

4.2. Construction of feature set

In the construction of Feature set phase, the DBN algorithm is applied to generate the features for the classification of documents into appropriate classes. The aim of DBN learns from given input data (in our case feature vectors) via multi-level neural network and reconstructs the content of input data with high probability. The DBN includes one input, N hidden, and one output (i.e. Top) layer with a set of several nodes. The number of hidden layers and nodes can be tuned by user accordingly. Eq. (1) describes the joint distribution of input and hidden layers.

$$P\left(v, h^1, \ldots, h^l\right) = P\left(v|h^1\right)\left(\prod_{k=1}^{l} P\left(h^k|h^{k+1}\right)\right) \tag{2}$$

where the term v is referred as feature vector from input layer, l is referred as total number of hidden layers and h^k is referred as a feature vector of k^{th} layer. The conditional distribution for two adjacent layers (for example k and $k+1$) is referred as $P\left(h^k|h^{k+1}\right)$ and calculated through Restricted Boltzmann Machines (RBM). In case of node, DBN learns probabilities from any current node to the upper level's nodes. The backward validation is performed by DBN to reconstruct the input feature set by tuning the weights between nodes across the layers.

4.3. Text classification

There are several supervised learning techniques that can be used in the text classification based automated system, however, it cannot be decided which one is better since no classifier can obtain better results for all problems. Consequently, it is recommended to find the best classifier for a new problem. In the text classification domain, researchers have used and reported the classification performance of C4.5 Decision Tree, Naïve Bayes, and Support Vector Machine (SVM) supervised learning techniques [37]. The details of each supervised learning algorithm is given in the Appendix. In this study, we used these classifiers to evaluate the effectiveness of the proposed approach.

5. Experimental setup

We performed a set of experiments to investigate the effectiveness of the proposed approach against the individual effect of four global filter-based feature selection methods named Document Frequency (DF) [42], Improved Gini Index [31], Gain Ratio and Information Gain [22,37], on the performance of classifiers for the organization of a target pattern catalog. We have summarized the experimental procedure and presented through the following pseudocode.

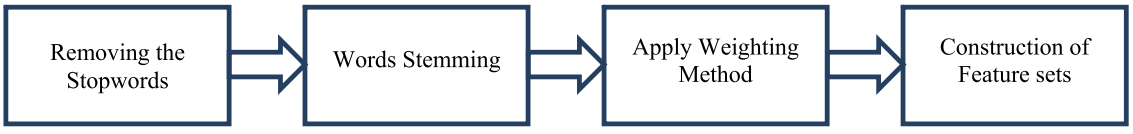


Fig. 2. Set of preprocessing activities.

Pseudocode: Pattern_Organization (PC, SC, N, M)

Input:

PC = {The set of problem domain description of design patterns in the given Pattern Collection}

SL = {Set of Supervised Learners under study, such as Naïve Bayes (NB), C4.5 Decision Tree, and Support Vector Machine (SVM))}

N = The number of design patterns in a collection, For example, in case of GoF and Douglass pattern collection the value of of N is 23 and 34 respectively.

M = The number of unique words, For example, in case of GoF and Douglass pattern collection the value of of M is 1465 and 1271 respectively.

Procedure Begin:

Step-1. Perform the first three activities of the preprocessing phase of the proposed approach to present N feature vectors with M unique words.

Step-2. Apply a weighting method (such as Binary, TF, TFIDF, TFC, LTC, and Entropy)

Step-3. Apply a Supervised learning technique to organize the N design patterns.

Step-4. Apply the evaluation criteria (Section 5.2) to assess the performance of the supervised learner with the corresponding weighting method.

Step-5. Repeat Steps 2, 3 and 4 until the all possible assessments (according to number of classifiers and weighting methods used in the proposed study) have done.

Step-6. For each Supervised learner select the best weighting method, and classifies the patterns into appropriate categories.

Step-7. Apply the evaluation criteria (Section 5.2) to assess the performance of the supervised learner with the their best weighting methods.

Step-8. In order to evaluate the improvement in the effectiveness of supervised learner, apply feature selection method (Such as Document Frequency (DF), Information Gain (IG), Mutual Information (MI), Correlation Coefficient (CC) and Chi-Square (CHI)) and proposed approach (Exploit through a deep learning algorithm DBN) to rank the top n features.

Step-9. Apply the evaluation criteria (Section 5.2) to assess the performance of the supervised learner with the their best weighting methods and top n ranked features (Step-8)

Procedure End

Output:

The outperform supervised learner with best weighting and feature selection method for the organization of design patterns catalog (i.e. PC).

5.1. Dataset

In software engineering, some groups of experts from different domains have categorized the design patterns on the basis of their intent, motivation, and applicability. We consider three frequently used design pattern collections named Gang-of-Four (GoF) [9], Douglass [6], and Security [30] pattern collections. The brief introduction and descriptive statistics of pattern collections [6,9,30] is given in the following sections.

