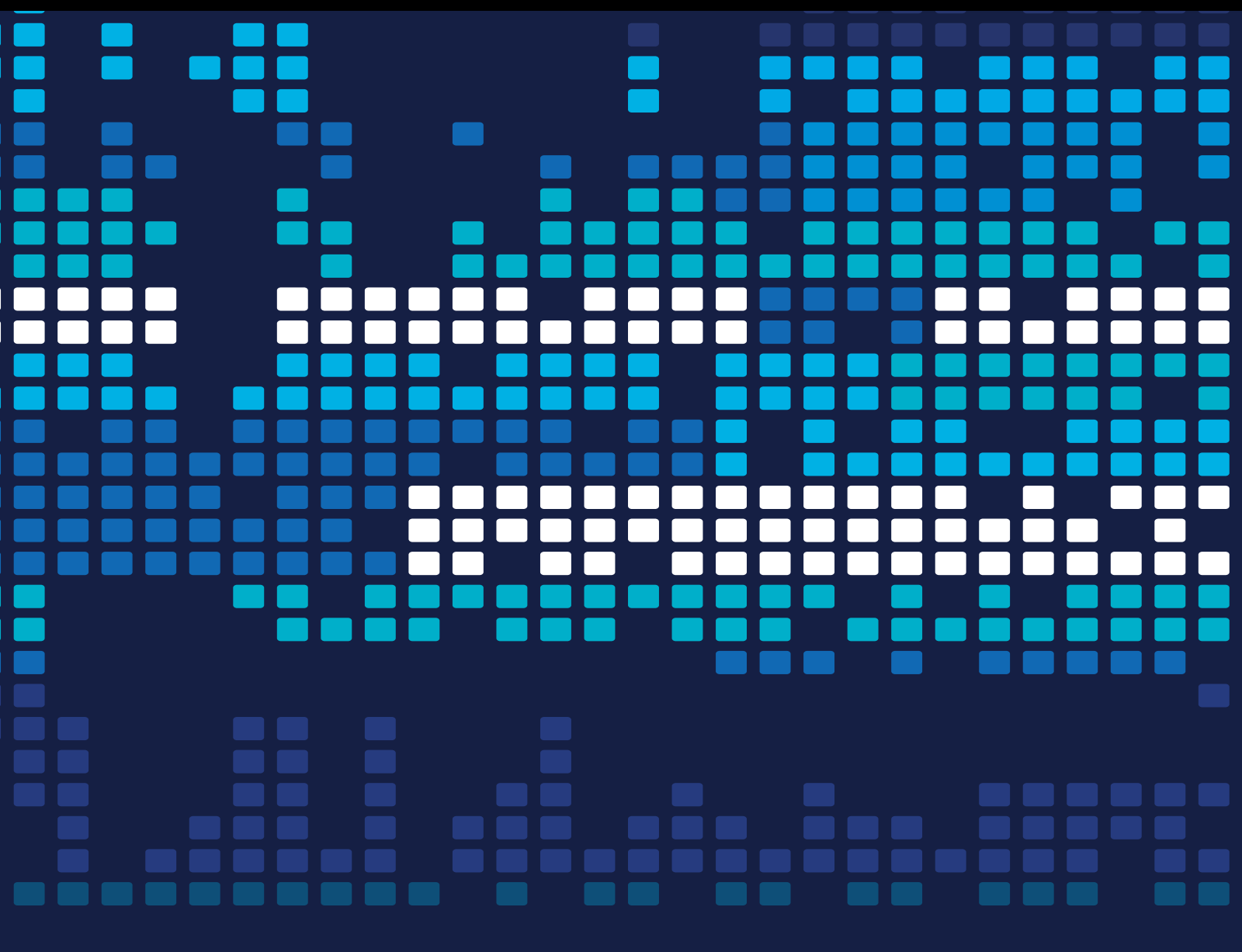
Scientific Programming

Novel Advances in the Development of Machine Learning Solutions for Scientific Programming

Lead Guest Editor: Vicente García-Díaz
Guest Editors: Edward R. Núñez-Valdez, Vijender K. Solanki, and Carlos Enrique Montenegro-Marin



Novel Advances in the Development of Machine Learning Solutions for Scientific Programming

Scientific Programming

Novel Advances in the Development of Machine Learning Solutions for Scientific Programming

Lead Guest Editor: Vicente García-Díaz
Guest Editors: Edward Rolando Núñez-Valdez,
Vijender K. Solanki, and Carlos Enrique Montenegro-Marin

Editorial Board

Contents

Hindawi
Scientific Programming
Volume 2019, Article ID 7896462, 2 pages
https://doi.org/10.1155/2019/7896462

Editorial

Novel Advances in the Development of Machine Learning Solutions for Scientific Programming

Vicente García-Díaz,[1] **Edward R. Núñez-Valdez**,[1] **Vijender K. Solanki**,[2] **and Carlos E. Montenegro-Marin**[3]

[1]*University of Oviedo, Oviedo, Spain*
[2]*CMR Institute of Technology, Hyderabad, India*
[3]*Universidad Distrital Francisco José de Caldas, Bogotá, Colombia*

Correspondence should be addressed to Vicente García-Díaz; garciavicente@uniovi.es

Received 26 May 2019; Accepted 26 May 2019; Published 1 July 2019

Scientific programming is a multidisciplinary field that uses advanced methods to understand and solve complex problems. Meanwhile, machine learning is the field that uses statistical techniques to give computer systems the ability to learn and extract knowledge from data, answering questions, and solving problems in various application domains, without the need of being explicitly programmed. Both are exciting, complex, and interrelated fields in which advances are taking place at a great pace.

This special issue aims to bring together some of the most important advances related to the union of scientific programming and progresses in the field of machine learning. Thus, after a rigorous selection process, this number includes 9 works that allow us to continue exploring the possibilities that machine learning can provide, particularly for scientific programming.

The first paper is entitled "A 64-Line Lidar-Based Road Obstacle Sensing Algorithm for Intelligent Vehicles" by H. Wang et al. In this work, the authors present a novel approach for road obstacle sensing through an effective and real-time-based algorithm. This work provides contributions to improve the safety of drivers using 64-line lidar sensors and classifiers based on support vector machines. The focus is on detecting obstacles with clustered object positions and specific features.

The second paper is entitled "A Low Cost Named Entity Recognition Research Based on Active Learning" by H. Huang et al. In this case, the authors propose advances in the field of natural language processing. They focus on named entity recognition by using active learning together with the conditional random field classifier, which serves to improve its performance. To this end, the authors apply clusters created with the *K*-means method. The testing data include Chinese judicial documents and Chinese electronic medical records.

The third paper is entitled "Design and Implementation of a Machine Learning-Based Authorship Identification Model" by W. Anwar et al. The paper focuses on authorship identification in English and Urdu languages using a latent Dirichlet allocation model. The presented approach is an unsupervised computational methodology that can handle the heterogeneity of datasets, diversity in writing, and the inherent ambiguity of the Urdu language. Finally, the authors used a big corpus to test the performance of the presented approach.

The fourth paper is entitled "A non-disturbing service to automatically customize notification sending using implicit-feedback" by F. López Hernández et al. The work addresses the problem of automatically customizing notifications in a nondisturbing way. The idea is to use implicit feedback in order not to ask preferences directly to users. To that end, a hybrid filter that combines both content and collaborative filtering to predict the best notifications for users in each concrete situation is also presented.

The fifth paper is entitled "Consensus Clustering-Based Undersampling Approach to Imbalanced Learning" by A. Onan. In this work, the author works on class imbalance, which is one important issue in machine learning. He presents a consensus clustering-based undersampling approach to imbalanced learning. The

work also shows an empirical analysis with 33 small-scale and 2 large-scale imbalanced classification benchmarks together with 5 clustering algorithms and their combinations, also with supervised learning methods.

The sixth paper is entitled "Towards the Construction of a User Unique Authentication Mechanism on LMS Platforms through Model-Driven Engineering (MDE)" by J. F. Herrera-Cubides et al. In this case, the authors work on authentication issues. They propose a security abstraction model on learning management systems based on model-driven engineering. To that end, the authors presented a metamodel with a set of guidelines on how to carry out authentication considering the different involved stakeholders.

The seventh paper is entitled "Design and Development of a Large Cross-Lingual Plagiarism Corpus for Urdu-English Language Pair" by I. Haneef et al. In this work, the authors focus on cross-lingual plagiarism that detects plagiarism of a text even if it is written in different languages by checking semantic similarities. The paper is intended to present the results working with English and Urdu languages. To meet the goals, the authors have created a corpus with 2398 source pairs, which is publicly available for research purposes.

The eighth paper is entitled "Avionics Graphics Hardware Performance Prediction with Machine Learning" by S. R. Girard et al. The authors create a system design tool to help predict the rendering performance of graphical hardware based on the OpenGL library. That is, they propose to replace expensive alternatives by a predictive software application running on a desktop computer. In addition, the authors render an industrial scene with features not used during the training phase, getting an error of less than 4 frames per second.

The last paper is entitled "Effects of challenging weather and illumination on learning-based License Plate Detection in non-controlled environments" by A. Rio-Alvarez et al. The authors work on automatic license plate recognition systems focusing on license plate detention. They study the differences in the effectiveness of different descriptors for concrete weather conditions. They state that images affected by raining should not be included in training sets, but images affected by low illumination conditions should be included because they increase performance.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The guest editors would like to thank all the authors for contributing their efforts and all the reviewers for their time and rigorous reviews, making this special issue possible.

Vicente García-Díaz
Edward R. Núñez-Valdez
Vijender K. Solanki
Carlos E. Montenegro-Marin

Hindawi
Scientific Programming
Volume 2019, Article ID 6897345, 16 pages
https://doi.org/10.1155/2019/6897345

Research Article

Effects of Challenging Weather and Illumination on Learning-Based License Plate Detection in Noncontrolled Environments

A. Rio-Alvarez, J. de Andres-Suarez, M. Gonzalez-Rodriguez, D. Fernandez-Lanvin, and B. López Pérez

Faculty of Computer Science, University of Oviedo, Oviedo, Spain

Correspondence should be addressed to A. Rio-Alvarez; delrioalvarez@gmail.com

Received 21 December 2018; Revised 30 March 2019; Accepted 14 May 2019; Published 1 July 2019

Guest Editor: Vijender K. Solanki

License Plate Detection (LPD) is one of the most important steps of an Automatic License Plate Recognition (ALPR) system because it is the seed of the entire recognition process. In indoor controlled environments, there are many effective methods for detecting license plates. However, outdoors LPD is still a challenge due to the large number of factors that may affect the process and the results obtained. It is an evidence that a complete training set of images including as many as possible license plates angles and sizes improves the performance of every classifier. On this line of work, numerous training sets contain images taken under different weather conditions. However, no studies tested the differences in the effectiveness of different descriptors for these different conditions. In this paper, various classifiers were trained with features extracted from a set of rainfall images using different kinds of texture-based descriptors. The accuracy of these specific trained classifiers over a test set of rainfall images was compared with the accuracy of the same descriptor-classifier pair trained with features extracted from an ideal conditions images set. In the same way, we repeat the experiment with images affected by challenging illumination. The research concludes, on one hand, that including images affected by rain, snow, or fog in the training sets does not improve the accuracy of the classifier detecting license plates over images affected by these weather conditions. Classifiers trained with ideal conditions images improve the accuracy of license plate detection in images affected by rainfalls up to 19% depending on the kind of extracted features. However, on the other hand, results evidence that including images affected by low illumination regardless of the kind of the selected feature increases the accuracy of the classifier up to 29%.

1. Introduction

This research work is aimed at studying the effect of two important issues for outdoor Automatic License Plate Recognition (ALPR) systems as the rainfalls (rain, fog, and snow) and the lack of light over the training stage of these ALPR systems. Our main goal is obtaining new information for improving the composition of training sets, achieving valid conclusions for every kind of scenario while being compatible with other image-processing techniques.

ALPR systems have played a prominent role in the literature over recent years due to their popular application in real-life scenarios like automatic coin collectors in tolls, supervision of traffic regulation, parking access, or traffic control, among others. However, efficiency of such approaches is usually limited to specific or controlled scenarios.

Figure 1 shows the traditional stages of any ALPR system. These stages are common both in controlled and uncontrolled environments. However, the algorithms that must be applied in each of these stages should be adapted to the particular conditions of the environment. The less controlled the environment, the more difficult the challenges faced by the ALPR system.

License Plate Detection (LPD) is the first stage of any ALPR system. A complete image or video frame is taken as

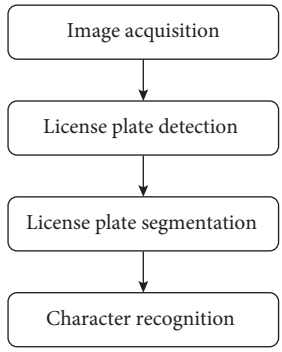


Figure 1: Stages of an ALPR system.

input. The output is the set of Regions of Interest (ROIs) that potentially contain a license plate. Therefore, LPD comprises two phases: license plates localization and ROIs cropping. The efficacy of the LPD significantly determines the accuracy of the entire ALPR system. Moreover, it is the most time-consuming stage.

LPD systems can be broadly categorized into two groups: those based on boundary/edge information and character detection and those based on Machine-Learning (ML) algorithms working on local features, mainly boosting-based approaches. It is also important to mention another kind of LPD systems based on the location of specific areas of the vehicles close to the license plates (like for example the braking lights). In any case, these context-aware methods are based on one of the previous alternatives [1].

When the ALPR works in a controlled environment, the region of the image where license plates can appear, the license plate angle or its size is usually bounded. In addition, many indoor scenarios such as parking accesses, lighting conditions, and other meteorological factors can also be controlled. In these scenarios, LPD methods based on edge detections and morphological operations can achieve good performance. These approaches are intuitive and powerful in scenarios where license plates are not noisy [2].

Nevertheless, when the ALPR works in an uncontrolled environment, no prior information can be used to support the detection process. The license plate can appear in any region of the image, and the detection algorithms usually require an approach based on ML algorithms with a training stage where all possible angles and sizes of license plates are taken into consideration. Traditionally, this kind of approaches have handled the problem of the angle and regional variations using a learning-based algorithm and including sufficient variety of images in the training set. When dealing with the issue of the scale, these systems use to sequentially apply single-scale classifiers over a pyramid of images. But, in outdoor environments, training can be also determined by environmental factors such as lighting and meteorology conditions.

An appropriate selection of the kind of descriptors is a determinant step for ML-based approaches. In the same way, the selection of images that will be used for the training is one of the most important steps of the entire process. In this paper, we evaluate the influence in the training process of the weather and the challenging illumination that occur outside uncontrolled environments. It is reasonable to think that a complete set of training images including light variations and different weather conditions would improve the accuracy of the classifier, especially if the system will be used in areas with frequent rainfalls. In this research, we wonder under which conditions this affirmation is true. To answer this question, we check the accuracy of different classifiers trained with images captured in optimal conditions and compare it with those of the same classifiers but trained with images affected by challenging weather or low illumination.

This research includes the testing of commonly used texture features such as Histogram of Oriented Gradients (HOG) [3], Local Binary Pattern (LBP) [4], and Haar-like features [5] in combination with a boosted cascade and a Support Vector Machine (SVM) in order to consider traditional object detection algorithms such as the algorithm of Viola and Johns [5], the HOG-based approach proposed by Dalal and Triggs [3], and the approach based on LBP features proposed by Ojala et al. [4]. In addition to the chosen representative descriptors, we consider various texture-based variants such as the combination HOG-LBP [6], Local Gradient Patterns (LGPs) [7], Multi-Block Local Binary Patterns (MB-LBPs) [8], Compound Local Binary Patterns (CLBPs) [9], Local Ternary Patterns (LTPs) [10], or features extracted from the Gray-Level Co-occurrence Matrix (GLCM) [11].

In addition, and in order to check the robustness of our research, we repeated our tests using three more classifiers: K-Nearest Neighbor (KNN), an Artificial Neural Network (ANN), and a Linear Regression (LR) approach.

The rest of the paper is organized as follows. In Section 2, we briefly review the related literature. Section 3 describes the methods and algorithms tested in this research and presents our proposed experiment in detail. The results of the experiment are discussed in Section 4, and finally, we conclude the paper in Section 5, explaining the limitations and future work in Section 6.

2. Related Work

Challenging weather and difficult illumination conditions are important issues that should be taken into consideration by every LPD system in outside uncontrolled environments.

Most of the existing LPD methods do not consider input images having challenging illumination. Any methods proposed a contrast enhancement step in their LPD step [12, 13] using techniques as the fuzzy-based contrast enhancement technique proposed by Raju and Nair [14] or the improved methods proposed by Xue et al. [15]. But these kinds of methods are not an effective technique for highly low contrast regions as night images. Several others consider uneven illumination and other low-contrast issues [16, 17] but do not consider all challenging illumination conditions as a great lack of light that happens at night in outside environments. The use of special hardware like IR cameras is a common solution utilized for many methods for detecting license plates at night time [18, 19].

In addition, preprocessing techniques based on Weber's Law can reduce the luminance effect and the high-intensity impulse noise [20] providing a better detection against illumination variation. In the same way, these techniques can be used in the description of the images improving the performance of object detection techniques, like WLD [21] that is a texture descriptor based on Weber's Law that presents robustness to noise and illumination changes.

Respecting the challenging weather, in 2016, Azam and Islam [22] considered that none of the existing LPD methods until then were able to handle the issue of weather condition. For support this assertion, they provide a table that summarizes the most important LPD techniques with their limitations from the view of hazardous conditions. Since then, few approaches have been taken into account in this issue. Azam and Islam themselves [22] presented a LPD system in hazardous conditions. This approach includes a novel method that uses a frequency domain mask to filter rain streaks from an image for rain removal.

Recently, Panahi and Gholampour [23] presented a complete ALPR system capable of detecting and recognizing Persian license plates obfuscated by stains and several levels of dirtiness numbers in different kind of scenarios, variable weather and illumination. This approach is also assisted by a monochrome camera and an IR projector for plate detection and achieves a 98.7%, 99.2%, and 97.6% accuracies for plate detection, character segmentation, and plate recognition, respectively.

Raghunandan et al. [19] recently proposed a novel mathematical model based on Riesz fractional operator for enhancing details of edge information in license plate images. Performing this operation on each input image allows to improve the quality of the images affected by multiple factors after applying detection and recognition methods.

The approaches listed above face the problem of challenging weather and illumination by different preprocessing techniques and specialized hardware. Otherwise, there are not researches about how diverse weather affects the training using different descriptors. In the training process, several approaches simply incorporate different weather and illumination conditions in its datasets. In the presented research, we wonder if any descriptor is capable to describe correctly the challenging weather for using this information in the detection process. In the same way, we analyze the influence of challenging illumination in the same descriptors.

3. Materials and Methods

He et al. in [24] consider Haar-like, HOG, and LBP as highly representative descriptors for license plate detection because Haar-like and LBP features are appropriate to represent character corners, while HOG features are suitable to represent outlines, such as horizontal and vertical relation of characters. For this reason, they proposed a fusion of LBP and HOG features as a suitable descriptor for representing license plates.

To be able to study the influence of weather conditions and illumination changes over representative image descriptors, we decided considering these three descriptors, motivated by the reasons given by He et al. [24] and supported by an extensive literature [10–18]. In addition, we consider various improved variants of the aforementioned representative descriptors proposed recently in the literature like Local Gradient Patterns (LGPs), Multi-Block Local Binary Patterns (MB-LBPs), Compound Local Binary Patterns (CLBPs), the combination of HOG and LBP (HOG-LBP) or Local Ternary Patterns (LTPs), and features extracted from the classical Gray-Level Co-occurrence Matrix (GLCM), one of the earliest techniques used for image texture analysis. Furthermore, we expect that descriptors based on the same kind of feature perform similar behaviours.

Color-based descriptors are also widely used in LPD but are not considered for the present research. Since some countries have specific colors for their license plates, the main idea of the color-based LPD methods is the location of this color pattern in the image, for example, using the blue rectangle that appears on the left side of European license plates. This kind of methods, in addition to be closely associated to certain kind of license plate, are not taken into consideration for this paper due to the fact that color features are sensitive to illumination variations, so some approaches also require special lighting [25], and they are not considered a good option for uncontrolled environments.

Each descriptor was trained using the original classifier proposed in the literature. These combinations feature-classifier are given in Table 1 marked with a check (✓). In addition to these original combinations, three more classifiers (A Neural Network, K-Nearest Neighbor, and a Linear Regression) are included in this research in order to support the results obtained by the originals combinations.

SVM has been taken as the reference classifier for training the other descriptors due to its widespread use for texture classification. In addition, to ensure the robustness of the experiment, we repeated the tests of each variant using the above-mentioned three additional classifiers.

For each combination of descriptor-classifier, one "generic" classifier was trained with a set of images obtained in ideal conditions (without challenging weather and adequate lighting) and another "specific" one was trained with a set of images affected by rainfalls (heavy rain, snow, or fog). Both classifiers were tested over another set of images affected by rainfalls with the goal of comparing the effectiveness of both classifiers detecting license plates under these challenging conditions. The objective is to determine which kind of descriptor can adapt better to challenging weather and how could we increase the performance of each descriptor by an appropriate selection of images for the training sets.

In the same way, we repeat the experiment using images affected by challenging illumination. One generic classifier and another specific classifier (trained with images taken without an adequate lighting) were trained for each combination descriptor-classifier and testing with a set of images taken in poor illumination conditions. The goal is the same—obtain information about how each descriptor is affected by low illumination and improve our knowledge about the composition of training sets.

Table 1: Original combinations of features and classifiers.

Features	SVM	AdaBoost (cascaded)
Haar-like		✓
HOG	✓	
LBP	✓	

3.1. Feature Extractors and Classifiers

3.1.1. Histogram of Oriented Gradients (HOG). HOG descriptor was introduced by Dalal and Triggs [3] in 2005 and is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid.

The image is divided into blocks which also consist of several cells. For each cell, a histogram that summarizes the gradient direction of each pixel is calculated. The traditional steps of this process are shown in Figure 2.

Every histogram is concatenated into one vector. This vector is a HOG descriptor of the image. Modifying the size of cells and blocks, it is possible to adapt the descriptor to different scales.

Figure 3(b) is the visualization of a HOG descriptor of the input image (Figure 3(a)) calculated using $8 * 8$ pixels per cell, the descriptor shown in Figure 3(c) is calculated using $16 * 16$ pixel per cell, and finally, the descriptor shown in Figure 3(d) is calculated using $32 * 32$ pixel per cell. This kind of visualization shows for each cell the visual representation of the gradient vectors that are summarized into its histogram.

The classifier selected by Dalal and Triggs it is a SVM. SVM is a supervised machine-learning algorithm which can be used for both classification and regression challenges. SVM was developed by Vladimir Vapnik and Alexey Ya. Chervonenkis in 1963 and with further improvements was published in 1995 by Cortes and Vapnik [26].

SVMs are binary classifications systems, only two classes are considered. In the case of LPD, these classes are License Plate (LP) and Not License Plate (NLP). In their simplest form, SVMs are hyperplanes that separate the training data by a maximal margin. The training samples that lie closest to the hyperplane are called support vectors.

In other words, given the training data $\{x_1, \ldots, x_l\}$, SVM finds the hyperplane leaving the largest possible number of samples of the same class in the same side, while maximizing the distance of either class from the hyperplane. Depending on the side of the hyperplane where they are located, the samples are labelled with 1 or -1:

$$(x_1, y_1)(x_2, y_2)\cdots(x_l, y_l), \quad x_i \in \mathbb{R}^N, y \in \{-1, 1\}. \tag{1}$$

LP and NLP classes are considered. Each positive sample (LP) is labelled as 1, and negative samples (NLP) are labelled as -1.

Finding the optimal hyperplane implies solving a constrained optimization problem using quadratic programming. The distance between positive and negative samples is the optimization criterion. The hyperplane is defined as

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i k(x, x_i) + b, \tag{2}$$

where k is the kernel function. Any data point x_i corresponding to a nonzero α_i is a support vector of the optimal hyperplane. Therefore, finding the optimal hyperplane is equivalent to finding the all nonzero α_i. When $f(x) \geq 0$, x is classified as 1 (LP); otherwise, x is classified as -1 (NLP).

Several different kernels are used to solve different problems. In this research, a linear kernel is used due its performance in the context of LPD [27].

Muhammad and Altun [28] utilized HOG features for detecting license plates by means of genetics algorithms with a success rate of over 98 percent. In [29], Sarfraz et al. proposed a method in which the license plate is previously bounded in a region of interest and localized by simple template matching using HOG descriptors. Khan et al. proposed an efficient method [30] using a fusion between HOG features and geometric features followed by a selection of different features selected using a novel entropy-based method. On the other hand, HOG descriptors are widely used in ALPR systems for detecting characters of the license plate in the recognition stage [31].

3.1.2. Local Binary Patterns (LBPs). The original idea proposed by Ojala et al. [4] was to label each pixel of an image with LBP codes. The first step for calculating each LBP code is subtracting the center pixel value from the value of its eight neighbors in a 3×3 square. Resulting strictly negative values are encoded with 0, and the others with 1. Concatenating all these binary codes in a clockwise direction starting from the top-left produces the LBP code associated to the center pixel, and this decimal value encodes the local structure around it. Figure 4 shows an example of the basic LBP operator.

Using the basic LBP operator, large-scale structures cannot be captured due the small 3×3 square. To deal with textures at different scales, the size of the neighborhood becomes variable [32] and is defined as a set of sampling points evenly spaced on a circle whose center is the pixel to be labelled. The sampling points that do not fall within the pixels are interpolated allowing for any radius and any number of sampling points in the neighborhood [33]. The notation (P, R) denotes a neighborhood of P sampling points on a circle of radius R. Figure 5 represents three examples of three LBP operators with different radius and numbers of sampling points.

Dividing the image in groups of pixels called blocks and, summarizing the LBP values of the pixels of each block in a histogram, a powerful texture descriptor is obtained. Now, the feature vector can be processed using a Support Vector Machine (SVM), K-nearest neighbor (K-NN), or some other machine-learning algorithm.

Since the publication of Ojala et al. [4], LBP methodology has been developed with plenty of variations for improved performance in different applications including license plate detection. Recently, Al-Shemarry et al. [34] proposed a novel LPD method based on AdaBoost cascades

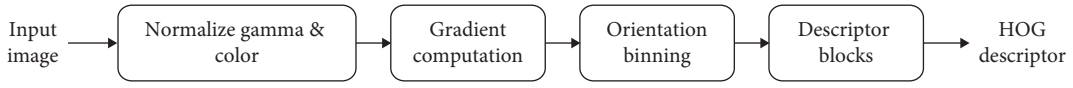


Figure 2: Traditional steps of HOG calculation.

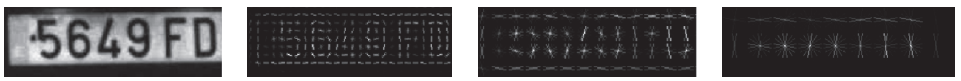


Figure 3: Visualization of three HOG descriptors (b–d) obtained from the same input image (a).

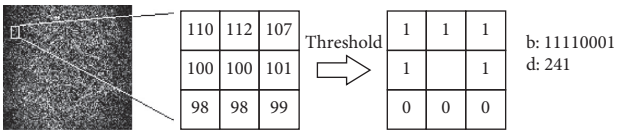


Figure 4: Original LBP codes calculation.

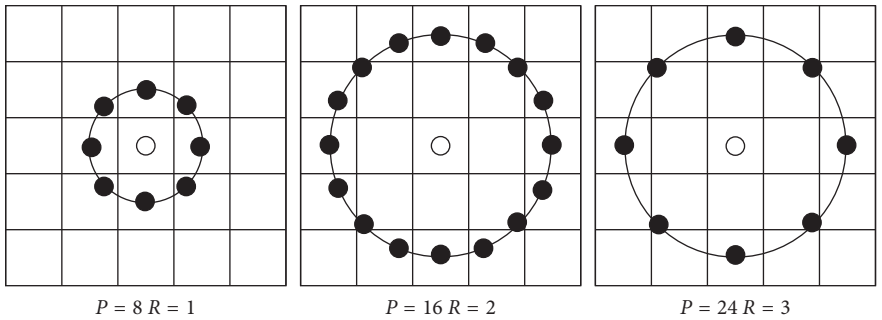


Figure 5: Examples of different extended LBP operators.

classifiers with three-level LBPs (3L-LBP) features. Rashedi and Nezamabadi-pour proposed in [35] a complete LPD solution employing a combination of four methods including one based on cascade classifiers and local binary pattern (LBP) features.

3.1.3. Haar-Like Features. Under the approach of Viola and Jones [5], rectangular regions with shaded and clear areas are extracted from the image. The four original kinds of feature are shown in Figure 6. These features are designed for detecting certain elements, like edges (Figures 6(a) and 6(b)), lines (Figure 6(c)), and diagonals (Figure 6(d)). The resulting value of the feature is calculated by subtracting the sum of all pixels within shaded rectangles from the sum of the clear rectangles.

These features were initially designed for the specific problem of face detection. In 2002, Lienhart et al. [36] presented an extended set of Haar-like features which add additional domain-knowledge to the framework.

The original work of Viola and Jones [5] was designed for detecting faces using a cascade classifier based on AdaBoost [37]. Its proven effectiveness detecting faces allows that it quickly became popular in every area of object detection. The Boosted Cascade of Simple Features is a supervised

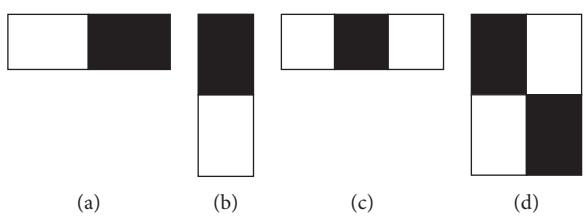


Figure 6: Original Haar-like features.

machine-learning method based on AdaBoost (Adaptative Boosting), required for training the cascade. Boosting techniques are based on the combination of weak classifiers for creating a strong classifier with the desired precision.

AdaBoost was introduced by Freund and Schapire [37] in 1995 in order to solve many challenges associated to boosting processes. For creating a cascade, AdaBoost is used both for selecting a set of features and training the classifier.

For selecting features, weak classifiers, each of them associated to a single feature, are trained. The main goal of these classifiers is to determine the value that minimizes the number of badly classified samples. Therefore, a weak classifier, $h_j(x)$, where x is an input image, can be determined by a feature f_j, a threshold θ_j, and a polarity $p_j \in \{-1, 1\}$

$$h_j(x) = \begin{cases} 1, & \text{if } p_j f_j(x) < p_j \theta_j, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

For each iteration of the AdaBoost algorithm ($t = 1, \ldots, T$), one weak classifier, and thereby one feature, is selected. The strong classifier is computed as a linear combination of the selected weak classifiers $h_t(x)$ which value is either 0 or 1 and is weighted by α_t

$$h_j(x) = \begin{cases} 1, & \text{if } \sum_{t=0}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=0}^{T} \alpha_t, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Instead of creating a single strong classifier by the above descripted algorithm, it is possible to create several efficient smaller classifiers capable of rejecting a high number of negative windows whilst continuing to ensure a high number of positive windows for its evaluation in further classifiers. In this way, a cascade of classifiers is obtained. This process is represented in Figure 7.

Respecting to LPD, Zheng et al. proposed in [38] an efficient cascade detector whose two first stages are based on global features in order to discard most clear background areas, and the four following stages are based on Haar-like features. Furthermore, Wang et al. [39] presented a cascade-based classifier for detecting and tracking license plates using an extended set of Haar-like features.

3.1.4. Variants. In order to confirm the results obtained for the three selected representative descriptors, we added to our research various related texture descriptors and improved variants: Compound Local Binary Patterns (CLBPs), Local Ternary Patterns (LTPs), Local Gradient Patterns (LGPs), Multi-Block Local Binary Patterns (MB-LBPs), HOG-LBP combination, and features extracted from the Gray-Leve Co-occurrence Matrix (GLCM).

Compound Local Binary Patterns [9] increases the robustness of simple LBP. In this method, a 2-bit code is used to encode the local texture property of an image. The first one encodes the difference between the center and neighboring pixel value, while the second bit is used to encode the magnitude of difference with respect to a threshold. The main disadvantage of CLBP algorithm is the size of the feature descriptor in the process of texture feature description, which is larger than other LBP variants and brings great difficulty to the calculation. Reducing the feature dimension will inevitably lead to the loss of texture features.

Local Ternary Patterns [10] was introduced in 2010 by Tan and Triggs for face recognition. From then, several LTP methods were developed for improving the original LPT ([40–43]). Original LTP used uses a group of ternary codes $\{+1, 0, -1\}$ to encode each pixel. Every ternary sequence is divided into two separate sequences of LBP: upper patterns and lower patterns. The algorithm generates the texture features through these binary codes.

LTP is capable to encode the relations "greater than," "equal to," and "less than" between a pixel and its neighbors; on the other hand, LBP could only reflect two of them "greater than" and "less than."

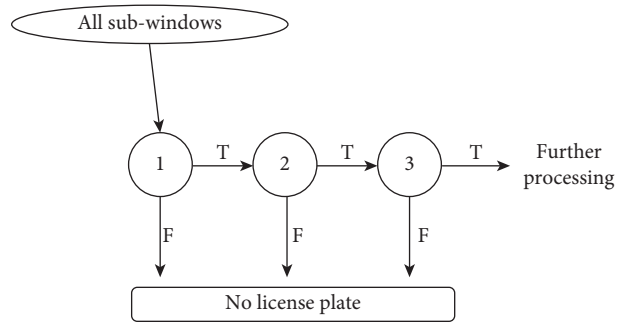


Figure 7: Cascade of classifiers.

Jun et al. [7] proposed the Local Gradient Patterns in 2013. The main goal of LGP was to overcome the problem of local intensity variations along the edge components. In order to achieve that, LGP considers the intensity gradient profile to emphasize the local variation in the neighborhood. If the intensity of the entire image is changed globally, there is no significant difference between the LGP and LBP operators (invariant patterns). If the intensity of the background or the foreground is changed locally, the LGP generates invariant patterns in contrast to the LBP operator due to gradient differences (not only by intensity differences).

Multi-Block LBP, proposed by Zhang et al. [8], is an extension of LBP. Equally sized subblocks are used to compute the features. Instead of taking the comparison between single pixel values, MB-LBP takes the comparison between mean pixel values of these subblocks and does well in describing the texture information in different scales allowing its computation on multiple scales in constant time using the integral image.

Combining Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) as the feature set, Wang et al. [6] proposed a novel approach for detecting pedestrians. The proposed method combines the HOG feature with the cell-structured LBP feature.

When the background contains a high amount of noisy edges, HOG performs poorly. However, LBP uses the concept of uniform pattern that can filter this kind of noises. This reason makes them complementary in this aspect. This combination brings together the advantages of HOG and LBP for detecting license plates. In this way, He et al. [24] recently published a part-based model using HOG for detecting the car and the combination between HOG and LBP for detecting the license plate.

The Gray Level Co-occurrence Matrix [11] is one of the earliest techniques used for image texture analysis. Given a grayscale image composed of pixels each with a specific gray level (intensity), the GLCM is a tabulation of how often different combinations of gray levels co-occur in the image or in a subimage. Using the content of a GLCM matrix, the associate descriptor calculates different texture properties as contrast, dissimilarity, homogeneity, energy, and correlation.

3.2. Dataset. An extensive and complete dataset provided by the City Council of Oviedo (Asturias, Spain) that includes more than 1,000,000 of images captured by the traffic cameras of the city between 2013 and 2016 was used. All images included at least one license plate, and they were captured from 37 different locations: restricted access areas, urban and interurban roads up to 4 lanes, intersections with traffic lights, and roundabouts.

Several camera positions were used. In some cases, such as restricted access areas, the camera is located on the side of the vehicle, while on roads and intersections, the camera is usually located hanging from poles, lampposts, or traffic lights.

Images were captured for 24 hours a day, and several degrees of illumination and meteorological conditions are included in the dataset. From the dataset described above, 21,000 images were extracted and sorted into three groups depending on the environmental conditions: rainfall (rain, snow, and fog), low illumination, and optimal conditions (without rainfall and with good illumination). Example images of low illumination set and rainfall set are shown in Figures 8(a) and 8(b), respectively. Every group is composed by a training set of 5,000 images and a test set of 2,000 images. The percentage of images captured from the different cameras and locations is the same in each group. Table 2 summarizes the composition and purpose of each set of images.

In addition, 7000 negative images were extracted by cropping nonlicense plates areas from images of the main set, in order to consider the same urban scenarios.

3.3. Methodology. The main goal of this experiment is to compare the accuracy of a classifier trained with images captured in optimal conditions with the same classifier trained with images affected by challenging weather or low illumination over the corresponding test images set.

Our experiment is composed by two phases, each one associated with one of the two issues considered for this paper: challenging weather and low illumination. First, we analysed the influence of the challenging weather. To achieve this goal, we proceeded as follows. For each combination of descriptor-classifier, one classifier was trained using the GenTrainingSet and another one was trained using the RainTrainingSet composed of images affected by challenging weather. Both classifiers were tested over the set RainTestSet which is composed of images affected by challenging weather. Comparing the accuracy of both classifiers detecting license plates under these challenging conditions, it is possible to determine which kind of descriptor can adapt better to challenging weather and how could we increase the performance of each descriptor by an appropriate selection of images for the training sets.

For the second phase, we repeat the experiment using images affected by challenging illumination. One classifier trained using the GenTrainingSet and another one using the NightTrainingSet (trained with images taken without an adequate lighting) were considered for each combination of descriptor-classifier and testing with the NightTestSet. The goal is the same—obtain information about how each descriptor is affected by low illumination and improve our knowledge about the composition of training sets.

The classifiers were named as follows: CLASSIFIER-DESCRIPTOR-{GEN/RAIN/NIGHT}. Where the suffix -GEN indicates that the classifier was trained with the optimal conditions training set (GenTrainingSet), the suffix -RAIN denotes that the classifier was trained with the rainfall training set (RainTrainingSet), and in the same way, the suffix -NIGHT indicates that the classifier was trained with challenging illumination images (NightTrainingSet).

As an example of the entire test for each pair classifier-descriptor, Figure 9 represents the model training step (Figure 9(a)) and the testing step (Figure 9(b)) for the pair SVM-HOG. This procedure is repeated for each possible pair of classifier-descriptor.

The HOG features extraction module was configured with the following parameters. The cell size was settled at 8×8 pixels, and every block is composed by 2×2 cells. 9 orientations are considered; this is the number of orientation bins that the gradients of the pixels of each cell will be split up in the histogram.

Regarding the extraction of LBP features, each LBP code is defined by 8 sampling points and a 2px radius. Images are divided into 16×16 px blocks, and the LBP codes of each block are summarized in a histogram. Each histogram has a separate bin for every pattern. We decided to use uniform patterns [44] in order to reduce the length of the histogram, and thus the dimension of the feature vector. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59.

Every Cascade classifier was trained with the same parameters. The training process was settled at 20 stages. Minimal desired hit rate for each stage of the classifier was settled at 0.998, and the maxima desired false alarm at 0.5.

In relation to linear SVM, all classifiers were also trained using a Regularization factor C settled in 0.1.

3.4. Evaluation. In order to compare detectors, miss rate versus FPPI (false positive per image) by varying the threshold on detection confidence are plotted. Both values are plotted on log axes according with the evaluation metrics proposed by Dollar et al. [45]. This is preferred to precision recall curves for tasks in which there is an upper limit on the acceptable FPPI rate independent of license plate density [45].

Miss rate or false positive rate (FPR) is the number of missed detections (license plates that the classifier failed to detect) in relation to the number of false positives. It is the opposite of recall or true positive rate (TPR).

$$\begin{aligned} \mathrm{TPR\,(recall)} &= \frac{\mathrm{TP}}{P}, \\ \mathrm{FPR\,(miss\,rate)} &= \frac{\mathrm{FN}}{P} = 1 - \mathrm{TPR}. \end{aligned} \tag{5}$$

Equation (5) shows the relation between recall or TPR (true positive rate) and miss rate or FPR. Where TP (true

(a) (b)

Figure 8: Example images extracted from the low illumination set (a) and rainfall set (b).

Table 2: Description of the sets of images used for the experiments.

Name	Purpose	Number of images	Description
GenTrainingSet	Training	5000	Images taken in optimal conditions (without rainfall and with good illumination)
GenTestSet	Test	2000	
RainTrainingSet	Training	5000	Images taken under rainfall conditions (heavy rain, fog, or snow)
RainTestSet	Test	2000	
NightTrainingSet	Training	5000	Images taken under difficult lighting conditions (lack of light)
NightTestSet	Test	2000	

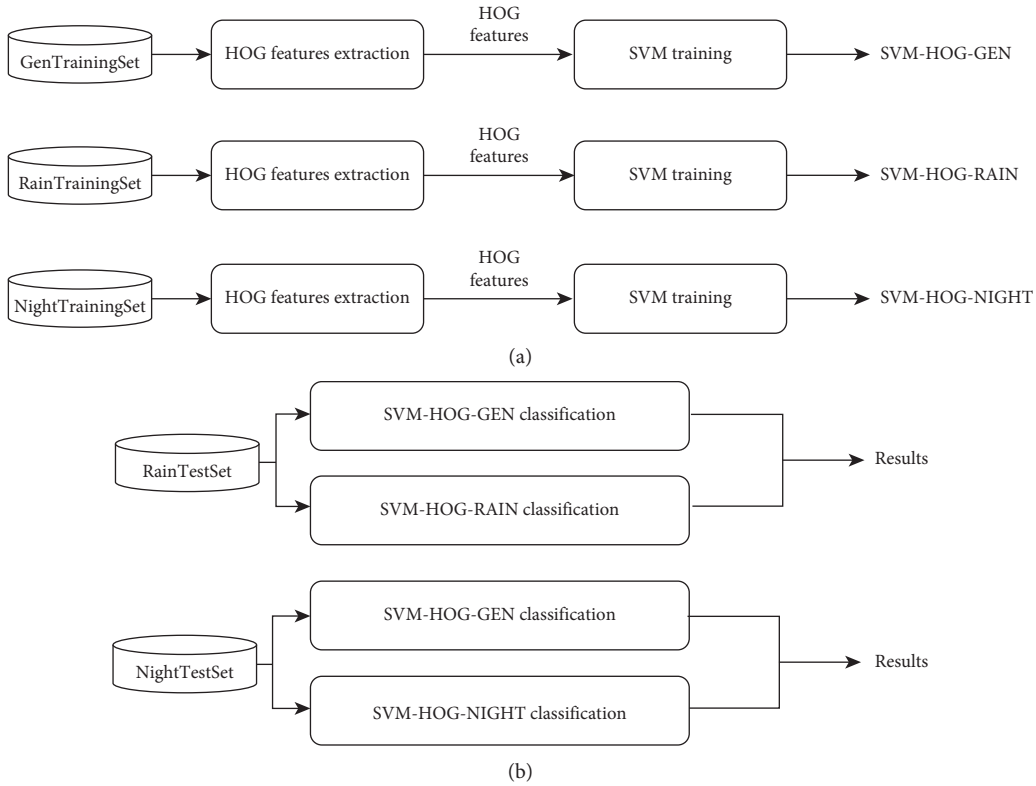


Figure 9: Complete experiment for the pair SVM-HOG including classifiers creation step (a) and testing step (b).

positives) is the number of correct detections, P is the total number of license plates, and FN (false negative) is the number of missed detections. False positives per image is the number of false positives in relation to the total number of images.

$$\text{FPPI} = \frac{\text{FP}}{\text{total number of images}}, \tag{6}$$

In this kind of graphs, lower curves indicate better accuracy. Miss rate equal to 1 FPPI is considered the common

reference point for comparing results, considering that there are 1.2 license plates/image in the selected test sets. In addition, the range 10^{-2}–10^{1} is considered the range of interest for evaluating which classifier performs better (the lower the curve, the better the performance).

4. Results and Discussion

This research focuses on two environmental factors as challenging weather and challenging illumination conditions. Eight tests were performed for each challenge, comparing the performance of specific trained classifiers with generic trained classifier according the metric exposed in Section 3.1.

In order to analyze the effects of the challenging weather conditions, eight tests were developed. First, HOG features are tested, in accordance with the approach of Dalal and Triggs [3], by comparing the performance of the SVM trained in optimal conditions and the SVM trained with challenging weather (Figure 10).

The classifier trained with the GenTrainingSet performs better than the classifier that has received a specific training for challenging weather (RainTrainingSet). The performance of the generic classifier improves up to 19% the recall at 1 FPPI.

Secondly, we tested the classical approach of LBP descriptor with an SVM classifier. Figure 11 shows the results of this test.

At first, we selected the nonrotation invariant LBP-Uniform version as representative LBP operator (Figure 11(a)). When using a SVM classifier, we did not detect a clear difference between the two curves. It is clearly seen that both curves performs better than the another one along two different parts of the rage of interest. In order to ensure reliable results, we decided to repeat the experiment using the original LBP operator (Figure 11(b)), using the same parameters as the NRI LBP-U except the number of bins of the histogram (255 instead of 59). The margin between both curves was already small, but in this last test, the performance of the classifier trained with the GenTrainingSet outperformed the specific one over the whole range of interest.

The following test compared the performance of the algorithm of Viola and Jones [5] trained with good weather conditions and trained with specific challenging weather images.

Figure 12 shows again that the curve of the classifier trained with optimal conditions images runs along the area of interest under the specific training classifier curve. The difference between them is not very high due to the high accuracy of this approach for LPD.

Regarding the variants, the results are in line with the three selected representative descriptors. It is important to note that every LBP variant (CLBP, LTP, LGP, and MB-LBP) obtained similar results at the range of interest. Our tests show (Figure 13) that the performance of the generic classifier of each LBP-based descriptor improves the recall of the model between 5% and 10% at the reference point

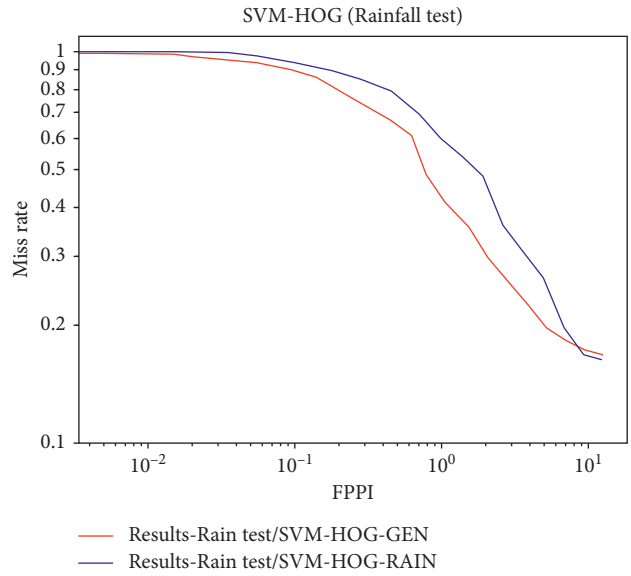


Figure 10: Results of the challenging weather test for HOG descriptors using an SVM classifier.

of 1 FPPI, compared to the specific one trained with RainTrainingSet.

Figure 14 shows the comparison between the performance of both classifiers trained with HOG-LBP descriptors and the SVM classifier. According to the results obtained by the HOG descriptor, the general performance of both curves outperforms the LBP variants. Results are similar to those of previous tests. In this case, the performance of the generic classifier trained with the GenTrainingSet improves up to 7% the recall of the classifier trained with the RainTrainingSet at the reference point of 1 FPPI and remains higher all along the range of interest.

Finally, we repeated the test with the GLCM descriptor. Because of the simplicity of the extracted features, the general performance of both curves is significantly worse than the rest of tests. Irrespective of general performance, Figure 15 shows that the curve of the generic classifier trained with the GenTrainingSet outperforms, again, the curve of the classifier trained with an specific training.

The second challenging environmental condition for evaluation in this research was the lack of light in ALPR scenarios. We used the same procedure as for evaluating the effects of challenging weather.

Again, the first test of this experiment was designed for evaluating the effects of the lack of light in HOG performance by comparing the performance of the SVM trained in optimal conditions and the SVM trained with challenging illumination (Figure 16). It is noticeable that the curve that relates Miss rates and FPPI corresponding to the classifier which has received a specific training with the NightTrainingSet performs better than the classifier trained with the GenTrainingSet all along the range of interest.

Figure 17 shows the results of the same test but using LBP descriptors. The curve corresponding to the classifier

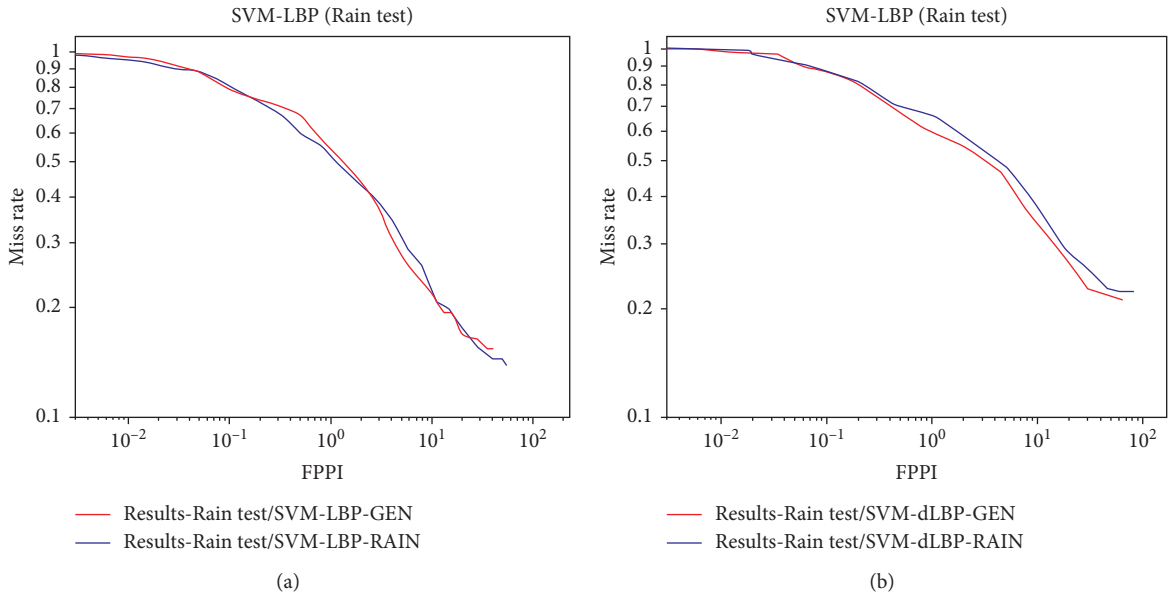


Figure 11: Results of the challenging weather test for two different kinds of LBP descriptors using an SVM classifier. The left-hand image (a) shows the results using the extension nonrotation invariant LBP-Uniform and the right-hand image (b) shows the results using the original simple LBP operator.

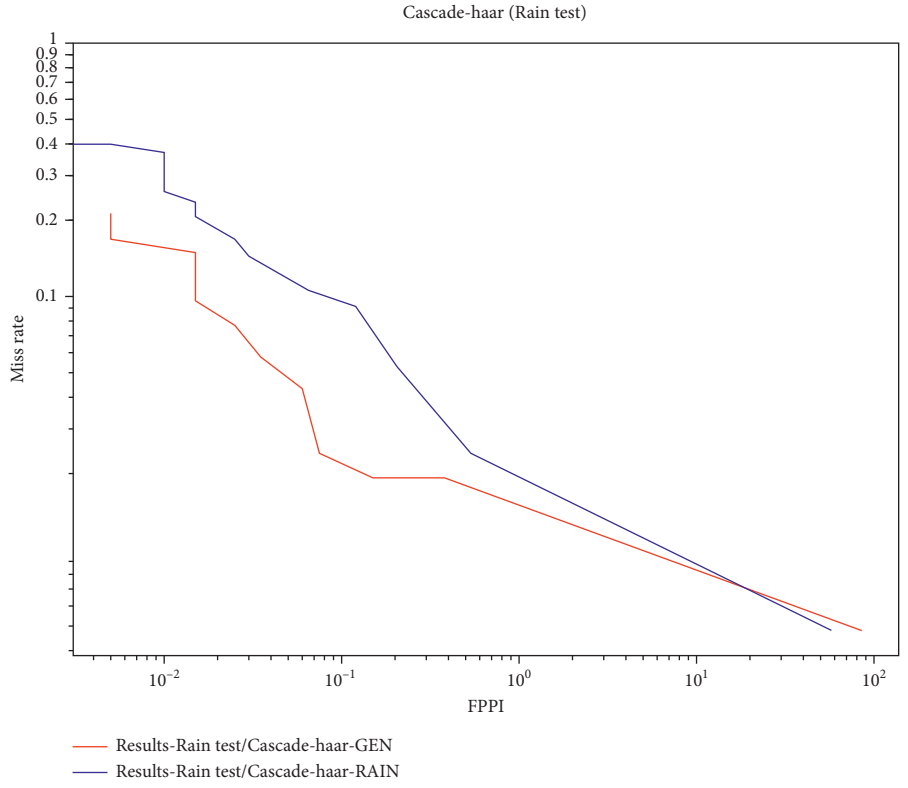


Figure 12: Results of the challenging weather test for the Viola and Jones [5] approach.

which has received a specific training with NightTrainingSet performs clearly better than the classifier trained with good conditions images along the range of interest. LBP is able to capture the light differences with high accuracy, and the difference between the classifiers with regard to recall is up to 19% taking 1 FPPI as reference.

Similar results were obtained in the next test. Results show again a clear difference between the performance of

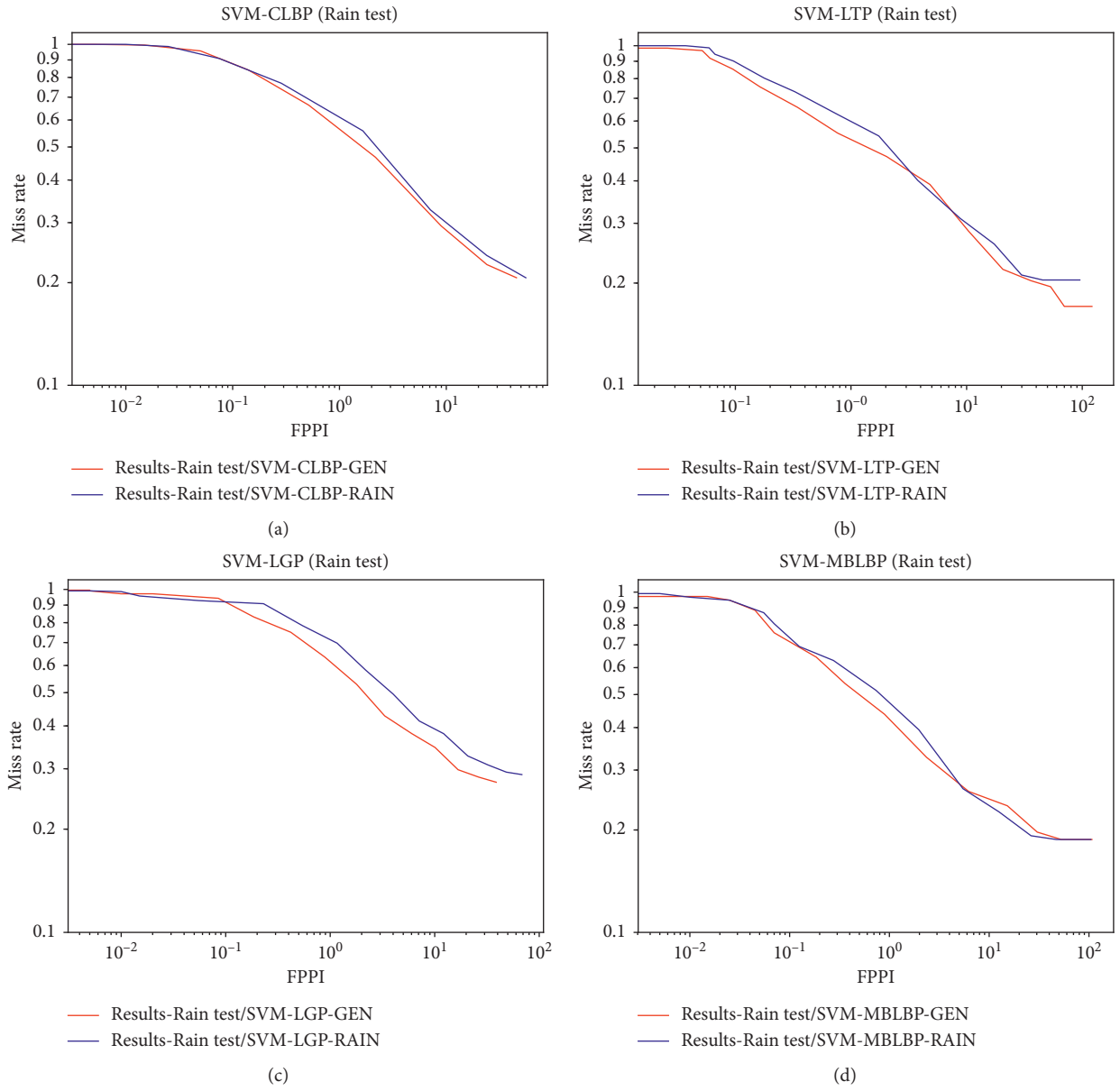


Figure 13: Results of the challenging weather test for LBP variants trained with SVM classifiers: CLBP (a), LTP (b), LGP (c), and MB-LBP (d).

the curve corresponding to the classifier which has received a specific training with NightTrainingSet and the curve corresponding to the classifier trained with good conditions images. Figure 18 shows the graph relative to the Cascade Classifier of Haar features. In this test, the performance of the specificly trained classifier improves the recall up to 29% at the reference point of 1 FPPI.

With regard to the variants, there are huge differences between the curve corresponding to the classifier which has received a specific training and the generic one for every test performed. The results were conclusive, and every variant test (Figure 19) confirmed the results of the representative descriptors tests.

4.1. Robustness Checks. In order to verify the robustness of our results, three additional classifiers were included in our tests: Linear Regression (LR), K-Nearest Neighbor (K-NN), and an Artificial Neural Network (ANN).

The LR classifier implements a simple logistic regression using a regulation factor $c = 1$.

The K-NN classifier defines the class of an element depending on the distance measured between these elements and its neighbors. We selected a minimum of 5 nearest neighbors and an Euclidean function for computing the distance. Respecting the ANN, we used a Multi-layer Perceptron with one hidden layer.

Similar differences can be observed between the curve corresponding to the model which received a specific

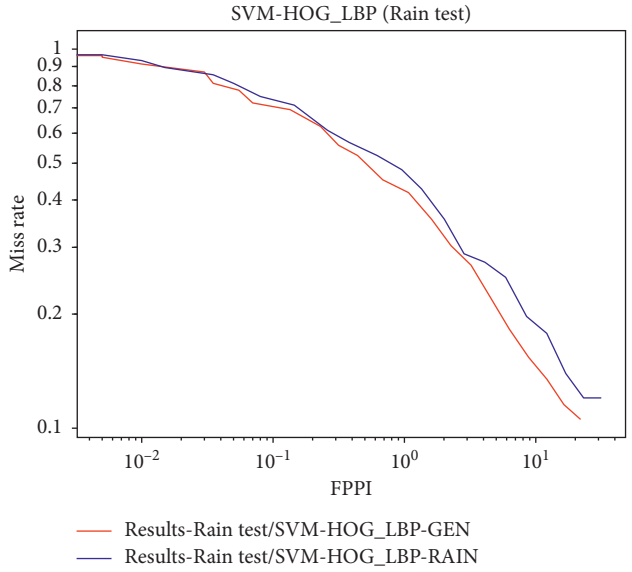


Figure 14: Results of the challenging weather test for HOG-LBP descriptor using an SVM classifier.

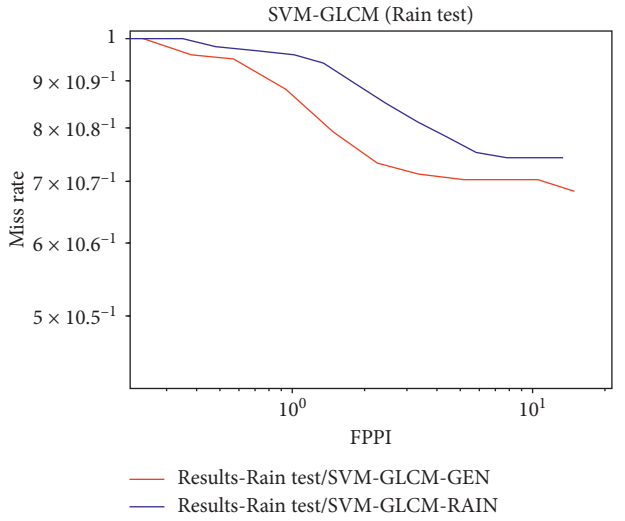


Figure 15: Results of the challenging weather test for GLCM descriptor using an SVM classifier.

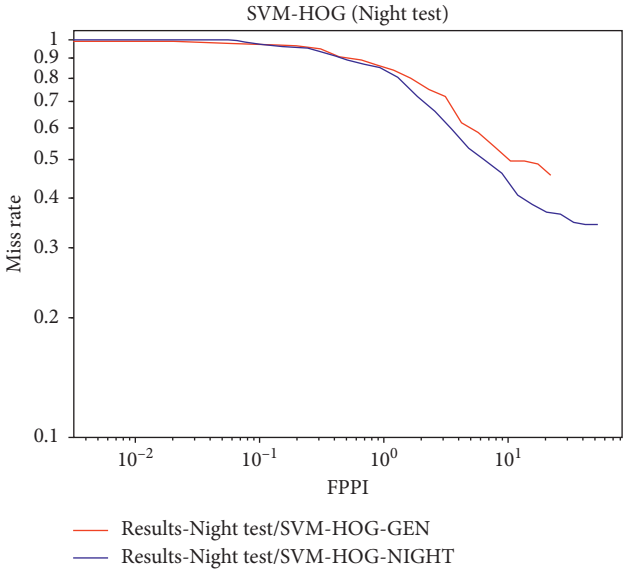


Figure 16: Results of the challenging illumination test for HOG descriptors using an SVM classifier.

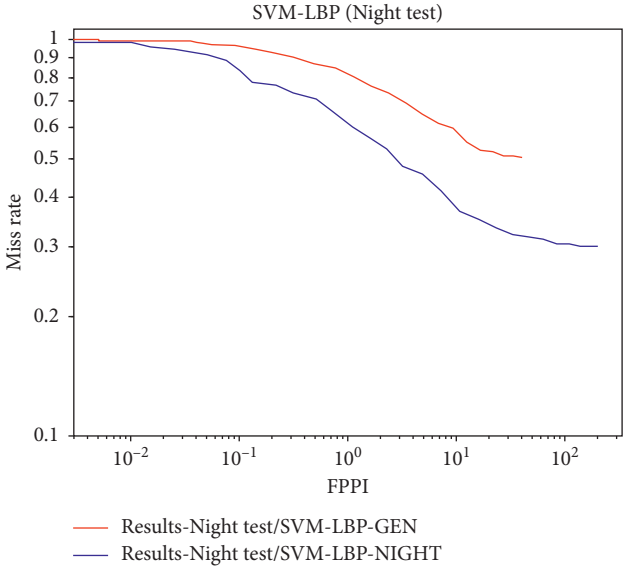


Figure 17: Results of the challenging illumination test for LBP descriptors using an SVM classifier.

training and the curve trained with the generic GenTrainingSet for each descriptor regarding the kind of classifier. The results confirm that the curves associated to a specific training tend to outperform the generic one for challenging illumination tests. On the other hand, the curve associated to a generic training tends to outperform the specific one for challenging weather tests.

Detailed data of every test performed is available in https://doi.org/10.6084/m9.figshare.7926920.

5. Conclusions

When an object detection system based on feature classification is executed in outdoor scenarios, it is reasonable to think that the best way to obtain a good performance is to consider every environmental condition in the training sets. If it is located in a place where it often rains, a feasible strategy is to train the classifier using rain images in proper proportion. In the same way, if the system operates during the night, it makes sense to use low-illumination images in proportion to the estimated time that the system works under these conditions or, if it is possible, to work with different classifiers specifically trained for each condition.

The results obtained suggest that illumination and weather are two diverse problems with a different origin. Thus, their influence in the description of objects is completely different.

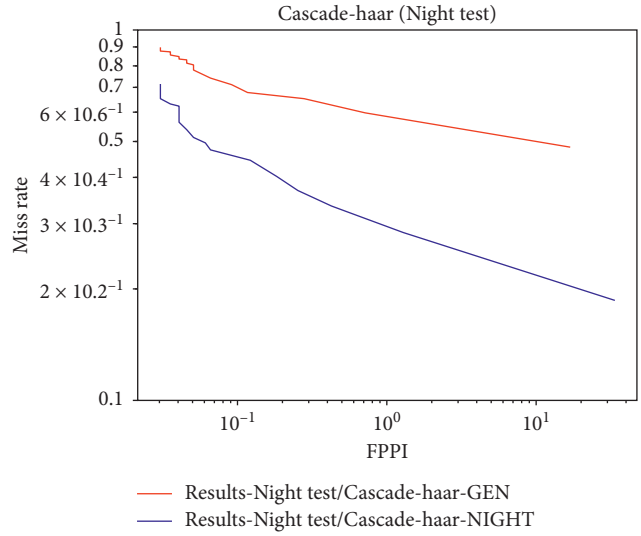


Figure 18: Results of the challenging illumination test for the Viola and Jones [5] approach.

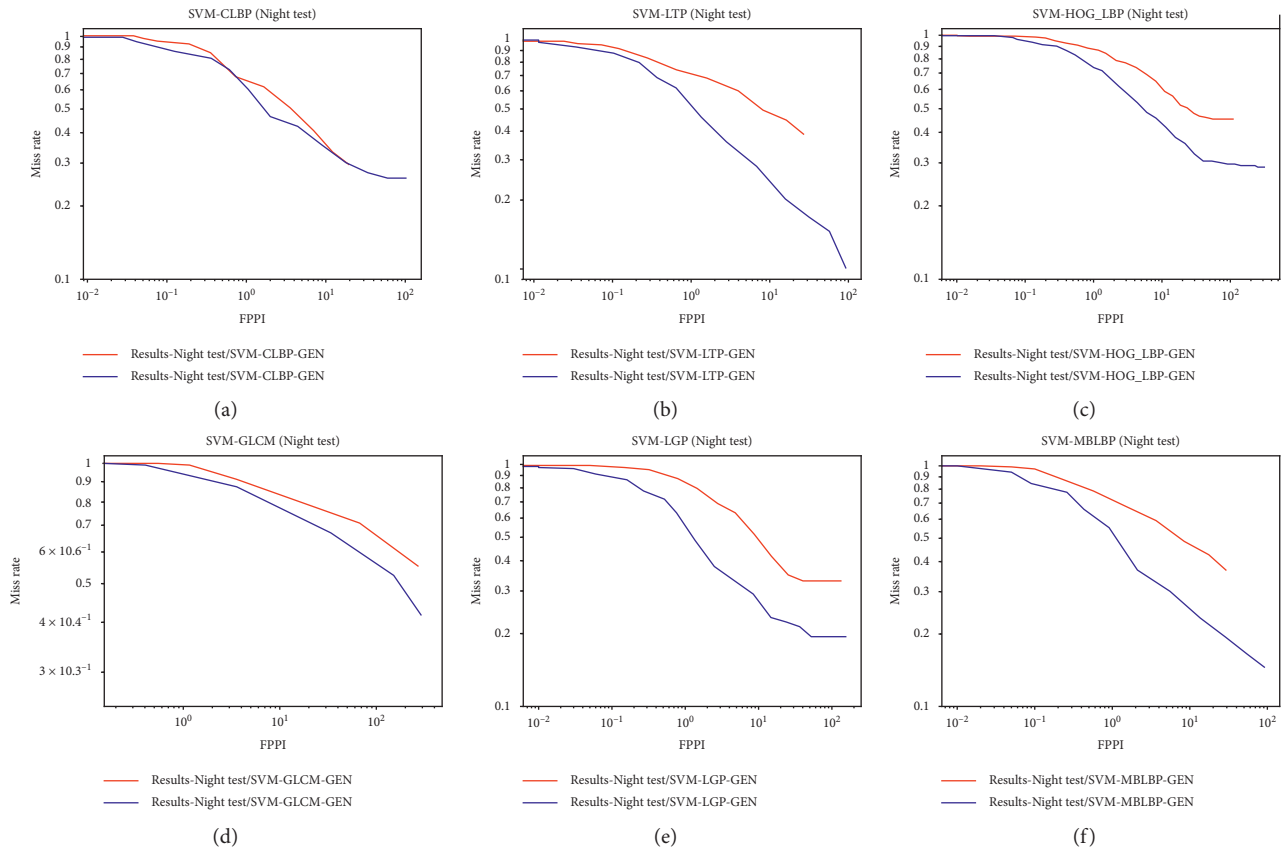


Figure 19: Results of every variant for the challenging illumination test using an SVM classifier: CLBP (a), LTP (b), HOG-LBP (c), GLCM (d), LGP (e), and MB-LBP (f).

Regardless the kind of feature, the effect of rainfall is not adequately reflected in the descriptors. After testing four different features in combination with two descriptors, only the combination LBP + SVM does not produce conclusive evidence. The remaining tests indicate that by removing the rainfall images from the training sets, it is possible to improve the performance of the classifier up to 19%. When the rain, fog, or snow is captured by a camera into a 2d image, suspended particles create a random pattern between the camera and the objective. These patterns differ, and it seems that the feature extraction process is not capable to recover relevant information to improve the classification. In

fact, the inclusion of this random information over the license plates could worsen the performance of the classifier because its effect is similar to that of digital noise.

On the contrary, the results suggest that the influence of light should be considered in the training process. Texture, color, or gradient patters produced in different illumination conditions are important information to be extracted because these patterns could be recognizable by the classifier.

HOG is the least-sensitive descriptor with regard to challenging illumination. HOG summarizes the gradient information into histograms grouped by its direction. These gradient directions remain constant regardless the intensity of the light, and therefore, the performance improvement of the classifiers is not high. With an LBP descriptor, it is possible to improve the classifier recall up to 20% by performing a proper training that considers images affected by challenging illumination. In a similar way, we could improve the performance of the Viola and Jones algorithm up to 29% by including different illumination conditions.

Comparing with other recent techniques [12, 17], which usually define all phases of LPD, our technique was tested with different texture-based descriptors and classifiers, allowing for a great adaptability to any algorithm based on ML. The conclusions of our study allow for a correct selection of the images that comprise the training sets; therefore, our technique is compatible with many other preprocessing techniques [12–23] that try to avoid the adverse effects of meteorology and lack of lighting.

6. Limitations and Future Work

In the presented research, several texture-based descriptors were tested under the assumption that other descriptors based on the same kind of features exhibit similar behaviour. It is an interesting research for us, expanding our work including descriptors like SURF, SIFT, FAST, or CSIFT. In addition, Content-Based Image Retrieval (CBIR) has attracted enormous attention over the last few years, and methods incorporating shape, spatial layout, and saliency to describe visual contents are gaining attention. In this line, novel descriptors which incorporate various kinds of features have been developed recently. Incorporating descriptors as MTH, CDH, SED, or MSD into our work is another interesting research for us.

We decided to test our classifiers using our own dataset because it is not a goal of this paper to compare the accuracy of the tested classifiers with the state of the art. Instead, the objective is to assess the variation in the accuracy of each classifier when the training set is modified in order to include challenging illumination/weather conditions. Comparing the performance of different approaches, considering the conclusions extracted from this paper, by testing them using the widely used benchmarks is an important avenue of research.

Data Availability

The image datasets that support the findings of this study are not publicly available due to them containing information that could compromise research participant privacy. The rest of the data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the European Union, through the European Regional Development Funds (ERDF), and the Principality of Asturias, through its Science, Technology and Innovation Plan. This work was partially funded by the Department of Science, Innovation and Universities (Spain) under the National Program for Research, Development and Innovation, project RTI2018-099235-B-I00.

References

[1] M. Molina-Moreno, I. Gonzalez-Diaz, and F. Diaz-de-Maria, "Efficient scale-adaptive license plate detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2109–2121, 2019.

[2] B.-G. Han, J. T. Lee, K.-T. Lim, and Y. Chung, "Real-time license plate detection in high-resolution videos using fastest available cascade classifier and core patterns," *ETRI Journal*, vol. 37, no. 2, pp. 251–261, 2015.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, San Diego, CA, USA, June 2005.

[4] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, pp. 582–585, Tsukuba, Japan, November 2012.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 511–518, Kauai, HI, USA, December 2001.

[6] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 32–39, Tampa, FL, USA, September 2009.

[7] B. Jun, I. Choi, and D. Kim, "Local transform features and hybridization for accurate face and human detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1423–1436, 2013.

[8] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Proceedings of the International Conference on Biometrics, ICB 2007: Advances in Biometrics*, pp. 11–18, Seoul, Republic of Korea, August 2007.

[9] F. Ahmed, H. Bari, and E. Hossain, "Person-independent facial expression recognition based on compound local binary pattern (CLBP)," *International Arab Journal of Information Technology*, vol. 11, no. 2, pp. 195–203, 2014.

[10] X. Y. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE*

Transactions on Image Processing, vol. 19, no. 6, pp. 1635–1650, 2010.

[11] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[12] M. K. Saini and S. Saini, "Multiwavelet transform based license plate detection," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 128–138, 2017.

[13] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," *Image and Vision Computing*, vol. 27, no. 8, pp. 1134–1142, 2009.

[14] G. Raju and M. S. Nair, "A fast and efficient color image enhancement method based on fuzzy-logic and histogram," *AEU—International Journal of Electronics and Communications*, vol. 68, no. 3, pp. 237–243, 2014.

[15] X. Xue, J. Ding, and Y. Shi, "Research and application of illumination processing method in vehicle color recognition," in *Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1662–1666, Chengdu, China, December 2017.

[16] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.

[17] D. Wang, Y. Tian, W. Geng, L. Zhao, and C. Gong, "LPR-Net: recognizing Chinese license plate in complex environments," *Pattern Recognition Letters*, In press.

[18] Yi-T. Chen, J.-H. Chuang, W.-C. Teng, H.-H. Lin, and H.-T. Chen, "Robust license plate detection in nighttime scenes using multiple intensity IR-illuminator," in *Proceedings of the 2012 IEEE International Symposium on Industrial Electronics*, pp. 893–898, Hangzhou, China, May 2012.

[19] K. S. Raghunandan, P. Shivakumara, H. A. Jalab et al., "Riesz fractional based model for enhancing license plate detection and recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2276–2288, 2018.

[20] H. Dawood, H. Dawood, and P. Guo, "Removal of high-intensity impulse noise by Weber's law noise identifier," *Pattern Recognition Letters*, vol. 49, pp. 121–130, 2014.

[21] J. Jie Chen, S. Shan, C. He et al., "WLD: a robust local image descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2010.

[22] S. Azam and M. M. Islam, "Automatic license plate detection in hazardous condition," *Journal of Visual Communication and Image Representation*, vol. 36, pp. 172–186, 2016.

[23] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.

[24] H. He, Z. Shao, and J. Tan, "Recognition of car makes and models from a single traffic-camera image," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3182–3192, 2015.

[25] F. Delmar Kurpiel, R. Minetto, and B. T. Nassu, "Convolutional neural networks for license plate detection in images," in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3395–3399, Beijing, China, September 2017.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[27] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.

[28] J. Muhammad and H. Altun, "Improved license plate detection using HOG-based features and genetic algorithm," in *Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 1269–1272, Zonguldak, Turkey, May 2016.

[29] M. S. Sarfraz, A. Shahzad, M. A. Elahi, M. Fraz, I. Zafar, and E. A. Edirisinghe, "Real-time automatic license plate recognition for CCTV forensic applications," *Journal of Real-Time Image Processing*, vol. 8, no. 3, pp. 285–295, 2011.

[30] M. A. Khan, M. Sharif, M. Y. Javed, T. Akram, M. Yasmin, and T. Saba, "License number plate recognition system using entropy-based features selection approach with SVM," *IET Image Processing*, vol. 12, no. 2, pp. 200–209, 2018.

[31] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.

[32] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[33] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 765–781, 2011.

[34] M. S. Al-Shemarry, Y. Li, and S. Abdulla, "Ensemble of adaboost cascades of 3L-LBPs classifiers for license plates detection with low quality images," *Expert Systems with Applications*, vol. 92, pp. 216–235, 2018.

[35] E. Rashedi and H. Nezamabadi-pour, "A hierarchical algorithm for vehicle license plate localization," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2771–2790, 2018.

[36] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings International Conference on Image Processing*, vol. 1, pp. 900–903, Rochester, NY, USA, September 2002.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[38] L. Zheng, X. He, B. Samali, and L. T. Yang, "An algorithm for accuracy enhancement of license plate recognition," *Journal of Computer and System Sciences*, vol. 79, no. 2, pp. 245–255, 2013.

[39] R. Wang, N. Sang, R. Wang, and L. Jiang, "Detection and tracking strategy for license plate detection in video," *Optik*, vol. 125, no. 10, pp. 2283–2288, 2014.

[40] M. Raja, M. Raja, and V. Sadasivam, "Optimized local ternary patterns: a new texture model with set of optimal patterns for texture analysis," *Journal of Computer Science*, vol. 9, no. 1, pp. 1–15, 2013.

[41] X. Wu, J. Sun, G. Fan, and Z. Wang, "Improved local ternary patterns for automatic target recognition in infrared imagery," *Sensors*, vol. 15, no. 3, pp. 6399–6418, 2015.

[42] S. Wu, L. Yang, W. Xu, J. Zheng, Z. Li, and Z. Fang, "A mutual local-ternary-pattern based method for aligning differently exposed images," *Computer Vision and Image Understanding*, vol. 152, pp. 67–78, 2016.

[43] X.-H. Han, Y.-W. Chen, and G. Xu, "Integration of spatial and orientation contexts in local ternary patterns for HEp-2 cell

classification," *Pattern Recognition Letters*, vol. 82, pp. 23–27, 2016.

[44] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, Tampa, FL, USA, December 2013.

[45] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: a benchmark," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304–311, Miami, FL, USA, June 2009.

Hindawi
Scientific Programming
Volume 2019, Article ID 9195845, 15 pages
https://doi.org/10.1155/2019/9195845

Research Article

Avionics Graphics Hardware Performance Prediction with Machine Learning

Simon R. Girard,[1] Vincent Legault,[1] Guy Bois,[1] and Jean-François Boland[2]

[1]*Computer Engineering Department, Polytechnique Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C 3A7*
[2]*Electrical Engineering Department, École de Technologie Supérieur (ETS), 1100 Rue Notre-Dame Ouest, Montréal, QC, Canada H3C 1K3*

Correspondence should be addressed to Guy Bois; guy.bois@polymtl.ca

Received 21 December 2018; Revised 15 April 2019; Accepted 6 May 2019; Published 3 June 2019

Guest Editor: Vicente García-Díaz

Within the strongly regulated avionic engineering field, conventional graphical desktop hardware and software application programming interface (API) cannot be used because they do not conform to the avionic certification standards. We observe the need for better avionic graphical hardware, but system engineers lack system design tools related to graphical hardware. The endorsement of an optimal hardware architecture by estimating the performance of a graphical software, when a stable rendering engine does not yet exist, represents a major challenge. As proven by previous hardware emulation tools, there is also a potential for development cost reduction, by enabling developers to have a first estimation of the performance of its graphical engine early in the development cycle. In this paper, we propose to replace expensive development platforms by predictive software running on a desktop computer. More precisely, we present a system design tool that helps predict the rendering performance of graphical hardware based on the OpenGL Safety Critical API. First, we create nonparametric models of the underlying hardware, with machine learning, by analyzing the instantaneous frames per second (FPS) of the rendering of a synthetic 3D scene and by drawing multiple times with various characteristics that are typically found in synthetic vision applications. The number of characteristic combinations used during this supervised training phase is a subset of all possible combinations, but performance predictions can be arbitrarily extrapolated. To validate our models, we render an industrial scene with characteristic combinations not used during the training phase and we compare the predictions to those real values. We find a median prediction error of less than 4 FPS.

1. Introduction

In recent years, there has been an increased interest in the avionics industry to implement high-performance graphical applications like synthetic vision systems (SVSs) that display pertinent and critical features of the environment external to the aircraft [1]. This has promoted the advent of faster graphical processing hardware. Because it is a highly regulated field, conventional desktop and embedded graphics hardware could not be used because they do not conform to the DO-178C and DO-254 avionic certification standards (the international standards titled "RTCA DO-178C—Software Considerations in Airborne Systems and Equipment Certification" and "DO-254—Design Assurance Guidance for Airborne Electronic Hardware" are the primary standards for commercial avionics software and hardware development) [2, 3]. Considering the need of avionic hardware with higher performance, we observe that graphical application development tools and hardware benchmarks and simulators available for conventional embedded or desktop graphical applications seem still to be missing for avionics applications. This fact is made especially clear when a quick search through the specifications of the most renowned tools such as Nvidia Nsight [4], AMD PerfStudio [5], or SPECViewPerf [6] lead to the same conclusions. Even though, there are some WYSIWYG ("what you see is what you get") GUI toolboxes available for the ARINC-661 standard [7, 8], it seems that there is no performance benchmark, performance prediction

tool, or performance-correct simulator available for graphical avionic hardware. The interest in having such tools is especially significant because most development processes include a design phase before the actual implementation. Taking the example on the classical V-Model [9], designers must make choices in regard to the purchase or the in-house development of graphical hardware. But as they want to evaluate the performance of such hardware relating to the choices made, they need some kind of performance metrics and benchmarks. This benchmarking tool should be provided by the software development team but as the project is still in the design phase, they have not yet necessarily implemented a graphical engine to enable performance testing. Performance prediction can be useful to (1) further extrapolate the benchmark performance results for any volume of graphical data sent to the hardware and (2) reduce the number of benchmarks required to evaluate various use cases. Going further, the performance models generated can then be used to develop a performance-correct hardware simulator that developers can use on their workstation, in order to have a general preview of the efficacy of their software, before executing it on the real system. As this is the case with the most hardware emulation tools, this reduces the development costs by facilitating functional verification of the system [10].

In this work, we demonstrate two prototype tools that can be used as a pipeline to evaluate and then predict the performance of graphical hardware. The first tool is a benchmark that can generate and then render custom procedural scenes according to a set of scene characteristics such as the number of vertices, size of textures, and more. It evaluates and outputs the number of frames generated per seconds (FPS). The second tool takes the output of a certain number of executions of the benchmark and generates a nonparametric performance model, by using machine learning algorithms on the performance data. Those performance models can then extrapolate predictions of performance for any dense 3D scene rendered on this piece of hardware. We evaluated the distribution of prediction errors experimentally to find that most prediction errors will not exceed 4 FPS.

In the rest of the paper, Section 2 presents the main problems which make inadequate the aforementioned existing tools for the avionics industry. It also presents the work related to the various algorithms and methods used by those standard tools. Section 3 presents the first contribution which is the graphical avionic application benchmarking tool. Then, Section 4 presents the second contribution which is the performance model generation tool. Section 5 presents the experimental method used to evaluate the prediction power of these models. Section 6 presents analyses and discusses the achieved prediction error distributions. Finally, Section 7 provides information for those who would like to repeat the experiment.

2. Background

Among the differences between conventional (consumer market) and avionics graphic hardware development, three are denoted as especially standing out. The first is the use of OpenGL SC instead of OpenGL ES or the full OpenGL API to communicate with the hardware [11]. The second is the use of a fixed graphics pipeline instead of letting the possibilities of using shaders or custom programs that can be sent to the graphics hardware to modify the functionalities of certain areas of the rendering pipeline [12]. Finally, there is the research interest in the development of DO-254-compliant graphical hardware, in the form of software GPUs, FPGAs, and CPU/GPU on-a-chip to name a few [2]. The nature of the hardware is then not necessarily a processor, and certain metrics specific to that nature cannot be applied, such as instruction count. Also, avionics graphical hardware is usually a very secured black box that cannot be intruded to actually perform the instruction count metrics on the internal programs. Thus, performance benchmark and prediction tools in an avionics context should account for these specificities. By only using the functions available in OpenGL SC to evaluate the performance of the hardware, we make sure of the following two points: (1) to use the standard fixed pipeline that accompanies this version of the API and (2) to be independent on the nature of the underlying hardware beyond that interface.

To the best of our knowledge, graphics hardware performance prediction in the avionics context does not exist in the literature. Looking for methods to closely related fields would thus be the best approach. It is then interesting to look over the literature to find the methods that have been used in a conventional desktop and embedded context. It is also interesting to widen this review to general computer hardware and microarchitecture, as well as graphical hardware performance prediction. Numerous benchmarks for graphic hardware exist in the conventional context, such as SPECViewPerf or Basemark [6, 13], to name a few. Even if they do not satisfy the special problematic of the avionics needs, their workflow can be a source of inspiration. For example, SPECViewPerf allows users to create a list of tests, each varying different characteristics of the scenes or the render state, such as local illumination models, culling, texture filters, simple, or double buffering. It then returns the average FPS attained during the rendering of the scene for each test. The use of the average FPS might be more significant for the user, but because the FPS distribution is not normal it loses a lot of statistical significance. As for the performance prediction tools, they tend to be made available by the graphic hardware manufacturers such as the NVIDIA® Nsight™ [4] or the AMD GPU PerfStudio [5]. The problem with these tools is that they are only available for the desktop and embedded domain and are not adapted to the needs of avionics, as explained previously. It is still interesting to review the literature to better understand how those profiling tools might work internally. There are three main approaches to generate models: analytical modeling, parametric modeling, and machine learning. Analytical models attempt to create mathematical models that represent performance as a set of functions describing the hardware. They often use metrics such as instruction count and properties such as frequency clock of the processor [14–17]. The main issue with these methods is that they require a good understanding of the hardware's inner workings, which is difficult in an avionics context because

they are secured black box entities. Also, because of the fixed pipeline of the graphics card, system engineers cannot obtain the inner programs operating the pipeline and thus cannot use analytical metrics such as instruction count. Also, the literature seems to indicate that it is very difficult to truly identify all factors influencing performance and thus to mathematically model them. However, there is one analytical model that has the potential to be used in an avionic context. It is a function that estimates the transfer time of data from the main to the graphic memory [18].