5.1.1. Gang-of-Four (GoF) design pattern collection

The GoF collection includes 23 object-oriented design patterns which are divided into three groups named Creational, Structural, and Behavioral. This case study includes 1465 non-repeated words of all 23 design patterns after removing the stop words and words stemming [9].

5.1.2. Douglass design pattern collection

The Douglass collection includes 34 real time systems relevant design patterns which are divided into five categories named

Table 2
Implication of Deep learning in different domains.

Study	Domain	Study objective
Wang et al. [39]	Defect Prediction	Authors leveraged the deep learning algorithm (i.e. Deep Belief Network (DBN)) to learn semantic features from the token vectors which are extracted from programs' Abstract Syntax Tree in order to build accurate prediction models.
Yang et al. [42]	Defect Prediction	Authors leverage the deep learning to produce the features from the existing set of features and used them to predict buggy commit.
Lam et al. [21]	Fault Localization	In this work, the authors combined the information retrieval techniques and deep learning algorithms in order to enhance the fault localization.
White et al. [40]	Code Suggestion	In this work, the authors used deep learning to model the programming languages to suggest the code.
Pascanu et al. [25]	Malware Classification	In this work, the authors present an approach similar to natural language modeling, which learn the language of malware spoken through the executed instructions and extract the features.

Concurrency, Safety, and Reliability, Distribution, Memory, and Resource. This case study includes 1271 non-repeated words of all 34 design patterns after removing the stop words and words stemming [6].

5.1.3. Security design pattern collection

The Security collection includes 46 security systems relevant design patterns which are divided into eight categories named SACA (System Access and Control Architecture), ACM (Access Control Model), IA (Identification and Authentication), OSAC (Operating System Access Control), SIA (Security Internet Application), FA (Firewall Architecture), ESRM (Enterprise Security and Risk Management) and Accounting [30]. This case study includes 1230 non-repeated words of all 34 design patterns after removing the stop words and words stemming.

5.2. Evaluation criteria

In the text categorization approach, Precision and Recall for learners can be estimated using micro-averaging or macro-averaging equations, however, the precision of micro-averaging (used in this study) is better than macro-averaging [8,37]. In order to select the best weighting method for each unsupervised learning technique, the effect of each method such as Binary, TFIDF, TF, TFC and Entropy on respective learning technique is computed in terms of F-measure. Subsequently, weighting method with highest F-measure value is chosen as the best for the corresponding learning technique.

$$P = \frac{\sum_{c=1}^{N} \mathrm{TP}_c}{\sum_{c=1}^{N} (\mathrm{TP}_c + \mathrm{FP}_c)}, \quad \text{Micro-Averaging} \tag{3}$$

$$R = \frac{\sum_{c=1}^{N} \mathrm{TP}_c}{\sum_{c=1}^{N} (\mathrm{TP}_c + \mathrm{FN}_c)}, \quad \text{Micro-Averaging} \tag{4}$$

$$F = \frac{2 \times P \times R}{(P + R)}. \tag{5}$$

In Eqs. (2)–(4), N is the number of design pattern classes decided to evaluate the performance of learning techniques. For example, in the case of GoF design pattern collection N is 3 (Section 6.3). The term TP is the number of design problems which are correctly identified for each design pattern class, FP is the number of design problems which are incorrectly identified for each design pattern class, and FN is the number of design problems which are missing from each corresponding design pattern class. The values of P, R, and F are computed using Eqs. (2)–(4) respectively. Subsequently, Eq. (4) can be used to evaluate the outperform supervised learner with their best weighting methods.

5.3. Experiment design for construction of feature set

In the proposed approach, we leverage DBN to construct a more illustrative feature set by considering the semantic relationship between patterns, and assess the improvement in the performance of supervised learners (under study) for the organization of design patterns. We performed all experiments using darch-CRAN package implemented in *R* tool as software environment for statistical computing. In each experiment, we used DBN with default parameters (i.e. the number of hidden layers, and the nodes in each layer and iteration) for each dataset. However, the values of these parameters can be tuned by the user to obtain more effective results. Subsequently, in each experiment, we performed widely used 10-fold cross validation [32], and comparatively assess the performance of classifiers on the feature set constructed through the proposed approach and existing filter-based feature selection methods. In order to maintain consistency, the number of features filtered from the feature set of each filter-based method is same as retrieved via tuning the DBN employed in the proposed approach.

6. Results and discussion

For each design pattern collection under study, we called the pseudocode (Section 5) to determine the (1) best weighting method for each supervised learner, (2) outperform classifiers with their best weighting method, (3) effect of the constructed feature set (via proposed approach and global filter-based feature selection method under study) on the performance of classifiers, and (4) improvement in the performance of classifiers by applying the proposed approach. Subsequently, in each experiment, we consider the pattern category as dependent variable while extracted features as independent variables. Subsequently, we performed widely used 10-fold cross validation [32] to improve the classification.