The creation of parametric models implies the use of parametric regressions such as linear or polynomial regressions. It has been used for microarchitecture and CPU design-space exploration [19–22], but because we only have access to the interface of the hardware, it is hard to identify all the factors influencing the performance. Thus, these methods would have difficulty to explain most of the variance of the performance data and can then perform poorly. This is usually solved by using nonparametric regression models created with machine learning.

Even though there are a large number of machine learning that can be used to create performance models, we identified four algorithms that are mainly used throughout the literature to generate performance models in the specific case of processors, microarchitecture explorations, or parallel applications: regression trees, random forest, multiple additive regression trees (MART), and artificial neural networks. Performance and power consumption prediction in the case of design-space exploration of general purpose CPUs has been achieved with random forests [23] and MART [24, 25]. Regression trees [26] have been used for performance and power prediction of a GPU. Artificial neural networks have successfully been used for performance prediction of a parallelized application [21], but also for workload characterization of general purpose processors [27, 28], superscalar processors [29], and microarchitectures [30]. A variant of the MART method has also been used for predicting performances of distributed systems [31]. Regression trees are usually less accurate, and its more robust version, the random forest, is usually preferred. Madougou et al. [25] used nvprof**,** a visual profiler, to collect performance metrics (cache miss, throughput, etc.) and events of CUDA kernels running on NVIDIA GPUs. The data are stored in a database and further used for model building. This approach seems very promising but cannot be easily adapted to the needs of avionics, as explained previously.

Another problem with tree-based methods is their low performance to predict values from predictors out of the range of the values of the observations used to train them (extrapolation). Recent work on hybrid models generated from a mix of machine learning and first-principle models has also yielded good results for similar applications [32–34].

3. Avionics Graphic Hardware Performance Benchmarking

There are two main steps in the creation of our performance models. First, a benchmark must be executed to gather performance data for various scene characteristics. The GPU benchmarking consists in itself in the generation of a customizable synthetic 3D scene and in the analysis of the render time of each frame. Second, performance models are generated with machine learning from the performance data obtained in the first step. For our experimental purposes, we add a model validation phase to evaluate the predictive power of those performance models by comparing the predictions with the render time of a customizable and distinct validation 3D scene. Figure 1 presents this dataflow. The remainder of this section will present the requirements and the implementation of our proposed avionics GPU benchmarking tool.

Performance data acquisition for a piece of hardware is achieved with our benchmarking tool as follows. First, a synthetic scene is generated according to various parameters. Then, the scene is rendered and explored by following a specific camera movement pattern. Finally, a last analysis step is performed to evaluate for each frame the percentage of the number of vertices of the scene that has been rendered. We use a study case from an industrial partner to enable us to enumerate the various characteristics of graphical data that might have an impact on the rendering performance of an avionic graphical application. The study case was a SVS using tile-based terrain rendering.

System performances can be measured in several ways by test benches. For a 3D graphics system, one of the most effective methods is to try to reproduce the behaviour of a real system [35]. Thus, we characterized our industrial partner's study case to extract an exhaustive list of all the graphical features to take into account while implementing our benchmarking tool. Our tool tests these features one by one, by inputting a set of values to test per feature. It then outputs results that give a precise idea of how each graphical feature impacts rendering performance. The graphical features involved in our industrial partner's study case and which we tested in our tool are as follows:

(1) Number of vertices per 3D object (terrain tile)

(2) Number of 3D objects per scene

(3) Size of the texture applied on 3D objects

(4) Local illumination model, either per-vertices (flat) or per-fragment (smooth)

(5) Presence or absence of fog effect

(6) Dimension of the camera frustum

(7) Degree of object occlusion in the scene

Those parameters follow the hypothesis that the principal factors that would influence the performances are directly related to the amount of data sent through the graphic pipeline. The amount of work required by the graphic hardware would be in relation to the amount of data needed to be rendered because of the amount of operations required to send all these data across the pipeline.

Compared to the list of features tested in a contemporary graphical benchmark, this set list brings us back to the beginnings of 3D graphics era. The reason why this list is made of basic graphic features is because critical graphical avionic software uses a version of OpenGL which is stripped

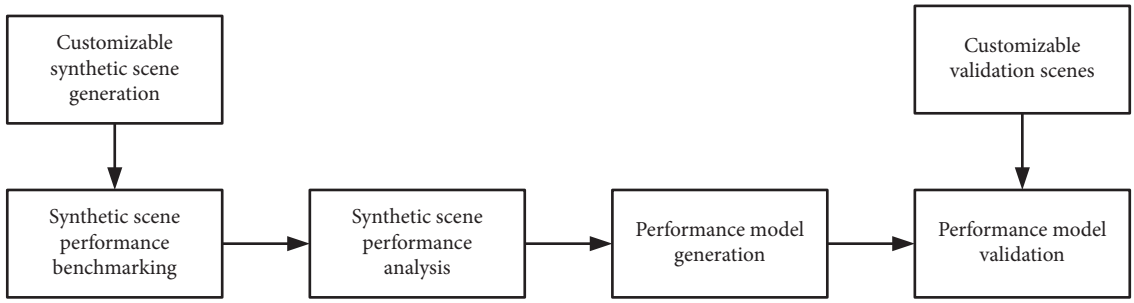


Figure 1: Dataflow of the proposed tool in an experimental context.

down to its simplest form: OpenGL Safety Critical (SC) [36]. This simple API was specifically designed for safety-critical environments because it is easier and faster to certify. OpenGL SC was also designed for embedded system limited hardware resources and processing power.

Nowadays, graphical benchmarks are designed for new generation graphic cards, game consoles, or smartphones. These platforms were designed to deliver a gigantic graphical processing power in a noncritical environment. It is thus normal that this kind of product has its own set of benchmarking tools. Basemark® ES 2.0 [37] (and now Basemark ES 3.0 [13]), one of the most popular graphical benchmarking suite for GPUs used in the mobile device industry, offers benchmarks that are not suitable for avionics platforms for several reasons:

(1) It uses OpenGL ES 2.0 which is an API too rich and complex to be a reasonable candidate for the full development of a certifiable driver on safety-critical platform [12].

(2) Sophisticated lighting methods are tested like per-pixel lighting (used to create bump mapping effects), lightmap (used to create static shadow effects), or shadow mapping (used to create dynamic shading effects). Although these lighting methods offer rich visual effects, they remain pointless within an avionic context, where rendering accuracy is the main concern for 3D graphics.

(3) Various image processing effects are tested like tilt and shift effect, bloom effect, or anamorphic lens flares effect. More complex lightning models such as Phong interpolation and particle effects are also tested (usually by implementing shaders). Once again, these visual effects will not enhance the accuracy of rendering, and also shaders are not supported in OpenGL SC.

(4) Results and metrics resulting from these kinds of benchmarks are usually undocumented nebulous scores. Those scores are useful when we want to compare GPUs between them, but they cannot be helpful when system architects want to figure out if a particular piece of hardware satisfied a graphical software processing requirement. When designing a safety-critical system, metrics like GPU's RAM loading time (bandwidth, latency), level of image detail (maximum number of polygons and 3D objects per scene), and maximum size of textures or the processing time per frame are much more significant data.

Since the current graphical benchmarking tools were not designed for a safety-critical environment, we thus decided to implement specialized benchmarking tool for the avionic industry.

Based on this analysis, from those 7 factors, we divide a tile-based synthetic scene that would evaluate rendering performance based on these factors. The benchmark tool takes a list of "tests" as input. Each test influences the generation of the procedural 3D scene by manipulating a combination of those factors. The output of the benchmark tool is a file with time performance according to the inputted characteristics. Each test is designed to evaluate the performance of the scene rendered by varying one of the characteristics and keeping fixed every other. Consider, for instance, the tile resolution test, for each value, the benchmark will be executed, and a vector of performance will be outputted. During this test every other characteristics (e.g., the number of tiles or the size of textures) shall be fixed. It is important to mention that tile-based scenes are stored as height maps or even dense 3D scenes, removing the need for analyzing the number of triangles or faces because it can always be derived or approximated from the number of vertices.

3.1. Synthetic Scene Generation. Each tile of our synthetic scene contains a single pyramidal-shaped mesh. We used this shape because it can model various ground topography by varying the height of the pyramid. Furthermore, it enables the possibility to have an object occlusion when the camera is at a low altitude and it is oriented perpendicularly to the ground. Also, this shape is easy to generate from a mathematical model. The remaining of this subsection presents how the visual components of the procedural 3D scene are generated and how they help to produce more representative performance data.

3.1.1. Tiles' Dimensions. Tiles have a fixed dimension in OpenGL units, but the number of vertices it can contain can vary depending on the corresponding benchmark input value. To simplify the vertices count of our models, we use a per-dimension count c for the square base of the pyramids,

meaning that each tile has a resolution of c^2 vertices. When the perspective distortion is not applied, each vertex of the pyramid is at equal distance of its neighbours in the XZ plane (Figure 2).

3.1.2. Noise. To further reproduce a realistic ground topography, we add random noise to the pyramid faces to unsmooth them. The quantity of noise applied is more or less 10% of the height of the pyramids and is only applied to the Y coordinates (attributed to the height) as shown in Figure 3. This proportionality helps to keep the general shape of the pyramid, regardless of its height.

3.1.3. Grid Generation. The grid of tiles is generated according to the corresponding benchmark input value. As for the tile resolutions, the grid size is measured as a per-dimension value v, meaning that the total grid size is v^2, and thus the grid has a square shape. In a real context, a LOD functionality is usually implemented, making farthest tiles load at a lower resolution and nearest tiles at a full resolution. However, because we evaluate the worst-case execution performance of the hardware, every tile has full resolution.

3.1.4. Pyramid Height. The height of the pyramids varies from tile to tile, depending on their position in the tile grid, but the maximum height will never exceed the quarter of the length of the scene. This constraint enables the possibility to have various degrees of object occlusion for the same scene, depending on the position and orientation of the camera. Because of the positioning of the camera and the movement pattern (explained previously), the bigger the tile grid is, the higher the pyramids are. To obtain consistent scene topologies for each benchmark test, the pyramid height is calculated from the index of the tile in the grid (Figure 4) and is always a factor of two from the maximum pyramid height.

3.1.5. Texture Generation. The OpenGL SC API requires the use of texture dimensions that are powers of two. For simplicity, we create RGB-24 procedural textures which consist of an alternation of white and black texels. For each vertex of the tile, the texture itself and the texture coordinates are computed before the frame rendering timer starts. In real cases, this information is normally already available in some kind of database and not generated in real time, so it should not be taken into account by the timer measuring the period taken to draw the frame.

3.2. Camera Movement Pattern. According to the case study, there are three typical use cases for the camera movement and position patterns:

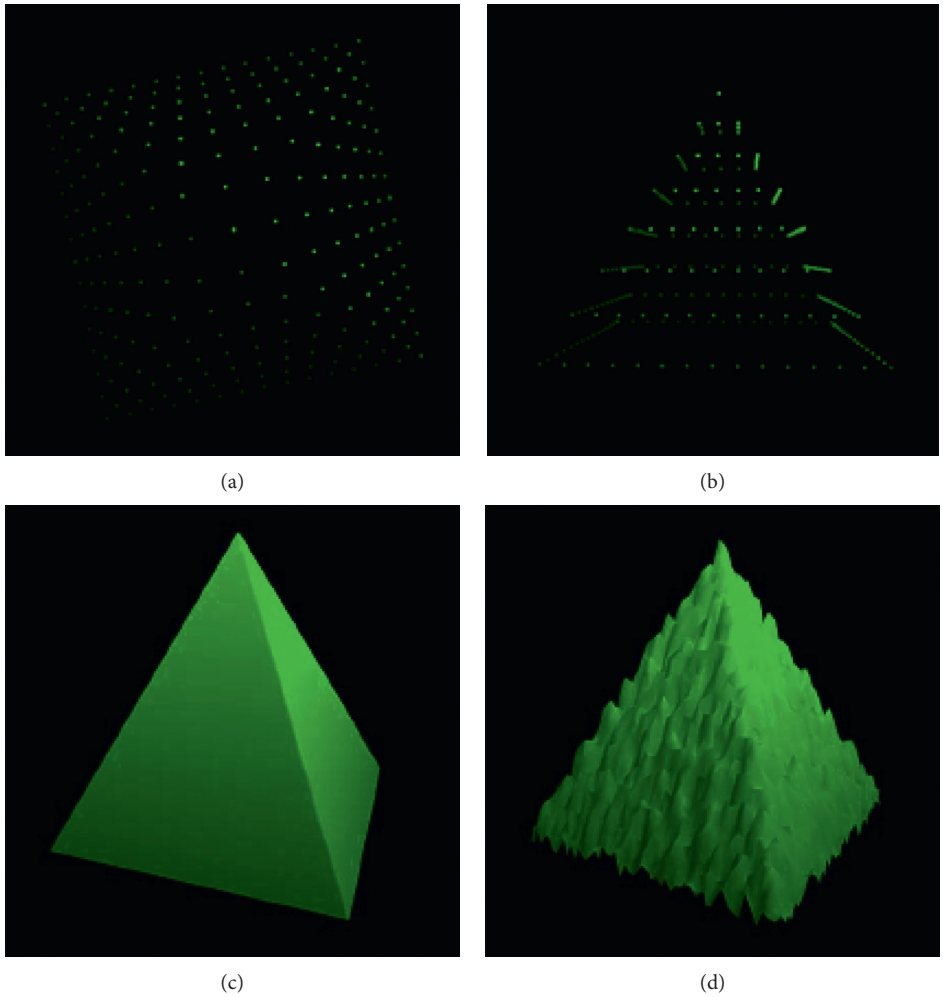
(1) Low altitude: a small percentage of the scene is rendered with the possibility of much object occlusions
(2) Midrange altitude: about half of the 3D objects are rendered with possibly less object occlusions
(3) High altitude: the whole scene is potentially rendered with low chances of object occlusions

For each test of a benchmark, the camera position goes through each of these use cases. To achieve this, the camera always starts at its maximum height over the tile at the middle of the grid. The camera then performs a 360 degrees rotation in the XZ plane, while also varying its inclination over the Y-axis depending on its height. After each 360 degrees rotation, the camera height is reduced and there are eight possible values for each test. The inclination angle over the Y-axis is not constant throughout the various heights taken by the camera, in order to cover the highest possible number of viewpoints of the scene. At the maximum height, the inclination leans towards the edges of the grid, and at the lowest height the camera points perpendicularly towards the ground. The camera inclination for every camera height is calculated with a linear interpolation between the inclinations, at maximum and minimum heights. Overall, each 360 degrees rotation of the camera will yield 32 frames for a total of 320 frames for each benchmark run.

The camera frustum is created to mimic the one used by the study case SVS. It implements a 45 degrees horizontal field of view and a vertical field of view corresponding to the 4 : 3 ratio of the screen, according to the OpenGL standard perspective matrix. Also, to maximize the precision of the depth buffer, it is desirable to show a maximum of vertices with the smallest frustum possible. The last important parameter is to define the maximum height of the camera. We set this limit to the value of the length of the scene in OpenGL units because the scene will be smaller than the size of the screen passed that length. Thus, the far plane of the frustum must carefully be chosen in regards of the scene length as the maximum depth of the scene will most likely vary accordingly.

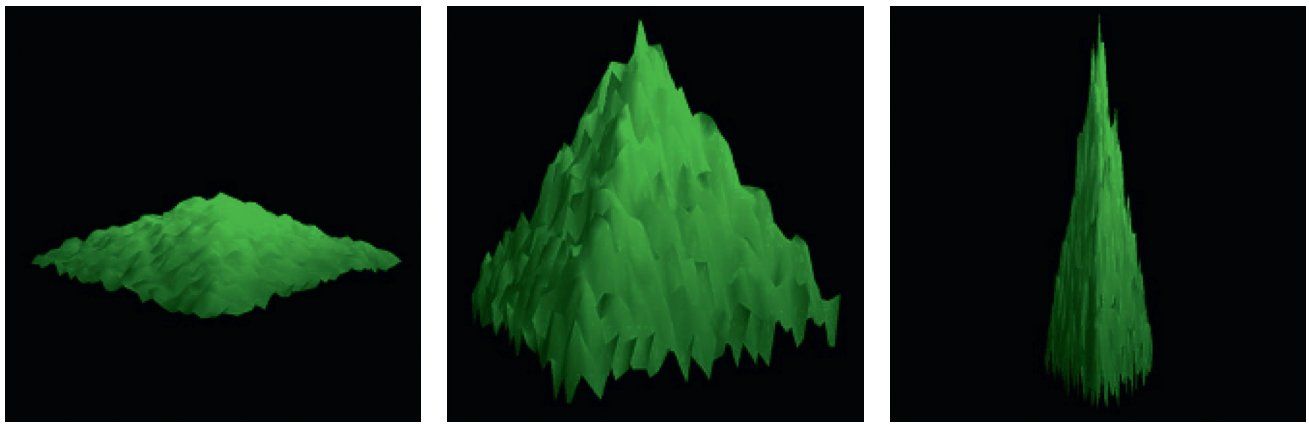
3.3. Loading Data to the Graphic Memory. To help reduce the randomness of the performance of the graphics hardware and due to the internal properties of most rendering pipelines, we apply the tipsify algorithm [38] to the vertex index buffer before sending it to the graphics pipeline. This should reduce the average cache miss ratio of the standard internal vertex program of the fixed pipeline. All the 3D data is loaded to the graphics memory before beginning the rendering and the performance timer. Because the scene is static, no further data need to be sent to the hardware, so the loading time does not influence the overall performance. This would not be the case in a real context, but as presented in Section 2, the literature presents at least one method to estimate the influence of data loading during the rendering process. If needed, the benchmark tool can return the time required to load this static data from the main memory to the graphics memory.

3.4. Analysis of the Percentage of Scene Drawn. The data sent to the graphical hardware for rendering usually contain 3D objects that could be ignored during the rendering process because they are either unseen or hidden by other 3D



(a) (b) (c) (d)

Figure 2: Pyramid vertices generated with a c-by-c dimension top facing (a) and front facing (b). Pyramid rendered mesh without added noise (c) and with added noise (d).



(a) (b) (c)

Figure 3: Various intensity of noise depending on the height of the pyramid. Low-noise amplitude (a) to high-noise amplitude (c).

objects, due to their spatial positioning. Thus, culling methods are commonly used to avoid the rendering of such objects. These methods are (1) the frustum culling which ignores the rendering of triangles outside of the camera frustum, (2) the back-face culling which ignores the rendering of triangles that are facing away from the camera orientation, and (3) the Z-test which ignores the per-fragment operations such as the smooth lighting.

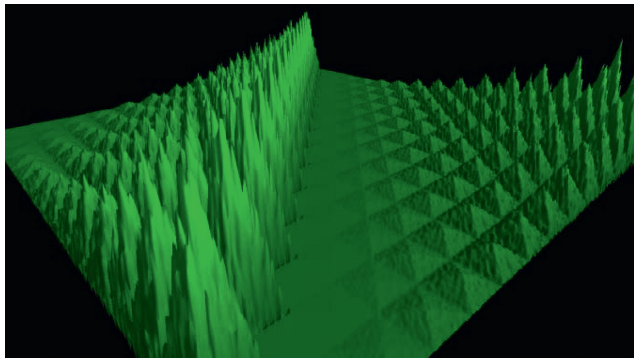


Figure 4: Overall generated synthetic scene with pyramids height varying according to their position in the grid.

As mentioned earlier, the final step of the benchmarking process is the analysis of the percentage of the vertices of the scene that were used during the rendering process. To do this, we count the number of vertices that are in the camera frustum and also that are part of front facing surfaces. We used a geometric approach by comparing the position of each vertex with the six planes of the frustum in order to determine if the vertex is inside it or not. If it is, the next step is to determine if it is front or back facing. To do this, we first transform the normal of the surface, of which the vertex is a part, from world-space to camera-space. Then, we compare the angle between the normal and the camera-space eye vector, which is a unit vector pointing in the Z-axis. If the angle is between 90 and 270 degrees, then the vertex is considered to be front facing. Finally, we can evaluate the percentage of vertices drawn as the size of the set of vertices that pass both tests, divided by the total number of vertices in the scene.

3.5. Rendering Performance Metrics. The performance metric returned by the benchmark is the instantaneous frame per second (IFPS), which is measured for each frame by inverting the time it took to render the frame. It is more desirable than a moving-average FPS over multiple frames because we can then apply more specialized smoothing operations to eliminate further any outliers. On the other hand, the benchmark uses a vertical sync of twice the standard North American screen refresh rate: 120 Hz. We found that not using vertical sync yields a very high rate of outliers for IFPS greater than 120. Also, using a vsync of 60 FPS may create less accurate models as most of the scene characteristics will yield the maximum frame rate. Finally, to ensure the proper calculation of the IFPS for each frame, we use the *glFinish* command to synchronize the high resolution timer with the end of the rendering.

4. Avionics Graphic Hardware Performance Modeling

Performance modeling of 3D scenes by using the characteristics of the latter is challenging because it is hard to take into account the noise made by random events in the abstracted hardware (processor pipeline stall, branching, etc.). The first step in the creation of a performance model is to evaluate which of the benchmark scene characteristics best explains the variance of FPS. In preliminary tests, we ran the benchmark by varying the values of one characteristic, while keeping fixed every other. This is repeated for each characteristic until all of them have been evaluated. We concluded that the size of the grid of tiles and the resolution of vertices in each tile are the most significantly well predicted characteristics by the machine learning algorithms. The size of the screen and the size of the texture also contribute to the variation of the FPS, but they were harder to model using our method. To predict IFPS in terms of texture sizes and screen sizes, a distinct performance model must be generated for each of their combinations, by training it with a subset of every possible combination of grid size, tile resolution, and percentage of vertices rendered. This limitation is being worked on as a future contribution. To generalize tile-based scenes to dense voxelized scenes without fixed resolutions in each voxel, the tile resolutions can be replaced by the mean number of vertices per voxel.

Besides, another key factor in the creation of good models is the fact that they can be generated efficiently without the need to feed the FPS obtained for every possible combination of scene characteristics to the learning algorithms. This number has been calculated to be about 58 million combinations of grid size and number of vertices. Compared to this number, the number of combinations of texture sizes (10) and of screen sizes (5) is relatively small. Generating a distinct performance model for each combination of texture and screen sizes would be a more trivial task, if the number of combinations of grid size, tile resolution, and percentage of vertices rendered could be reduced. To organize those performance models, we create a three-level tree, where the first two levels represent the combinations of texture and grid size. The third level contains a leaf pointing to a nonparametric model trained with machine learning that predicts IFPS in terms of grid sizes and tile resolutions. Those nonparametric models are created by feeding the machine learning algorithms with only a small percentage of the performance of the whole combinations of grid sizes, tile resolutions, and percentage of vertices, while keeping fixed the texture and screen size characteristics according to the leaf parents. The choice of the subset of total grid sizes and tile resolution combinations to use is chosen by selecting those inducing the worst-case rendering performance. We run the benchmark only twice by running the tile resolution and grid size variation tests and by concatenating the output performance vectors of each. The characteristics evaluated by the benchmark for the training dataset are shown in Table 1.

4.1. Machine Learning Algorithms Configuration. As stated in Section 2, three machine learning algorithms are of special interest for the task of performance prediction in the case of processors and parallel applications: random forest, MART, and artificial neural networks. We offer the comparison between the predictive powers of nonparametric models trained with each of these algorithms in order to determine

Table 1: Values used for the tile resolution grid size tests.

Variation test	Values of tile resolution	Values of grid size
Tile resolution	7; 9; 13; 17; 21; 25; 31; 37; 45	25
Grid size	25	7; 9; 13; 17; 21; 25; 31; 37; 45

Table 2: Parameters for the machine learning algorithms used.

Algorithm	Parameter nature	Optimal range
Artificial neural network	Number of hidden layers	1
	Number of nodes in the hidden layer	[5; 15]
MART	Number of weak learners	[500; 1000]
Random forest	Number of bootstrapped trees	[50; 150]

which one is the most suited for this application. We chose those algorithms to reflect the previous work of the scientific programming community as we felt they would be the best fit for GPU prediction. Other machine learning methods are present in the literature such as Bayesian networks and gradient boosting machines, to name a few, but have not been considered in the current experiment. Each of these algorithms has to be configured before its use: the number of hidden layers and the number of nodes per layers in the artificial neural networks, the number of bootstrapped trees in the case of random forest, or the number of weak learners in the case of MART. Most of the time, there is no single optimal parameter. It usually takes the form of a range of values. These ranges were found during preliminary experimentation and are given in Table 2. In the case of artificial neural networks, we used a multilayer feedforward perceptron trained with backpropagation based on the Levenberg–Marquardt algorithm. We also try to improve the problem generalization and reduce overfitting by using early stopping methods and scaling the input performance values in the range $[-1, 1]$. Early stopping methods consist in stopping prematurely the training of the neural network when some conditions are met. It can be after a certain number of epoch (1000 in our case), when the performance metric has reached a certain threshold (mean squared error is 0.0). But the most commonly used early stopping method is to divide the training dataset in two subsets: a training subset (70% of the initial dataset) and a validation subset (remaining 30%). After each epoch, the neural network makes predictions using the validation dataset and training is stopped when those predictions are accurate enough (e.g., error metric between the prediction and the predicted value is lower than a certain threshold).

Because of the random nature of the optimality of a parameter value, we create three performance models for each machine learning algorithm. The first model uses the lowest parameter value, the second model uses the middle range one, and the last model uses the upper one. During the validation phase, we retain the model which yields the lowest prediction error.

4.2. Performance Data Smoothing. The performance data output by the benchmark itself is randomly biased because it cannot explain some of the variance of IFPS, which can vary even for scenes with similar characteristics. The fact that we only analyze the input and output of a graphical hardware black box partially explains the variance because many internal factors can influence the output for the same input: cache misses ratio, processor instruction pipeline, and instruction branching unpredictability to name a few. Because the program running on the hardware is fixed, we can assume that these factors are not enough random to make the analysis of input/output unusable for performance prediction. To reduce this noise in the benchmark output data, we apply an outlier-robust locally weighted scatterplot smoothing. This method, known as LOESS, requires larger datasets than the moving average method but yields a more accurate smoothing. Similar to the moving average method, this smoothing procedure will average the value of a data point by analyzing its k-nearest points. In the case of the LOESS method, for each point of the dataset, a local linear polynomial regression of one or two degrees is computed with their k-nearest points, by using a weighted least square giving more importance to data points near the analyzed initial point. The analyzed data point value is then corrected to the value predicted by the regression model. More information is available in [39]. In our case, the k-nearest points correspond to the IFPS of the 6 frames preceding and the 6 frames following the analyzed frame. The use of the k-nearest frames is possible because the characteristics of the scene between adjacent frames are spatially related. This can be generalized to most graphical applications because the movement of the camera is usually continuous.

4.3. Quantifying Scene Characteristics Equivalency. To further help the machine in the performance modeling, we transform the output format of the benchmark (IFPS in *terms of* the number of points, scene or grid size, and percentage of scene drawn), into a format that is more similar to the scene characteristics that will be given by the system designer (IFPS *in terms of* the number of points and scenes or grid size). Also, the tool can be more easily used if the percentages of scene drawn parameter could be omitted: to have to choose a percentage of scene drawn when querying the tool for predictions might lead to confusion. Thus, it is necessary to internally find a proportionality factor that can help evaluate the equivalency of performance between various points in the training data. The basic assumption is that the IFPS of a scene drawn with a certain set of characteristics ($IFPS_1$) will be somewhat equivalent to the IFPS of the same scene drawn with another set of characteristics ($IFPS_2$) if the characteristics of both scenes follow a certain proportionality. The first characteristic in this case is the size of the scene without accounting for depth: (v_1 for the first set of characteristics and v_2 for the other) either in OpenGL units or in the size of the grid of voxels or tiles in the case of tile-based applications. The other characteristic is the tile resolutions in each tile or voxel c_1 and c_2. As mentioned earlier, those concentrations and those sizes in the case of

our benchmark are expressed as a per-dimension value. Thus, they are always equal in 2D (length and width). For simplicity, we use the following notation:

$$c_i = c^2_{\text{width}i} = c^2_{\text{depth}i}, \tag{1}$$

and

$$v_i = v^2_{\text{width}i} = v^2_{\text{depth}i}. \tag{2}$$

It implies that

$$\text{IFPS}_1 \approx \text{IFPS}_2 \Leftrightarrow \frac{c_1}{c_2} \propto \frac{v_1}{v_2}. \tag{3}$$

Considering that a scene drawn at a certain percentage p_1 with a set of characteristics, then v_1 represents the fraction of the total v_2 area that is drawn as

$$v_1 = p_1 * v_2. \tag{4}$$

In this case, since v_1 and v_2 are taken from the same scene, we have $c_1 = c_2$. Therefore,

$$\frac{v_1}{v_2} = p_1 * \frac{c_1}{c_2}, \tag{5}$$

where p_1 is the proportionality factor.

This example uses a single scene with a single set of characteristics to help find the proportionality factor, but the formula can also be used to compare scenes with different initial characteristics, which is a powerful metric for extrapolation. During the design of the typical use case of our tool, we assumed that the designer would want to request a performance estimation of the rendering of the scene when it is entirely drawn, and not just drawn at a certain percentage (worst-case scenario). A way had to be found to use the p_1 factor during the training phase, but to remove the need to use it in the performance queries, once the model is generated.

From equation (5), we obtain

$$p_1 * \frac{c_1}{v_1} = \frac{c_2}{v_2}. \tag{6}$$

Then, we found that the machine learning can create slightly more precise models if p_1 is expressed in terms of the concentration of triangles instead of in terms of the concentration of vertices. Considering that in our tile-based application the number of *facesPerTile* is obtained as follows:

$$\#facesPerTile = (\sqrt{c} - 1)^2 * 2. \tag{7}$$

From the proportionality function (6) and from (7), we deduce

$$p_1 * \frac{\left(\sqrt{c_1} - 1\right)^2 * 2}{v_1} = \frac{\left(\sqrt{c_2} - 1\right)^2 * 2}{v_2} = K. \tag{8}$$

The left part of equation (8) can be obtained by the scene characteristics, and the benchmark output performance metrics provide a value K which in turn allows to find values for c_2 and v_2. We are thus capable of obtaining approximately equal IFPS values between that scene rendered with c_2 and v_2 at 100% and the same scene drawn with c_1 and v_1 at p_1 percent. The machine learning algorithms are then trained with a vector containing a certain number of tuples (IFPS *in terms of* K and the number of points drawn) where

$$\#pointsDrawn = p_1 * \#totalNumberOfPoints, \tag{9}$$

and

$$\#totalNumberOfPoints = c_1 * v_1. \tag{10}$$

The designer can then create a prediction query by inputting K and the number of points he desires to render without having to mind about a percentage of scene drawn p_1. The tool would then output an IFPS prediction for the input parameters.

4.4. Identifying the Percentage of Space Parameters Evaluated. The best way to predict the performance would be to evaluate the rendering speed of the scene with every combination of characteristics. In this case, we would not even need to create nonparametric models, but this could require the evaluation of millions of possibilities, ending in way too long computation times (about a year). This performance prediction tool thus evaluates a very small subset of all those combinations in reasonable time (about half an hour). In the following, we determine the percentage of the number of combinations that our tool needs to evaluate.

Given:

(i) $n_{\text{screen}} = \{640 \times 480,\ 800 \times 600,\ 1024 \times 768,\ 1152 \times 864,\ 1920 \times 960\}$, the discrete number of studied screen sizes

(ii) $n_{\text{texture}} = \{x \mid x = 2^i \wedge 1 \leq i \leq 10 \wedge x \in \mathbb{N}\}$, the set of studied texture sizes

(iii) $n_{\text{vertices}} = \{x \mid 1 \leq x \leq 1{,}300{,}000 \wedge x \in \mathbb{N}\}$, the set of all possible quantity of points

(iv) $n_{\text{grid}} = \{x^2 \mid 1 \leq x \leq 45 \wedge x \in \mathbb{N}\}$, the set of studied tile grid sizes such that each tile has the same size in OpenGL coordinates

We generalized the concept of tile-based scenes for any dense scene by removing the tile resolution concept and replacing it by the concentration of any number of vertices lower than 1,300,000 divided by any grid size in n_{grid}. We chose this maximum number of vertices and also this maximum grid size arbitrarily as it should cover most data volumes in most of the hardware use cases. We can then evaluate the total number of combinations of characteristics influencing the density of points for every studied screen and texture sizes N_{Total} as

$$N_{\text{Total}} = \left|n_{\text{vertices}}\right| * \left|n_{\text{grid}}\right| * \left|n_{\text{screen}}\right| * \left|n_{\text{texture}}\right| = 2{,}925{,}000{,}000. \tag{11}$$

The tool then needs only to test a small fraction of all those combinations to produce adequate performance models. As mentioned earlier, the tool only needs to run two tests of the benchmark to construct adequate models for a fixed screen and texture size. Each test is configured initially to execute the benchmark with nine varying test parameters as shown in Table 1.

Given:

(i) N_{Tool}, the total number of combinations of characteristics analyzed by the tool

(ii) $S_{\text{TrainingSet}} = 5760$, the size of any training dataset output by the benchmark for a grid size and tile resolution test with fixed texture and screen size which corresponds to the number of frames rendered for both tests of the benchmark

We then suppose that each frame rendered during the benchmarking represents one unique combination of those billions and find the number of combinations tested by the tool N_{Tool} as

$$N_{\text{Tool}} = S_{\text{TrainingSet}} * n_{\text{screen}} * n_{\text{texture}} = P_{\text{TrainingSet}} * 288{,}000. \tag{12}$$

The tool is then guaranteed to train successful models by using only about 0.0098% of the total combinations of characteristics.

5. Prediction Error Evaluation

This section presents the experimental setup and also the experimental considerations used to validate the predictive power of the performance model.

5.1. Experimental Setup. We used a Nvidia QuadroFX570, a graphic card model which should have consistent performance with the current avionic hardware. Since OpenGL SC consists of a subset of the full OpenGL API's functions and that this subset of functions is very similar to the OpenGL ES 1.x specification, we worked around the absence of an OpenGL SC driver by using an OpenGL ES 1.x simulator which transforms the application's OpenGL ES 1.x function calls to the current installed drivers which is OpenGL. A meticulous care has been taken to make sure to use only functions available in the current OpenGL SC specification with the exception of one very important method, named *vertex buffer objects*.

The experiment begins in the training phase with the generation of the performance models as presented in Section 4. The prediction power of those models is then validated during the validation phase for which an industrial scene is benchmarked many times with varying characteristics. Those performances are then compared to the predictions generated by the models. The following sections explain this validation phase in detail.

5.2. Performance Model Validation. To validate the performance model created, we used a 3D scene from the World CDB representing a hilly location in Quebec, Canada. The scene was subsampled to various degrees to enable the validation of the model at various resolutions. The models were first validated for their interpolation predictive power by comparing the predictions with the validation scene rendered with characteristics similar to the ones used to train the models. The models were then validated for their extrapolation predictive power with the same method but by rendering the validation scene with characteristics untreated during the model training. It is also important to select well those characteristics in order to produce scenes that are not too easy to render. Because it is easier for the models to predict the maximum V-synced FPS, the validation scene characteristics should be selected in a way that makes the rendering operations generate various percentages of frames drawn with maximum IFPS. We analyzed the influence of having about 0%, 50%, or 100% of frames in the dataset drawn with maximum IFPS. As with the synthetic scenes, the whole validation scene is loaded in graphic memory prior to rendering the scene. Therefore, the loading time is not taken into account during the FPS calculation.

To validate a model, the benchmark is executed, but instead of displaying the usual synthetic scene, it renders the one from the World CDB subsampled according to the various parameters shown in Table 3. Figure 5 shows an example of the rendering of a CDB tile in our application at various resolutions. The output of the validation dataset is then smoothed in the same way as the training dataset. The size of each dataset is 5760 observations and is closely related to the number of frames produced by each run of the benchmarking tool from Section 3 (320 frames per run).

We then compare the smoothed instantaneous FPS of each frame to the predicted ones with the following metrics.

5.3. Metric Choice and Interpretation. The choice of a metric to evaluate the prediction errors is still subject to debate in the literature [40]. Especially in the case of models generated with machine learning, the error distribution are rarely normal-like and thus more than one metrics are commonly used to help understand and quantify the central tendency of prediction errors. We use the mean absolute prediction error MAE presented in equation (9) and also the rooted-mean-squared prediction error RMSE presented in equation (10). To conform to the literature, we also give the MAE in terms of relative prediction errors PE presented in equation (13). Because the error distributions will be most likely not normal, we also give the median error value which could yield the most significant central tendency. This last metric contribution to the understanding of the distribution is to indicate that 50% of the prediction errors are lesser than its value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{\text{FPS}}_i - \text{FPS}_i\right|, \tag{13}$$

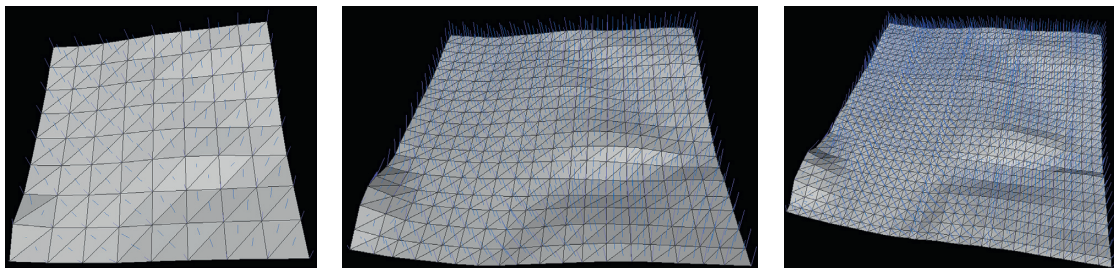
$$\text{RMSE} = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}\left(\widehat{\text{FPS}}_i - \text{FPS}_i\right)^2}, \tag{14}$$

$$\text{PE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\widehat{\text{FPS}}_i - \text{FPS}_i\right|}{\text{FPS}_i} * 100\%, \tag{15}$$

where $\widehat{\text{FPS}}_i$ is the i-th prediction, FPS_i is the i-th measured value, and n is the number of prediction/measurement pairs.

Table 3: Values of the World CDB scene characteristics.

Dataset	Name	Varied parameter	Tile resolution	Grid size	% of frames with maximum IFPS
Validation (CDB scene)	#1	Grid size	25	7; 9; 13; 17; 21; 25; 31; 37; 45	44.38
	#2	Grid size	17	7; 9; 13; 17; 21; 25; 31; 37; 45	44.51
	#3	Tile resolution	7; 9; 13; 17; 21; 25; 31; 37; 45	25	0%
	#4	Tile resolution	7; 9; 13; 17; 21; 25; 31; 37; 45	17	100%



(a) (b) (c)

Figure 5: Mesh and normals of one tile of the validation scene sampled at a resolution of 9×9 (a), 19×19 (b), and 31×31 (c) shown before applying the randomized texture.

6. Results

Results (e.g., Figures 6–9) show that the prediction errors do not follow a normal distribution, but even though there is some skewness in them, they still retain a half-bell like appearance. The most adequate central tendency metric is thus the median. Furthermore, the artificial neural network has a better prediction than the two other algorithms most of the time, followed closely by random forest. Table 4 shows that the gap between the median prediction errors of both of these algorithms never exceeds 1 FPS. On the other hand, the MART method performed poorly on all datasets with a gap of up to about 12 FPS. Also, the performance models trained with artificial neural networks made quite good predictions in an interpolation and extrapolation context, as shown by the central tendencies of errors of all validation datasets confounded. Another explanation for the low performance of the tree-based methods comes from the fact that they do not perform well for extrapolation as mentioned in Section 2. The central tendency gap between those two sets never exceeds 4 FPS in this experiment. By analyzing the maximum absolute prediction error of most datasets, we see that there is a small chance that a very high prediction error is produced. These high errors can be as high as 43 FPS in the third dataset. Even though the general accuracy of the model is pretty good, the precision can be improved. Hopefully, the mode of each error distribution is always the first bin (range of values) of the histogram, which means that the highest probability of a prediction error is always the lowest error.

7. Discussion

Figure 10 somehow illustrates why parametric modeling would perform poorly as it would be hard to assume a geometric relationship between the IFPS and the predictors. Preliminary work also demonstrated the inefficiency of those methods, and thus they were not included in this work.

Because there is a very small chance (less than 1%) that a prediction might have a high error, the final prediction offered by the tool for a combination of characteristics should be a weighted average or a robust local regression of a small set of performance with similar scene characteristics, in order to help reduce the influence of these prediction outliers. Also, the scene used to train and validate our models are all dense, and thus our experiment cannot imply any significance for sparse scenes. But, as most graphical applications in an avionics context uses dense scenes, this should not be a major issue.

We also do not use fog effects in our tools which is a feature that could be used in a real industrial context as this feature will be part of next releases of OpenGL SC.

On the other hand, because of the high costs of avionics hardware, we had to abstract desktop graphic hardware behind an OpenGL ES 1.x environment to simulate an OpenGL SC environment which might weaken the correlation of our results to the ones that would be obtained with real avionics hardware. Related to this issue, we used standard desktop graphics hardware for the same reason. The presented method has been developed to abstract the nature of the hardware underlying the API, and the use of full-fledged avionics graphics hardware would improve the credibility of the results. However, this does not mean that our method would work for any use case of standard desktop graphics hardware. It has been designed for the specific problematic of the avionics industry: fixed graphics pipeline, use of OpenGL SC (or equivalent), and abstraction of the underlying hardware. This is fundamentally the opposite of the average desktop graphics hardware use case.

We also used only one validation scene subsampled into four distinct scenes. It could be of interest to reproduce

Table 4: Central tendencies of the prediction error distributions for each machine learning algorithm and for each validation dataset.

Validation dataset name	Supervised learning algorithm	RMSE (FPS)	Relative prediction error (%)	Mean absolute prediction error (FPS)	Median absolute prediction error (FPS)
#1	Random forest	2.71	2.12	1.36	0.45
	MART	9.37	10.15	6.85	4.86
	Neural net	2.29	1.99	1.26	0.50
#2	Random forest	10.87	8.74	5.85	2.06
	MART	16.53	17.65	12.16	10.39
	Neural net	7.90	8.50	5.16	3.51
#3	Random forest	5.94	5.98	4.10	2.79
	MART	9.99	10.60	7.51	6.15
	Neural net	2.97	3.03	2.39	2.13
#4	Random forest	15.80	9.12	10.94	3.98
	MART	30.20	18.78	22.53	15.02
	Neural net	4.02	2.58	3.10	3.01

Figures 6–9 present the error distribution of each nonparametric performance model for the four validation scenes.

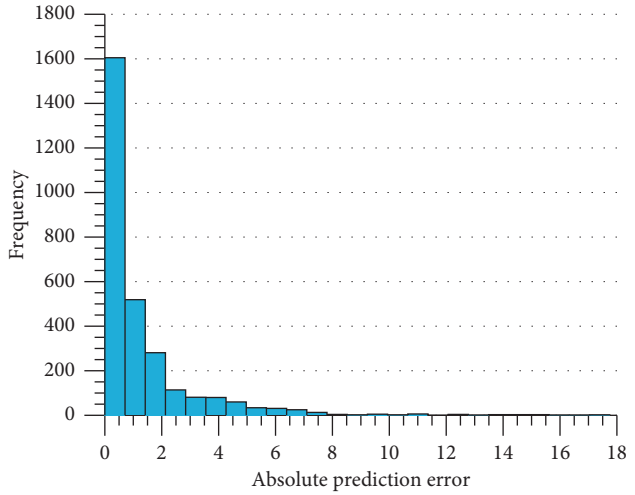


Figure 6: Prediction error distribution of the validation dataset #1 for the artificial neural network.

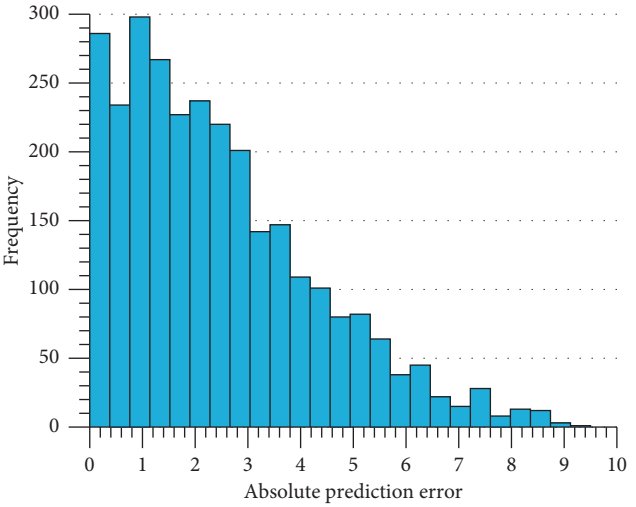


Figure 8: Prediction error distribution of the validation dataset #3 for the artificial neural network.

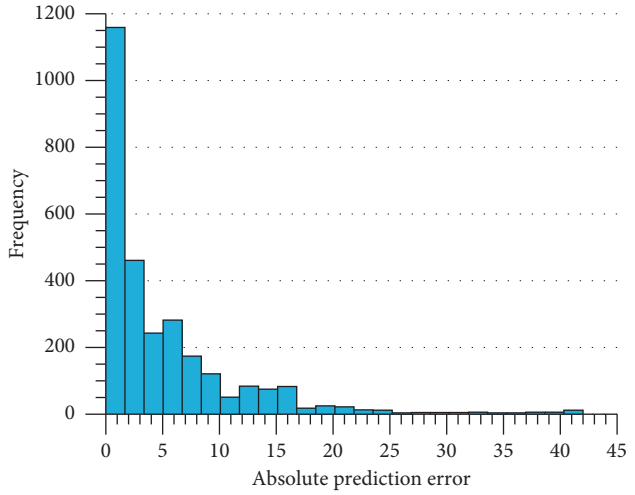


Figure 7: Prediction error distribution of the validation dataset #2 for the artificial neural network.

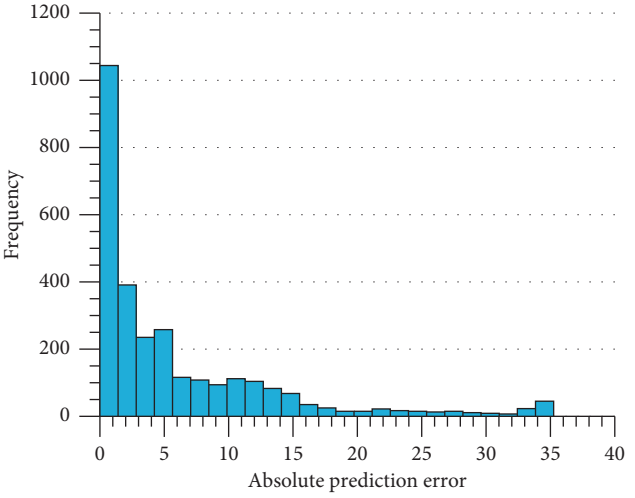


Figure 9: Prediction error distribution of the validation dataset #4 for the artificial neural network.

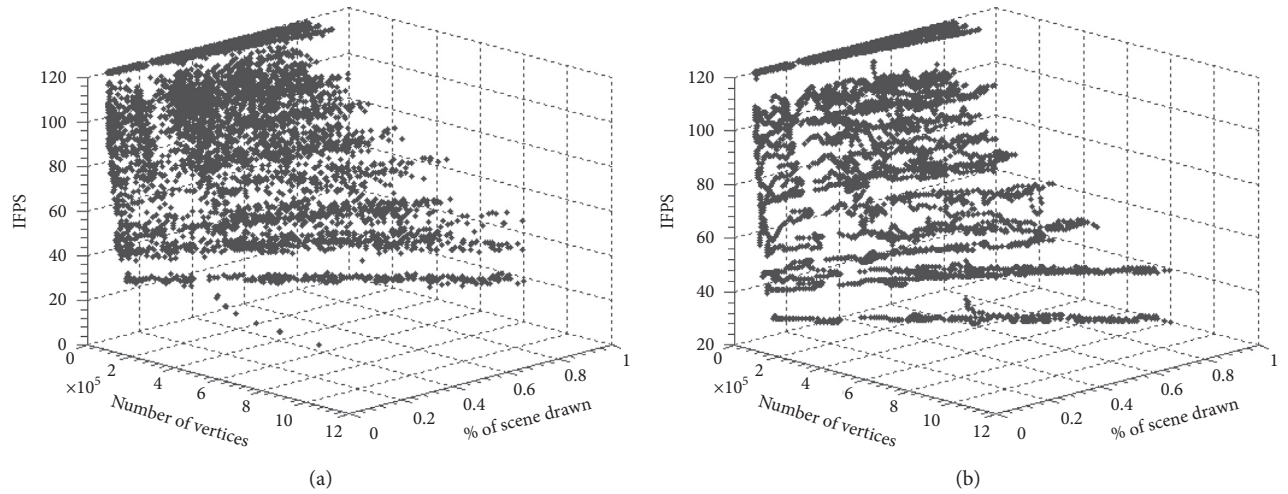


Figure 10: Comparison between unsmoothed (a) and smoothed (b) performance data.

the experiment with a bigger dataset of dense scenes. Considering our results as reproducible, the prediction errors made by our tool would be low enough for industrial use. Otherwise, we confirm that the central tendency of our prediction error distributions is similar to the ones presented in the literature, when these methods are applied to other kinds of hardware or software performance predictions.

8. Reproducing the Results

As our method includes two important experimental steps: dataset generation and machine learning, we foresee that the reader may want to reproduce experimentally either one or both of these experimental steps by using our dataset or by regenerating their own datasets to train the neural network.

To generate our training dataset, we used our procedurally generated 3D scenes. The reader may either want to create its own rendering tool following our specifications or we could make the C++ code available [41]. To generate the validation dataset, we rendered a private 3D scene, not available to the public, but any 3D scene consisting of a rendered height field could be used instead. We provide our raw datasets [41], more precisely the performance data generated by our tool for both 3D scenes, raw and unsmoothed.

Using our datasets or with a dataset generated by a third party, the actual training of performance models can be reproduced. The reader will have to smooth the data as described using the LOESS algorithm and use the Matlab Curve Fitting Toolbox to achieve this. Then, the reader will have to generate the performance models with the Neural Network ToolBox. Our Matlab code could also be made available [41].

9. Conclusion

We have presented a set of tools that enable performance benchmarking and prediction in an avionics context. These were missing or not offered in the literature. We believe that avionics system designers and architects could benefit from these tools as none other are available in the literature. Also, the performance prediction errors were shown to be reasonably low, thus demonstrating the efficacy of our method.

Future work will include the development of a performance-correct simulator for avionics graphics hardware and also the addition of other scene characteristics like fog effects or antialiasing in the performance models. Also, it is of interest to evaluate the possibility to enhance our modeling by using Bayesian networks, gradient boosting machines, and hybrid models made from machine learning. Finally, we plan to automate more our processes and then experiment different use cases and parameters. This will help us to determine with more precision the upper bound of the cost reduction.

Data Availability

The data that support the findings of this study not available in [41] can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interests.

Acknowledgments

Special thanks are due to CAE Inc. for providing experimental material, industrial 3D scenes from the World CDB. We would also like to thank our other partners, contributors, and sponsors (CMC Electronics, CRIAQ, Mitacs, and NSERC) who particularly helped for the completion of two master theses [42, 43], certain results of which inspired this publication.

References

[1] L. J. Prinzel and L. J. Kramer, "Synthetic vision system," US Patent LF99-1309, 2009, https://ntrs.nasa.gov/search.jsp?R=20090007635.

[2] M. Dutton and D. Keezer, "The challenges of graphics processing in the avionic industry," in *Proceedings of the 29th Digital Avionics Systems Conference*, Salt Lake City, UT, USA, October 2010.
[3] V. Hilderman, T. Baghai, L. Buckwalter et al., *Avionics Certification: A Complete Guide to DO-178 (Software), DO-254 (Hardware)*, Avionics Communication Inc., Leesburg, VA, USA, 2007.
[4] Nsight, Nvidia, https://developer.nvidia.com/nsight-graphics.
[5] PerfStudio, GPU, https://gpuopen.com/archive/gpu-perfstudio/.
[6] Standard Performance Evaluation Corporation, "What is this thing called "SPECviewperf®"? https://www.spec.org/gwpg/gpc.static/whatis_vp8.html.
[7] "ARINC-661," https://www.presagis.com/en/product/arinc-661/.
[8] Z. Wang, Y. Hui, and X. Zhou, "A simulation method of reconfigurable airborne display and control system," in *Proceedings of the First Symposium on Aviation Maintenance and Management-Volume 2*, pp. 255–263, Springer, Berlin, Germany, 2014.
[9] C. Haskins, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities (ver. 3)*, International Council on Systems Engineering, San Diego, CA, USA, 2006.
[10] L. Lavagno, G. Martin, and L. Scheffer, *Electronic Design Automation for Integrated Circuits Handbook-2 Volume Set*, CRC Press, Boca Raton, FL, USA, 2006.
[11] *OpenGL SC Overview*, https://www.khronos.org/openglsc/.
[12] P. Cole, "OpenGL ES SC-open standard embedded graphics API for safety critical applications," in *Proceedings of the 24th Digital Avionics Systems Conference*, Washington, DC, USA, October 2005.
[13] "Basemark GPU1.1," https://www.basemark.com.
[14] S. S. Baghsorkhi, M. Delahaye, S. J. Patel, W. D. Gropp, and W.-M. W. Hwu, "An adaptive performance modeling tool for GPU architectures," in *Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming-PPoPP '10*, Bangalore, India, January 2010.
[15] S. Hong and H. Kim, "An analytical model for a GPU architecture with memory-level and thread level parallelism awareness," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, Austin, TX, USA, June 2009.
[16] K. Kanter, "Predicting AMD and Nvidia GPU performance," April 2011, https://www.realworldtech.com/amd-nvidia-gpu-performance/.
[17] Y. Zhang and J. Owens, "A quantitative performance analysis model for GPU architectures," in *Proceedings of the 2011 IEEE 17th International Symposium on High Performance Computer Architecture*, San Antonio, TX, USA, February 2011.
[18] M. Boyer, J. Meng, and K. Kumaran, "Improving GPU performance prediction with data transfer modeling," in *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*, Cambridge, MA, USA, May 2013.
[19] P. Joseph, K. Vaswani, and M. Thazhuthaveetil, "A predictive performance model for superscalar processors," in *Proceedings of the 2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*, Orlando, FL, USA, December 2006.
[20] B. C. Lee and D. Brooks, "Accurate and efficient regression modeling for microarchitectural performance and power prediction," in *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems-ASPLOS-XII*, San Jose, CA, USA, October 2006.
[21] B. Lee and D. Brooks, "Illustrative design space studies with microarchitectural regression models," in *Proceedings of the IEEE 13th International Symposium on High Performance Computer Architecture*, Scottsdale, AZ, USA, February 2007.
[22] G. Marin and J. Mellor-Crummy, "Cross-architecture performance predictions for scientific applications using parameterized models," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, New York, NY, USA, June 2004.
[23] Y. Zhang, Y. Hu, B. Li, and L. Peng, "Performance and power analysis of ATI GPU: a statistical approach," in *Proceedings of the IEEE Sixth International Conference on Networking, Architecture, and Storage*, Dalian, Liaoning, China, July 2011.
[24] B. Li, L. Peng, and B. Ramadass, "Accurate and efficient processor performance prediction via regression tree based modeling," *Journal of Systems Architecture*, vol. 55, no. 10–12, pp. 457–467, 2009.
[25] S. Madougou, A. L. Vabanescu, C. D. Laat, and R. V. Nieuwpoort, "A tool for bottleneck analysis and performance prediction for GPU-accelerated applications," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Chicago, IL, USA, May 2016.
[26] E. Ould-Ahmed-Vall, J. Woodlee, C. Yount, K. A. Doshi, and S. Abraham, "Using model trees for computer architecture performance analysis of software applications," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, San Jose, CA, USA, April 2007.
[27] R. M. Yoo, H. Lee, K. Chow, and H.-H. Lee, "Constructing a non-linear model with neural networks for workload characterization," in *Proceedings of the IEEE International Symposium on Workload Characterization*, San Jose, CA, USA, October 2006.
[28] D. Nemirovsky, T. Arkose, and N. Markovic, "A machine learning approach for performance prediction and scheduling on heterogeneous CPUs," in *Proceedings of the 29th International Symposium on Computer Architecture and High Performance Computing*, Campinas Sao Paulo, Brazil, October 2017.
[29] P. J. Joseph, K. Vaswani, and M. Thazhuthaveetil, "Construction and use of linear regression models for processor performance analysis," in *Proceedings of the Twelfth International Symposium on High-Performance Computer Architecture*, Austin, TX, USA, February 2006.
[30] E. İpek, S. A. Mckee, R. Caruana, B. R. de Supinski, and M. Schulz, "Efficiently exploring architectural design spaces via predictive modeling," in *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII)*, San Jose, CA, USA, October 2006.
[31] E. Thereska and G. R. Ganger, "Ironmodel: robust performance models in the wild," in *Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Annapolis, MD, USA, June 2008.
[32] S. Ren, Y. He, and K. S. Elnikety, "Exploiting processor heterogeneity in interactive services," in *Proceedings of the 10th International Conference on Autonomic Computing (ICAC)*, San Jose, CA, USA, June 2013.
[33] C. Stewart, T. Kelly, A. Zhang, and K. Shen, "A dollar from 15 cents: cross-platform management for internet services," in

Proceedings of the ATC'08 USENIX 2008 Annual Technical Conference, Boston, MA, USA, June 2008.

[34] A. Verma, L. Cherkasova, and R. H. Campbell, "ARIA: automatic resource inference and allocation for Map Reduce environments," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*, Karlsruhe, Germany, June 2011.

[35] G. Fatini and C. A. P. D. S. Martins, "A configurable and portable benchmark for 3D graphics," in *Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing*, Florianopolis, Brazil, October 2001.

[36] "Safety-Critical Profile Specification," March 2009, https://www.khronos.org/registry/OpenGL/specs/sc/sc_spec_1_0_1.pdf.

[37] B. Klug, *Right Ware Lauches Basemark ES 2.0 Taiji Free for Android, iOS*, December 2011, https://www.anandtech.com/show/5263/rightware-launches-basemark-es-20-taiji-free-for-android-ios.

[38] P. Sander, D. Nehab, and J. Barczak, "Fast triangle reordering for vertex locality and reduced overdraw," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[39] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[40] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?-arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[41] https://github.com/guybois/HWPerfPredwithML.

[42] R.-G. Simon, "Prédiction de performance de matériel graphique dans un contexte avionique par apprentissage automatique," Masters thesis, École Polytechnique de Montréal, Montreal, Canada, 2015.

[43] L. Vincent, "Méthodologie expérimentale pour évaluer les caractéristiques des plateformes graphiques avioniques," Masters thesis, É Sters Thesisacte de MontratPe, Montreal, Canada, 2014.

Hindawi
Scientific Programming
Volume 2019, Article ID 2962040, 11 pages
https://doi.org/10.1155/2019/2962040

Research Article

Design and Development of a Large Cross-Lingual Plagiarism Corpus for Urdu-English Language Pair

Israr Haneef,[1] Rao Muhammad Adeel Nawab,[2] Ehsan Ullah Munir,[1] and Imran Sarwar Bajwa[3]

[1]*Department of Computer Science, COMSATS Institute of Information Technology, Wah Campus, Wah Cantonment, Pakistan*
[2]*Department of Computer Science, COMSATS Institute of Information Technology, Lahore Campus, Lahore, Pakistan*
[3]*Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur, Pakistan*

Correspondence should be addressed to Imran Sarwar Bajwa; imran.sarwar@iub.edu.pk

Received 17 December 2018; Accepted 25 February 2019; Published 17 March 2019

Guest Editor: Vicente García-Díaz

Cross-lingual plagiarism occurs when the source (or original) text(s) is in one language and the plagiarized text is in another language. In recent years, cross-lingual plagiarism detection has attracted the attention of the research community because a large amount of digital text is easily accessible in many languages through online digital repositories and machine translation systems are readily available, making it easier to perform cross-lingual plagiarism and harder to detect it. To develop and evaluate cross-lingual plagiarism detection systems, standard evaluation resources are needed. The majority of earlier studies have developed cross-lingual plagiarism corpora for English and other European language pairs. However, for Urdu-English language pair, the problem of cross-lingual plagiarism detection has not been thoroughly explored although a large amount of digital text is readily available in Urdu and it is spoken in many countries of the world (particularly in Pakistan, India, and Bangladesh). To fulfill this gap, this paper presents a large benchmark cross-lingual corpus for Urdu-English language pair. The proposed corpus contains 2,395 source-suspicious document pairs (540 are automatic translation, 539 are artificially paraphrased, 508 are manually paraphrased, and 808 are nonplagiarized). Furthermore, our proposed corpus contains three types of cross-lingual examples including artificial (automatic translation and artificially paraphrased), simulated (manually paraphrased), and real (non-plagiarized), which have not been previously reported in the development of cross-lingual corpora. Detailed analysis of our proposed corpus was carried out using n-gram overlap and longest common subsequence approaches. Using Word unigrams, mean similarity scores of 1.00, 0.68, 0.52, and 0.22 were obtained for automatic translation, artificially paraphrased, manually paraphrased, and nonplagiarized documents, respectively. These results show that documents in the proposed corpus are created using different obfuscation techniques, which makes the dataset more realistic and challenging. We believe that the corpus developed in this study will help to foster research in an underresourced language of Urdu and will be useful in the development, comparison, and evaluation of cross-lingual plagiarism detection systems for Urdu-English language pair. Our proposed corpus is free and publicly available for research purposes.

1. Introduction

In cross-lingual plagiarism, a piece of text in one (or source) language is translated into another (or target) language by neither changing the semantics and content nor referring the origin [1, 2]. Cross-lingual plagiarism detection is a challenging research problem due to various reasons. Firstly, machine translation systems are available online free of cost such as Google Translator (https://translate.google.com/) to translate a document written in one language into another language. Secondly, the Web has become a hub of multi-lingual resources. For example, Wikipedia contains articles in more than 200 languages on same topics (http://en.wikipedia.org/wiki/wikipedia Last visited 10-02-2019). Thirdly, people might be often interested to write in another language which is different from their native language. Consequently, all these factors contribute to an environment, which makes it easier to commit cross-lingual plagiarism and difficult to detect it.

The task of plagiarism can be broadly categorized into two categories [3]: (1) intrinsic plagiarism analysis and (2) extrinsic plagiarism analysis. In the former case, a single document is examined to identify plagiarism in terms of variation of an author(s)'s writing style. The fragment(s) for text which is significantly different from other fragments in a document is a trigger of plagiarism. Mostly stylometric-based features are modeled to detect such plagiarism. In the latter case, we are provided with a document which is suspected to contain plagiarism (suspicious document) and source collection. The aim is to identify fragments of text(s) in the suspicious document which are plagiarized and their corresponding source fragments from the source collection. Extrinsic plagiarism can be further divided into (1) monolingual–both source and plagiarized texts are in the same language and (2) cross-lingual plagiarism–source and plagiarized texts are in different languages. In case of cross-lingual plagiarism, a source text can be translated either automatically or manually, and after translation, it can be either used verbatim or rewritten for plagiarism [4].

To develop and evaluate Cross-Lingual Plagiarism Detection (CLPD) methods, standard evaluation resources are needed. Majority of CLPD corpora are developed for English, European, and some other languages (http://www.webis.de/research/corpora-Last-visited-10-02-2019). In addition, none of the existing cross-lingual corpus contains a mix of artificial, simulated, and real examples, which is necessary to make a realistic and challenging corpus. The problem of CLPD has not been thoroughly explored for South Asian languages such as Urdu, which is a widely spoken by a large number of people around the globe. Urdu is the first language of about 175 million people around the world and particularly spoken in Pakistan, India, Bangladesh, South Africa, and Nepal (http://www.ethnologue.com/language/urd, last visited: 20-02-2019). It is written from right to left like Arabic script. Urdu language usually follows Nastalique writing style [5]. However, Urdu is an under-resourced language in terms of computational and evaluation resources.

The main objectives of this study are threefold: (1) to develop a large benchmark cross-lingual corpus for Urdu-English language pair, which contains a mix of artificial, simulated, and real examples, (2) to carry out linguistic analysis of the proposed corpus to get insights into the edit operations used in cross-lingual plagiarism, and (3) to carry out detailed empirical analysis of the proposed corpus using n-gram Overlap and Longest Common Subsequence approaches to investigate whether the documents in the corpus are created using different obfuscation techniques. There are total 2,398 source-suspicious document pairs in our proposed corpus. Source documents are in Urdu language, and suspicious ones are in English. The source-suspicious document pairs are categorized into two main categories: (1) plagiarized (1,588 document pairs) and (2) nonplagiarized (810 document pairs). The plagiarized documents are created using three obfuscation strategies: (1) automatic translation (540 document pairs), (2) artificial paraphrasing (540 document pairs), and (3) manual paraphrasing (508 document pairs). The documents in our proposed corpus are from various domains including Computer Science, Management Science, Electrical Engineering, Physics, Psychology, Countries, Pakistan Studies, General Topics, Zoology, and Biology, which makes the corpus more realistic and challenging. We also carried out linguistic and empirical analysis of our proposed corpus.

Our proposed corpus will be beneficial for (1) fostering and promoting research in a low resourced language—Urdu, (2) enabling us to make a direct comparison of existing and new CLPD methods for Urdu-English language pair, (3) developing and evaluate new methods for CLPD for Urdu-English language pairs, and (4) developing a bilingual Urdu-English dictionary using our proposed corpus. Furthermore, our proposed corpus is free and publicly available for research purposes.

The rest of this paper is organized as follows: Section 2 summarizes the related work on existing corpora for CLPD. Section 3 describes the corpus generation process, including source documents collection, levels of rewriting, creation of suspicious documents, and standardization of the corpus. Section 4 presents the linguistic analysis of our proposed corpus. Section 5 presents a deeper empirical analysis of the corpus. Finally, Section 6 concludes the paper.

2. Related Work

In the literature, efforts have been made to develop benchmark corpora for CLPD. One of the prominent efforts is the series of PAN (http://pan.webis.de/, last visited: 20-02-2019) (a forum of scientific events and shared tasks on digital text forensic) competitions. A number of frameworks for cross-lingual plagiarism evaluation are also proposed by researchers for this forum [6, 7]. The main outcome of these competitions is a set of benchmark corpora for mono- and cross-lingual plagiarism detection. The majority of plagiarism cases, in these corpora, are monolingual (90%), and remaining 10% are cross-lingual such as English-Persian and English-Arabic and other language pairs. Almost 80% of cross-lingual plagiarism cases, in these corpora, are generated using automatic translation, and the rest are generated using manual translation. PAN cross-lingual corpora have been developed for two language pairs: English-Spanish and English-German.

The relevant literature presents a number of benchmark CLPD corpora for languages like Indonesian-English [8], Arabic-English [9], Persian-English [10], and English-Hindi [11]. Developing such a resource for especially under-resourced languages is an active research area [12, 13]. Parallel corpora have also been developed and used in [14] for the automatic translation purpose in cross-lingual domain. CLPD systems based on these corpora and other approaches are also proposed in the literature [15]. Most of these approaches used syntax-based plagiarism detection methods, but at the same time, semantic-based plagiarism detection approaches were also applied for the purpose. Savador et al. used semantic plagiarism detection approach using the graph analysis method for cross-language plagiarism detection. It is a language-independent model for plagiarism detection applied to the Spanish-English and German-English domains [16].

Cross Language Indian Text Reuse (CLITR) task has been designed in conjunction with Forum for Information Retrieval Evaluation (FIRE) to detect cross-lingual plagiarism for English-Hindi language pair. The corpus is divided into training and test segments in which source documents are in English and suspicious documents are in Hindi.

The training and test collection both include 5032 source files in English while 198 suspicious files in training and 190 suspicious files are in Hindi (http://www.uni-weimar.de/medien/webis/events/panfire-11/panfire11-web/, last visited: 20-08-2018). Corpora have also been developed for performance evaluation of cross-language information retrieval (CLIR) systems [17], while Kishida [18] raised technical issues of this domain. Moreover, different plagiarism detection tasks like text alignment and source retrieval are designed based on these corpora's, and overview of these tasks are being consistently (yearly) been published by PAN@ CLEF forum [19, 20].

The JRC-Acquis Multilingual Parallel Corpus has been used by Potthast et al., to apply CLPD approaches. As many as 23,564 parallel documents are constructed in the corpus that is extracted from legal documents of European Union [21, 22]. Out of 22 languages in legal document collection, only 5 including French, Germen, Polish, Dutch, and Spanish were selected to generate source-suspicious document pair (English language was used as source language). Comparable Wikipedia Corpus is another dataset used for the evaluation of CLPD methods. The corpus contains 45,984 documents.

Benchmark cross-lingual corpora have been developed using two approaches: (1) automatic translation and (2) manual translation. PAN corpora are created using both approaches for English-Spanish and English-German language pairs. However, the majority of cross-lingual cases are generated using automatic translation, and only a small number of them are generated using manual translation.

CLITR Corpus is generated using both automatic and manual translations: Near copy/exact copy documents are created using automatic translation, whereas heavy revision (HR) documents are created using manual paraphrasing of automatic translations of source texts. Again, this corpus only contains 388 suspicious documents, and it is created for English-Hindi language pair.

Two cross-lingual corpora used in plagiarism detection task are (1) JRC-EU Corpus and (2) Fairy Tale Corpus [21, 22]. JRC-EU cross-lingual corpus consists of randomly extracted 400 documents from the legislation reports of European Union which includes 200 English source documents and 200 Czech documents. Fairy-tale corpus contains 54 documents: 27 in English and 27 in Czech. Ceska et al. also used these corpuses for CLPD task [23].

In a previous study, we developed a corpus for the PAN 2015 Text Alignment task (we named it CLUE Corpus) [24]. In that corpus, there are total 1000 documents (500 are source documents and 500 are suspicious documents). Among the suspicious collections, 270 documents are plagiarized using 90 source-plagiarized fragment pairs, while the remaining 230 suspicious documents are nonplagiarized. Note that this corpus contains simulated cases of plagiarism, which were inserted into suspicious document to generate plagiarized documents. The CLUE Corpus can be used for the development and evaluation of CLPD systems for English-Urdu language pair for the text alignment task only as described by PAN organizers.

To conclude, the relevant literature presents the majority of CLPD corpora for English and other European languages. Moreover, these are mainly created using comparable documents, parallel documents, and automatic translations, which are not realistic examples for cross-lingual plagiarism. This study contributes a large benchmark corpus (containing 2,398 source-suspicious document pairs) for CLPD in Urdu-English language domain. Note that the 270 fragment pairs used in the development of CLUE Corpus are also included in this corpus.

3. Corpus Generation

This section describes the process for construction of a benchmark corpus for CLPD for Urdu-English language pair (hereafter called CLPD-UE-19 Corpus) including collection of source texts, levels of rewrite used in creating suspicious documents, creation of suspicious documents, and standardization of corpus and corpus characteristics.

3.1. Collection of Source Texts. Urdu is an underresourced language as large repositories of digital texts in this language are not readily available for the research purposes. Urdu newspapers in Pakistan mostly publish news stories in images format which is not suitable for text processing. Therefore, to collect realistic, high-quality, and diversified source articles for generating CLPD-UE-19 Corpus, we selected Wikipedia[1] as a source. Wikipedia is a free and publicly available, multitopic, and multilingual resource. Initially, Wikipedia contains an article in multiple languages which makes it possible to be considered as a comparable corpus. AJ Head investigated the potential use of Wikipedia for course-related search by students [25]. Martinez also investigated the cases where Wikipedia is mainly used for copy and paste plagiarism cases [26]. Wikipedia articles are taken as source documents for generating cross-lingual plagiarism detection corpus for Hindi-English language pair [27].

Plagiarism is a serious problem, particularly in higher educational institutions [28–31]. Therefore, CLPD-UE-19 Corpus focuses on plagiarism cases generated by university students. Table 1 shows the domains from which Wikipedia (http://ur.wikipedia.org/wiki/urdu) source articles are collected to generate CLPD-UE-19 Corpus. Apart of it, 270 source-suspicious document pairs were used in the creation of the CLUE Corpus [24].

These domains include Computer Science, Management Science, Electrical Engineering, Physics, Psychology, Countries, Pakistan Studies, General Topics, Zoology, and Biology. As can be noted, these articles are on a wide range of topics, which makes the CLPD-UE-19 Corpus more realistic and challenging.

The amount of text reused for creating a plagiarized document can vary from a phrase, sentence, and paragraph to the entire document. It is also likely that to hide

Table 1: Domains from which Wikipedia source articles were selected in creating our proposed CLPD-UE-19 Corpus.

Domain	Major topics
Computer science	Free software, binary numbers, open source, database normalization, robotics, artificial intelligence, MSN, Google, Yahoo, WhatsApp, Android, Facebook, Twitter, RUBY language, daily motion, HTML, mobile apps, Gmail, Skype, and others
General topics	Globalization, muhammad iqbal, global warming, capitalism, mosque, bookselling, Pakistan air force, cricket, fashion, Lahore Fort, capitalism, Badshahi Masjid, and two-nation theory
Electrical engineering	Electricity, magnetism, and conducting materials
Management science	Trade and finance
Physics	Atoms and scientists
Psychology	Neurology, psycho diseases, and enlightenment
Countries	Politics and trade of different countries (mostly African)
Pakistan studies	History of Pakistan and Indo-Pak partition
Zoology	Animals, food, and living styles
Biology	Natural organisms, living cells, and DNA

plagiarism, a plagiarist may reuse the texts of different sizes from different sources. Therefore, the size of source documents is varied. The length of a source text may fall into one of the three categories: (1) small (1–50 words), (2) medium (50–100 words), and (3) large (100–200 words).

3.2. Levels of Rewrite. The proposed corpus contains two types of suspicious documents: (1) plagiarized and (2) nonplagiarized. The details of these are as follows.

3.2.1. Plagiarized Documents. A plagiarized document in CLPD-UE-19 Corpus falls into one of the three categories: (1) automatic translation, (2) artificially paraphrased copy, and (3) manually paraphrased copy. The reason for creating plagiarized documents with three different levels of rewrite is that a plagiarist is likely to use one of the three above-mentioned approaches for creating a plagiarized document using existing document(s) for cross-lingual settings.

(a) Automatic Translation. Using this approach, plagiarized documents (in English) are created by automatically translating the source texts (in Urdu) using Google Translator (https://translate.google.com/, last visited: 20-02-2019). Note that Google Translator has been effectively used in earlier research studies [32, 33].

(b) Artificially Paraphrased Copy. This approach aims to create artificially paraphrased cases of cross-lingual plagiarism in two steps. A source text (in Urdu) is translated automatically into English using Google Translator in the first step. After that, an automatic text rewriting tool is used to paraphrase the translated text, which results in an artificially paraphrased copy of the original text. For this study, we explore various free and publicly available text rewriting tools. Among the available tools, we found that two of them have the highest number of visitors per day: (1) Spinbot text rewriting tool (http://www.spinbot.net/) with an average number of 26 k visitors per day and (2) Article Rewriter text rewriting tool (http://articlerewritertool.com/) with an average number of 45 k visitors per day reported by Alexa (this is a ranking system set by alexa.com (a subsidiary of amazon.com) that basically audits and makes public the frequency of visits on various websites) as compared to other tools like http://paraphrasing-tool.com/, etc.

(c) Manually Paraphrased Copy. Using this approach, the plagiarized document were created by manually translating and paraphrasing the original texts.

3.2.2. Nonplagiarized. Wikipedia is a comparable corpus and contains an article in multiple languages. It is notable that these articles are not translations of each other. To generate nonplagiarized cases, similar fragments of texts were manually identified from English and Urdu Wikipedia articles on the same topic.

The assumption is that although English and Urdu Wikipedia articles are written on the same topic, they are independently written by two different authors. Therefore, similar fragments of English-Urdu texts can serve as independently written cross-lingual document pairs.

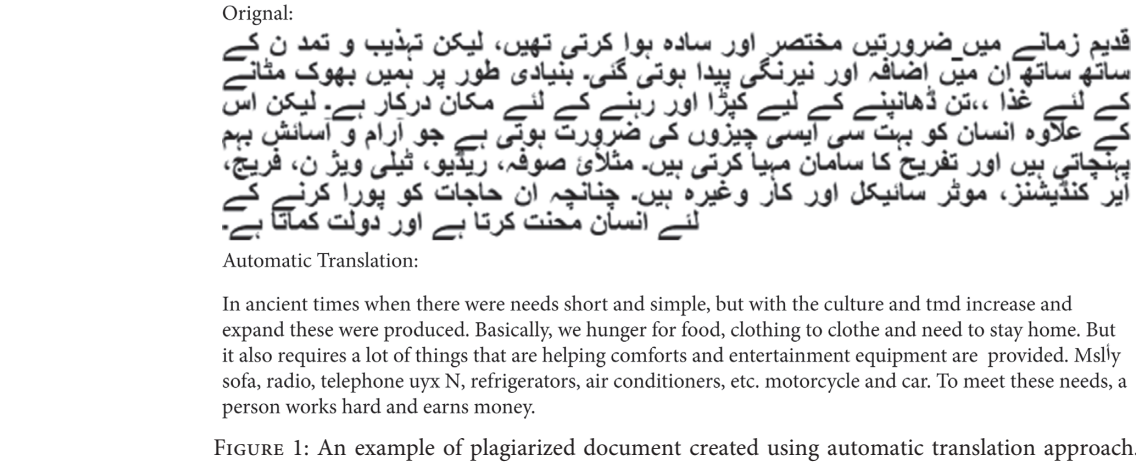
As far as we are aware, the proposed methods used for creating cross-lingual plagiarism cases of artificially paraphrased plagiarism and Nonplagiarism have not been previously used for creating cross-lingual plagiarism cases in any other language pair.

3.3. Generation of Suspicious Texts. Crowdsourcing is a process of performing a task in collaboration of a large number of people usually working as a remote user. It can be done with a group of people, small teams or even individuals. Generating a large benchmark CLPD corpus is not a trivial task. Therefore, we use the crowdsourcing approach to generate suspicious texts with four levels of rewriting. Examples of manually paraphrased copy and nonplagiarized are generated by participants (volunteers), who are graduate-level university students (masters and M Phil). All the participants are native speakers of Urdu. As the medium of instruction in university and colleges is English, students have a high level of proficiency in English language too.

The majority of the participants are from the English department, and hence are well aware of paraphrasing techniques.

However, for better quality, they were provided with examples of paraphrasing. The plagiarized documents generated by volunteers were manually examined, and low-quality documents were discarded.

3.4. Examples of Cross-Lingual Plagiarism Cases from CLPD-UE-19 Corpus. Figure 1 presents an example of source-


Orignal:

قدیم زمانے میں ضرورتیں مختصر اور سادہ ہوا کرتی تھیں، لیکن تہذیب و تمد ن کے ساتھ ساتھ ان میں اضافہ اور نیرنگی پیدا ہوتی گئی۔ بنیادی طور پر ہمیں بھوک مٹانے کے لئے غذا ،،تن ڈھانپنے کے لیے کپڑا اور رہنے کے لئے مکان درکار ہے۔ لیکن اس کے علاوہ انسان کو بہت سی ایسی چیزوں کی ضرورت ہوتی ہے جو آرام و آسائش بہم پہنچاتی ہیں اور تفریح کا سامان مہیا کرتی ہیں۔ مثلاً صوفہ، ریڈیو، ٹیلی ویژ ن، فریج، ایر کنڈیشنز، موٹر سائیکل اور کار وغیرہ ہیں۔ چنانچہ ان حاجات کو پورا کرنے کے لئے انسان محنت کرتا ہے اور دولت کماتا ہے۔

Automatic Translation:

In ancient times when there were needs short and simple, but with the culture and tmd increase and expand these were produced. Basically, we hunger for food, clothing to clothe and need to stay home. But it also requires a lot of things that are helping comforts and entertainment equipment are provided. Msly sofa, radio, telephone uyx N, refrigerators, air conditioners, etc. motorcycle and car. To meet these needs, a person works hard and earns money.

Figure 1: An example of plagiarized document created using automatic translation approach.

plagiarized document pair from CLPD-UE-19 Corpus created using automatic translation approach. As can be noted, the translated text is not an exact copy of the original one. The possible reason for this is that Urdu is an underresourced language, and machine translation systems for Urdu-English language pair are not matured compared to other language pairs. Consequently, the translated text seems to be a near copy of the original text instead of an exact copy. Moreover, it can also be observed from the translated document that for few words for which Google Translator does not find any equivalent word in English, it merely replaces the pronunciation of that word with English homonyms, for instance, تمد ن is replaced with *tmd* and مثلأئ is replaced with *Msly*. To conclude, the overall quality of Google Translator seems to be good considering the complexity in translating Urdu text to English.

Figure 2 shows an example of plagiarism document where automatic translation of a source document is further altered by an automatic rewriting tool to get artificially plagiarized copy of the source document. It can be observed from this example that automatic text rewriting tool has replaced the words by appropriate synonyms (the words presented in Italics are synonyms of original words). However, the text rewriting tool does not alter the order of text. The alteration in the translated text is carried out by rewriting tool which further increases the level of rewriting and makes it difficult to identify similarity between source-plagiarized text pairs.

A sample plagiarized document generated using the manually paraphrased copy approach is shown in Figure 3, which is a very well paraphrased content. Different text rewriting operations have been applied by the participants to paraphrase the original text including synonym replacement, sentence merging/splitting, insertion/deletion of text, word reordering. Consequently, the source-plagiarized text pairs are semantically similar but different at surface level, which makes the CLPD task even more challenging.

A nonplagiarized source-suspicious document pair from the CLPD-UE-19 Corpus is shown in Figure 4. The text is topically related, but independently written. The inclusion of more introductory sentences and last sentence reflects that both texts are written in different contexts.

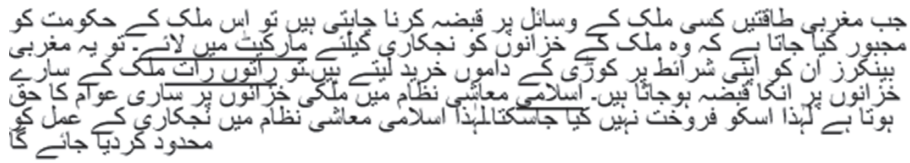
3.5. Corpus Characteristics. Table 2 presents the detailed statistics of the proposed corpus. In this table, AT, APC, MPC, and NP represent automatic translation, artificially paraphrased copy, manually paraphrased copy, and non-plagiarized, respectively. There are total 2,398 source-suspicious document pairs in the corpus, 810 are non-plagiarized and 1,588 are plagiarized. Among the plagiarized document pairs, 540 are automatically translated, 540 are artificially paraphrased, and 508 are manually paraphrased. Above statistics show that the corpus contains a large number of documents for both plagiarized and nonplagiarized cases. Also, the documents for four different levels of rewrite in the proposed corpus are almost balanced. The CLPD-UE-19 Corpus is standardized in XML format and publicly available for research purposes (the CLPD-UE-19 Corpus is distributed under the terms of the Creative Common Attribution 4.0 International License and can be downloaded from the following link: https://www.dropbox.com/sh/p9e00rxjj9r7cbk/AACj3gtVEy5T74rfP58_BtP6a?dl=0).

4. Linguistic Analysis of CLPD-UE-19 Corpus

This section presents the linguistic analysis of the CLPD-UE-19 Corpus. As reported in [34, 35], various edit operations are performed on the source text to create plagiarized text, particularly when it the source text is reused for paraphrased plagiarism. Below we discuss the various edit operations which we observed while carrying out linguistic analysis on a subset of CLPD-UE-19 Corpus (note that, we used 50 source-suspicious document pairs for the linguistic analysis presented in this section) (Figures 5–9).

4.1. Replacing Pronoun with Noun. In these edit operations, a pronoun is replaced by actual name or vice versa in source and suspicious document, for instance:

Original:

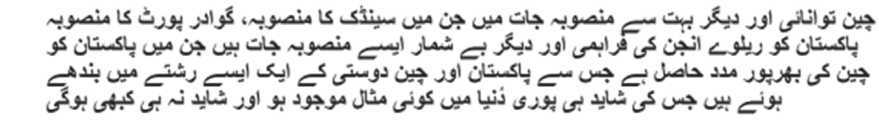

جب مغربی طاقتیں کسی ملک کے وسائل پر قبضہ کرنا چاہتی ہیں تو اس ملک کے حکومت کو مجبور کیا جاتا ہے کہ وہ ملک کے خزانوں کو نجکاری کیلئے مارکیٹ میں لائے۔ تو یہ مغربی بینکرز ان کو اپنی شرائط پر کوڑی کے داموں خرید لیتے ہیں۔تو راتوں رات ملک کے سارے خزانوں پر انکا قبضہ ہوجاتا ہیں۔ اسلامی معاشی نظام میں ملکی خزانوں پر ساری عوام کا حق ہوتا ہے لہذا اسکو فروخت نہیں کیا جاسکتا۔لہذا اسلامی معاشی نظام میں نجکاری کے عمل کو محدود کردیا جائے گا

Artificially Plagiarized Copy:

When the Western powers wish to capture a country's resources for the country's government is forced to denationalize the country's treasures dropped at market. If the worth of West Bankers obtain penny on their own terms as a result of the night they're treasures of the country. monotheism financial set-up and also the public's right to national treasuries, therefore it didn't sell. so privatization method within the monotheism financial set-up would be restricted.

Figure 2: An example of plagiarized document created using artificial paraphrasing approach.