6.1. Organization of GoF design patterns

The patterns in the GoF pattern collection are divided into three categories. The preprocessing activities of the proposed approach (Section 3.1) are applied to construct the dataset. We follow the procedure described in pseudocode and use the F-measure criterion (Section 5.2) to find the best weighting method for each supervised learner. According to the F-measure criterion, the evaluation result in the selection of best weighting method for each supervised learner is shown in Fig. 3. Subsequently, the results regarding the best weighting for each classifier are shown in Table 2. The main consequences of the results of Fig. 3 and Table 2 are as follows.

- We observe the better performance of NB, SVM, and DT with TFC, TFC and TDIDF weighting methods respectively.

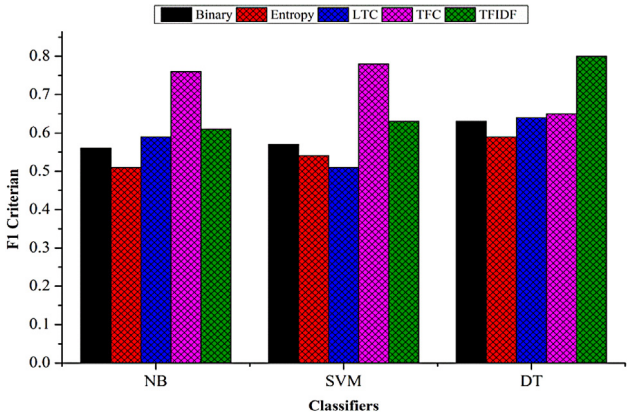


Fig. 3. Performance evaluation of classifiers for the weighting methods on the GoF Design pattern collection.

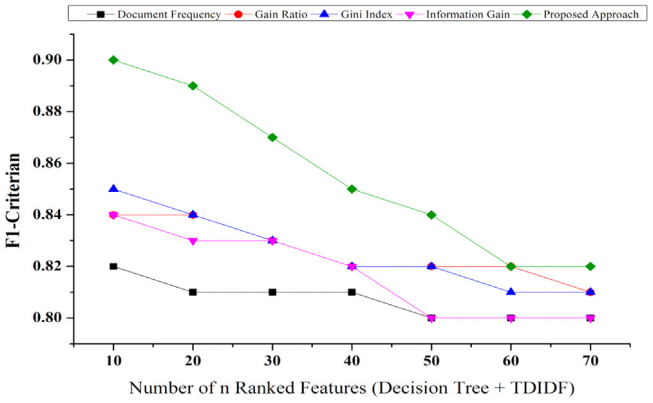


Fig. 4. Performance evaluation of classifiers with feature sets constructed through feature selection method and proposed approach GoF pattern collection.

- We observe the DT (F-measure = 0.80) with its best weighting method TDIDF as outperform classifiers as compared to NB (F-measure = 0.76) and SVM (F-measure = 0.78).

Subsequently, in order to evaluate the effectiveness and improvement (in terms of the F-measure criterion) in the performance of supervised learners used in the proposed framework, we applied the feature selection methods and the proposed approach to construct a more representative feature set. Though, we performed experiments with all supervised learners, we depict the effects of feature selection methods on the performance of outperform supervised learner, that is the Decision Tree (with TDIDF) in Fig. 4. The *x*-axis of Fig. 4 (labeled with Number of Rank Features) refers to top n features ranked by each feature selection method and the proposed approach. The *y*-axis refers to the corresponding F-measure values. The main consequences of the result of Fig. 4 are as follows.

- We observed the significant impact of the feature set constructed through the employed feature selection methods and proposed approach on the performance of classifiers.
- The average improvement in the classification performance of outperform supervised learner (i.e. Decision Tree) ranges from 2.5% to 11.25% (in terms of F-measure) with top 10 features ranked by each feature selection method under study.
- The highest increase in the performance (i.e. 12.5%) of Decision Tree with top 10 features ranked through the proposed approach, which aid us to describe the significance of the proposed approach in order to construct a more representative feature set.

6.2. Organization of Douglass design patterns

The patterns in the Douglass pattern collection are divided into five categories. The preprocessing activities of the proposed approach (Section 3.1) are applied to generate the dataset. We follow the procedure described in pseudocode and use the F-measure criterion (Section 5.2) to find the best weighting method for each supervised learner. According to the F-measure criterion, the evaluation result in the selection of best weighting method for each supervised learner is shown in Fig. 5. Subsequently, the results regarding the best weighting for each classifier are shown in Table 3. The main consequences of the results of Fig. 5 and Table 3 are as follows.

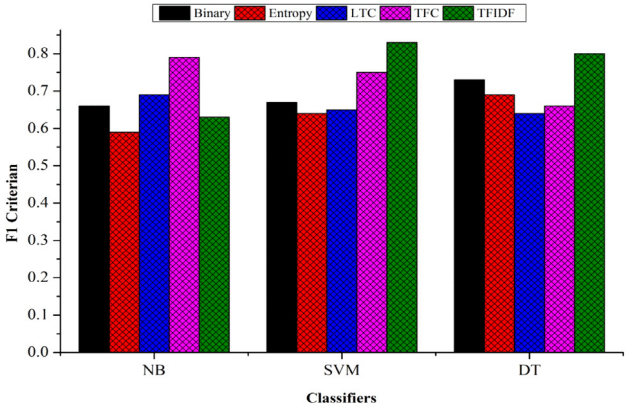


Fig. 5. Performance evaluation of classifiers for the weighting methods on the Douglass Design pattern collection.