Original:


چین توانائی اور دیگر بہت سے منصوبہ جات میں جن میں سینڈک کا منصوبہ، گوادر پورٹ کا منصوبہ پاکستان کو ریلوے انجن کی فراہمی اور دیگر بے شمار ایسے منصوبہ جات ہیں جن میں پاکستان کو چین کی بھرپور مدد حاصل ہے جس سے پاکستان اور چین دوستی کے ایک ایسے رشتے میں بندھے ہوئے ہیں جس کی شاید ہی پوری دنیا میں کوئی مثال موجود ہو اور شاید نہ ہی کبھی ہوگی

Manually Plagiarized Copy:

There are many projects in which China provided their full aid, these projects not only include energy, but also gawadar port, sindic, Pakistan Railway engine supply, and such are other types of huge projects are part of planning. So, in this way, Pakistan and China have good harmony between them. This instance of friendship will and might not seen on anywhere.

Figure 3: An example of plagiarized document created using manual paraphrasing approach.

Original:


ڈاکٹر سر علامہ محمد اقبال (9 نومبر 1877ء تا 21 اپریل 1938ء) بیسویں صدی کے ایک معروف شاعر، مصنف، قانون دان، سیاستدان، مسلم صوفی اور تحریک پاکستان کی اہم ترین شخصیات میں سے ایک تھے۔ اردو اور فارسی میں شاعری کرتے تھے اور یہی ان کی بنیادی وجہ شہرت ہے۔

Non-Plagiarized

Sir Muhammad Iqbal (9 November 1877 –21 April 1938), widely known as Allama Iqbal was a philosopher, poet, mystic and politician in British India who is widely regarded as having inspired the Pakistan Movement. He is considered one of the most important figures in Urdu literature.

Figure 4: An example of nonplagiarized document from our proposed corpus.

Table 2: Corpus statistics.

Size (count of words)	Level name/plagiarized and nonplagiarized/ plagiarized version (total count)			Subject domains								
				CS	*GT*	*Phy*	*Bio*	*EE*	*Zol*	*Psy*	*PS*	*MS*
≤50	(Small)	NP: 450*		100	50	75				25		
		Plagiarized	AT (300)	100	50	99				51		
			AP (300)	100	50	99				51		
			MP (290)	100	50	90				50		
>50 and ≤100	Paragraph (medium)	NP: 225		50	25			20	75		15	40
		Plagiarized	AT (150)	50	25			15			10	50
			AP (150)	50	25			15			10	50
			MP (148)	50	25			15			10	48
≥100 and ≤200	Essay (large)	NP: 135		30	15				33		57	
		Plagiarized	AT (90)	30	15		45					
			AP (90)	30	15		45					
			MP (70)	30	15		25					
			Total	720	360	363	115	65	108	177	102	188

CS: Computer science, *GT*: General Topics, *Phy*: Physics, *Bio*: Biology, *EE*: Electrical Engineering, *Zol*: Zoology, *Psy*: Psychology, *PS*: Pak Studies, *MS*: Management Sciences (200 nonplagiarized documents are from countries domain).

S: گو کہ انہوں نے اس نئے ملک کے قیام کو اپنی آنکھوں سے نہیں دیکھا لیکن انہیں پاکستان کے قومی شاعر کی حیثیت حاصل ہے۔

D: Iqbal has been a national poet of Pakistan Although, he did not see the establishment of the new country with his own eyes

Figure 5: An example of replacing pronoun with noun.

S: ایک چھوٹا سا مکان لے کر اس میں رہنے لگے ، مرتے دم تک یہیں رہے

D:He spent the rest of his life in a small house which he took for rent

Figure 6: An example of changing order of text paraphrasing.

S: بین الاقوامی تجارت معاشیات کی ایک شاخ ہے۔ بنیادی طور پر یہ بین الاقوامی معاشیات کی ایک ذیلی شاخ ہے۔

D: In the branch of economics there exist international trade but basically it is a sub-branch of international economics.

Figure 7: An example of changing source text by adding words.

S: پاکستان فوج کا قیام ۱۹۴۷ میں پاکستان کی آزادی پر عمل میں آیا۔یہ ایک رضاکارپیشہ ور جنگجوقوت ہے۔

D: It was established on August 14, 1947. Brave, volunteer and sacrificing warriors are main features of them.

Figure 8: An example of paraphrasing text by date completion.

S: جیسا کے جال یا ویب کا مفہوم ہے کہ یہ تمام اطراف پھیلا ہوا ہوتا ہے یعنی باالفاظ دیگر ہر طرف رابطے میں ہوتا ہے اسی طرح رابط کا لفظ بھی اسی مفہوم کی ترجمانی کرتا ہے۔

D: Like the meaning of net spread every where so as web

Figure 9: An example of summarizing source text in plagiarized document.

4.2. Order Change with Add/Delete Words. It is also a common approach used in edit operation. In this approach, later part of the source text is quoted first in the suspicious text and vice versa like.

4.3. Continuing Sentences: Adding Words. Combining two sentences by using an additional word is the most used approach in rewriting text, for example.

4.4. Date Completed. It is another approach where an event in the source text is rewritten in context of the event date and place in suspicious document.

4.5. Summary. In this category, an abstract description of the rewritten text in suspicious document is used in place of long narrations in the source document.

The corpus contains a number of examples of order changes and changing active to passive and direct to indirect and vice versa. Such examples reflect that edit operations change the source text so that it is not a verbatim case. It is not an easy case for plagiarism detection.

5. Translation + Monolingual Analysis of CLPD-UE-19 Corpus

For convenience, this section is further divided into three Sections: starting with experimental setup, next two sections describe detailed and comprehensive analysis of the corpus.

5.1. Experimental Setup. To analyze the quality of artificially and manually paraphrased levels of rewritten cases, we applied translation + monolingual analysis approach on our proposed corpus. Using this approach, we automatically translated source documents (in Urdu) into English using Google Translator. Now, both source and suspicious documents are in the same language, i.e., English. After that, we computed mean similarity scores for source-suspicious document pairs for all four categories (automatic translation copy, artificially paraphrased copy, manually paraphrased copy, and nonplagiarized) using n-gram overlap and longest common subsequence approaches.

To compute similarity scores between source-suspicious document pairs, we applied containment similarity measure [36] (equation (1)). Using the n-gram overlap approach, similarity score between source-suspicious document pair is computed by counting common n-grams between two

documents divided by the number of n-grams in both or any one of the documents. If $S(X, n)$ and $S(Y, n)$ represent word n-grams of length n in source and suspicious document, respectively, then similarity between them using containment similarity measure is computed as follows:

$$\text{Scontainment}(X, Y) = \frac{|S(X, n) \cap S(Y, n)|}{|S(X, n)|}. \quad (1)$$

We used another simple and popular similarity estimation model, longest common subsequence (LCS), to compute the mean similarity scores for four levels of rewrite in CLPD-UE-19 Corpus. Using the LCS approach, for a given pair of source-suspicious text (X and Y), we first computed the LCS between source-suspicious strings and then divided the LCS score with the length of smaller document to get a normalized score between 0 and 1 (equation (2)). Note that LCS method is order-preserving, and LCS score is affected by edit operations performed on source text to generate plagiarized text:

$$\text{LCSnorm}(X, Y) = \frac{|\text{LCS}(X, Y\ )|}{\min(|X|, |Y|)}. \quad (2)$$

5.2. Partial (Domainwise) Analysis. This dimension provides us an opportunity for microlevel and size-oriented domain analysis. Size is one of the dimensions in the rewritten cases. For this purpose, few sample documents from different domains have been randomly selected. Automatic translation copy (ATC) of a source document is compared with artificial and manual paraphrased versions of the same document. Bi, tri, and tetragram split has been applied to identify word to the sentence level similarity between different levels of the rewritten text. An empirical based analysis has been carried out for documents related to all the domains, but only results of only three domains for all size documents are listed here. Almost all results showing that n-gram similarity between both levels of rewrite decreases gradually as values of n increase.

5.2.1. Discussion. It is observed that overall average word n-gram similarity in small-sized manually paraphrased copies of documents is less than large- and medium-sized cases similarity. It also reflects that paraphrasing small-sized text using different edit operations is more paraphrased as compared to other sizes of suspicious documents and hence difficult to detect as well.

In Tables 3–5 and Figure 10, it is noteworthy that 4-gram value or even 3-gram value in most of the cases approaches to zero. It reflects that how well a source document has gradually been altered in both APC and MPC levels of rewrite across the entire corpus. Only a few documents out of such a large corpus have high value of similarity between the source and its MPC level because plagiaries have not used any major paraphrasing techniques for rewriting the source text. But, in such a large corpus of more than 2300 documents, these are only a few such cases.

To have a better view of rewriting levels, we apply APC- and MPC-wise average n-gram approach also, the results of which are presented in Table 6. As per Figure 11, the similarity ratio in most of the APC cases is higher than MPC cases. It also indicates that artificial paraphrasing techniques are still slightly not as precise in paraphrasing source text as compared to the manual effort.

Table 3: Comparison of rewrite levels of *medium* documents from *Pak Study* domain.

	MPC			APC		
	2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
Document 0002.txt	0.153	0.042	0	0.625	0.521	0.457
Document 0005.txt	0.110	0.049	0.025	0.659	0.519	0.388
Document 0006.txt	0.143	0.040	0.008	0.587	0.448	0.347
Document 0009.txt	0.114	0.023	0	0.466	0.322	0.209
Document 0011.txt	0.210	0.066	0	0.387	0.262	0.167

Table 4: Comparison of rewrite levels of *large-sized* documents from the *Biology* domain.

	MPC			APC		
	2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
Document-0041.txt	0.111	0.038	0	0.370	0.231	0.120
Document-0042.txt	0.120	0.042	0	0.280	0.042	0
Document-0087.txt	0.324	0.182	0.063	0.588	0.515	0.469
Document-0094.txt	0.455	0.286	0.150	0.364	0.190	0.050
Document-0095.txt	0.381	0.250	0.105	0.429	0.250	0.053

Table 5: Comparison of rewrite levels of *small sized* documents from the *Physics* domain.

	MPC			APC		
	2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
Document-0066.txt	0.113	0.025	0	0.463	0.329	0.231
Document-0068.txt	0.103	0.026	0	0.449	0.234	0.105
Document-0070.txt	0.218	0.091	0.066	0.487	0.338	0.211
Document-0072.txt	0.121	0.031	0	0.803	0.708	0.609
Document-0075.txt	0.133	0.068	0.014	0.547	0.419	0.329

5.3. Complete (Corpus-Based) Analysis. Table 7 shows the mean similarity scores obtained using n-gram overlap and LCS approaches. AT refers to automatic translation, APC refers to artificial paraphrased copy, MPC refers to manually paraphrased copy, and NP refers to nonplagiarized. 1-gram refers to mean similarity scores generated using n-gram overlap approach, where $n = 1$ (i.e., unigram). Similarly, 2-gram refers to mean similarity scores generated using n-gram overlap approach, where $n = 2$ (i.e., bigram) and so on. Mean similarity scores obtained using LCS approach are referred as LCS. Note that mean similarity score for AT is 1.00 for all methods. The reason is that we used Google Translator for both creating AT cases of plagiarism (Section 3.2) and M + TA analysis (presented in this section). Therefore, the two translations are exactly same generating a similarity score of 1.00 for AT.

As expected, similarity score drops as the level of rewrite increases (from AT to NP). This shows that it is hard to

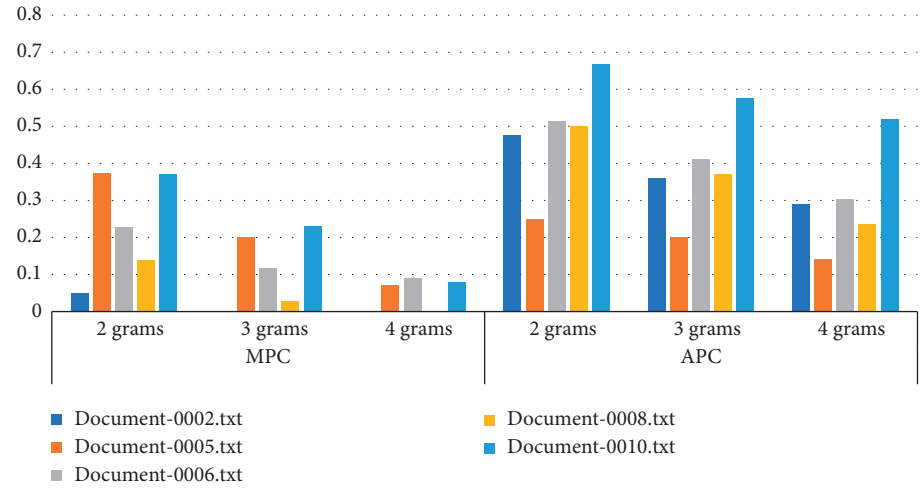


Figure 10: Comparison of manually paraphrased copy (MPC) and artificially paraphrased copy (APC) based on small-sized documents from the Psychology domain.

Table 6: Rewrite level-wise averaged *n*-gram-based small-sized documents from the Psychology domain.

Documents/rewrite levels	MPC	APC
Document-0002.txt	0.017	0.374
Document-0005.txt	0.215	0.198
Document-0006.txt	0.146	0.41
Document-0008.txt	0.056	0.369
Document-0010.txt	0.227	0.588

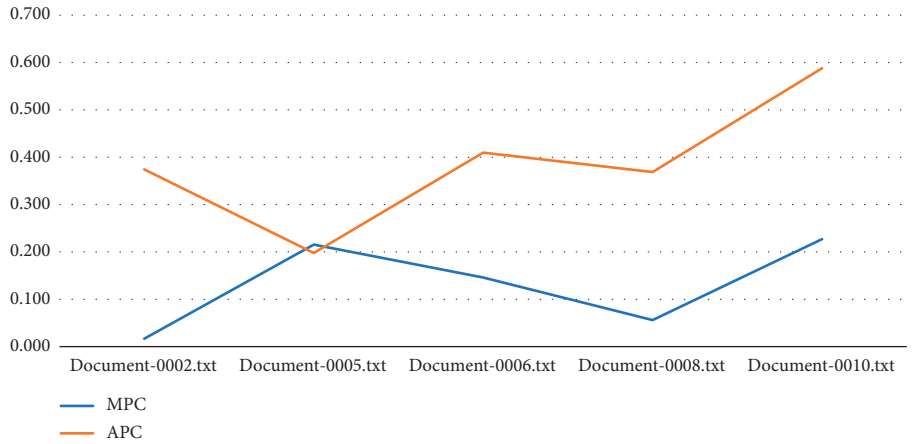


Figure 11: Averaged *n*-gram overlap scores for manually paraphrased copy (mpc) and artificially paraphrased copy (APC) documents.

Table 7: Mean similarity scores for four levels of rewrite in the CLPD-UE-19 Corpus using *n*-gram overlap and LCS approaches.

Method\rewrite levels	At	APC	MPC	NP
1-gram	1.00	0.68	0.52	0.22
2-gram	1.00	0.44	0.21	0.01
3-gram	1.00	0.31	0.11	0.00
4-gram	1.00	0.22	0.07	0
5-gram	1.00	0.16	0.04	0
LCS	1.00	0.20	0.15	0.05

detect plagiarism when the level of rewrite increases. This also shows that suspicious documents in the CLPD-UE-19 Corpus are generated using different obfuscation strategies. For *n*-gram overlap approach, mean similarity scores drops as the length of *n* increases, indicating that it is hard to find long exact matches in the source-suspicious document pairs. For LCS approach, the score is quite low compared to 1-gram approach. This highlights the fact that the order of texts in the source and suspicious document pair is significantly different which makes it hard to find longer matches.

6. Conclusion

The main goal of this study was to develop a large benchmark corpus of cross-lingual cases of plagiarism for Urdu-English language pair at four levels of rewrite including automatic translation, artificial paraphrasing, manual paraphrasing, and nonplagiarized. There are total 2,398 document pairs in our proposed corpus: 1,588 are plagiarized and 810 are nonplagiarized. Plagiarized documents are created using three obfuscation strategies: automatic translation (540 documents), artificial paraphrasing (540 documents), and manual paraphrasing (508 documents). Wikipedia articles are used as source texts and categorized into small, medium, and large documents. Crowdsourcing approach has been applied to create our proposed corpus. We also performed linguistic analysis and translation + monolingual analysis of our proposed corpus. Our empirical analysis showed that there is a clear distinction in four levels of rewrite in our proposed corpus, which makes the corpus more realistic and challenging. Being an emerging area of research [37], in future, we plan to apply cross-lingual plagiarism detection techniques on our proposed corpus.

Data Availability

The authors declare that the data mentioned and discussed in this paper will be provided, if required.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are thankful to all the volunteers for their valuable contribution in construction of the CLPD-UE-19 Corpus.

References

[1] A. Barrón-Cedeno, P. Rosso, E. Agirre, and G. Labaka, "Plagiarism detection across distant language pairs," in *Proceedings of the 23rd International Conference on Computational Linguistics Association for Computational Linguistics*, pp. 37–45, Beijing, China, August 2010.

[2] A. Barrón-Cedeño, P. Gupta, and P. Rosso, "Methods for cross-language plagiarism detection," *Knowledge-Based Systems*, vol. 50, pp. 211–217, 2013.

[3] B. Stein and S. M. zu Eissen, "Intrinsic plagiarism analysis with meta learning," in *Proceedings of the PAN 2007*, p. 276, Amsterdam, Netherlands, July 2007.

[4] B. Martin, "Plagiarism: a misplaced emphasis," *Journal of Information Ethics*, vol. 3, no. 2, p. 36, 1994.

[5] S. Hussain, "Complexity of Asian writing systems: a case study of Nafees Nasta'leeq for Urdu," in *Proceedings of the 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society*, Asian Media Information Center, Singapore, June 2003.

[6] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, "An evaluation framework for plagiarism detection," in *Proceedings of the 23rd international conference on computational linguistics: Association for Computational Linguistics*, pp. 997–1005, Beijing, China, August 2010.

[7] C. H. Lee, C. H. Wu, and H. C. Yang, "A platform framework for cross-lingual text relatedness evaluation and plagiarism detection," in *Proceedings of the 3rd International Conference on Innovative Computing Information and Control ICICIC'08*, p. 303, Dalian, China, June 2008.

[8] Z. F. Alfikri and A. Purwarianti, "The construction of Indonesian-English cross language plagiarism detection system using fingerprinting technique," *Jurnal Ilmu Komputer dan Informasi*, vol. 5, no. 1, pp. 16–23, 2012.

[9] A. Aljohani and M. Mohd, "Arabic-English cross-language plagiarism detection using winnowing algorithm," *Information Technology Journal*, vol. 13, no. 14, pp. 2349–2355, 2014.

[10] H. Asghari, K. Khoshnava, O. Fatemi, and H. Faili, "Developing bilingual plagiarism detection corpus using sentence aligned parallel corpus," in *Proceedings of the CLEF 2015*, Toulouse, France, September 2015.

[11] R. Kothwal and V. Varma, "Cross lingual text reuse detection based on keyphrase extraction and similarity measures," in *Multilingual Information Access in South Asian Languages*, pp. 71–78, Springer, Berlin, Germany, 2013.

[12] M. El-Haj, U. Kruschwitz, and C. Fox, "Creating language resources for under-resourced languages: methodologies, and experiments with Arabic," *Language Resources and Evaluation*, vol. 49, no. 3, pp. 549–580, 2015.

[13] J. Ferrero, F. Agnes, L. Besacier, and D. Schwab, "A multilingual, multi-style and multi-granularity dataset for cross-language textual similarity detection," in *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, Portorož, Slovenia, May 2016.

[14] P. E. Koehn, "A parallel corpus for statistical machine translation," in *Proceedings of the MT summit*, vol. 5, pp. 79–86, Phuket, Thailand, September 2005.

[15] C. K. Kent and N. Salim, "Web based cross language plagiarism detection," in *Proceedings of the Second International Conference on Computational Intelligence, Modelling and Simulation (CIMSIM)*, pp. 199–204, Bali, Indonesia, September 2010.

[16] M. Franco-Salvador, P. Rosso, and M. Montes-y-Gómez, "A systematic study of knowledge graph analysis for cross-language plagiarism detection," *Information Processing & Management*, vol. 52, no. 4, pp. 550–570, 2016.

[17] M. L. Littman, S. T. Dumais, and T. K. Landauer, "Automatic cross-language information retrieval using latent semantic indexing," in *Cross-Language Information Retrieval*, pp. 51–62, Springer, Boston, MA, USA, 1998.

[18] K. Kishida, "Technical issues of cross-language information retrieval: a review," *Information Processing & Management*, vol. 41, no. 3, pp. 433–455, 2005.

[19] M. Hagen, M. Potthast, and B. Stein, "Source retrieval for plagiarism detection from large web corpora: recent approaches," in *Proceedings of the CLEF 2015*, Toulouse, France, September 2015.

[20] E. Stamatatos, M. Potthast, F. Rangel, P. Rosso, and B. Stein, "Overview of the PAN/CLEF 2015 evaluation lab," in *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 518–538, Toulouse, France, September 2015.

[21] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 45–62, 2011.

[22] R. Steinberger, B. Pouliquen, A. Widiger et al., "The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages," https://arxiv.org/abs/cs/0609058.

[23] Z. Ceska, M. Toman, and K. Jezek, "Multilingual plagiarism detection," in *Artificial Intelligence: Methodology, Systems, and Applications*, pp. 83–92, Springer, Berlin, Germany, 2008.
[24] I. Hanif, R. M. A. Nawab, A. Arbab, H. Jamshed, S. Riaz, and E. U. Munir, "Cross-language Urdu-english (CLUE) text alignment corpus," in *Proceedings of the CLEF 2015*, Toulouse, France, September 2015.
[25] A. J. Head and M. B. Eisenberg, "How today's college students use wikipedia for course-related research," *First Monday*, vol. 15, no. 3, 2010.
[26] I. Martınez, "Wikipedia usage by Mexican students. The constant usage of copy and paste," in *Proceedings of the Wikimania 2009*, Buenos Aires, Argentina, August 2009.
[27] A. Barrón-Cedeno, P. Rosso, S. L. Devi, P. Clough, and M. Stevenson, "Pana fire: overview of the cross-language Indian text re-use detection competition," in *Multilingual Information Access in South Asian Languages*, pp. 59–70, Springer, Berlin, Germany, 2013.
[28] G. Judge, "Plagiarism: bringing economics and education together (with a little help from it)," *Computers in Higher Education Economics Reviews (Virtual edition)*, vol. 20, pp. 21–26, 2008.
[29] D. L. McCabe, "Cheating among college and university students: a North American perspective," *International Journal for Educational Integrity*, vol. 1, no. 1, 2005.
[30] C. Park, "In other (people's) words: plagiarism by university students–literature and lessons," *Assessment & Evaluation in Higher Education*, vol. 28, no. 5, pp. 471–488, 2003.
[31] P. Clough and M. Stevenson, "Developing a corpus of plagiarised short answers," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 5–24, 2011.
[32] J. Nair, K. A. Krishnan, and R. Deetha, "An efficient English to Hindi machine translation system using hybrid mechanism," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2109–2113, Jaipur, India, September 2016.
[33] E. M. Balk, M. Chung, M. L. Chen, T. A. Trikalinos, and L. K. W. Chang, "Assessing the accuracy of google translate to allow data extraction from trials published in non-english languages," Agency for Healthcare Research and Quality (US), Rockville, MD, USA, Report no: 12(13)-EHC145-EF, 2013.
[34] M. Vila, M. A. Martí, and H. Rodríguez, "Is this a paraphrase? What kind? Paraphrase boundaries and typology," *Open Journal of Modern Linguistics*, vol. 4, no. 1, pp. 205–218, 2014.
[35] M. Sharjeel, R. M. A. Nawab, and P. Rayson, "COUNTER: corpus of Urdu news text reuse," *Language Resources and Evaluation*, vol. 51, no. 3, pp. 777–803, 2017.
[36] R. M. A. Nawab, *Mono-lingual paraphrased text reuse and plagiarism detection*, University of Sheffield, Sheffield, England, Ph.D. thesis, 2012.
[37] S. Sameen, M. Sharjeel, R. M. A. Nawab, P. Rayson, and I. Muneer, "Measuring short text reuse for the Urdu language," *IEEE Access*, vol. 6, pp. 7412–7421, 2018.

Hindawi
Scientific Programming
Volume 2019, Article ID 9313571, 16 pages
https://doi.org/10.1155/2019/9313571

Research Article

Towards the Construction of a User Unique Authentication Mechanism on LMS Platforms through Model-Driven Engineering (MDE)

Jhon Francined Herrera-Cubides, Paulo Alonso Gaona-García, and Geiner Alexis Salcedo-Salgado

Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Correspondence should be addressed to Jhon Francined Herrera-Cubides; jfherrerac@udistrital.edu.co

Received 20 October 2018; Accepted 26 December 2018; Published 11 March 2019

Guest Editor: Vicente García-Díaz

In LOD, authentication is a key factor in the security dimension of linked data quality models. This is the case of (a) LMS that manages open educational resources (OERs), in training process, and (b) LMS integrated platforms, which also require authenticating users. Authentication tackles a range of problems such as users forgetting passwords and time consumption in repetitive logins in different applications. In the context of linked OERs that are developed in LMS, it is necessary to design guidelines in order to carry out the authentication process. This process authorizes access to different linked resources platforms. Therefore, to provide abstraction methods for this authentication process, it is proposed to work with model-driven architecture (MDA) approach. This paper proposes a security abstraction model on LMS, based on MDA. The proposed metamodel seeks to provide a set of guidelines on how to carry out unified authentication, establishing a common dialogue among stakeholders. Conclusion and future work are proposed in order to generate authentication instances that allow access to resources managed in different platforms.

1. Introduction

Model-driven engineering (MDE) [1–5] has as one of its first phases the search for the understanding of the problem to be addressed, that is, the knowledge of the problem domain. From this knowledge, an initial abstraction is constructed, using model schemes. These models are intended to be essential components for communication between the problem domain experts and software developers.

Based on the principle that the fundamental artifacts of development software are the models and not the programs, MDE proposes model-driven software development (MDD) based on the models that are generated from the most abstract to the most concrete through transformation or refinements steps, until arriving at the code applying a last transformation. Taking into account the above, the model-driven architecture (MDA), a concept promoted by object management group (OMG), is configured as an architecture that provides a set of guidelines to structure specifications expressed as models, following the MDD process [6]. On the other hand, linked open data (LOD) is a strategy to link open data licensed under one of the several open licenses that allow reuse [7]. In order to verify the linkage process, in addition to carry out different studies about the status of the data web [8–12], different linked data quality models have been proposed in LOD. These models define a dimensions and metrics set, such as those proposed by Zaveri et al. [13]. Within these dimensions, security has been considered, as a measure in which data access can be restricted and, therefore, protected against its alteration and misuse.

The security priority level depends on whether the data should be protected and whether there is a cost to the data that is unwittingly available. It should be noted that, although in the case of open data, the security aspect is often

not very developed given that the data are freely accessible under a specific license. However, today's students want to access the training at their own time rhythm and place, which has motivated them to implement LMS to join students' personalized needs who seek an online learning with resources in mobile and dynamic environments. For this reason, LMS platforms, and their integration with other applications, pose a security challenge in aspects such as (a) user authentication, which consumes linked resources in these applications; (b) the management and remembrance of multiple users and passwords; (c) continuous calls to reset passwords; (d) time consumption by continuous logins in different platforms; and (e) managing different authentication schemes according to the platforms or applications used, among other factors [14, 15]. With the purpose of analyzing and establishing criteria about the integration of security as a quality dimension in the linked open educational resources, the question arises about How to structure a metamodel that provides an authentication abstraction process for linked open educational resources, supported on linked data quality dimensions? To address this research question, the use of MDA is proposed, in order to design a metamodel, which allows the generation of authentication instances for the linked open educational resources, supported by quality dimensions.

In Section 2, the background about the research subject is described. Subsequently, the methodological approach is presented in Section 3. The methodological development used for the metamodel approach is described in Section 4. In section 5, the results obtained and the discussion about them are exposed. Conclusion and future work are presented in Section 6.

2. Related Work

The main references that support the theoretical foundations used in this research are described below.

2.1. Identification and Authentication. As described in [16–18], when a user connects to a computer system, he/she must provide

(i) User name or identification: identification is the ability to identify a user of a system or an application that is running in the system [19]

(ii) Password or authentication

Authentication is the ability to demonstrate that a user or an application is really who the said person or application claims to be [19]. To carry out this identity verification process, there are different proposals such as [17, 20, 21]

(i) Something that is known (e.g., password): the most basic authentication model is to decide if a user proves he is who he says he is. In this case, it is possible to use a knowledge test that only that user can answer.

(ii) Something you have (e.g., badge, token, and smart card): an example of these are smart cards, which have a chip embedded in the card itself that can implement an encrypted file system and cryptographic functions and can also detect actively invalid attempts to access stored information.

(iii) Something that one is (e.g., fingerprint, DNA, and iris): for example, the so-called biometric systems based on the physical characteristics of the user to be identified.

(iv) Where you are (e.g., using a particular terminal).

To carry out authentication, the steps of (a) obtaining the authentication information of an entity, (b) analyzing the data, and (c) determining whether the authentication information is effectively associated with the entity are carried out [16]. For this process, there are authentication mechanisms such as passwords, challenge-response, alternative mechanisms (information, tags, cards, biometrics, signatures, etc.), and multiple methods, among others [22].

2.2. Linked Data Quality. As discussed in [23], published data may suffer from different problems given the heterogeneity of the data source, such as redundancy, inconsistencies, or may be incomplete. Therefore, queries made by applications that consume LOD may be inaccurate, ambiguous, or unreliable. Different authors [24–28] have proposed models and metrics to evaluate LOD instances published on the Web. Some of the criteria worked on in these proposals are oriented to provenance, content, RDF structure, and links, among other factors. It is important to note that in most of the references, a treatment of the proposed quality dimensions is identified, on the data instances already published on the Web.

On the other hand, some of these authors emphasize that quality must not only operate in the resource construction but also in the metadata itself, in order to seek the interoperability of said resources [29]. For the work in this investigation, the model proposed by Zaveri et al. [30] is proposed as a fundamental basis. This model qualitatively analyzes 30 main approaches to quality assurance and 12 tools using a set of attributes. As a result of this review, data quality dimensions and metrics model in LOD are proposed by the authors. Dimensions of (a) accessibility, (b) representation, (c) contextual, (d) intrinsic, (e) trust, and (f) dataset dynamicity are identified in this model.

2.3. MDE-MDA. MDE is a software development approach focused on the model generation to describe the elements of a system. Its main objective is the separation of the system design both from the architecture and from the construction technologies, facilitating that design and architecture can be modified independently. In this section, it is important to present some essential concepts in MDE [31–36]:

(i) Meta-metamodel describes the proposed metamodels, generating a supremely high degree of abstraction in which all models coincide.

(ii) Metamodel is a general structure in which entities are managed but not instances of them. The metamodel guides the model construction, through the description of the basic structure to follow, in addition to showing

the interaction rules between defined entities. In summary, they are tools (rules, restrictions, models, and theories) that allow the model construction.

(iii) Model is the application of the metamodel in a particular case, in other words, a structure in which general entities are not managed, but rather with specific instances of them. In general, it is a description (simplified representation) of one or more domain elements or real world.

As for MDA, this architecture provides a set of guidelines for structuring specifications expressed as models [5]. MDA focuses on the following three principles:

(i) Direct representation focuses on the ideas and concepts of the problem domain and decreases the distance between the domain semantics and its representation, applies principles of abstraction, and separates relevant aspects from the problem domain of technology decisions

(ii) Automation promotes the use of functionalities such as the exchange of models, the management of metamodels, the verification of consistency, and the transformation of models

(iii) The use of open standards: the purpose is to achieve the interoperability in different tools and platforms, promoting the applications development

Metamodels in the context of MDA are expressed using meta-object facility (MOF), which proposes a 4-layer scheme: M0 (instances), M1 (the model), M2 (the metamodel), which define the elements of the model, and M3 (the meta-metamodel), which defines concepts, attributes, and relationships for the elements. Now, MDA adds to the model-driven approach the inclusion of several levels of abstraction (CIM, computation independent model; PIM, platform independent model; PSM, platform specific model) and several transformations between levels, thus carrying out system descriptions to several levels [2, 4, 35]. The development steps proposed in MDA are as follows [6]:

(a) The system requirements are presented in a CIM model, which describes the situation in which the system will be used

(b) The CIM model is transformed into a PIM model that describes the system, but does not show the details of its use in a particular technological platform

(c) After obtaining the PIM model, another transformation is made to the PSM model, which contains the necessary detail to use the technological platform in which the system will operate

(d) Finally, having the PSM model, a transformation is performed which results in the generation of code to achieve a solution or executable model

3. Methodological Approach

This research works on a quasiexperimental methodology. This design, as an empirical method, allows the analysis of the properties resulting from the application of the technological process, to obtain an analysis of the proposed variables. Considering that metamodels in the context of MDA are expressed in a 4-layer scheme, for this research process, M0: the instances; M1: the model; and M2: the metamodel layers are considered. From this perspective, the need to carry out a quasiexperimental design was proposed, in order to conceptually consider the abstraction process of the security dimension in the context of linked open educational resources, based on metamodels. This methodology is shown in Figure 1.

To develop this proposal, a methodological design structured in phases was defined, which allows to determine the processes that led to this research. In order to carry out the proposed design, the following phases are described succinctly:

(a) Context approach: in this phase, in order to identify the theoretical elements of the research, the following question was designed to guide the research process: How to structure a metamodel that provides an authentication abstraction process for the linked open educational resources, supported on linked data quality dimensions?

Based on the design of a metamodel for the abstraction of the authentication process for the linked open educational resources, it is important to take into account aspects of model-driven engineering and linked data quality, as key elements for the metamodel design, which allows generating guidelines for the construction of this abstraction.

(b) LMS authentication: in this phase, the research problem and the strategies identification used for the authentication in LMS are proposed, as the main axis in the abstraction process of the knowledge domain. In the same way, the key requirements are defined, to feed the build process of the proposed metamodel.

(c) Metamodel layers definition: after the requirements definition, the identification of the proposed layers for the elaboration of the conceptual design that will guide the metamodel implementation is carried out.

(d) Verification example design: in this phase, the verification example is carried out that will show the follow-up to the conceptual design that will guide the metamodel implementation.

(e) Results analysis: with the abstraction of the protocol to be followed and the corresponding application in the case of study, the proposed design is evaluated.

4. Methodological Development

4.1. LMS Authentication. In general, information is stored in an authentication process to carry out this process, such as user and password. As described in [37], in order to authenticate itself, a user generally provides at least two elements: (a) its identifier that allows its definition and (b) one or more elements that guarantee the authentication itself.

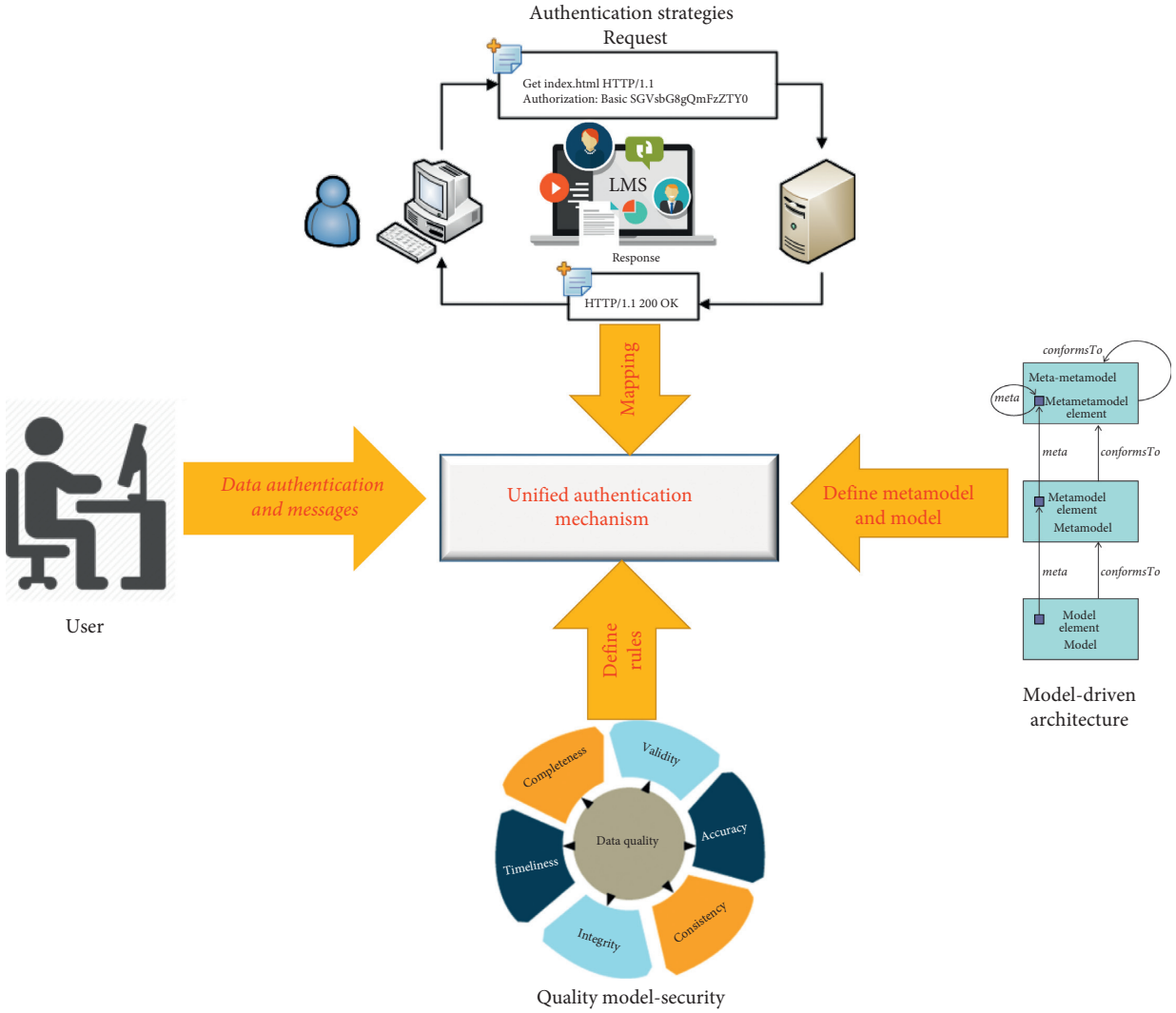


Figure 1: Methodological approach.

Thus, independent of the mechanism, the same elements are found in different forms, for example,

(i) Identifier and password.

(ii) PKI certificates on smart card or USB token: identifier is a public certificate that is signed and consequently guaranteed by a recognized certification authority. The user must provide a secret element to be able to use the different cryptographic elements, for example, "PIN code of your card or your USB key."

(iii) Identifier and password on a smart card.

(iv) Authentication by biometrics is based on the verification of an element of the user's body (usually the fingerprint).

(v) In multifactor authentication, different combinations such as smart card + PIN code, smart card + biometric, and biometric + password, among others, can be registered.

As for the learning management systems (LMSs), these are systems whose main function corresponds to providing sufficient support for the mediation of appropriation of knowledge and its administration, access to didactic and communication tools, and reuse of contents, among others. To carry out their objective, LMS must provide services and tools such as authentication. For such purpose, LMS must have an infrastructure to guarantee the users authentication [38]. In LMS such as Moodle, different authentication mechanisms through modules are developed, which allow easy integration with existing systems. Within these mechanisms are the following [39]:

(i) Standard method of e-mail registration: students can create their own access accounts. The e-mail address is verified by confirmation.

(ii) Lightweight Directory Access Protocol (LDAP) method: access accounts can be verified on an LDAP server. The administrator can specify which fields to use.

(iii) Internet Message Access Protocol (IMAP), Post Office Protocol (POP3), and Network News Transport Protocol (NNTP): access accounts are

verified against a mail or news server (news). Supports secure sockets layer (SSL) certificates and transport layer security (TLS).

(iv) External database: any database, which contains at least two fields, can be used as an external authentication source.

An example of user authentication interfaces which access different applications of the District University (Academic Management and LMS) is shown in Figure 2. Each of them has an independent authentication mechanism.

4.2. Metamodel Layers Definition. According to the layer scheme defined for the metamodels in the MDA context, layers below were proposed to work on the framework project: (a) M0 (the authentication), (b) M1 (the authentication model), and (c) M2 (the metamodel). The structure of the meta-object facility (MOF) proposed for the project is shown in Figure 3. In this architecture, the following is proposed:

(i) A metamodel that defines restrictions, rules, and theories set must be followed by the representations made about it. This layer will define requirements and restrictions, which must be met in any authentication process.

(ii) A model that characterizes both the defined authentication strategy, supported on the quality dimension, and the requirements definition that these must meet in their design.

(iii) A model instance, which represents the abstraction made of the knowledge domain, adjusted under metamodel constraints, in an own domain of linked data quality.

4.3. Verification Example Design. For the verification example design, the metamodel requirements must be identified in the first instance.

Case study: "A digital educational resources bank is connected to several institutional repositories. When a user enters the resource bank, after logging in, he should be able to access the different LMSs that manage the resources there related, even more so when said LMS corresponds to different providers. This single access provides the user with a username and password for each of the repositories, in order to access and consume the resources published there."

For this problem context, metadata or data model requirements are not handled. The domains used correspond to security and LMS.

4.3.1. Knowledge Domain. The knowledge domain for the problem posed presents the following knowledge instances:

(a) *Quality Dimension*. For this verification exercise, the security dimension was worked on, grouped in the accessibility category, based on the quality model proposed by Zaveri et al. [13]. According to the authors, this aspect describes the following:

(i) Accessibility: the dimensions that belong to this category involve aspects related to the access and retrieval of data to obtain all or part of the data for a particular use case. There are five dimensions that are part of this group, which are availability, licensing, interconnection, security, and performance.

(ii) Security: security is the metrics to which data access can be restricted and, therefore, protected against its alteration and misuse. Security is measured according to whether the data have an owner or require web security techniques (e.g., SSL or SSH) for accessing, acquiring, or reusing the data by users. The importance of security depends on whether the data should be protected and whether there is a cost of data that is unwittingly available. For open data, security is often not very developed since the data are freely accessible under a specific license.

(b) *Digital Repositories*. As described in [41], a learning object repository (ROA) is a software system that stores educational resources and their metadata (or, only, the latter) and provides some type of resource search interface, either for interaction with humans or with other software systems. Additionally, there are learning management systems (LMSs), which allow designing courses based on the reuse and integration of learning objects. These resources have been searched and previously selected in repositories. Regarding authentication, an LMS must have an infrastructure to guarantee the authentication of its users.

(c) *Identification and Authentication*. Identification is the ability to uniquely identify a system user or an application that is running on the system. Authentication is the ability to demonstrate that a user or an application is really who the said person or application claims to be [16].

(d) *Single Sign-On*. It is an authentication mechanism [42], which allows users to access different systems through a single identification instance. In other words, single sign-on (SSO) is a concept to delegate the authentication of an end-user on a service provider (SP) to a third party, the so-called identity provider (IdP) [43]. The behavior proposed by single sign-on is shown in Figure 4.

In SSO, there are common configurations:

(i) Enterprise single sign-on (E-SSO): It operates as a primary authentication and intercepts the login requirements that are required by the secondary applications in order to complete the user and password fields. For the correct operation of E-SSO, it is necessary that the underlying applications allow to disable the login interface.