- We observe the better performance of NB, SVM, and DT with TFC, TDIDF and TDIDF weighting methods respectively.
- We observe the SVM (F-measure = 0.83) with its best weighting method TDIDF as outperform classifiers as compared to NB (F-measure = 0.79) and DT (F-measure = 0.80).

Subsequently, in order to evaluate the effectiveness and improvement (in terms of the F-measure criterion) in the performance of supervised learners, we applied all feature selection methods and the proposed approach to construct a more representative feature set. Though, we performed experiments with all supervised learners, we depict the effect of feature selection methods on the performance of outperform supervised learner that is SVM (with TDIDF) in Fig. 6. The *x*-axis of Fig. 6 (labeled with Number of Rank Features) refers to top *N* features ranked by each feature selection method and the proposed approach. The *y*-axis refers to the corresponding F-measure values. The main consequences of the result of Fig. 6 are given below.

- We observed the significant impact of the feature set constructed through the employed feature selection methods and proposed approach on the performance of classifiers.
- The average improvement in the classification performance of outperform supervised learner (i.e. SVM) is ranges from 2.40% to 10.84% (in terms of F-measure) with top 10 features ranked by each feature selection method under study.
- The highest increase in the performance (i.e. 10.84%) of SVM with top 10 features ranked through the proposed approach,

Table 3
Performance evaluation of classifiers on different design pattern catalogs.

Pattern catalog (i.e. Dataset)	Classifiers	Best weighting method	F1-criterion
GoF Pattern Catalog	DT	TDIDF	**0.80**
	NB	TFC	0.76
	SVM	TFC	0.78
Douglass Pattern Catalog	DT	TDIDF	0.80
	NB	TFC	0.79
	SVM	TFIDF	**0.83**
Security Pattern Catalog	DT	TFC	0.75
	NB	LTC	0.79
	SVM	TFC	**0.82**

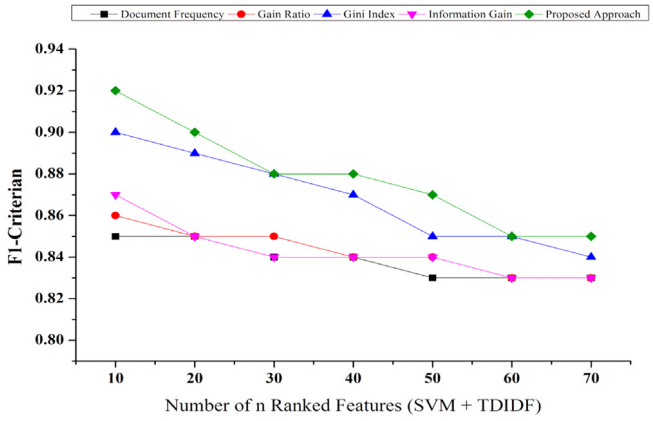


Fig. 6. Performance evaluation of classifiers with the feature sets constructed through feature selection method and proposed approach on Douglass pattern collection.

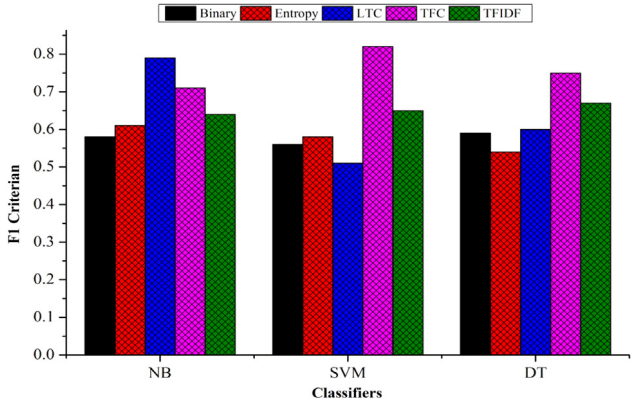


Fig. 7. Performance evaluation of classifiers for the weighting methods on the Security Design pattern collection.

which aid us to describe the significance of the proposed approach in order to construct a more representative feature set.

6.3. Organization of security design patterns

The patterns in the Security pattern collection are divided into eight categories. The preprocessing activities of the proposed approach (Section 3.1) are applied to generate the dataset. We follow the procedure described in pseudocode and use the F-measure criterion (Section 5.2) to find the best weighting method for each supervised learner. According to the F-measure criterion, the evaluation result in the selection of best weighting method for each supervised learner is shown in Fig. 7. Subsequently, the results regarding the best weighting for each classifier are shown in Table 2. The main consequences of the results of Fig. 7 and Table 2 are given below.

- We observe the better performance of NB, SVM, and DT with TFC, TDIDF and TDIDF weighting methods respectively.
- We observe the SVM (F-measure = 0.82) with its best weighting method TFC as outperform classifiers as compared to NB (F-measure = 0.79) and DT (F-measure = 0.75).

Subsequently, in order to evaluate the effectiveness and improvement (in terms of the F-measure criterion) in the performance of supervised learners, we applied all feature selection methods and the proposed approach to construct a more representative feature set. Though, we performed experiments with all supervised learners, we depict the effect of feature selection methods on the performance of outperform supervised learner that is SVM (with TFC) in Fig. 8. The *x*-axis of Fig. 8 (labeled with Number of Rank Features) refers to top N features ranked by each feature selection method and the proposed approach. The *y*-axis refers to

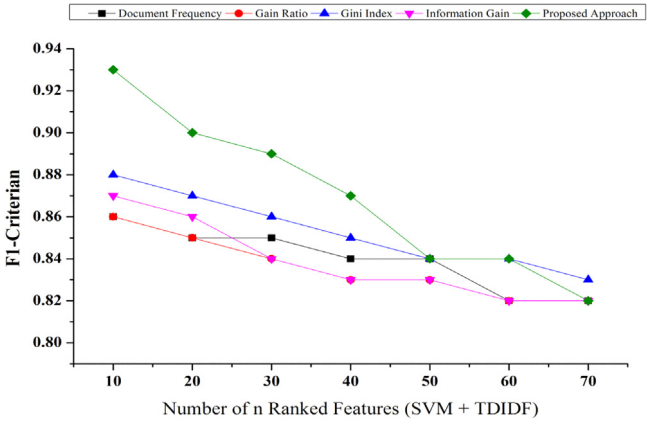


Fig. 8. Performance evaluation of classifiers with the feature sets constructed through feature selection method and proposed approach on Security pattern collection.

the corresponding F-measure values. The main consequences of the result of Fig. 8 are as follows.

- We observed the significant impact of the feature set constructed through the employed feature selection methods and proposed approach on the performance of classifiers.
- The average improvement in the classification performance of outperform supervised learner (i.e. SVM) ranges from 4.80% to 31.41% (in terms of F-measure) with top 10 features ranked by each feature selection method under study.
- The highest increase in the performance (i.e. 13.41%) of SVM with top 10 features ranked through the proposed approach, aid us to describe the significance of the proposed approach in order to construct a more representative feature set.

7. Evaluation summary

The main objective of the proposed framework is to employ a well-known deep learning algorithm to construct a more representative feature set and evaluate its impact on the organization of design patterns. We employed the proposed approach within the context of three widely used design pattern collections. The main consequences of our proposed study are as follows.

- We applied three supervised learning techniques to evaluate the effectiveness of the proposed approach. In the evaluation results, we observe a significant increase in the performance of classifiers for the organization of design patterns. For example, for the organization of Security design patterns, we observe 13.41% (in terms of F-measure) improvement in the actual performance (F-measure = 0.82) of SVM.
- The proposed approach is employed through supervised learning techniques. Consequently, before performing comparative investigations, we used different weighting methods (such as TFIDF, TFC, LTC, Binary, and Entropy) to increase the efficacy of learning techniques. The evaluation results indicate that no single weighting method can be suggested for all supervised learners. For example, in the case of the organization of Douglass and Security design patterns, SVM outperform with different weighting methods, such as TFC for security patterns and TFIDF for the Douglass patterns.
- From the experimental results, we observed the effect of feature selection methods on the performance of supervised learners. However, it depends on the target pattern catalog. For example, the improvement in the performance of outperform classifier SVM is 10.84% and 13.41% for Douglass and Security pattern collection respectively.
- We observed the significant increase in the performance of classifiers with feature set constructed through the proposed approach as compared to feature sets constructed through other feature selection methods. For example, in case of Security patterns, the performance of SVM increases up to 13.41% with feature set constructed through the proposed approach.

8. Threats to validity

In our study, we also find some threats. The first threat is related to the generalization of experimental results. We performed experiments and concluded the results with a limited number of pattern collection and classifiers. In order to generalize the applicability of proposed approach, we need to extend our work with a large number of datasets and classifiers. The second threat is related to use of the default values adjusted for the parameter of DBN. The effectiveness of proposed approach can be improved by tuning the parameters of DBN, such as a number of hidden layers, nodes and iterations. The third threat is related to the use of performance measure in order to assess the effectiveness of the proposed approach. We used only micro-F1 measure, while macro-F1 measure can be used to evaluate the effectiveness of the proposed approach in terms of class discrimination.

9. Conclusion

The experimental results illustrate that the proposed approach can be applied to construct a more illustrative feature set via a deep learning named Deep Belief Network (DBN) and aid to improve the classification performance. We leveraged a DBN algorithm to construct a more illustrative feature set on the basis of semantic relationship between pattern across the collection. The aim of the proposed approach is to improve the classifier's performance in order to organize the design patterns. We evaluate the effectiveness of the proposed approach with three widely-used design pattern collections and three well-known classifiers Decision Tree (DT), Naïve Bayes (NB) and Support Vector Machine (SVM). The experimental results indicate the applicability of the proposed approach in order to improve the performance of classifiers to organize the design patterns. However, the effectiveness of proposed approach can be increased by tuning the parameters of the leveraged DBN algorithm. In the future work, we will empirically investigate the applicability of the proposed approach to handle the multi-class problem by tuning the DBN parameters.