(ii) Web-based single sign-on (Web-SSO): this type of solution operates only with applications and

(a) (b)

Figure 2: Examples of authentication interfaces. (a) Academic management system. (b) LMS system. Source: [40].

resources that are accessed through the web. The access data are intercepted through a proxy, a component on the server, or the portion of software running on the client. Users, who have not been authenticated yet, are redirected to an authentication service from which they must return with a token or successful access.

(iii) Kerberos: users register on a server and obtain a ticket (TGT: ticket-granting ticket), which is used by client applications to gain access.

(iv) Federated identity: it corresponds to an identity management solution or identity management, which allows using the credentials available in one authentication system in others, either from the same organization or even from other companies. The above is done with standards that define mechanisms to share information between domains.

After a review, different SSO implementations were identified. These *related works* are shown below:

(i) Web-SSO is a used technique to allow users to easily register and sign-in to websites with the use of social media accounts. These websites can be associated with new applications downloaded from Apple's App Store, Android's Google Play store, or even accessing website accounts [45].

(ii) In Austria, most public sector applications use an open-source identity provider called MOA-ID. However, due to the sectorial identity management, MOA-ID has not been single sign-on capable. A security architecture that enables single sign-on between different governmental applications using MOA-ID as identity provider while meeting the requirements for sectorial data privacy protection at the same time is presented in [46]. This research achieves this by transforming unique sectorial identifiers of users with the help of an additional trusted attribute provider.

(iii) A system which wants to integrate information systems by using web services should provide a unified identity authentication single sign-on scheme for heterogeneous platforms. This research introduces the characteristics of Kerberos-based single sign-on and SAML-based single sign-on [47].

(iv) Federated access control schemes such as SAML and OpenID for authentication or OAuth for authorization enable secure, cross-domain single sign-on for web and mobile applications regardless of where users are located or what device they are using to start the authentication or authorization flows. Using federated schemes allow users to avoid managing as many user names and passwords as services they want to interact with. Instead, they ask for a token at some kind of identity provider or verifier and all services participating in the federation trust these tokens to solve authentication and/or authorization [48].

(v) OpenID Connect is the OAuth 2.0-based replacement for OpenID 2.0 (OpenID) and one of the most important single sign-on (SSO) protocols used for delegated authentication. It is used by companies like Amazon, Google, Microsoft, and PayPal [43].

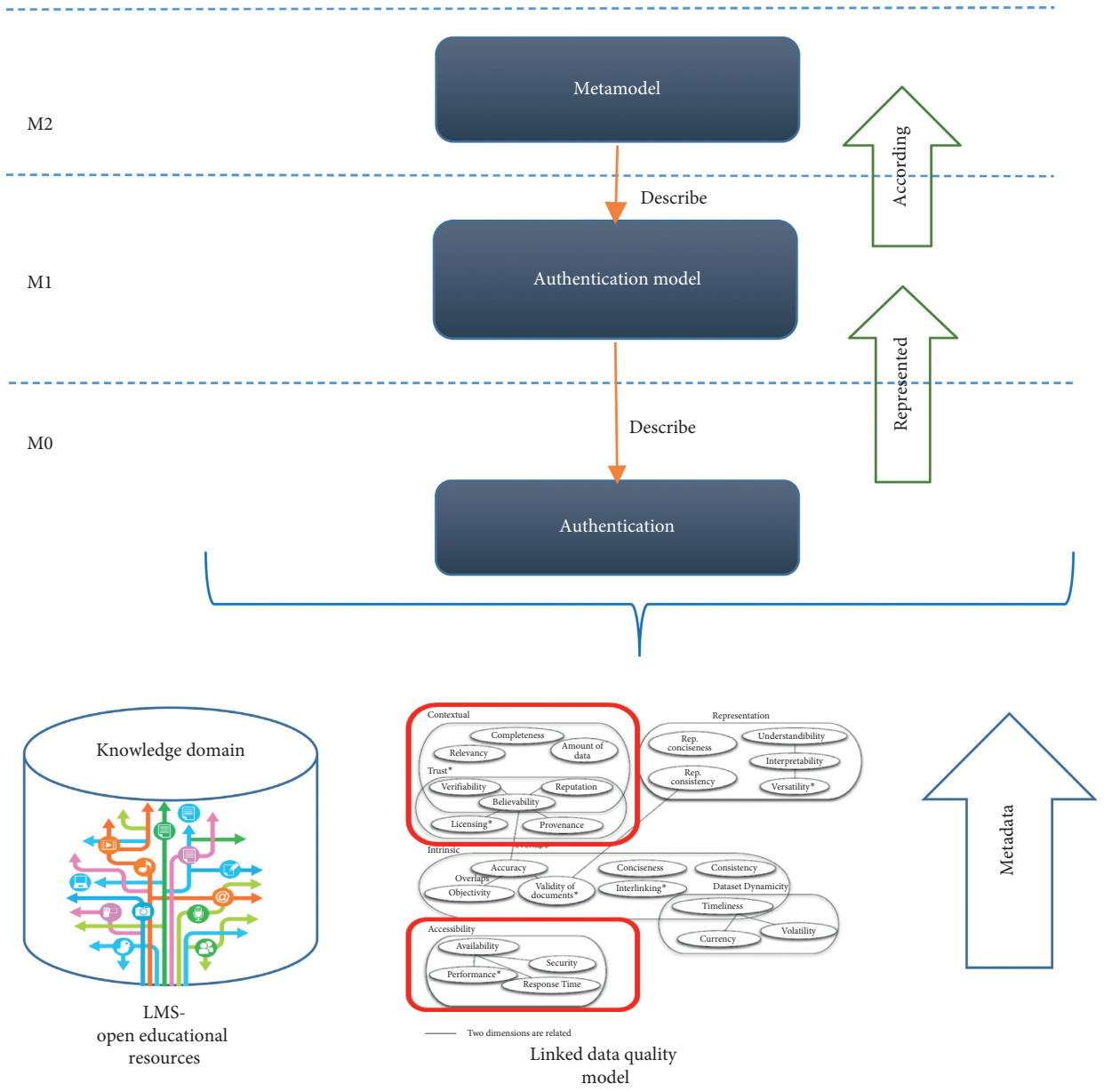


Figure 3: Project MOF.

5. Results and Discussion

5.1. Requirements. Identified requirements are as follows:

(i) Requirement R1 provides a security mechanism for accessing users and acquiring or reusing data.

(ii) Requirement R2 provides an authentication mechanism that allows a user to access the LMS instances, where resources are managed.

5.2. Metamodel. Taking into account the above requirements, it is necessary to consider some situations described below, in order to build the metamodel:

(a) Accessibility in a secure way to different LMSs, using the same access point, involves an authentication layer, which allows authenticating the access of consumers to exposed digital resources in each LMS.

(b) In addition to the accessibility to different LMSs, it is possible to integrate the LMS with other tools, such as customer relationship management (CRM), electronic commerce, or any other application. This authentication scheme in different applications should also have a single authentication layer.

In other words, the task of this layer is twofold: on the one hand, integrating each authentication scheme from different LMSs, and on the other hand, offering an authentication service to users. Since each integrated LMS has its own authentication model, it is necessary to make an abstraction over multiple authentication models, unifying them under this basic metamodel. The proposed general

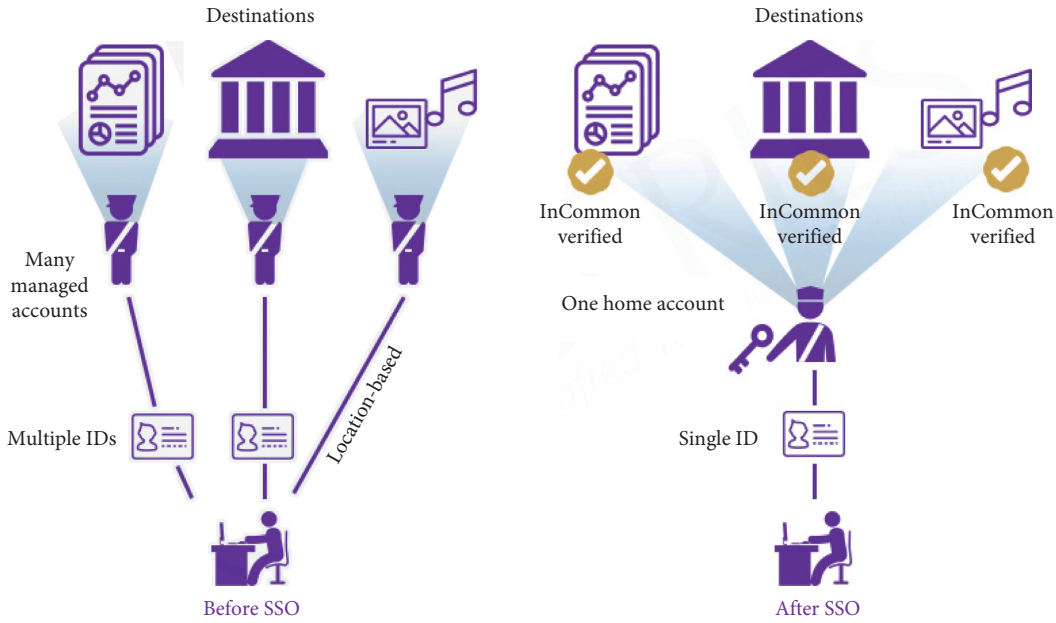


Figure 4: Single sign-on. Source: [44].

metamodel is illustrated in Figure 5. The implementation of this authentication layer uses a metamodel that consists of the following three simple classes:

(i) *Entity*. They are both the user who wishes to carry out the authentication and the authenticating entity, which handles the unified authentication mechanism and which can map the attributes to each LMS authentication scheme or integrated applications.

(ii) *Attribute*. These factors are captured to carry out the identification (e.g., the nick) and the authentication (e.g., the token).

(iii) *Message*. These are the request and response messages made between user and unified authentication system.

The restrictions that the metamodel handles are defined according to the unified authentication factors. As described in [24], different criteria can be configured, which can be taken as factors for authentication. An example of configuring basic criteria for IMAP is shown in Table 1.

According to the characteristics of the LMS authentication modules, the use of an identifier factor and an authenticating factor (pin, biometric, or a simple password) can be abstracted. To specify a little more, the requirements proposed by the domain abstraction, a more detailed metamodel where the entities, attributes, and messages are specified in textual form, are shown in Figure 6.

As seen in Figure 6, after receiving the message from the "User" entity, "Authenticator" entity processes the authentication of the factors submitted as attributes (credentials), generating a response message, either the authorization or the rejection of the user.

5.3. Model. To perform the interaction between entity, attribute, and message, the model design is proposed, which, in addition to responding to the criteria established in the metamodel, encapsulates access to identity management functions and provides a single session on. For this, delegator pattern in [51] is proposed, which allows an independent evolution of the weakly coupled identity management services while providing system availability. The class diagram, which visualizes the unique authentication model implementation based on a delegator pattern, is shown in Figure 7.

With this pattern implementation, the client does not interact directly with identity management service interfaces. The delegate prepares for the single session on, configures security session, looks for the physical security service interfaces, invokes the appropriate security service interfaces, and performs the global logout at the end [51].

5.4. Instances. The proposed model instance is based on single sign-on. An example of the login process in the TalentLms domain, through the SSO service, is shown in Figure 8.

Implementations of SSO in Canvas, WordPress, Atlassian, Jooomla, Drupal, and Magento, among other platforms, are described in [53, 54]. Research on SSO, which is being carried out at the Universidad Distrital Francisco José de Caldas, is shown in Figures 9 and 10. This proposal is based on The OAuth 2.0 Authorization Framework (RFC 6749) [55], a framework based on refresh tokens which are credentials used to obtain access tokens.

The developed interface is shown in Figure 11.

The aim of this research is to integrate academic applications, used at the Universidad Distrital. Subsequently, learning resources repositories will be integrated. For this integration, its metamodel proposes a set of recommendations. These recommendations will allow mapping generated tokens, to each learning objects repositories. This process lets

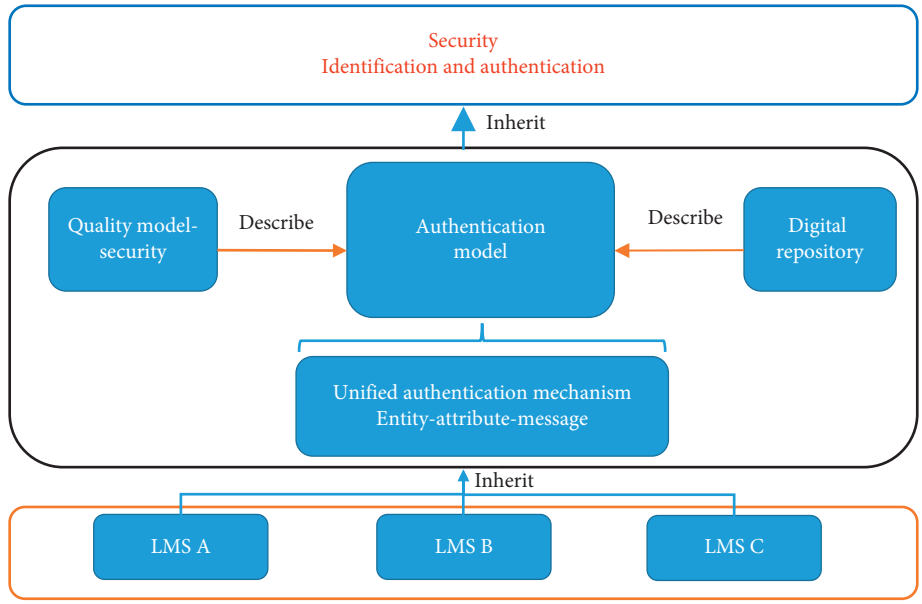


Figure 5: General metamodel proposal.

Table 1: IMAP configuration parameters.

Username	Personal name (e.g., Juan Carlos Pérez)
E-mail	Main address, for example, juan.perez@alumnos.unican.es
Answer	The same
Incoming IMAP mail server	imap.alumnos.unican.es with SSL protocol (port 993)
Outgoing SMTP mail server	smtp.alumnos.unican.es with TLS protocol (port 587)
Account name	username@alumnos.unican.es (e.g., xyz01@alumnos.unican.es)
Password	User password
Other parameters	Enable outbound authentication (SMTP) with the same username and password

Source: [49].

perform respective authentication. In other words, for each authentication model, respectively, instantiated, the unified authentication mechanism will manage tokens for authentication process and according to their validity will allow access to the respective repository.

The proposed model focuses on user authentication and validation process that will allow them to access educational resources. Basic architecture for the credential verification process and after that access for query or management of educational resources is shown in Figure 12. Different points are shown in this model. Some of them are relationship between Web applications that share educational resources and messages passing to access control to resources. Access control is done by credentials, which are validated in SSO.

Token must be sent with each request that is made for queries or management of educational resources. When a request is made in the API manager, validation process starts, taking the token from the request header to query it in the authentication server and grant access permissions and consumption service in the API. By obtaining an educational resources list located in other applications, resource access will be transparent to users, since message flow for access authorization will be managed with the same implicit flow.

This model allows controlling access to educational resources on different applications that require credentials validation. In addition to manage permissions to repositories access, this model allows to edit resource information (if the application allows it). The flow is controlled and is supported through the most used authentication protocols, such as OAuth2.

As a discussion about what is the broad application of the SSO model beyond the case study proposed, a single authentication design has different advantages and disadvantages, which are exposed [39, 43, 45–48, 51, 58, 59]. Among the advantages, the following advantages are identified:

(i) Minimizing the amount of passwords and usernames, which are used in password-based authentication for instance, and the ease in signing up for new websites and apps.

(ii) With a single session record, factors such as the use and possible forgetting of multiple access keys are attacked, as well as reducing the time in different authentication processes.

(iii) Using model implementation patterns, such as delegation, allows to improve the handling of the sessions, since the single sign-on service creates a secure SSO session and delegates the service requests to the relevant security services.

(iv) By avoiding centralized management of users and authorizations, the applications themselves have to implement these mechanisms (it is delegated to the SSO itself), streamlining the process of provisioning user credentials for the different applications, and

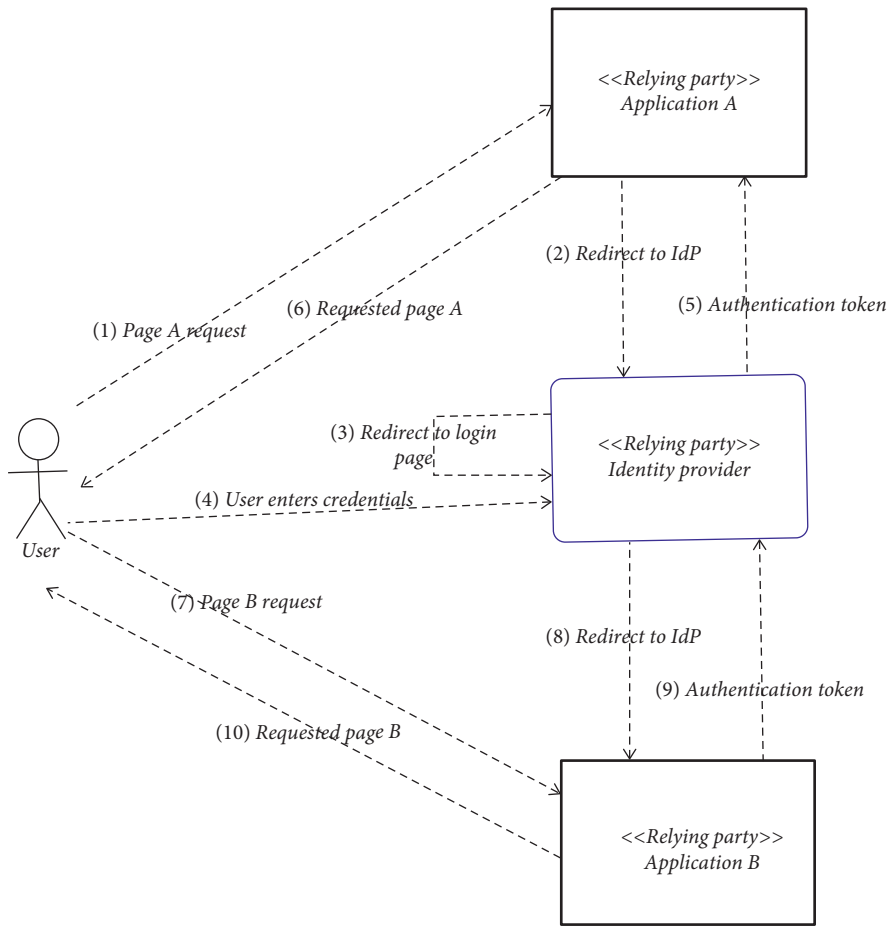


Figure 6: Detailed metamodel proposal. Source: [50].

automating this process considerably reduces the error probability.

(v) Multiplatform system allows the integration of different systems from different manufacturers in a single user authentication mechanism.

Disadvantages are as follows:

(i) The SSO system accessing process allows an intruder to access all the systems covered by the SSO system. This inconvenience is usually mitigated by making the authentication process much stricter than in the usual processes.

(ii) When SSO is implemented, it must be used very carefully. This is especially true if there is not complete control over who is authenticated by the identity provider. Authentication only provides information about the user's identity; for this reason, it should be verified separately in each application what should be visible to him/her.

However, some *contributions and findings* were identified in the review as follows:

(i) Using single sign-on delegator pattern, multiple instances of the remote security services will help improve scalability and support a standards-based single sign-on framework that does not require users to sign-on multiple times [51].

(ii) The SSO process for web applications can use two techniques (a) using passive redirection mechanism: applications that are involved in the process do not communicate directly with each other, but rely on browser's redirection and standard HTTP GET and POST messages. (b) Using "active SSO": when a relying party application talks directly (e.g., via a Web Service) to the identity provider to validate the user's identity and obtain the related security token [50].

(iii) Under a corporate environment, the user needs to remember only one set of credentials to access various resources in and out of the organization's network, in addition to increasing productivity by avoiding reentering your password to authenticate yourself in various resources repeatedly [52].

(iv) Burden on the IT is cut down due to fewer helpdesk requests for password resets. Centralized authentication center and identity management allows quicker and better control over the accesses granted to each user [53].

(v) SSO can be mixed with any of authentication methods (e.g., security questions, mobile authentication, and

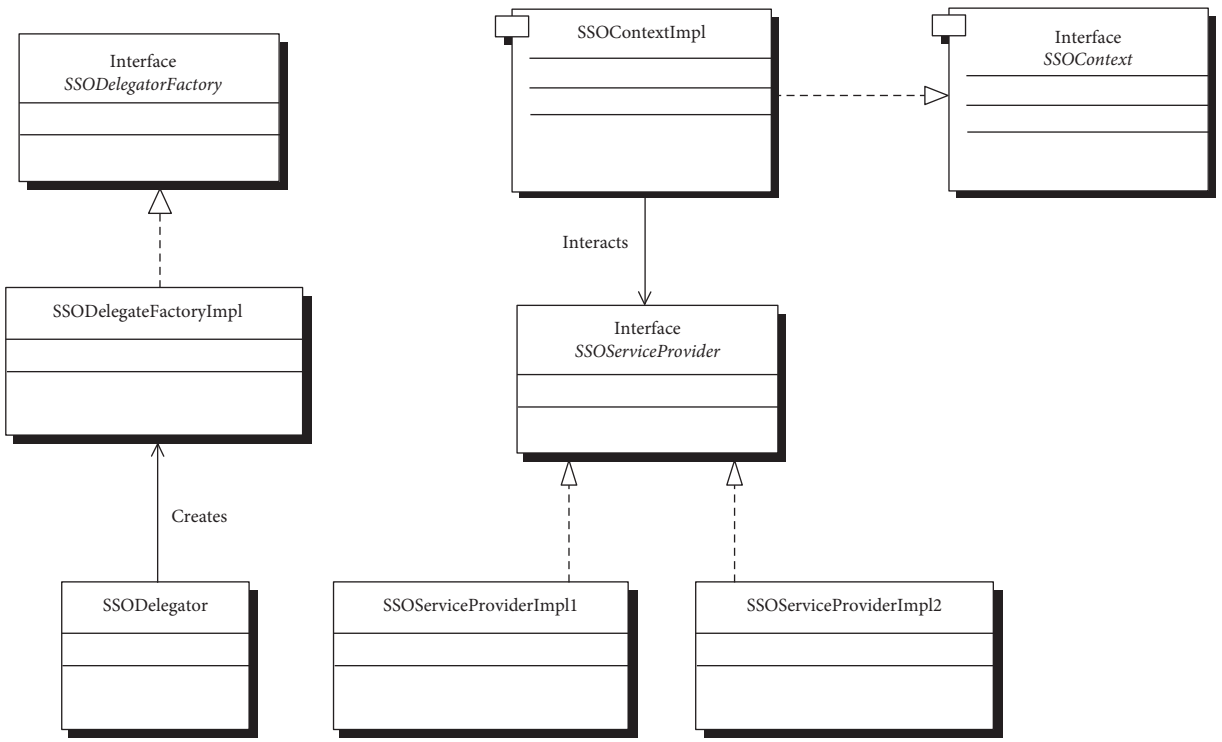


Figure 7: Proposed model class diagram. Source: [51].

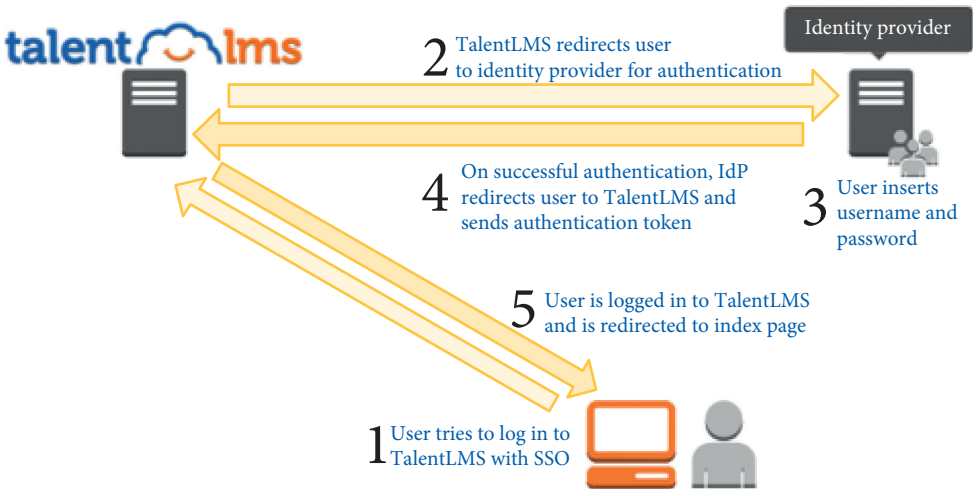


Figure 8: Single sign-on example. Source: [52].

voice authentication) to augment your password-based authentication [54].

Finally, regarding the management of educational resources, which are stored in repositories with access control, SSO implementation guarantees access to these resources, performing a validation process of credentials from a centralized node. In this node, each one of the LMS that wants to query resources is registered. In this registration process, each application requests credentials validation to carry out access to resources. When a user wants to query a resource in a repository (which has valid credentials), one of the following scenarios could happen:

(a) User is logged in into the centralized authentication node: if the user has previously accessed the centralized authentication module, this user will have access credentials for all those applications or repositories registered in the centralized node, so that credentials validation would be transparent to the user, allowing him/her to access to the educational resource.

(b) User is not logged in, but if he/she is registered in centralized authentication node: when user tries to query the educational resource, a credential restriction will not allow him/her to query this

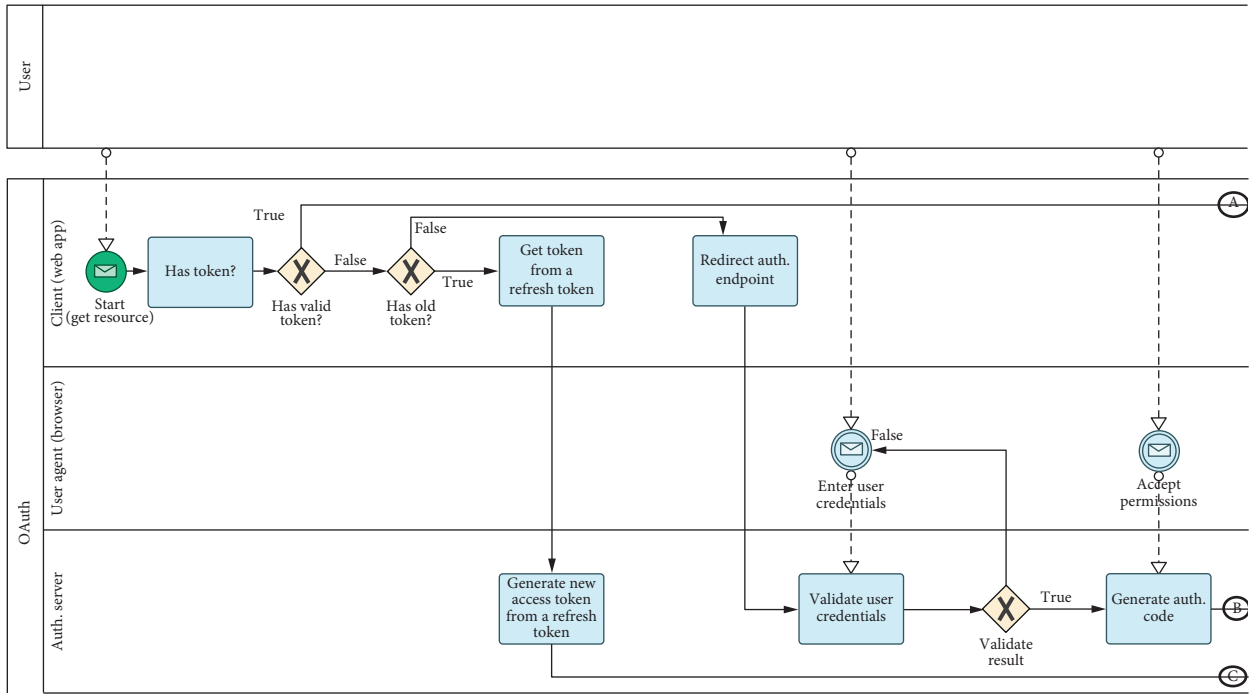


Figure 9: OAuth 2.0 authorization code grant: first part. Source: [55, 56].

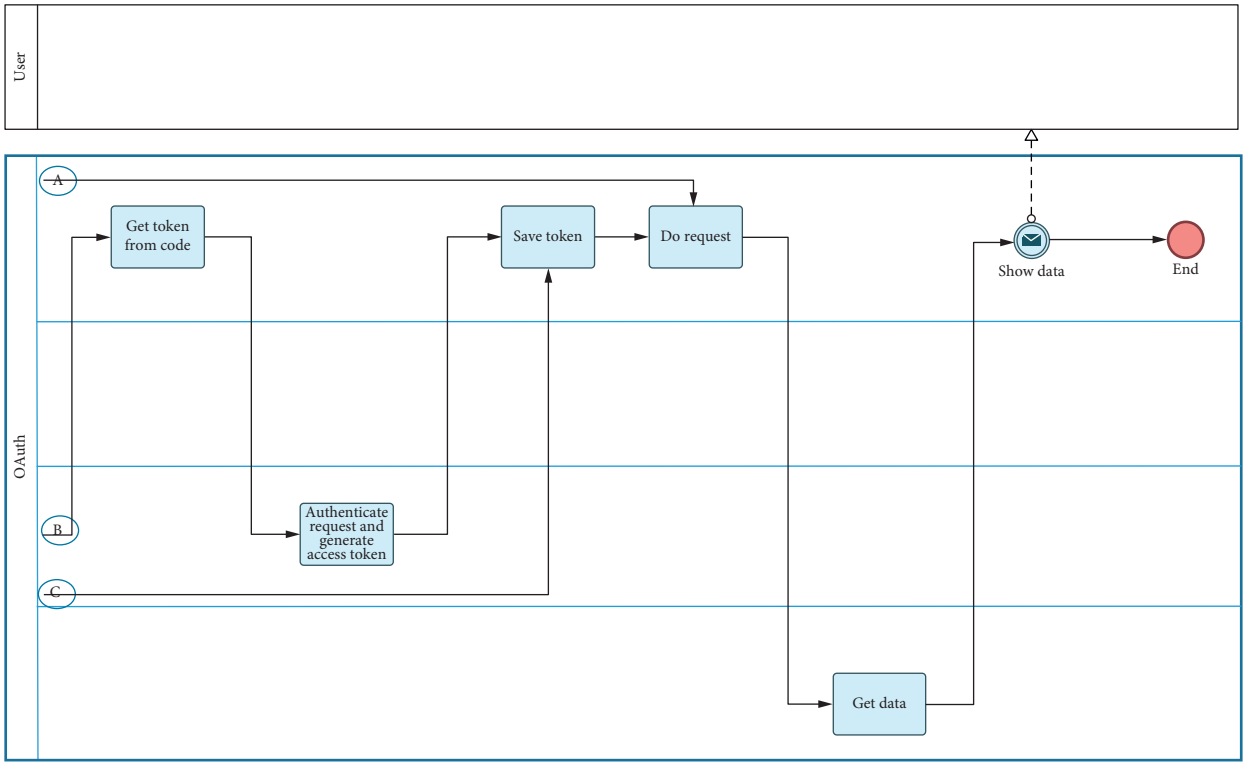


Figure 10: OAuth 2.0 authorization code grant: second part. Source: [55, 56].

resource, unless he/she accesses the application from the centralized node or from the application itself.

(c) User is neither logged nor registered: this case is similar to the previous one; therefore, the process will be the same. Additionally, an option to register and query the educational resources will be offered by the centralized node.

In a nutshell, metamodel design for the security dimension abstraction, and specifically different LMS and

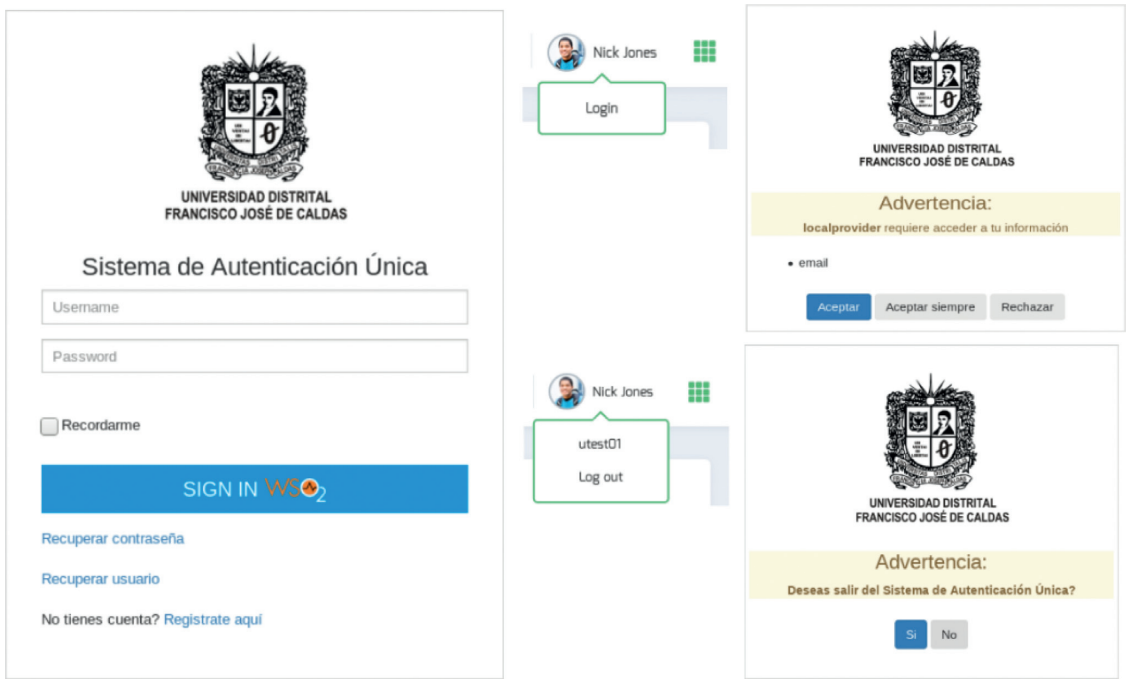


Figure 11: Proposed SSO: UDistrital. Left: login. Right: access authorization (above) and logout confirmation (below). Source: [57].

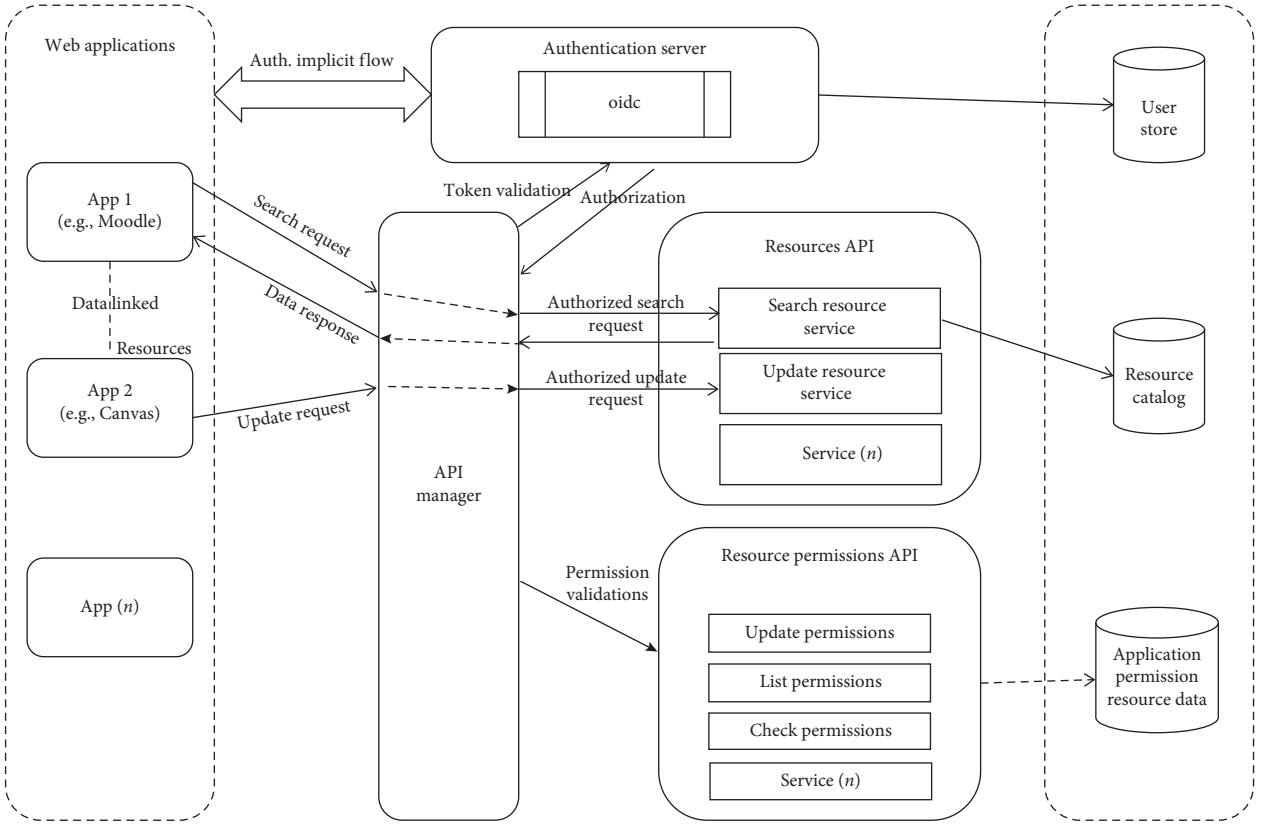


Figure 12: Architecture SSO: UDistrital. Source: authors.

authentication of integrated applications, is configured as a strategy, which provides an authentication unifying model. This model provides users with a single set of authenticating factors, which authorize applications under them authenticated. However, task of integrating each authentication scheme from different LMSs, where each one handles its own authentication model, is configured as restriction basis that metamodel must carefully manage, so that entities, attributes, and messages exposed perform corresponding mapping to each of the authentication instances. Derived

from this, use of design patterns becomes a priority strategy when designing models.

Briefly, abstraction level offered by MDA becomes a useful tool when planning authentication scenarios, not only for access and authorization in LMS, but in different environments where applications are integrated and required to simplify user identification and authentication process.

6. Conclusion and Future Works

MDA, besides raising different abstraction levels, which are represented in models, allows the automatic code generation using built models. For this purpose, MDA makes use of metamodels, which correspond to a set of domain concepts to be modeled and the existing relationships between them, defined in an abstract way. The metamodels allow carrying out a better abstraction of the knowledge domain, through the identification of concepts, rules, restrictions, etc., which operate in the domain, facilitating their understanding.

Regarding the main aspects to be taken into account for the authentication metamodel definition for accessing the LMS, and the use of linked open educational resources, for the model back-end, the metamodel should consider (1) the identification and characterization of the authentication schemes from the different applications that are to be integrated and (2) the authentication factors identification and the mechanisms used to carry out this task. These elements provide relevant criteria and restrictions that are raised in the metamodel. These criteria are implemented in the design of the proposed model, through (a) entities that participate in the authentication process, (b) attributes or authentication factors which are mapped to different schemes, and (c) messages which are sent and received among different entities that participate in model instances.

Regarding the model's front-end, the metamodel must consider the parameterization of restrictions on authentication factors, which are requested from the user. Using this parameterization process, the factors can meet all necessary requirements in order to be mapped, by the integrated authentication unit, with each application authentication scheme, which are integrated into the model.

In the authentication metamodel, the linked educational resources taxonomy, or the metadata provided, is not considered relevant, since at this level what is sought is to provide an access and authorization mechanism to the platforms that manage those resources. In the generation of model instances, considerations such as the following should be taken into account:

(i) With a single authentication point, authentication factors that authorize access and use of resources in different platforms are provided, according to the defined profiles. Therefore, the use of stronger authentication mechanisms should be considered, such as the use of multifactor authentication methods, which prevent unauthorized users from accessing information and resources that only have a password as an authentication factor.

(ii) In the authentication process, only information about the user's identity is provided. Authorizations about what is visible to each user should be verified separately in each application.

Taking into account the above, the use of metamodels for the authentication abstraction is configured as a strategy of the knowledge domain representation, which will allow to define restrictions and necessary components so that an administrator can add new authentication mechanisms, in addition to providing new applications to the model that implements these authentication mechanisms. Future works are proposed (a) to extend the metamodeling process to the other linked data quality dimensions, proposed in the framework research of this project; (b) to design a metamodel for the process of generating data models for linked open educational resources, based on linked data quality dimensions; and (c) to carry out the framework implementation, which allows verification of the proposed metamodel for the linked open educational resources; and finally, to develop a machine learning solution to measure the level of trust in different queried repositories, after validation of access credentials.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The current work has been developed within the doctoral research project framework on Linked Data at the Universidad Distrital Francisco José de Caldas. In the same way, Linked Data is also being worked as a research topic of the GIIRA Research Group.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison-esley, Boston, MA, USA, 2003, ISBN: 0-321-19442-X, https://dl.acm.org/citation.cfm?id=829557.

[2] Z. Bizonova, D. Ranc, and M. Drozdova, "Model driven e-learning platform integration," in *Proceedings of CEUR Workshop*, Busan, Korea, November 2007, http://ceur-ws.org/Vol-288/p02.pdf.

[3] V. García-Díaz, J. Tolosa, B. G-Bustelo, E. Palacios-González, O. Sanjuan-Martínez, and R. Crespo, "TALISMAN MDE framework: an architecture for intelligent model-driven engineering," in *Lecture Notes in Computer Science*, vol. 5518, Springer, Berlin, Germany, 2009.

[4] A. Rodrigues da Silva, "Model-driven engineering: a survey supported by the unified conceptual model," *Computer Languages, Systems & Structures*, vol. 43, pp. 139–155, 2015.

[5] D. Orozco, W. Giraldo, and H. Trefftz, *MDE; MDA; Transformaciones y DSLs. Una breve introducción*, Universidad Eafit, Medellín, Colombia, 2013, https://repository.eafit.edu.co/bitstream/handle/10784/5107/Articulo8CCC.pdf?sequence=4&isAllowed=y.

[6] F. Aguillón Martínez and M. Mateus Gómez, *Automatización del desarrollo de aplicaciones web mediante el enfoque MDA-MDE*, Facultad de Ingeniería, Pontificia Universidad Javeriana, Colombia, Bogotá, Colombia, 2014, https://repository.javeriana.edu.co/handle/10554/15572.

[7] B. Hyland, G. Atemezing, M. Pendleton, and B. Srivastava, *Linked Data Glossary*, W3C Working Group, Dublin, Ireland, 2013, https://www.w3.org/TR/ld-glossary/#linked-open-data.

[8] J. Herrera-Cubides, P. Gaona-Garcia, and S. Sánchez-Alonso, "The web of data: past, present and ¿future?," in *Proceedings of XI Latin American Conference on Learning Objects and Technology (LACLO)*, pp. 1–8, San Carlos, AL, Costa Rica, October 2016.

[9] J. Herrera-Cubides, P. Gaona-García, J. Alonso Echeverri, K. R. Vargas, and A. Gómez Acosta, "A Fuzzy logic system to evaluate levels of trust on linked open data resources," *Revista Facultad de Ingeniería*, no. 86, pp. 40–53, 2018.