Acknowledgments

This work is supported in part by the General Research Fund of the Research Grants Council of Hong Kong (No. 11208017 and 11214116), the research funds of City University of Hong Kong (No. 7004683 and 7004474), and the NRF of Korea Grant funded by the Korean Government (2015R1D1A1A01058171).

Appendix

A.1. Naïve Bayes (NB)

The Naïve Bayes (NB) is a probabilistic classifier which is based on Bayes theorem. Naïve Bayes works on an assumption that all features are independent of each other and calculate a probability score by multiplying the conditional probabilities with each other. It is supposed that the pattern class $L(p_i)$ of pattern p_i, has some relation to words appearing in the description of Problem Definition of design patterns and design problems. The Bayesian formula describes in Eq. (5) yields the probability of a pattern class given the words used in the description of patterns and design problem.

$$p\left(L_i|t_1,\ldots,t_{m_i}\right)=\frac{p\left(t_1,\ldots t_{m_i}|L_k\right)p(L_k)}{\sum_{l\in L}p\left(t_1\ldots t_{m_i}|l\right)p\left(l\right)}. \tag{6}$$

The prior probability $p(l)$ depicts the probability that design problem belongs to a class label $l \in L$ before its known words. The conditional probability of words (includes in the description of patterns and design problem) given a class L_k is described in Eq. (5).

$$p\left(t_1,\ldots,t_{m_i}|L_k\right)=\prod_{j=1}^{m_i}p\left(t_j|L_k\right). \tag{7}$$

The estimation of the probabilities of words $p(t_j|L_k)$ is required to make the learning of Naïve Bayes classifier which depends on their frequencies in documents (i.e. Design patterns) of the training set.

A.2. C4.5 decision tree

The decision tree based algorithms generate a set of rules in order to make the classification decisions. The C4.5 algorithm used the concept of information entropy and makes the decision from the training set. In order to predict the document's class, the word t_i is selected from a training set T of labeled documents. Subsequently, training set T is partitioned into two subsets $T+$ (documents with word t_i) and $T-$ (documents without the word t_i). The same step is repeated and applied to $T+$ and $T-$. The recursive procedure is stopped when all documents belong to the same class L_k.

A.3. Support Vector Machines (SVM)

A Support Vector Machine (SVM) is considered as one of the most effective text categorization algorithms, which looks for a decision surface and determines a margin through its closest data points. Usually, an SVM algorithm can separate documents of a training set into two classes, $y = +1$ for the positive and $y = -1$ for the negative class. Subsequently, Eq. (8) is used to define a hyper plane at $y = 0$ for the set of input vectors. Each input vector for the document d is represented as a count of words as depicted in Eq. (7).

$$\vec{t_d} = t_{d1}, \ldots, t_{\mathrm{dM}}. \tag{8}$$

$$y = f(\vec{t_d}) = b_0 + \sum_{j=1}^{M} b_j t_{\mathrm{dj}}. \tag{9}$$

The SVM algorithm classifies a new document with $\vec{t_d}$ into a positive class if $f(\vec{t_d}) > 0$ otherwise into negative class.