[10] P. Gaona-García, A. Fermosa-García, and S. Sánchez-Alonso, "Exploring the relevance of europeana digital resources: preliminary ideas on europeana metadata quality," *Revista Interamericana de Bibliotecología*, vol. 40, no. 1, pp. 59–69, 2017.

[11] P. Gaona-Garcia, K. Gordillo, C. Montenegro-Marin, and A. Gómez-Acosta, "Visualizing security principles to access resources based on linked open data: case study DBpedia," *Information: An International Interdisciplinary Journal*, vol. 21, no. 1, pp. 109–122, 2018.

[12] J. Herrera-Cubides, P. Gaona-García, and K. Gordillo-Orjuela, "A view of the web of data. case study: use of services CKAN," *Revista Ingeniería*, vol. 22, no. 1, pp. 111–124, 2017.

[13] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, "Quality assessment methodologies for linked open data. a systematic literature review and conceptual framework," *Semantic Web Journal*, vol. 7, no. 1, pp. 63–93, 2012.

[14] F. McSweeney, "Five reasons to use single sign-on (SSO) with Workable," Workable, 2018, https://blog.workable.com/use-sso-with-workable/.

[15] E. McKeown, "What is single sign-on (SSO)? Ping identity, 2017, https://www.pingidentity.com/en/company/blog/2017/08/23/what_is_single_sign-on_sso.html.

[16] GSI, "Seguridad informática," Grupo de Seguridad informática, 2018, https://eva.fing.edu.uy/pluginfile.php/58016/mod_resource/content/6/FSI-2018-IAA.pdf.

[17] J. Lanza Calderón and L. Sánchez González, "Seguridad en Redes de Comunicación," in *Grupo de Ingeniería Telemática*, Departamento de Ingeniería de Comunicaciones, Universidad de Cantabria, Santander, Spain, 2015, https://ocw.unican.es/course/view.php?id=28.

[18] Mentor, "Mecanismos básicos de Seguridad," in *Seguridad Informática*, Torrelavega, Spainhttp://descargas.pntic.mec.es/mentor/visitas/demoSeguridadInformatica/mecanismos_bsicos_de_seguridad.html.

[19] IBM, *Identificación y Autenticación*, IBM Knowledge Center, New York, NY, USA, 2016, https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009740_.htm.

[20] J. Montoya and Z. Restrepo, "Gestión de identidades y control de acceso desde una perspectiva organizacional," *Ingenierías USBMed*, vol. 3, no. 1, pp. 23–34, 2012.

[21] RedIris, *Autenticación de usuarios*, Red Académica y de Investigación Nacional Iris, Madrid, Spain, 2008, https://www.rediris.es/cert/doc/unixsec/node14.html.

[22] Oracle, *Guía de administración del sistema: servicios de seguridad*, Oracle, Redwood City, CA, USA, 2011, https://docs.oracle.com/cd/E24842_01/html/E23286/toc.html.

[23] E. Ruckhaus, M. Vidal, S. Castillo, O. Burguillos, and O. Baldizan, "Analyzing linked data quality with LiQuate," in *The Semantic Web: ESWC 2014, Lecture Notes in Computer Science*, vol. 8798, Springer, Berlin, Germany, 2014.

[24] J. Pattanaphanchai, "DC proposal: evaluating trustworthiness of web content using semantic web technologies," in *Lecture Notes in Computer Science*, vol. 7032, Springer, Berlin, Germany, 2011.

[25] A. Rula and A. Zaveri, "Methodology for assessment of linked data quality," in *Proceedings of LDQ 2014, 1st Workshop on Linked Data Quality*, pp. 1–4, Leipzig, Germany, September 2014, http://ceur-ws.org/Vol-1215/paper-04.pdf.

[26] F. Radulovic, N. Mihindukulasooriya, R. García-Castro, and A. Gómez-Pérez, "A comprehensive quality model for Linked Data," *Semantic Web*, vol. 9, pp. 3–24, 2018.

[27] C. C. T. Di Noia, B. Marcu, and M. Matera, "A quality model for linked data exploration," *Web Engineering, Lecture Notes in Computer Science*, vol. 9671, pp. 397–404, Springer, Berlin, Germany, 2016.

[28] C. Bizer, P. Mendes, Z. Miklos, J. Calbimonte, A. Moraru, and G. Flouris, "D2.1 conceptual model and best practices for high-quality metadata publishing," Technical Report, Planet Data, 2012, https://www.planet-data.eu/results/deliverables.html.

[29] D. Pons, J. Hilera, and C. Pagés, "La estandarización para la calidad en los metadatos de recursos educativos virtuales," in *Proceedings of IV Congresso Internacional sobre Qualidade e Acessibilidade da Formação Virtual*, Leiria, Portugal, July 2013, http://www.esvial.org/wp-content/files/estandarizacionmetadatosPonsHileraPages.pdf.

[30] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, "Quality assessment for linked data: a survey, a systematic literature review and conceptual framework," 2012, http://www.semantic-web-journal.net/system/files/swj773.pdf.

[31] C. Castro, *Montoya. Configuración de software basada en metamodelos y modelos*, Repositorio Universidad de los Andes, Bogotá, Colombia, 1992, http://repositorio.uniandes.edu.co/xmlui/handle/1992/3991.

[32] V. García Díaz, E. Núñez Valdez, J. Espada, C. Pelayo García, J. Cueva Lovelle, and C. Montenegro Marín, "Introducción breve a la ingeniería dirigida por modelos," *Revista Tecnura*, vol. 18, no. 40, 2014.

[33] V. García Díaz, H. Fernández-Fernández, E. Palacios-González, C. Pelayo, O. Sanjuán-Martínez, and J. Cueva Lovelle, "TALISMAN MDE: mixing MDE principles," *Journal of Systems and Software*, vol. 83, no. 7, pp. 1179–1191, 2010.

[34] V. García Díaz, *MDCI: Model Driven Continuous Integration*, Departamento de Informática, Universidad de Oviedo, Oviedo, Spain, 2011, http://www.tdx.cat/handle/10803/80298.

[35] C. Montenegro Marín, P. Gaona García, J. Cueva Lovelle, and O. Sanjuan Martínez, "Aplicación de ingeniería dirigida por modelos (mda), para la construcción de una herramienta de modelado de dominio específico (dsm) y la creación de módulos en sistemas de gestión de aprendizaje (lms) independientes de la plataforma," *Revista Dyna*, vol. 78, no. 169, 2011, http://www.scielo.org.co/pdf/dyna/v78n169/a05v78n169.pdf.

[36] C. Montenegro, J. Cueva, O. Sanjuán, and P. Gaona, "Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de

implementación KiwiDSM mediante ingeniería dirigida por modelos," *Revista Ingeniería*, vol. 15, no. 2, pp. 67–81, 2010.

[37] Evidian, *Los 7 métodos de autenticación más utilizados*, Evidian, New York, NY, USA, 2015, https://www.evidian.com/pdf/wp-strongauth-es.pdf.

[38] F. Sotelo Gómez and M. Solarte, "Incorporación de recursos web como servicios de e-learning al sistema de gestión de aprendizaje. LRN: una revisión," *Tecnura*, vol. 18, no. 39, pp. 165–180, 2014.

[39] WizHosting, *Soluciones Web enlatadas*, WizHosting InternetServices, London, UK, 2015, http://www.wizhosting.com/e-learning.

[40] UDistrital, *Interfaces de Logueo Sistema Académico y LMS*, UDistrital, Bogotá, Colombia, https://funcionarios.portaloas.udistrital.edu.co/urano/.

[41] M. Rojas, J. Montilva, and M. Hurtado, "Diseño de repositorios de objetos de aprendizaje como estrategia de reutilización e integración de contenidos en modelos de educación virtual," in *Proceedings of 11th LACCEI Latin American and Caribbean Conference for Engineering and Technology*, Cancun, Mexico, August 2013, http://www.laccei.org/LACCEI2013-Cancun/RefereedPapers/RP240.pdf.

[42] Tecnoinver, *Qué es Single Sign-On o Autenticación Única*, Tecnoinver: Cloud, Datacenter y Hosting, Santiago, Chile, 2015, https://www.tecnoinver.cl/que-es-single-sign-on-o-autenticacion-unica/.

[43] C. Mainka, V. Mladenov, J. Schwenk, and T. Wich, "SoK: single sign-on security—an evaluation of openID connect," in *Procedings of 2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 251–266, Paris, France, April 2017.

[44] 9Series, *How Single Sign On Authentication Work?*, 9Series HandCrafted Technology Solutions, Ahmedabad, Gujarat, 2017, https://www.9spl.com/blog/how-single-sign-on-authentication-work/.

[45] C. Scott, D. Wynne, and C. Boonthum-Denecke, "Examining the privacy of login credentials using web-based single sign-on–are we giving up security and privacy for convenience?," in *Proceedings of 2016 Cybersecurity Symposium (CYBERSEC)*, pp. 74–79, Coeur d'Alene, Idaho, USA, April 2016, https://www.computer.org/csdl/proceedings/cybersecsym/2016/5771/00/07942428.pdf.

[46] B. Zwattendorfer, A. Tauber, and T. Zefferer, "A privacy-preserving eID based Single Sign-On solution," in *Proceedings of 2011 5th International Conference on Network and System Security*, pp. 295–299, Milan, Italy, September 2011.

[47] Y. Chen, B. Xia, B. Wu, and L. Shi, "Design of web service single sign-on based on ticket and assertion," in *Proceedings of 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce AIMSEC 2011*, pp. 297–300, Zhengzhou, China, August 2011.

[48] M. Beltrán, M. Calvo, and S. González, "Federated system-to-service authentication and authorization combining PUFs and tokens," in *Proceedings of 2017 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pp. 1–8, Madrid, Spain, July 2017.

[49] UC, *Configuración correo IMAP*, Universidad de Cantabria, Santander, Spain, https://sdei.unican.es/paginas/servicios/correo/manual_imap.aspx.

[50] J. Szczegielniak, *Introducing Single Sign-on to an Existing ASP.NET MVC Application*, RedGate Hub, Cambridge, UK, 2015, https://www.red-gate.com/simple-talk/dotnet/asp-net/introducing-single-sign-on-to-an-existing-asp-net-mvc-application/.

[51] C. Steel, R. Lai, and R. Nagappan, *Core Security Patterns: Securing the Identity--Design Strategies and Best Practices*, InformIT, Pearson, Carmel, Indiana, 2009, http://www.informit.com/articles/article.aspx?p=1398626.

[52] D. Kaplanis and TalentLMS, *Integrating Single Sign-On with your Cloud LMS*, TalentLMS Features & Updates, London, UK, 2014, https://www.talentlms.com/blog/integrating-single-sign-on-with-cloud-lms/.

[53] D. Parr, *LMS SSO with ONELOGIN*, Paradiso Solutions, Maharashtra, India, 2017, https://www.paradisosolutions.com/blog/lms-sso/.

[54] MiniOrange, *Single Sign On (SSO)*, MiniOrange, Maharashtra, India, 2018, https://www.miniorange.com/canvas-single-sign-on-(sso).

[55] D. Hardt, *The OAuth 2.0 Authorization Framework*, RFC 6749. Internet Engineering Task Force (IETF), Fremont, CA, USA, 2012, https://tools.ietf.org/html/rfc6749.

[56] Pradas, *BPMN OAuth 2 0 Authorization Code Grant, GenMyModel*, Pradas, Milan, Italy, 2018, https://repository.genmymodel.com/pradas/BPMN-OAuth.2.0.Authorization.Code.Grant.

[57] G. Salcedo, *Dashboard Proyecto de Investigación SSO*, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, 2018, https://autenticacion.udistrital.edu.co/dashboard.

[58] J. Martin, *Implantación de un SSO (Single Sign On), Master interuniversitario en Seguridad de las tecnologías de la información y de las Comunicaciones (MISTIC)*, Universidad Oberta de Cataluña, Barcelona, Spain, 2008, http://openaccess.uoc.edu/webapps/o2/bitstream/10609/28021/6/nacho_martinTFM0114memoria.pdf.

[59] P. Sheriff, "Single Sign-On Enterprise Security for Web Applications," *Microsoft Developer Network*, PDSA, Inc., Bristol, UK, 2004, https://msdn.microsoft.com/en-us/library/ms972971.aspx#singlesignon_topic10.

Hindawi
Scientific Programming
Volume 2019, Article ID 5901087, 14 pages
https://doi.org/10.1155/2019/5901087

Research Article

Consensus Clustering-Based Undersampling Approach to Imbalanced Learning

Aytuğ Onan

İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Computer Engineering, 35620 İzmir, Turkey

Correspondence should be addressed to Aytuğ Onan; aytugonan@gmail.com

Received 11 November 2018; Revised 15 January 2019; Accepted 10 February 2019; Published 3 March 2019

Guest Editor: Vicente García-Díaz

Class imbalance is an important problem, encountered in machine learning applications, where one class (named as, the minority class) has extremely small number of instances and the other class (referred as, the majority class) has immense quantity of instances. Imbalanced datasets can be of great importance in several real-world applications, including medical diagnosis, malware detection, anomaly identification, bankruptcy prediction, and spam filtering. In this paper, we present a consensus clustering based-undersampling approach to imbalanced learning. In this scheme, the number of instances in the majority class was undersampled by utilizing a consensus clustering-based scheme. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification benchmarks have been utilized. In the consensus clustering schemes, five clustering algorithms (namely, *k*-means, *k*-modes, *k*-means++, self-organizing maps, and DIANA algorithm) and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and *k*-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. The empirical results indicate that the proposed heterogeneous consensus clustering-based undersampling scheme yields better predictive performance.

1. Introduction

Class imbalance is an important research problem in machine learning, where the proportion of instances belonging to one class (referred as, the minority class) is extremely small, whereas the proportion of instances of the other class or classes (referred as, the majority class) is extremely high. Imbalanced datasets pose several challenges to the conventional supervised learning methods. Conventional supervised learning methods (such as support vector machines and decision trees) can build viable classification models for balanced datasets. Since imbalanced datasets suffer from outnumbering the instances of majority class and underrepresenting the instances of minority class, skewed distributions may lead to degradation of predictive performance [1, 2]. Supervised learning process is based on the use of global evaluation measures (such as classification accuracy). Hence, learning from imbalanced datasets can have bias towards the majority class, and classification models may tend to misclassify the instances of minority class [3]. Supervised learning algorithms may regard the instances of minority class as noise or outlier, and noisy data and outlier may be regarded as the instances of minority class [4]. In addition, classification models for datasets with skewed sample distributions may be challenging to learn due to the overlapping nature of the instances of minority class with the instances of other classes [5].

Imbalanced datasets can be encountered in several real-world problems and applications, including software fault identification [6], medical diagnosis [7], malware detection [8], anomaly identification [9], bankruptcy prediction [10], and spam filtering [11]. For data mining problems mentioned in advance, the number of instances for minority class is scarce. However, the identification of the instances of minority class may be more critical. For instance, the misclassification of cancerous (malignant) tumors as noncancerous (benign) in medical diagnosis can

have severe effects. Similarly, the number of instances for fraudulent transactions can be scarce. However, it is critical to build prediction models that can identify fraudulent transactions in finance. Hence, handling imbalanced datasets properly is an important research problem in machine learning.

To deal efficiently with the datasets with imbalanced distribution and to build robust and efficient classification schemes, data preprocessing methods have been utilized in conjunction with machine learning algorithms. The methods utilized to tackle with class imbalance problem can be mainly divided into four categories as algorithm level approaches, data-level approaches, cost-sensitive approaches, and ensemble learning-based approaches [12]. Algorithm level approaches seek to adapt supervised learning algorithms to bias learning towards the instances of minority class [13]. Data-level approaches seek to rebalance the instances of the imbalanced dataset so that the effects of skewed distributions can be eliminated in the learning process [14]. In order to do so, data-level approaches utilize resampling on the training datasets. Cost-sensitive approaches aim at minimizing total cost of errors for minority and majority classes by defining misclassification costs [15]. In addition, ensemble learning-based approaches have been also utilized for class imbalance. Ensemble classifiers aim at enhancing the predictive performance of a single learning algorithm by combining the predictions of several learning algorithms. In ensemble approaches to imbalanced learning, several strategies (such as bagging and undersampling, undersampling and cost-sensitive learning, boosting and resampling) have been combined [12]. In data-level approaches, data preprocessing and learning process of supervised learning algorithm are handled independently. In addition, compared to the cost-sensitive approaches, which involve to set cost matrix for imbalanced datasets, data-level preprocessing (resampling) is a viable tool to apply for researchers who are not expert in the field [1]. Hence, regarding different approaches to imbalanced learning, data-level approaches, which are based on resampling the imbalanced datasets, are frequently employed. The two main directions on data-level approaches are undersampling and oversampling. In order to obtain a dataset with balanced class distribution, the original imbalanced dataset can be resampled by oversampling the minority class or undersampling the majority class [16, 17]. In addition, there are several hybrid approaches, which combine undersampling and oversampling methods, such as SMOTEBoost, OverBagging, and UnderBagging [18–20]. Compared to the oversampling, undersampling yields better predictive performance [21]. However, undersampling may result in elimination of some useful representative instances of majority class [22]. Hence, the identification of useful representative instances in undersampling is of great performance to the predictive performance of supervised learning algorithms on imbalanced learning. In response, clustering methods can be utilized to identify useful representative instances of majority class in undersampling for imbalanced learning [23–25].

In this paper, we present a consensus clustering-based undersampling approach to imbalanced learning. In this scheme, the number of instances in the majority class was undersampled by utilizing a consensus clustering-based scheme. There are a large number of clustering algorithms in the literature. However, there is no single clustering algorithm that can yield the best clustering results under all scenarios, as the no free lunch theorem claims [26]. In this regard, the presented scheme aims at combining the decisions of different clustering algorithms, to overcome the limitations of individual clustering algorithms to achieve more robust/efficient clustering results. In this way, the presented scheme aims at identifying better representative instances of majority class in undersampling for imbalanced learning. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification (with imbalance ratios ranged between 1.8 and 163.19) were utilized. In the empirical analysis, the predictive performances of two clustering-based framework (namely, homogeneous and heterogeneous consensus clustering schemes) were compared with three data-level methods (namely, SMOTEBoost algorithm [16], RUSBoost [27], and underBagging algorithm [28, 29]). In the consensus clustering schemes, five clustering algorithms (namely, k-means, k-modes [30], k-means++ [31], self-organizing maps [32], and DIANA algorithm [33] and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. The empirical results indicate that the proposed heterogeneous consensus clustering-based undersampling scheme yields better predictive performance. To the best of our knowledge, the presented scheme is the first to use the paradigm of consensus clustering for imbalanced learning. The remainder of this paper is organized as follows. Section 2 briefly reviews the state of the art in imbalanced learning. Section 3 presents the proposed consensus clustering based-undersampling schemes. Section 4 presents the empirical analysis results, and Section 5 presents the concluding remarks.

2. Related Works

Imbalanced learning has attracted great research interest. As mentioned in advance, the methods to deal with imbalanced datasets can be broadly categorized as data-level methods, algorithm level methods, cost-sensitive methods, and ensemble learning-based methods. Compared to the other approaches, data-level approaches have greater potential use on imbalanced learning since they seek to improve the distribution of datasets, rather than relying on supervised learning-based enhancements [34]. This section briefly reviews the related work on imbalanced learning with emphasis on data-level approaches. Data-level approaches (sampling methods) can be mainly divided into two categories as undersampling and oversampling. Oversampling

and undersampling approaches can be employed effectively for class imbalance.

Oversampling approaches aim at obtaining a balanced dataset by generating synthetic instances for the minority class. In contrast, undersampling approaches aim at obtaining a balanced dataset by removing the instances of the majority class from the training set. For instance, Anand et al. [35] introduced a distance-based undersampling approach for class imbalance. Supervised learning methods can easily construct learning models for instances that are far from the decision boundaries. In response, the presented scheme aims at eliminating the instances of majority class that are far from decision boundaries, while preserving the instances near to the decision boundaries in the training set. In this way, the balanced training set was constructed and the balanced dataset was utilized in conjunction with the weighted support vector machines. Similarly, Li et al. [36] utilized vector quantization algorithm to decrease the instances of majority class. The presented scheme employed support vector machines for imbalanced learning. In another study, Kumar et al. [37] empirically examined the effect of undersampling on the performance of clustering algorithms. In another study, Sun et al. [22] presented an ensemble classification scheme based on undersampling for imbalanced learning. In the presented scheme, the instances of majority class were first divided into several partitions with similar number of instances with the minority class. In this way, balanced datasets were generated. The balanced datasets were trained on binary classifiers to build classification models. Finally, the predictions of binary classifiers were combined by an ensemble scheme to identify the final outcome. In another study, D'Addabbo and Maglietta [38] presented a selective sampling-based approach for imbalanced learning. Based on the observation that the instances near to decision boundaries are relevant/critical, the instances of majority class near to decision boundaries are preserved. In another study, Ha and Lee [39] presented an evolutionary undersampling scheme for class imbalance. In this scheme, genetic algorithm was utilized to select the informative instances of majority class by minimizing the loss between the distributions between original and balanced datasets. In another study, Lin et al. [24] introduced two clustering-based undersampling schemes for imbalanced learning. In this scheme, the number of clusters was determined based on the number of instances of minority class, and k-means algorithm was employed to undersample the instances of majority class. More recently, Shobana and Battula [40] presented an undersampling scheme based on diversified distribution and clustering for imbalanced learning. In this scheme, k-means algorithm was employed to identify and remove rare instances and outliers.

In a recent study, Guo and Wei [41] presented a hybrid scheme based on clustering and logistic regression for imbalanced learning. In the presented scheme, clustering was utilized to partition instances of the majority class into clusters. Similarly, Douzas et al. [42] integrated k-means clustering algorithm and synthetic minority oversampling technique to eliminate noisy data and to effectively obtain a balanced dataset within classes. Recently, Han et al. [43] presented a distribution-based approach for imbalanced learning. In the presented scheme, the instances of minority class were divided into groups as noisy instances, unstable instances, boundary instances, and stable instances based on the location information for the instances. The presented scheme has been utilized to improve the predictive performance on medical diagnosis. In another study, Tsai et al. [44] introduced an undersampling approach for imbalanced learning, which integrates clustering analysis and instance selection.

As mentioned in advance, undersampling is a simple resampling strategy to deal with class imbalance problem. However, undersampling may remove potentially useful/informative instances of the majority class, which may lead to the degradation of the predictive performance of classification schemes. In this paper, a consensus clustering-based framework is presented to identify the informative instances of majority class through the use of a cluster ensemble method.

3. Proposed Consensus Clustering-Based Undersampling Framework

Undersampling and oversampling methods can be successfully employed for class imbalance. In order to obtain a robust classification scheme with high predictive performance, undersampling methods should retain useful and informative representative instances of the majority class in the training set. Clustering (cluster analysis) is an unsupervised technique which assigns similar instances (objects) into the same cluster in terms of their proximity or similarity. Hence, clustering algorithms can be employed to identify useful instances of majority class in undersampling. With the use of clustering on undersampling, the majority class yields a distribution of instances into clusters such that similar instances are grouped together within the same cluster. One of the main problems encountered in applying clustering algorithms is the selection of an appropriate algorithm for a given problem. Each clustering algorithm has strong and weak characteristics, and the results obtained by clustering algorithms are greatly influenced based on the characteristics of dataset, parameters of algorithm, etc. The clustering algorithms suffer from instability, and the same clustering algorithm can yield a particularly different partition for different parameter settings. One possible solution to this problem is to use multiple clustering algorithms on the same dataset and to combine the outputs of individual clustering algorithms. The process is referred as consensus clustering (or cluster ensembles). Consensus clustering aims at combining the clustering results of different clustering algorithms so that a final clustering with better clustering quality can be obtained [45]. In this paper, two ensemble generation schemes are presented to undersample the instances of majority class based on consensus clustering, namely, homogeneous and heterogeneous ensemble schemes are introduced.

3.1. Consensus Function. Consensus clustering involves a staged procedure: in Stage 1, cluster ensemble is generated, and in Stage 2, consensus function is utilized to obtain the final partition from the individual clustering algorithms. There are direct approaches (such as simple voting, incremental voting, and label correspondence search), feature -based approaches (such as iterative voting consensus, mixture model, clustering aggregation, and quadratic mutual information), pairwise similarity-based approaches (such as agglomerative hierarchical models), and graph-based approaches (such as cluster-based similarity partitioning algorithm and shared nearest neighbors-based combiner) [45]. Motivated by the success of clustering algorithms on imbalanced learning [24] and the enhanced clustering quality obtained by consensus clustering schemes [46], we seek to find an efficient consensus clustering-based scheme for imbalanced learning. In this regard, we have conducted an experimental analysis with several different consensus functions. Since the highest predictive performance is obtained by direct approaches, of the wide range of consensus functions available, three consensus functions were chosen for the study.

3.1.1. Simple Voting Function (SV). Let π_r denote the reference partition and let π_g denote to be relabelled partitions, a contingency matrix $\Omega \in R^{K\times K}$ is obtained, in which K corresponds to the number of clusters. The contingency matrix entries $(\Omega(l, l'))$ are filled by co-occurrence statistics computed based on the following equation [45,43]:

$$\Omega(l,l') = \sum_{\forall x_{i\in X}} w(x_i), \tag{1}$$

where $w(x_i) = 1$ if $(C^r(x_i) = l) \wedge (C^g(x_i) = l')$ and $w(x_i) = 0$ otherwise. Based on the label correspondence obtained based on equation (1), the aim of the simple voting consensus is to maximize the objective function, given by

$$\sum_{l=1}^{K}\sum_{l'=1}^{K} \Omega(l,l')\Theta(l,l'), \tag{2}$$

where $\Theta(l,l') \in R^{K\times K}$ is a label correspondence matrix amongst the labels of partitions π_r and π_g. First, the reference partition (π_r) is randomly selected among the partitions of the cluster ensemble. Then, the remaining partitions are relabelled based on the reference partition by following the procedure outlined above. Finally, a majority voting scheme is employed to identify the consensus label of each instance.

3.1.2. Incremental Voting Function (IV). In incremental voting scheme (IV), data partitions are repeatedly added to the cluster ensemble. Let $P_g \in R^{N\times K}$ denote gth partition $(\pi_g \in \Pi)$. $P_g(x_i, C_t^g)$ takes the value of 1 if a data point $x_i \in X$ belongs to cluster $C_t^g \in \pi_g$. Otherwise, it takes the value of 0. Let V_g denote the matrix of intermediate g partitions $(\pi_1, \ldots, \pi_g)$ and $V_g(x_i, L_j)$ denote the number of partitions in which label L_j is corresponds to data point x_i. The process of incremental voting-based consensus is initialized with the construction of contingency matrix $\Omega \in R^{K\times K}$. The contingency matrix entries $(\Omega(l,l'))$ are filled by the following equation [48]:

$$\Omega(l,l') = \sum_{\forall x_{i\in X}} w(x_i), \tag{3}$$

where $w(x_i) = 1$ if $(V_g(x_i, L_j) \geq 1) \wedge (P_g(x_i, l') = 1)$. Otherwise, it takes the value of 0. After obtaining the contingency matrix, the entries of matrix for the $(g+1)$th partition (denoted by V_{g+1}) are computed as given by

$$\mathrm{V}_{g+1}(x_i, l) = \mathrm{V}_g(x_i, l) + \mathrm{P}_{g+1}(x_i, l'). \tag{4}$$

Based on the incremental combinations of M data partitions, the consensus label of each data point $x_i \in X$ is determined based on following equation [45]:

$$C^*(x_i) = \mathrm{argmax}_l V_M(x_i, l). \tag{5}$$

3.1.3. Label Correspondence Search. In label correspondence search (LCS), the problem of correspondence is modelled as an optimization problem [49]. The aim of the method is to obtain a consensus partition such that overall agreement among the different partitions is maximized. Let $R_{\{c,s\}}$ denote the vector representation of cluster c of system s. The $k-$th element of $R_{\{c,s\}}$ represents the posterior probabilities of cluster c for the data points. The agreement between clusters $\{c, s\}$ and $\{c', s'\}$ can be defined as given by the following equation:

$$g\{c,s\},\{c',s'\} = R_{\{c,s\}}^T \cdot R_{\{c',s'\}}. \tag{6}$$

If a cluster c of system s is assigned to metacluster m, $\lambda_{\{c,s\}}^{\{m\}}$ takes the value of 1 and it takes the value of 0 otherwise. $r_{\{c,s\}}^{\{m\}}$ denotes the reward of assigning cluster c to metacluster m, and it can be defined as given by the following equation:

$$r_{\{c,s\}}^{\{m\}} = \frac{1}{|I(m)|} \sum_{\{c',s'\}\in I(m)} g\{c,s\}, \quad \{c',s'\} \in I(m) \leftrightarrow \lambda_{\{c,s\}}^{\{m\}} \neq 0. \tag{7}$$

Based on equations (6) and (7), the objective of label correspondence is to maximize the argument defined in the following equation [49]:

$$\lambda^* = \mathrm{argmax}_\lambda \sum_{m=1}^{M}\sum_{s=1}^{S}\sum_{c=1}^{C_s} \lambda_{\{c,s\}}^{\{m\}} r_{\{c,s\}}^{\{m\}}, \tag{8}$$

subject to

$$\sum_{m=1}^{M} \lambda_{\{c,s\}}^{\{m\}} = 1, \forall c, s. \tag{9}$$

3.2. Homogeneous Consensus Clustering-Based Undersampling Framework. Let D denote an imbalanced dataset with two classes, where there is one class (referred as, the minority class) containing the small number of instances and there is

another class (referred as, the majority class) containing extremely high quantity of instances. Let us denote the number of instances corresponding to majority and minority classes as n and m, respectively. Initially, k-fold cross-validation scheme is utilized for dividing the imbalanced dataset into subsets as training and test sets. Then, the number of instances in the majority class (n) is undersampled so that it contains equal number of instances to the minority class (m). In the undersampling, homogeneous consensus clustering scheme is utilized to undersample the majority class. Clustering algorithms require the number of clusters as the input parameter. We adopted the clustering framework presented in [24]. Hence, the number of instances in the minority class (m) is taken as the number of clusters (k). In homogeneous consensus clustering scheme, the same clustering algorithm is utilized as the base clustering algorithm, with different parameter settings. In this scheme, five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm) are utilized as the base clustering algorithms.

In this way, diversified partitions are obtained by the base clustering algorithms. The partitions obtained by the base clustering algorithms are combined by consensus function to obtain the final partition. For obtaining final partition with consensus function, three consensus functions (namely, simple voting function, incremental voting function, and label correspondence search algorithm) are utilized. The center of each cluster of the final partition is selected as the instance for the majority class. In this way, a balanced training set is obtained. The balanced training set is utilized to train supervised learning algorithms (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and ensemble learning methods (namely, AdaBoost, bagging, and random subspace algorithm). The general stages of this scheme is depicted in Figure 1. In Figure 2, the general steps of homogeneous consensus clustering-based undersampling scheme (CONS1) are outlined.

3.3. Heterogeneous Consensus Clustering-Based Undersampling Framework. In heterogeneous consensus clustering scheme (CONS2), diversity among the clustering algorithms is achieved with the use of different clustering algorithms as the base clustering algorithms. As stated in advance, each clustering algorithm has its own strengths and weaknesses and can yield promising results on different datasets. The partitions obtained by different clustering algorithms may complement each other and can yield higher clustering quality. The heterogeneous consensus clustering-based undersampling framework follows the same stages as outlined in Figure 1. The only difference is that the heterogeneous consensus clustering framework utilizes 5 different clustering algorithms, as the base clustering algorithms, whereas the homogeneous consensus clustering framework utilizes the same clustering algorithm with different parameter settings, as the base clustering algorithms. The general structure of heterogeneous consensus clustering-based undersampling scheme is summarized in Figure 3. In the heterogeneous consensus clustering-based undersampling scheme, k-fold cross-validation is employed for dividing the imbalanced dataset into training set and test set. Then, the number of instances in the majority class is undersampled with the use of heterogeneous consensus clustering scheme. In this scheme, different clustering algorithms are utilized as the base clustering algorithms. The presented scheme can be configured with different clustering algorithms, yet, we have combined the five base clustering algorithms (namely, K-means, K-modes, K-means++, self-organizing maps, and DIANA algorithm). The partitions obtained by different clustering algorithms are combined by the consensus function. The center of each cluster of the final partition is selected as the instance for the majority class. In this way, a balanced training set is obtained. The predictive performance of undersampling scheme is examined with the use of supervised learning methods and ensemble learning methods.

4. Experimental Analysis and Results

This section presents the empirical analysis of the proposed consensus clustering-based undersampling schemes.

4.1. Datasets. To examine the effectiveness of the proposed undersampling approaches, we have utilized 44 small-scale and 2 large-scale imbalanced classification benchmarks. The imbalanced classification benchmarks were utilized in Galar et al. [12]. The imbalance ratios of small-scale benchmarks range from 1.8 to 129, and the number of instances ranges from 130 to 5500. The imbalance ratios of large-scale benchmarks range from 111.46 to 163.19, and the number of instances ranges from 102294 to 145751. For obtaining test and training sets for the supervised learning methods, we utilized k-fold cross-validation scheme, where we were partitioned the 80% and 20% training and testing sets with 5-fold cross-validation scheme. The basic descriptive information regarding the imbalanced classification benchmarks is presented in Table 1.

4.2. Experimental Procedure. In the empirical analysis, the presented consensus clustering-based undersampling schemes have been compared by seven state-of-the-art methods. The utilized methods in the analysis include UnderBagging4 (UB4), UnderBagging24 (UB24), RusBoost1 (Rus1), SMOTEBagging4 (SBAG4), UnderBagging1 (UB1), clustering-based undersampling based on cluster centers (Centers), and clustering-based undersampling based on the nearest neighbors of cluster centers (Centers_NN) [12, 24]. In order to examine the predictive performance changes obtained by data balancing strategies, the results obtained by C4.5 algorithm without data balancing have also been presented as the baseline results. In the consensus clustering schemes, five clustering algorithms (namely, k-means, k-modes, k-means++,

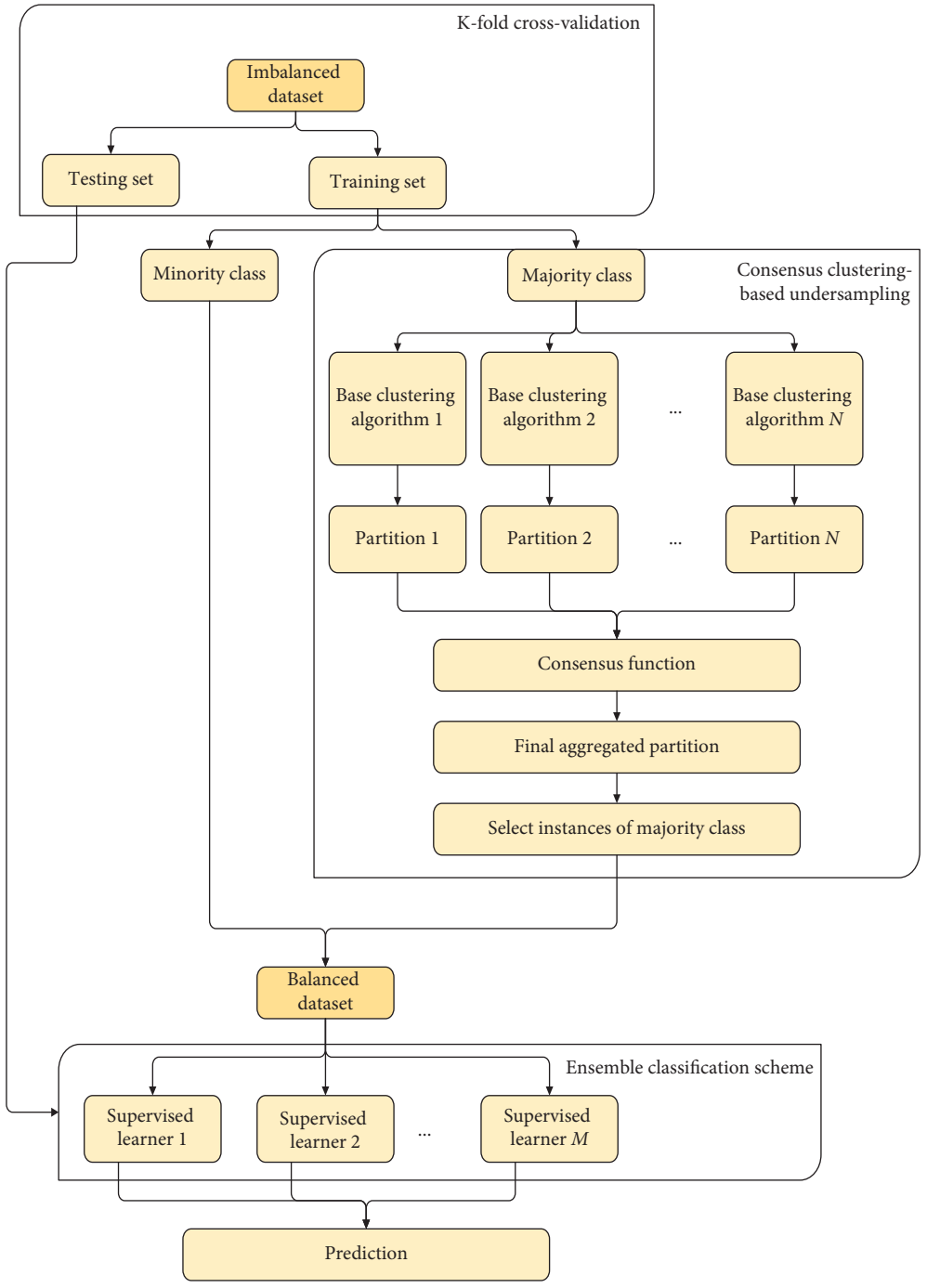


Figure 1: Homogeneous consensus clustering-based undersampling scheme (CONS1).

self-organizing maps, and DIANA algorithm) and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. In the empirical analysis, area under roc curve was utilized as the evaluation metric. For the supervised learning methods and state-of-the-art data preprocessing methods, the default parameters were employed. For the homogeneous consensus clustering-based undersampling scheme, i parameter (the number of base clustering algorithms) is taken as five.

4.3. Experimental Results and Discussions. In Table 2, average AUC values of the state-of-the-art methods and conventional clustering algorithms (namely, K-means, K-means++, K-modes, self-organizing maps, and DIANA algorithm) are presented. As it can be observed from the results presented in Table 2, the application of data balancing strategies enhance the predictive performance in terms of

Input: An imbalanced dataset D

Output: A classification model obtained from a balanced dataset D'

Undersampling Phase

1. Let n denote the number instances of the majority class in the training set and let m denote the number of instances of the minority class in the training set.
2. Let i denote user-defined parameter for the number of base clustering algorithms in homogeneous clustering scheme.
3. Utilize k-fold cross validation scheme to divide D into training and test sets.
4. Set the number of clusters equal to m.
5. Apply i times the same clustering algorithm (K-means, K-modes, K-means++, self-organizing maps or DIANA algorithm) on the instances of majority class of training set to undersample the majority class.
6. Obtain the partitions of base clustering algorithms on the majority class.
7. Obtain the final partition from the individual clustering algorithms by employing consensus function (simple voting function, incremental voting function or label correspondence search algorithm).
8. Compute the center of each cluster of the final partition.
9. Take cluster centers of the final partition as the instances of the majority class.
10. Combine the instances of majority class and the instances of minority class to obtain the balanced training set D'.

Classification Phase

1. Employ supervised learning algorithms (Naïve Bayes, logistic regression, support vector machines, random forests and k-nearest neighbor algorithm) and ensemble learners (AdaBoost, Bagging and Random Subspace method) on the balanced training set D'.
2. Use test set to evaluate the predictive performance of supervised learning methods in terms of area under roc curve and classification accuracy.

Figure 2: The general structure of the homogeneous consensus clustering-based undersampling scheme (CONS1).

Input: An imbalanced dataset D

Output: A classification model obtained from a balanced dataset D'

Undersampling Phase

1. Let n denote the number instances of the majority class in the training set and let m denote the number of instances of the minority class in the training set.
2. Utilize k-fold cross validation scheme to divide D into training and test sets.
3. Set the number of clusters equal to m.
4. Apply five different clustering algorithm (K-means, K-modes, K-means++, self-organizing maps and DIANA algorithm) on the instances of majority class of training set to undersample the majority class.
5. Obtain the partitions of base clustering algorithms on the majority class.
6. Obtain the final partition from the individual clustering algorithms by employing consensus function (simple voting function, incremental voting function or label correspondence search algorithm).
7. Compute the center of each cluster of the final partition.
8. Take cluster centers of the final partition as the instances of the majority class.
9. Combine the instances of majority class and the instances of minority class to obtain the balanced training set D'.

Classification Phase

1. Employ supervised learning algorithms (Naïve Bayes, logistic regression, support vector machines, random forests and k-nearest neighbor algorithm) and ensemble learners (AdaBoost, Bagging and Random Subspace method) on the balanced training set D'.
2. Use test set to evaluate the predictive performance of supervised learning methods in terms of area under roc curve and classification accuracy.

Figure 3: The general structure of the heterogeneous consensus clustering-based undersampling scheme (CONS2).

AUC values. The lowest average AUC values obtained by C4.5 algorithm without data balancing have been applied. The highest average AUC values are generally obtained by UnderBagging4 algorithm, and the second highest average AUC values are generally obtained by UnderBagging24 algorithm. In the empirical analysis, five base clustering

Table 1: Descriptive information for the datasets [12, 24].

Dataset	Number of data samples	Number of features	Imbalance ratio
Small-scale datasets			
Abalone9-18	731	8	16.68
Abalone19	4174	8	128.87
Ecoli-0_vs_1	220	7	1.86
Ecoli-0-1-3-7_vs_2-6	281	7	39.15
Ecoli1	336	7	3.36
Ecoli2	336	7	5.46
Ecoli3	336	7	8.19
Ecoli4	336	7	13.84
Glass0	214	9	3.19
Glass0123vs456	192	9	10.29
Glass016vs2	184	9	19.44
Glass016vs5	214	9	1.82
Glass1	214	9	10.39
Glass2	214	9	15.47
Glass4	214	9	22.81
Glass5	214	9	22.81
Glass6	214	9	6.38
Haberman	306	3	2.68
Iris0	150	4	2
New-thyroid1	215	5	5.14
New-thyroid2	215	5	4.92
Page-blocks0	5472	10	8.77
Page-blocks13vs2	472	10	15.85
Pima	768	8	1.9
Segment	2308	19	6.01
Shuttle0vs4	1829	9	13.87
Shuttle2vs4	129	9	20.5
Vehicle0	846	18	3.23
Vehicle1	846	18	2.52
Vehicle2	846	18	2.52
Vehicle3	846	18	2.52
Vowel0	988	13	10.1
Wisconsin	683	9	1.86
Yeast05679vs4	528	8	9.35
Yeast1	1484	8	2.46
Yeast1vs7	459	8	13.87
Yeast1289vs7	947	8	30.56
Yeast1458vs7	693	8	22.1
Yeast2vs4	514	8	9.08
Yeast2vs8	482	8	23.1
Yeast3	1484	8	8.11
Yeast4	1484	8	28.41
Yeast5	1484	8	32.78
Yeast6	1484	8	39.15
Large-scale datasets			
Breast cancer	102294	117	163.19
Protein homology prediction	145751	74	111.46

algorithms have been taken into consideration. Among the base clustering algorithms, the highest average AUC values are obtained by DIANA clustering algorithm.