References

[1] T. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[2] A. Birukou, A survey of existing approaches for pattern search and selection, in: Proceeding of PLoP, 2010.
[3] G. Booch, Handbook of Software Architecture, 2006. http://handbookofsoftwarearchitecture.com/.
[4] C. Bouhours, H. Leblance, C. Percebois, Spoiled patterns: How to extend the GoF, Softw. Qual. J. 23 (2015) 661–694.
[5] P. Coad, D. North, M. Mayfield, Object Models: Strategies, Patterns, Applications, Yourdon Press, NJ, USA, 1995.
[6] B.P. Douglass, Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, Addison-Wesley/Longman Publishing Co. Inc., Boston, MA, USA, 2002.
[7] C. Edwards, Growing pains for deep learning, Commun. ACM 58 (7) (2015). http://dx.doi.org/10.1145/2771283.
[8] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (2003) 1289–1305.
[9] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Boston, MA, 1995.
[10] S. Gunal, Hybrid feature selection for text classification, Turk. J. Electr. Eng. Comput. Sci. 20 (2012) 1296–1311.
[11] S. Hasso, C.R. Carlson, A theoretically-based process for organizing design patterns, in: Proceedings of 12th Pattern Language of Patterns, 2005.
[12] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[13] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[14] A. Hotho, A. Nurnberger, G. Paab, A brief survey of text mining, J. Comput. Linguist. Lang. Technol. 20 (2005) 19–62.
[15] S. Hussain, J. Keung, A.A. Khan, Software design patterns classification and selection using text categorization approach, Appl. Soft Comput. (2017). http://dx.doi.org/10.1016/j.asoc.2017.04.043.
[16] S. Hussain, J. Keung, A.A. Khan, K.E. Bennin, A methodology to automate the selection of design patterns, in: Proceedings of International Conference on Computers, Software and Application, IEEE, 2016.
[17] I. Idris, A. Selamat, Improved email spam detection model with negative selection algorithm and particles warm optimization, Appl. Soft Comput. 22 (2014) 11–27.
[18] G.J. Kim, J.S. Han, Clustering algorithm of design pattern using object-oriented relationship, in: Proceeding of Computational Science and Its Applications, ICCSA, in: LNCS, Springer, 2007, pp. 997–1006.
[19] D.K. Kim, C.E. Khawand, An approach to precisely specifying the problem domain of design patterns, J. Vis. Lang. Comput. 18 (2007) 560–591.
[20] D.K. Kim, W. Shen, Evaluating pattern conformance of UML models: A divide and conquer approach and case studies, Softw. Qual. J. 16 (3) (2008) 329–359.
[21] A. Lam, A. Nguyen, H. Nguyen, T. Nguyen, Combining deep learning with information retrieval to localize buggy files for bug reports, in: Proceedings of 30th IEEE ASE Conference, pp. 476–481.
[22] C. Lee, G.G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Inf. Process. Manage. 42 (1) (2006) 155–165.
[23] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Eng. J. 5 (2014) 1093–1113.
[24] H. Ogura, H. Amano, M. Kondo, Distinctive characteristics of a metric using deviation from poisson for feature selection, J. Expert Syst. Appl. 37 (2010) 2273–2281.
[25] R. Pascanu, J.W. Stokes, H. Sanossian, M. Marinescu, A. Thomas, Malware classification with recurrent networks, in: Proceeding of ICASSP, 2015, pp. 1916–1920.
[26] M.F. Porter, An algorithm for suffix stripping, J. Program Electron. Libr. Inf. Syst. 40 (2006) 211–218.
[27] W. Pree, Design Patterns for Object-Oriented Software Development, ACM Press/Addison-Wesley, 1995.
[28] L. Rising, The Pattern Almanac 2000, Addison-Wesley, 2000.
[29] E. Sarac, S.A. Ozel, An ant colony optimization based feature selection for web page classification, 2014, pp. 1–16.
[30] M. Schumacher, E. Fernandez, D. Hybertson, F. Buschmann, Security Patterns: Integrating Security and Systems Engineering, John Wiley and Sons, 2006.
[31] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Syst. Appl. 33 (1) (2007) 1–5.
[32] C. Tantithamthavorn, S. Mclntosh, A.E. Hassan, K. Matsumoto, An empirical comparison of model validation techniques for defect prediction models, IEEE Trans. Softw. Eng. PP (99) (2016).
[33] S. Tascı, T. Güngör, Comparison of text feature selection policies and using an adaptive framework, Expert Syst. Appl. 40 (2013) 4871–4886.
[34] W.F. Tichy, A catalogue of general-purpose software design patterns, in: Proceedings of Technology of Object-Oriented Languages and Systems, 1997, pp. 330–339.
[35] P.D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, J. Artif. Intell. Res. 37 (2010) 141–188.
[36] H. Uguz, A two-stage feature selection method for text classification by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (7) (2011) 1024–1032.
[37] A.K. Uysal, An improved global feature selection scheme for text classification, Expert Syst. Appl. 43 (2016) 82–92.
[38] P. Velasco-Elizondo, R. Marín-Piña, S. Vazquez-Reyes, A. Mora-Soto, Mejia, Knowledge representation and information extraction for analyzing architectural patterns, Sci. Comput. Programming 121 (2016) 176–189.
[39] S. Wang, T. Liu, L. Tan, Automatically learning semantic features for defect prediction, in: Proceeding of IEEE International Conference on Software Engineering, ICSE, 2016.
[40] M. White, C. Vendome, M.L. Vasquez, D. Poshyvanyk, Toward deep learning software repositories, in: Proceedings of MSR'15, 2015, pp. 334–345.
[41] X. Yang, D. Lo, X. xia, Y. Zhang, J. Sun, Deep learning for just-in-time defect prediction, in: Proceedings of QRS, 2015, pp. 17–26.
[42] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412–420.
[43] C. Zhang, X. Wu, Z. Niu, W. Ding, Authorship identification from unstructured texts, Knowl. Based Syst. 66 (2014) 99–111.
[44] W. Zimmer, Relationships between design patterns, J. Pattern Lang. Program. Des. 1 (1995) 345–364.

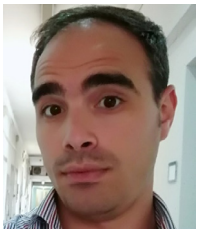
Shahid Hussain received his M.Sc. (Computer Science) and M.S. (Software Engineering) degrees from Gomal University, DIK and City University of Science and Information Technology (CUSIT), Peshawar, Pakistan. Currently, he is pursuing Ph.D. (Software Engineering) with the Department of Computer Science, City University of Hong Kong. His research interests include the Software Design patterns and metrics, text mining, empirical studies, and software defect prediction. Mr. Shahid Hussain is a student member of ACM and IEEE.

Jacky Keung received his B.Sc. (Hons) in Computer Science from the University of Sydney, and his Ph.D. in Software Engineering from the University of New South Wales, Australia. He is Assistant Professor in the Department of Computer Science, City University of Hong Kong. His research interests include software effort and cost estimation, empirical modeling and evaluation of complex systems, and intensive data mining for software engineering datasets. He has published papers in prestigious journals including IEEE-TSE, IEEE-SOFTWARE, EMSE and many other leading journals and conferences.

Arif Ali Khan received his B.S. in Software Engineering from University of Science and Technology Bannu, Pakistan in 2010. Arif has M.Sc. by research in Information Technology from Universiti Teknologi PETRONAS, Malaysia. He is currently pursuing Ph.D. degree with the Department of Computer Science, City University of Hong Kong. He is an active researcher in the field of empirical software engineering. His is interested in Software Process Improvement, 3C's (Communication, Coordination, Control), Global Software Development and Systematic Reviews. He has participated in and managed several software Engineering related research projects. He is a student member of IEEE and ACM.

Awais Ahmad received his B.S. in Computer Science and M.S. in Telecommunication and Networking from University of Peshawar and Bahria University Islamabad in 2008 and 2010, respectively. During his masters research work, he worked on energy efficient congestion control schemes in Mobile Wireless Sensor Networks (WSN). In 2011, he was appointed as a Lecturer in Comsats Institute of Information Technology, Islamabad. In the mid of 2012, he moved to Republic of Korea for his Ph.D. course at Kyungpook National University (KNU), Daegu. Throughout his time at KNU, as a student and researcher in the field of Big Data, Social Internet of Things, Internet of Things, Vehicular Communication, WSN, and Machine-to-Machine Communication. Dr. Awais has been a key contributor in the said fields. He was also serving as a Lab Admin of CCMP Labs since 2013. He was also awarded as a Best Outgoing Researcher of CCMP labs. Dr. Awais was awarded a Ph.D. in February 2017. After his graduations, he was selected as an Assistant Professor in the Department of Information and Communication Engineering, Yeungnam University Korea. He serves as Guest Editor in various Elsevier and Springer Journals. He is an invited reviewer in IEEE Communication Letters, IEEE JSTAR, IEEE Transactions on Intelligent Transportation Systems, and several other IEEE and Elsevier Journals. He has also published more than 65 research papers (Journals and conferences) and also several book chapters related to big data and IoT. Dr. Ahmad was the recipient of three prestigious awards: (1) Research Award from President of Bahria University Islamabad, Pakistan in 2011, (2) best Paper Nomination Award in WCECS 2011 at UCLA, USA, and (3) best Paper Award in 1st Symposium on CS&E, Moju Resort, South Korea in 2013.

Salvatore Cuomo is an Assistant Professor in Numerical Analysis at University of Naples Federico II with interests in several Applied Mathematics topics. More in detail, he deals with: (i) numerical approximation problems (theory, practice and applications); (ii) parallel, GPU and scientific computing issues; (iii) Inverse problems. Finally, his passion is the technology transfer in data processing context and smart education environments. He is a co-founder of Educabile start-up innovative company.



Francesco Piccialli is a researcher at the University of Naples "Federico II", Department of Mathematics and Applications "Renato Caccioppoli", Italy. His research topics are Smart Environment design, Internet Technologies, Data Mining on Internet of Things systems with application in the Smart City framework and the Cultural Heritage domain.

Gwanggil Jeon received the B.S., M.S., and Ph.D. (summa cum laude) degrees from the Department of Electronics and Computer Engineering from Hanyang University, Seoul, Korea, in 2003, 2005, and 2008, respectively. From 2008 to 2009, he was with the Department of Electronics and Computer Engineering, Hanyang University, from 2009 to 2011, he was with the School of Information Technology and Engineering (SITE), University of Ottawa, as a postdoctoral fellow, and from 2011 to 2012, he was with the Graduate School of Science & Technology, Niigata University, as an Assistant Professor. He is currently an Associate Professor with the Department of Embedded Systems Engineering, Incheon National University, Incheon, Korea. His research interests fall under the umbrella of image processing, particularly image compression, motion estimation, demosaicking, and image enhancement as well as computational intelligence such as fuzzy and rough sets theories.

Adnan Akhunzada is currently an Assistant Professor at CIIT Islamabad. He received his Ph.D. from University of Malaya, Malaysia. He had a great experience in teaching international modules of the University of Bradford, United Kingdom. He has published several high impact research journal papers. His current research interests include secure design and modeling of software defined networks, man-at-the-end attacks, lightweight cryptography, human attacker attribution and profiling, and remote data auditing.