The homogeneous consensus clustering scheme utilizes a single clustering algorithm (of the same type) as the base clustering method. In the empirical analysis, five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm) are considered as the base clustering methods. For aggregating the clustering results of individual clustering results, we considered three consensus functions (namely, simple voting function, incremental voting function, and label correspondence search algorithm). In this way, 15 different homogeneous consensus clustering-based schemes are evaluated for imbalanced learning. In Table 3, average AUC values obtained by homogeneous consensus clustering schemes are presented. Compared to the results presented in Table 2 for conventional data-level methods and conventional clustering-based schemes, homogeneous consensus clustering schemes yield better predictive performance in terms of AUC values. Among the compared homogeneous consensus clustering schemes, the highest predictive performance is obtained by utilizing self-organizing map algorithm as the base clustering algorithm. In this scheme, simple voting function is employed as the consensus function.

For the heterogeneous consensus clustering scheme, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm methods were utilized to identify individual partitions. Similar to the homogeneous scheme, we considered three consensus functions (namely, simple voting function, incremental voting function, or label correspondence search algorithm). In this way, 3 different heterogeneous consensus clustering-based schemes are taken into consideration. In Table 4, average AUC values obtained by heterogeneous consensus clustering schemes are presented. As it can be observed from the results listed in Table 4, heterogeneous consensus clustering schemes outperform homogeneous consensus clustering schemes, conventional data-level methods, and conventional clustering-based schemes. Regarding the average AUC values analyzed in the empirical analysis, the highest predictive performance is obtained by heterogeneous clustering scheme with label correspondence search-based consensus function. The second highest predictive performance is obtained by heterogeneous clustering scheme with simple voting-based consensus function.

In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. In order to summarize the main findings of the empirical analysis, boxplots for undersampling methods and supervised learning methods are presented in Figures 4 and 5, respectively.

As it can be observed from Figure 4, average AUC values obtained from the presented heterogeneous clustering scheme is higher compared to the conventional data-level methods ($p < 0.05$). In Figure 5, the predictive performance analysis of conventional supervised learning methods and their ensembles are taken into consideration. As it can be observed, ensemble learning methods yield higher predictive performance in terms of AUC values compared to the conventional supervised learning methods. The highest predictive performance for supervised learning methods is achieved by random subspace ensemble of random forest, and the second highest predictive performance is obtained

Table 2: Average AUC values of state-of-the-art methods with C4.5 classifier.

	C4.5	UB4	UB24	Rus1	SBAG4	UB1	Centers	Centers_NN	KM	KM++	KMOD	SOM	DIANA
Abalone19	0.500	0.721	0.680	0.631	0.572	0.695	0.639	0.648	0.743	0.744	0.744	0.745	0.745
Abalone9-18	0.598	0.719	0.710	0.693	0.745	0.710	0.699	0.704	0.769	0.769	0.769	0.769	0.770
Breast cancer	0.867	0.927	0.929	0.929	0.925	0.922	0.889	0.914	0.839	0.847	0.854	0.845	0.857
Ecoli-0_vs_1	0.983	0.980	0.980	0.969	0.983	0.969	0.983	0.983	0.920	0.910	0.950	0.880	0.920
Ecoli-0-1-3-7_vs_2-6	0.748	0.745	0.781	0.794	0.828	0.726	0.715	0.726	0.774	0.774	0.775	0.775	0.775
Ecoli1	0.859	0.900	0.902	0.883	0.900	0.898	0.895	0.923	0.810	0.820	0.820	0.830	0.0.840
Ecoli2	0.864	0.884	0.881	0.899	0.888	0.870	0.864	0.878	0.800	0.810	0.820	0.820	0.830
Ecoli3	0.728	0.908	0.894	0.856	0.885	0.882	0.847	0.900	0.800	0.810	0.820	0.820	0.830
Ecoli4	0.844	0.888	0.899	0.942	0.933	0.891	0.905	0.862	0.800	0.810	0.810	0.820	0.820
Glass0	0.817	0.814	0.824	0.813	0.839	0.818	0.772	0.744	0.780	0.780	0.780	0.780	0.780
Glass0123vs456	0.916	0.904	0.917	0.930	0.946	0.894	0.914	0.902	0.810	0.810	0.820	0.830	0.840
Glass016vs2	0.594	0.754	0.625	0.617	0.559	0.636	0.645	0.708	0.773	0.773	0.773	0.773	0.774
Glass016vs5	0.894	0.943	0.943	0.989	0.866	0.943	0.943	0.943	0.810	0.820	0.830	0.840	0.850
Glass1	0.740	0.737	0.752	0.763	0.728	0.748	0.713	0.647	0.734	0.737	0.739	0.739	0.739
Glass2	0.719	0.769	0.706	0.780	0.779	0.758	0.658	0.756	0.783	0.783	0.783	0.783	0.783
Glass4	0.754	0.846	0.871	0.915	0.874	0.853	0.651	0.803	0.800	0.800	0.800	0.800	0.810
Glass5	0.898	0.949	0.949	0.943	0.878	0.949	0.888	0.949	0.820	0.830	0.840	0.840	0.850
Glass6	0.813	0.904	0.926	0.918	0.931	0.885	0.858	0.847	0.800	0.800	0.810	0.810	0.820
Haberman	0.576	0.664	0.668	0.655	0.656	0.658	0.620	0.595	0.715	0.715	0.716	0.717	0.718
Iris0	0.990	0.990	0.980	0.990	0.980	0.990	0.990	0.990	0.940	0.950	0.960	0.890	0.940
New-thyroid1	0.914	0.964	0.969	0.958	0.975	0.955	0.938	0.947	0.820	0.830	0.830	0.840	0.850
New-thyroid2	0.937	0.958	0.938	0.938	0.961	0.947	0.938	0.924	0.810	0.820	0.820	0.830	0.840
Page-blocks0	0.922	0.958	0.959	0.948	0.953	0.952	0.934	0.958	0.820	0.850	0.850	0.850	0.860
Page-blocks13vs2	0.998	0.978	0.975	0.987	0.988	0.975	0.911	0.992	0.980	0.980	0.980	0.930	0.950
Pima	0.701	0.760	0.753	0.726	0.751	0.758	0.753	0.727	0.776	0.776	0.776	0.776	0.777
Segmemt0	0.983	0.988	0.986	0.993	0.994	0.985	0.981	0.980	0.890	0.890	0.910	0.870	0.900
Shuttle0vs4	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.990	0.980	0.950
Shuttle2vs4	0.950	1.000	1.000	1.000	1.000	0.988	1.000	0.988	0.920	0.940	0.950	0.880	0.930
Vehicle0	0.930	0.952	0.954	0.958	0.965	0.945	0.942	0.948	0.820	0.830	0.840	0.840	0.850
Vehicle1	0.672	0.787	0.761	0.747	0.769	0.765	0.722	0.703	0.767	0.768	0.768	0.768	0.768
Vehicle2	0.956	0.964	0.964	0.970	0.966	0.957	0.942	0.956	0.820	0.840	0.840	0.850	0.860
Vehicle3	0.664	0.802	0.784	0.765	0.763	0.764	0.757	0.731	0.778	0.778	0.778	0.778	0.778
Vowel0	0.971	0.947	0.947	0.943	0.988	0.944	0.941	0.910	0.810	0.820	0.820	0.830	0.840
Wisconsin	0.945	0.960	0.971	0.964	0.960	0.957	0.945	0.945	0.820	0.820	0.830	0.840	0.850
Yeast05679vs4	0.680	0.794	0.814	0.803	0.818	0.782	0.756	0.769	0.826	0.826	0.826	0.826	0.826
Yeast1	0.664	0.722	0.721	0.719	0.734	0.716	0.741	0.738	0.779	0.779	0.779	0.779	0.779
Yeast1289vs7	0.616	0.734	0.689	0.721	0.658	0.675	0.632	0.700	0.754	0.755	0.755	0.755	0.755
Yeast1458vs7	0.500	0.606	0.617	0.567	0.623	0.563	0.559	0.603	0.727	0.727	0.728	0.728	0.730
Yeast1vs7	0.628	0.786	0.773	0.715	0.697	0.747	0.660	0.704	0.770	0.770	0.770	0.771	0.771
Yeast2vs4	0.831	0.936	0.929	0.933	0.897	0.940	0.914	0.882	0.800	0.810	0.820	0.820	0.830
Yeast2vs8	0.525	0.783	0.747	0.789	0.784	0.761	0.629	0.778	0.826	0.826	0.827	0.827	0.827
Yeast3	0.860	0.934	0.944	0.925	0.944	0.940	0.901	0.926	0.810	0.820	0.830	0.840	0.840
Yeast4	0.614	0.855	0.854	0.812	0.773	0.860	0.722	0.857	0.800	0.810	0.810	0.810	0.820
Yeast5	0.883	0.952	0.956	0.959	0.962	0.964	0.954	0.960	0.840	0.870	0.910	0.860	0.870
Yeast6	0.712	0.869	0.878	0.823	0.836	0.864	0.691	0.818	0.800	0.800	0.810	0.810	0.820
Protein homology prediction	0.922	0.956	0.961	0.956	0.945	0.952	0.928	0.947	0.820	0.828	0.835	0.840	0.850
Twitter-sentiment	0.962	0.979	0.978	0.980	0.981	0.976	0.966	0.979	0.903	0.914	0.927	0.888	0.909
Average	0.801	*0.870*	0.865	0.862	0.859	0.858	0.826	0.847	0.815	0.821	0.826	0.820	0.828

by random subspace ensemble of support vector machines ($p < 0.05$). Regarding the predictive performance of conventional clustering algorithms, naïve Bayes demonstrated the lowest predictive performance, whereas random forest algorithm demonstrated the best (the highest) predictive performance ($p < 0.05$).

In Figure 6, the confidence intervals for the mean values of average AUC values obtained by the compared algorithms for a confidence level of 95% are presented. Based on the statistical significances between the compared results, Figure 6 is divided into two regions denoted by red dashed line. As it can be observed from Figure 6, the predictive performance differences obtained by the proposed consensus clustering-based schemes are statistically significant.

5. Conclusion

Class imbalance is an important problem of machine learning. Imbalanced datasets can be seen in a wide variety of

Table 3: Average AUC values of homogeneous clustering schemes with C4.5 classifier.

Consensus function	Simple voting	Simple voting	Simple voting	Simple voting	Simple voting	Incremental voting	Incremental voting	Incremental voting	Incremental voting	LCS	Incremental voting	LCS	LCS	LCS	LCS
Method	CONS1 (KM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (SOM)	CONS1 (DIANA)	CONS1 (KM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (DIANA)	CONS1 (KM)	CONS1 (SOM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (SOM)	CONS1 (DIANA)
Abalone19	0.746	0.746	0.746	0.766	0.747	0.747	0.747	0.748	0.766	0.766	0.766	0.766	0.766	0.746	0.766
Abalone9-18	0.770	0.770	0.770	0.794	0.770	0.792	0.792	0.793	0.793	0.793	0.793	0.793	0.793	0.770	0.811
Breast cancer	0.855	0.867	0.870	0.940	0.882	0.879	0.891	0.887	0.903	0.909	0.921	0.926	0.918	0.888	0.931
Ecoli-0_vs_1	0.870	0.880	0.920	0.970	0.900	0.910	0.930	0.930	0.950	0.950	0.960	0.960	0.960	0.940	0.980
Ecoli-0-1-3-7_vs_2-6	0.775	0.775	0.775	0.782	0.775	0.778	0.779	0.780	0.780	0.780	0.781	0.781	0.782	0.775	0.788
Ecoli1	0.850	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.910	0.920	0.930	0.930	0.870	0.950
Ecoli2	0.830	0.840	0.850	0.930	0.860	0.860	0.870	0.860	0.870	0.890	0.900	0.910	0.910	0.860	0.910
Ecoli3	0.840	0.850	0.850	0.940	0.870	0.870	0.870	0.870	0.890	0.900	0.900	0.910	0.920	0.860	0.940
Ecoli4	0.830	0.840	0.840	0.930	0.860	0.860	0.870	0.860	0.870	0.890	0.890	0.910	0.850	0.850	0.850
Glass0	0.780	0.781	0.781	0.823	0.784	0.822	0.822	0.822	0.822	0.823	0.823	0.823	0.823	0.781	0.824
Glass0123vs456	0.840	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.900	0.900	0.910	0.930	0.860	0.940
Glass016vs2	0.774	0.774	0.774	0.789	0.774	0.786	0.787	0.787	0.788	0.788	0.789	0.789	0.789	0.774	0.790
Glass016vs5	0.850	0.860	0.860	0.960	0.880	0.880	0.890	0.890	0.910	0.920	0.940	0.940	0.950	0.890	0.960
Glass1	0.740	0.740	0.740	0.765	0.741	0.742	0.743	0.743	0.764	0.765	0.765	0.765	0.765	0.741	0.765
Glass2	0.784	0.784	0.784	0.842	0.784	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.784	0.842
Glass4	0.800	0.810	0.800	0.840	0.840	0.800	0.840	0.810	0.830	0.800	0.820	0.850	0.800	0.840	0.810
Glass5	0.850	0.870	0.880	0.960	0.890	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.900	0.970
Glass6	0.820	0.820	0.840	0.900	0.860	0.860	0.860	0.820	0.870	0.820	0.880	0.890	0.810	0.850	0.820
Haberman	0.718	0.722	0.722	0.759	0.725	0.725	0.725	0.727	0.757	0.757	0.758	0.758	0.759	0.724	0.759
Iris0	0.900	0.960	0.930	0.980	0.930	0.910	0.950	0.940	0.950	0.950	0.960	0.970	0.970	0.960	0.990
New-thyroid1	0.850	0.860	0.870	0.960	0.890	0.880	0.900	0.910	0.910	0.930	0.940	0.950	0.950	0.890	0.970
New-thyroid2	0.850	0.850	0.850	0.950	0.880	0.870	0.880	0.880	0.900	0.920	0.930	0.930	0.930	0.870	0.960
Page-blocks0	0.860	0.880	0.890	0.970	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.960	0.920	0.970
Page-blocks13vs2	0.960	0.960	0.950	0.990	0.970	0.940	0.950	0.940	0.950	0.970	0.990	0.980	0.970	0.960	0.990
Pima	0.777	0.777	0.777	0.792	0.777	0.790	0.790	0.791	0.791	0.791	0.792	0.792	0.792	0.777	0.792
Segmemt0	0.870	0.880	0.890	0.970	0.900	0.900	0.920	0.930	0.940	0.940	0.950	0.960	0.960	0.920	0.980
Shuttle0vs4	0.980	0.990	0.980	1.000	0.980	0.990	0.970	0.940	0.970	1.000	1.000	0.980	0.990	0.990	1.000
Shuttle2vs4	0.890	0.950	0.920	0.970	0.910	0.910	0.940	0.930	0.950	0.950	0.960	0.960	0.960	0.950	0.980
Vehicle0	0.850	0.870	0.880	0.960	0.890	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.890	0.970
Vehicle1	0.768	0.768	0.768	0.766	0.769	0.760	0.761	0.762	0.762	0.763	0.763	0.765	0.765	0.768	0.766
Vehicle2	0.860	0.880	0.880	0.970	0.890	0.900	0.900	0.910	0.920	0.940	0.950	0.950	0.950	0.900	0.970
Vehicle3	0.779	0.779	0.779	0.801	0.779	0.799	0.799	0.800	0.800	0.800	0.801	0.801	0.801	0.779	0.803
Vowel0	0.840	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.910	0.910	0.920	0.930	0.870	0.940
Wisconsin	0.850	0.860	0.870	0.960	0.880	0.880	0.890	0.890	0.910	0.930	0.940	0.950	0.950	0.890	0.960
Yeast05679vs4	0.826	0.826	0.826	0.842	0.826	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.826	0.842
Yeast1	0.779	0.780	0.780	0.811	0.780	0.809	0.810	0.810	0.810	0.810	0.811	0.811	0.811	0.780	0.813
Yeast1289vs7	0.756	0.756	0.756	0.767	0.756	0.757	0.757	0.757	0.767	0.767	0.767	0.767	0.767	0.756	0.770
Yeast1458vs7	0.730	0.731	0.731	0.762	0.732	0.732	0.733	0.734	0.760	0.762	0.762	0.762	0.762	0.732	0.762
Yeast1vs7	0.771	0.772	0.772	0.787	0.772	0.782	0.783	0.784	0.784	0.785	0.785	0.786	0.786	0.772	0.787
Yeast2vs4	0.840	0.840	0.850	0.940	0.870	0.870	0.870	0.870	0.890	0.900	0.900	0.910	0.910	0.860	0.920
Yeast2vs8	0.827	0.827	0.827	0.851	0.827	0.850	0.850	0.850	0.850	0.851	0.851	0.851	0.851	0.827	0.851
Yeast3	0.850	0.850	0.860	0.950	0.880	0.870	0.890	0.890	0.900	0.920	0.930	0.930	0.940	0.880	0.960
Yeast4	0.830	0.840	0.840	0.910	0.860	0.860	0.870	0.860	0.870	0.880	0.890	0.900	0.840	0.850	0.840
Yeast5	0.870	0.880	0.890	0.970	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.960	0.920	0.980

Table 3: Continued.

Consensus function	Simple voting	Simple voting	Simple voting	Simple voting	Simple voting	Incremental voting	Incremental voting	Incremental voting	Incremental voting	LCS	Incremental voting	LCS	LCS	LCS	LCS
Yeast6	0.810	0.820	0.830	0.850	0.850	0.800	0.840	0.810	0.850	0.810	0.870	0.870	0.810	0.850	0.810
Protein homology prediction	0.850	0.865	0.875	0.960	0.888	0.885	0.898	0.905	0.915	0.930	0.940	0.950	0.950	0.893	0.968
Twitter-sentiment	0.896	0.918	0.917	0.977	0.918	0.918	0.931	0.929	0.943	0.953	0.963	0.962	0.964	0.940	0.982
Average	0.826	0.835	0.837	0.893	0.845	0.848	0.856	0.854	0.867	0.871	0.879	0.883	0.878	0.849	0.888

Table 4: Average AUC values of heterogeneous clustering schemes with C4.5 classifier.

Consensus function	IV	SV	LCS
Method	CONS2	CONS2	CONS2
Abalone19	0.766	0.767	0.782
Abalone9-18	0.812	0.812	0.812
Breast cancer	0.945	0.946	0.954
Ecoli-0_vs_1	0.990	0.990	1.000
Ecoli-0-1-3-7_vs_2-6	0.789	0.789	0.797
Ecoli1	0.970	0.980	0.980
Ecoli2	0.920	0.920	0.940
Ecoli3	0.960	0.960	0.980
Ecoli4	0.900	0.880	0.890
Glass0	0.824	0.824	0.826
Glass0123vs456	0.960	0.960	0.980
Glass016vs2	0.790	0.791	0.791
Glass016vs5	0.970	0.980	0.990
Glass1	0.765	0.765	0.782
Glass2	0.842	0.842	0.842
Glass4	0.820	0.800	0.800
Glass5	0.970	0.980	1.000
Glass6	0.870	0.860	0.850
Haberman	0.760	0.762	0.772
Iris0	0.990	1.000	1.000
New-thyroid1	0.970	0.980	0.990
New-thyroid2	0.970	0.980	0.990
Page-blocks0	0.980	0.990	1.000
Page-blocks13vs2	0.990	1.000	1.000
Pima	0.793	0.793	0.793
Segmemt0	0.990	0.990	1.000
Shuttle0vs4	1.000	1.000	1.000
Shuttle2vs4	0.990	1.000	1.000
Vehicle0	0.970	0.980	0.990
Vehicle1	0.767	0.767	0.768
Vehicle2	0.980	0.990	1.000
Vehicle3	0.803	0.804	0.806
Vowel0	0.970	0.980	0.980
Wisconsin	0.970	0.980	0.990
Yeast05679vs4	0.843	0.843	0.843
Yeast1	0.813	0.813	0.815
Yeast1289vs7	0.770	0.770	0.782
Yeast1458vs7	0.762	0.763	0.781
Yeast1vs7	0.787	0.788	0.812
Yeast2vs4	0.950	0.940	0.940
Yeast2vs8	0.851	0.851	0.851
Yeast3	0.970	0.980	0.990
Yeast4	0.880	0.860	0.890
Yeast5	0.980	0.990	1.000
Yeast6	0.820	0.800	0.810
Protein homology prediction	0.970	0.980	0.993
Twitter-sentiment	0.988	0.994	1.000
Average	0.897	0.898	0.906

applications, including medical diagnosis, malware detection, anomaly identification, bankruptcy prediction, and spam filtering. In order to build efficient and robust classification schemes, data preprocessing methods can be utilized in conjunction with supervised learning methods. Undersampling- and oversampling-based methods can be successfully utilized for class imbalance. However, the identification of informative instances to be included in the training set is a critical issue for undersampling. In this

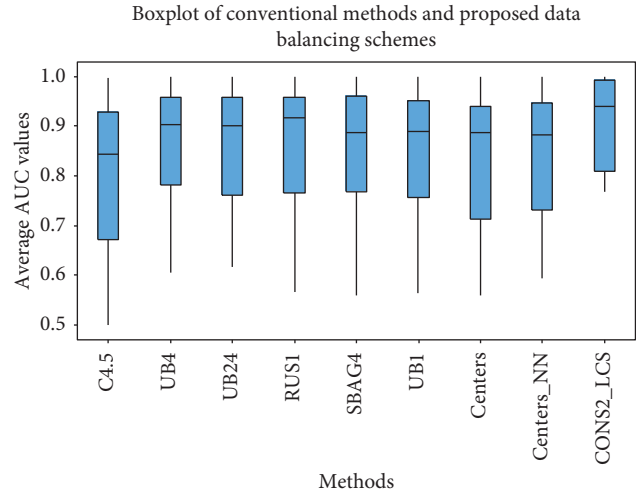


Figure 4: Boxplot distributions of AUC values for conventional data balancing methods and the proposed scheme.

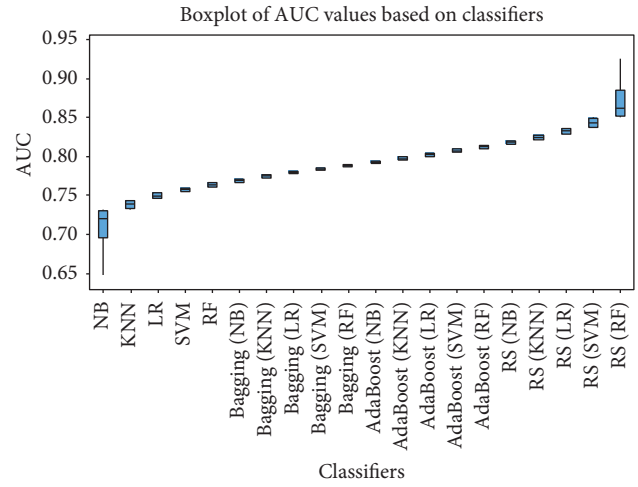


Figure 5: Boxplot distributions of AUC values for supervised learning methods and ensemble methods.

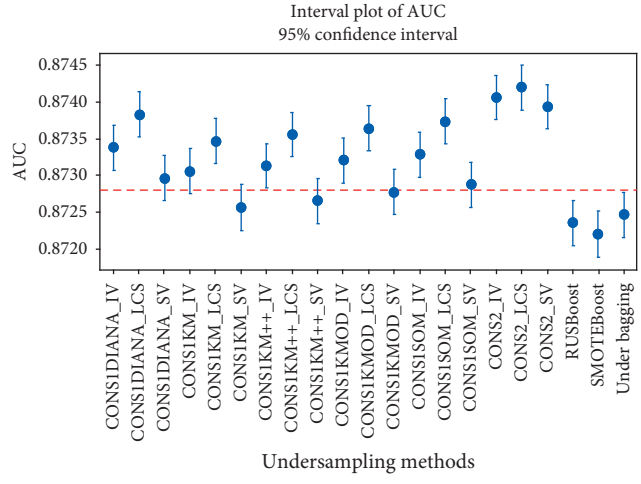


Figure 6: Interval plots for the compared algorithms.

regard, this paper empirically examines the predictive performance of two consensus clustering-based undersampling schemes for imbalanced learning. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification benchmarks (with imbalance ratios ranged between 1.8 and 163.19) were utilized. The experimental analysis indicates that clustering-based undersampling schemes can outperform conventional data-level preprocessing methods for class imbalance. In addition, consensus clustering, which aggregates the partitions of individual clustering algorithms, can further enhance the predictive performance of clustering-based undersampling schemes.

There are a number of issues that should be beneficial to extend in the future. The presented consensus clustering based undersampling scheme utilizes five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm). The clustering algorithms have been integrated with the use of three consensus functions, namely, simple voting-based consensus function, incremental voting function, and label correspondence search. Hence, the predictive performance of other conventional and swarm-based clustering algorithms (such as ant clustering, particle swarm-based clustering, firefly clustering) can be examined for imbalanced learning. In addition, recent proposals on the field indicate that imbalancing schemes which integrate instance selection and clustering may yield higher predictive performance. Hence, the performance of consensus clustering-based undersampling scheme should be taken into consideration in conjunction with conventional instance selection methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The study was performed as part of the employment of the author at Izmir Katip Celebi University.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[2] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[3] G. M. Weiss, "Mining with rarity," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[4] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, vol. 48, no. 5, pp. 1653–1672, 2015.

[5] M. Denil and T. Trappenberg, "Overlap versus imbalance," in *Proceedings of Canadian Conference on Artificial Intelligence*, pp. 220–231, Springer, Ottawa, Canada, May 2010.

[6] D. Rodriguez, I. Herraiz, R. Harrison, J. Dolado, and J. C. Riquelme, "Preliminary comparison of techniques for dealing with imbalance in software defect prediction," in *Proceedings of 18th International Conference on Evaluation and Assessment in Software Engineering*, p. 43, ACM, London, UK, May 2014.

[7] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of European Conference on Machine Learning ECML 2004*, pp. 39–50, Prague, Czech Republic, September 2004.

[8] N. Peiravian and X. Zhu, "Machine learning for android malware detection using permission and api calls," in *Proceedings of IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 300–305, IEEE, Herndon, VA, USA, November 2013.

[9] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Iterative Boolean combination of classifiers in the ROC space: an application to anomaly detection with HMMs," *Pattern Recognition*, vol. 43, no. 8, pp. 2732–2752, 2010.

[10] M.-J. Kim, D.-K. Kang, and H. B. Kim, "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1074–1082, 2015.

[11] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012.

[12] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[13] B. Liu, Y. Ma, and C. K. Wong, "Improving an association rule based classifier," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 504–509, Springer, Lyon, France, September 2000.

[14] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[15] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 225–252, 2008.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority oversampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] N. Japkowicz, "The class imbalance problem: significance and strategies," in *Proceedings of International Conference on Artificial Intelligence*, Las Vegas, NV, USA, June 2000.

[18] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis and Applications*, vol. 6, no. 3, pp. 245–256, 2003.

[19] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Workshop learning from imbalanced data sets II," in *Proceedings of International Conference on Machine Learning*, Washington, DC, USA, August 2003.

[20] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*

CIDM'09, pp. 324–331, IEEE, Nashville, TN, USA, March 2009.

[21] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, 2015.

[22] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.

[23] J. Kwak, T. Lee, and C. O. Kim, "An incremental clustering-based fault detection algorithm for class-imbalanced process data," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 3, pp. 318–328, 2015.

[24] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409-410, pp. 17–26, 2017.

[25] V. Vigneron and H. Chen, "A multi-scale seriation algorithm for clustering sparse imbalanced data: application to spike sorting," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 885–903, 2016.

[26] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," vol. 10, Santa Fe Institute, Santa Fe, NM, USA, 1995, Technical Report SFI-TR-95-02-010.

[27] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[28] S. Wang, K. Tang, and X. Yao, "Diversity exploration and negative correlation learning on imbalanced data sets," in *Proceedings of International Joint Conference on Neural Networks, IJCNN 2009*, pp. 3259–3266, IEEE, Atlanta, GA, USA, June 2009.

[29] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: improving prediction of the minority class in boosting," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 107–119, Springer, Cavtat-Dubrovnik, Croatia, September 2003.

[30] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," *DMKD*, vol. 3, no. 8, pp. 34–39, 1997.

[31] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 2007.

[32] T. Kohonen, *Self-Organising Maps Berlin*, Springer, Berlin, Germany, 2001.

[33] H. Chipman and R. Tibshirani, "Hybrid hierarchical clustering with applications to microarray data," *Biostatistics*, vol. 7, no. 2, pp. 286–301, 2005.

[34] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[35] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan, "An approach for classification of highly imbalanced data using weighting and undersampling," *Amino Acids*, vol. 39, no. 5, pp. 1385–1391, 2010.

[36] Q. Li, B. Yang, Y. Li, N. Deng, and L. Jing, "Constructing support vector machine ensemble with segmentation for imbalanced datasets," *Neural Computing and Applications*, vol. 22, no. S1, pp. 249–256, 2013.

[37] N. S. Kumar, K. N. Rao, A. Govardhan, K. S. Reddy, and A. M. Mahmood, "Undersampled K-means approach for handling imbalanced distributed data," *Progress in Artificial Intelligence*, vol. 3, no. 1, pp. 29–38, 2014.

[38] A. D'Addabbo and R. Maglietta, "Parallel selective sampling method for imbalanced and large data classification," *Pattern Recognition Letters*, vol. 62, pp. 61–67, 2015.

[39] J. Ha and J. S. Lee, "A new under-sampling method using genetic algorithm for imbalanced data classification," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, p. 95, January 2016.

[40] G. Shobana and B. P. Battula, "An under sampled *k*-means approach for handlingimbalanced data using diversified distribution," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 1.8, pp. 113–117, 2018.

[41] H. Guo and T. Wei, "Logistic regression for imbalanced learning based on clustering," *International Journal of Computational Science and Engineering*, vol. 18, no. 1, pp. 54–64, 2019.

[42] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on *k*-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, 2018.

[43] W. Han, Z. Huang, S. Li, and Y. Jia, "Distribution-sensitive unbalanced data oversampling method for medical diagnosis," *Journal of medical Systems*, vol. 43, no. 2, p. 39, 2019.

[44] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, and G.-T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," *Information Sciences*, vol. 477, pp. 47–54, 2019.

[45] T. Boongoen and N. Iam-On, "Cluster ensembles: a survey of approaches with recent extensions and applications," *Computer Science Review*, vol. 28, pp. 1–25, 2018.

[46] N. Nguyen and R. Caruana, "Consensus clusterings," in *Proceedings of Seventh IEEE International Conference on Data Mining ICDM 2007*, pp. 607–612, IEEE, Omaha, NE, USA, October 2007.

[47] A. P. Topchy, M. H. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in *Proceedings of Fourth IEEE International Conference on Data Mining ICDM'04*, pp. 225–232, IEEE, Brighton, UK, November 2004.

[48] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intellzigence*, vol. 30, no. 1, pp. 160–173, 2008.

[49] C. Boulis and M. Ostendorf, "Combining multiple clustering systems," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 63–74, Springer, Cavtat-Dubrovnik, Croatia, September 2003.

Hindawi
Scientific Programming
Volume 2019, Article ID 1293194, 17 pages
https://doi.org/10.1155/2019/1293194

Research Article

A Nondisturbing Service to Automatically Customize Notification Sending Using Implicit-Feedback

Fernando López Hernández, Elena Verdú Pérez, J. Javier Rainer Granados, and Rubén González Crespo

Universidad International de la Rioja (UNIR), Av. de la Paz 136, 26006 Logroño, Spain

Correspondence should be addressed to Rubén González Crespo; ruben.gonzalez@unir.net

Received 19 December 2018; Accepted 4 February 2019; Published 3 March 2019

Guest Editor: Vijender K. Solanki

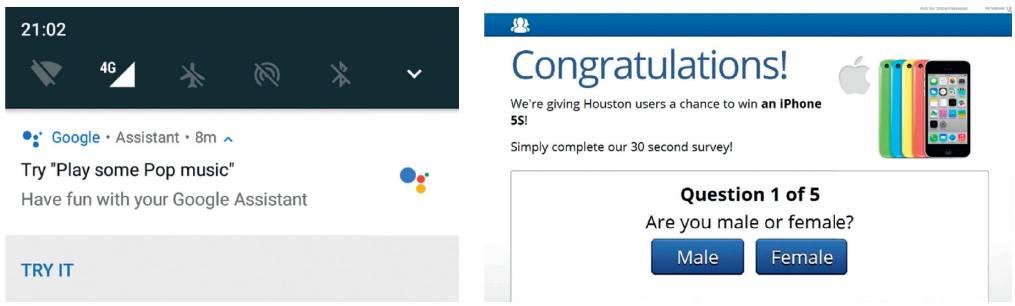
This paper addresses the problem of automatically customizing the sending of notifications in a nondisturbing way, that is, by using only implicit-feedback. Then, we build a hybrid filter that combines text mining content filtering and collaborative filtering to predict the notifications that are most interesting for each user. The content-based filter clusters notifications to find content with topics for which the user has shown interest. The collaborative filter increases diversity by discovering new topics of interest for the user, because these are of interest to other users with similar concerns. The paper reports the result of measuring the performance of this recommender and includes a validation of the topics-based approach used for content selection. Finally, we demonstrate how the recommender uses implicit-feedback to personalize the content to be delivered to each user.

1. Introduction

Companies want to keep their customers informed about the availability of new services and products. However, the continuous sending of notifications to the user's devices can produce the opposite effect [1] if these notifications end up bothering users, who in turn ignore, remove, or block them (Figure 1). To mitigate these adverse effects, it is sensible to be selective and only send the notifications that we are aware that really interest each user. Machine-learning techniques enable the analysis, summarization, and classification of text in a massive and automatic way. Therefore, this paper surveys the use of these tools to identify which clients are interested in which notifications.

The clues used to identify whether a notification subject matters to a user may be explicit or implicit [2]. The *explicit-feedback* constructs the user profile by asking the users for their personal characteristics and preferences for different topics or items. The *implicit-feedback* constructs the user profile by silently observing the behavior of the users (e.g., the time spent on a page, the amount of scrolling, or the number of mouse-clicks) and then inferring their rating [3]. Collecting explicit-feedback conflicts with modern marketing trends and policies. In particular, companies want to maximize the user experience by minimizing the cognitive effort and information-filling burden during any web interaction. An interaction that should be especially agile is registration, because this maximizes the number of new users who complete their registration. However, this policy results in collecting poor explicit-feedback.

This paper studies to which extent a recommender is able to extract clues that significantly improve the sending of notifications when only implicit-feedback is available (i.e., user actions which result in nondisturbing collection of information). For this purpose, on one hand, we have developed mechanisms that translate the implicit-feedback (i.e., user interactions) into topics of interest of the user. On the other hand, we have applied machine-learning techniques (i.e., text mining, summarization, and classification) to extract the topics of each notification. We hypothesize that these pieces of implicit-feedback, along with the automatically classified notifications, enable us to select the interesting notifications for each user. To analyze this hypothesis, we have conducted an experimental evaluation.



(a) (b)

Figure 1: Examples of disturbing notifications. (a) Android notification. (b) Web page campaign.

First, we have collected indicators that act as pieces of implicit-feedback. Second, we have measured the performance of the *recommender* (i.e., the automatic recommendation system) implemented in our service and related our findings with those of other state-of-the-art works.

This paper makes the following main contributions to the field:

(1) Determines to what extent implicit-feedback suffices to make automatic recommendation of notifications.

(2) Develops a mechanism to determine the user preference for the notifications by iteratively reducing dimensionality: terms, characteristic words, and topics.

(3) Demonstrates how text mining, summarization, and classification techniques are effectively combined to create a hybrid filter: a collaborative + a content-based filter that

  (a) Mitigates the cold-start problem (a well-known problem of collaborative filters) because once the user has positively rated one notification, we use the content-based filter to start recommending notifications of same topic.
  (b) Effectively integrates content similarity (the content-based filter finds notifications with the same topic) and diversity (the collaborative filter finds users with the same tastes).
  (c) Does not require explicit domain knowledge of the items (as is the case of knowledge-based techniques) and so does not require metadata-annotated items.
  (d) Completely automates customized notification sending, without any user intervention.

(4) Publishes the source code and the anonymized dataset to enable future research and repetitiveness of the reported experiments (https://github.com/ifernandolopez/hybrid-filter).

The rest of this paper is organized as follows: Section 2 reviews comparable state-of-the-art recommenders and background machine-learning techniques. Section 3 describes the proposed solution. Section 4 provides the design of the experiments. Section 5 provides and discusses the results of our approach and relates it to other approaches. Finally, Section 6 presents the conclusions and possible future work.

2. Literature Review

This section reviews related state-of-the-art recommenders in several application domains, as well as the basis of background machine-learning techniques used to conduct this research.

2.1. State-of-the-Art. Recommenders are popular and widely used. Well-known e-companies use them to improve the user experience [2, 4]. There is a wide variety of items recommend, including movies [5], books [2], music [6], research articles [7], clothes [8], travels [9], events [10], and many more.

Although these systems filter the relevant content and improve the user's satisfaction, some challenges remain; for instance, nondisturbing the user with the feedback collecting mechanism [11], monotonous recommendations [12], or the cold-start problem [2, 13]. Regarding the first challenge, currently many recommending solutions still rely on the explicit-feedback provided by the user, typically the user ratings [14–18]. However, not every user is willing to provide ratings, especially when ratings are optional. Moreover, ratings may be biased by user's contextual and emotional states.

Implicit-feedback is also available. The combination of both implicit-feedback and explicit-feedback may mitigate the problem of lack of explicit ratings. Therefore, different systems in the literature use both types of feedbacks. For example, the sports news recommender described in [19] bases its recommendations on the user's readings (i.e., implicit-feedback) and ratings (i.e., explicit-feedback). Bagherifard et al. [20] also use both implicit and explicit-feedback in a hybrid approach for movie recommendations, though their solution uses ontologies. A hybrid and ontology-based approach is surveyed in [17] using ratings for news recommendations. Also, Agarwal and Singhal [21] propose a solution based on a domain ontology, which uses explicit and implicit data of users; the registered user provides the explicit information, while the implicit information includes mouse behavior and user session data. The system proposed in [22] retrieves keywords from external user and

item sources to generate implicit-feedback to diminish the cold-start problem.

However, although using implicit-feedback and explicit-feedback can enhance recommendations, there are situations in which users are not willing to provide explicit-feedback. Fortunately, there are abundant implicit data that may serve to model the user preferences. In fact, there are authors that target the problem where only implicit-feedback is provided [23, 24]. Núñez-Valdez et al. [24] propose a system that converts implicit behavioral data into explicit-feedback to recommend books. Typical user actions are considered, such as highlighting content, adding notes, or suggesting content to other contacts. The mobile application advertising recommender in [25] also follows the approach of mapping the implicit-feedback (click, view, download, and installation information) into explicit ratings. Besides, they use slowly changing features of the mobile context for recommendations. The news recommender described in [26] uses the clicks on news items to create the user's profile. Also, Li and Li [27] use only user's reading actions for news recommendations, creating a hypergraph to model correlations among items and implicit relations among users. The system proposed by Zheng et al. [28] also keeps track of users' reading actions. Soft clustering is used to group users to enrich the user profile with general reading interest and relates users with similar behavior. For each user group, the system identifies the latent topics, and creates hierarchies. Lu et al. [29] consider implicit actions like browsing, commenting, and publishing. They target scalability and data sparsity by using Jaccard K-means based clustering technique. The users' similarities are calculated considering multiple dimensions, such as the user interactions with content and their eigenvectors of topics. Extracting information from the users' tweets is another strategy to build the user's models for recommendations [30]. Gu et al. [31] propose an approach that uses content in microblog or tweets of users and users' social network preferences (popularity) for news recommendations. Also, content in microblogs is taken into account for recommendations by Zheng and Wang [32], but from a sentimental point of view. Retweeting of news are the implicit activities considered in a content-based recommendation system that models user trends [33].

The second abovementioned challenge relates to monotony in recommendations, which is inherent in pure content-based approaches. Hybrid approaches target the lack of diversity in recommendations by combining content-based and collaborative filtering. For instance, Lenhart and Herzog [19] introduce a collaborative filter to target diversity in their sport news recommender. The keywords are automatically obtained from the pieces of news, as well as from the users' reading data, and are used to estimate the interest of users in topics and provide the content-based recommendation. All of this is complemented with a collaborative-based recommendation that uses users' ratings. Li et al. [26] use hierarchical clustering for grouping news articles and long- and short-term user profiles based on clicks on news, and they incorporate the absorbing random walk model to achieve a diversity of topics in recommendations. Lastly, some systems provide and merge multiple recommendation lists based on different criteria to enhance diversification, which is called *result diversification* [11].

While content-based approaches have the drawback of monotony, they effectively address the cold-start problem [13]. On one hand, apart from facilitating diversity, a clear advantage of collaborative-based approaches is that they do not need domain specific data. However, collaborative filtering alone suffers from the sparsity problem. On the other hand, content-based filtering is not very effective in recommending news items, because usually users only read a small part of these [17]. Therefore, our approach proposes combining collaborative-based filtering, which provides diversity, with content-based filtering, which mitigates the cold-start problem, and addresses the sparsity problem through aggregation and clustering techniques. In addition, our content-based filter does not have domain-dependency, as is the case with other approaches (e.g. [20]).

2.1.1. Datasets for Evaluation. This section reviews potential datasets to evaluate the performance of recommendation with notifications. Although we have not found a dataset of users rating notifications, we have found some datasets of users rating text documents. For instance, the authors of [34] studied topic diversification and have published their dataset with 278,858 users providing ratings about 271,379 books. The authors of [35] have studied dimensionality reduction for offline clustering, and they have published their dataset with 4.1 million ratings of 100 jokes from 73,421 users. Yahoo has published two datasets about news visits: "R6A-Yahoo! Front Page Today Module User Click Log Dataset, version 1.0" and "R6B-Yahoo! Front Page Today Module User Click Log Dataset, version 2.0" (https://webscope.sandbox.yahoo.com/). The first one includes 45,811,883 unique user visits to news articles displayed in the Featured Tab of the Today Module on Yahoo! Front Page during ten consecutive days. The second includes 15 consecutive days of data gathering with 28,041,015 user visits to the same module. Even though they are well populated with users' interactions, they lack other types of interactions (e.g., sharing, printing, skipping, and deletion). In addition, the short periods cause the events to repeat themselves, thus producing certain biases [13]. "Outbrain Click Prediction" (https://www.kaggle.com/c/outbrain-click-prediction/data) dataset also corresponds to a 2-week period, and its main limitation is that it does not provide the content, but only some semantic attributes of the documents.

In summary, after making this survey, we found that (1) most of the studies are based in unmentioned or private datasets [19, 21, 26–28, 36], and, what is worse, (2) we have not been able to find a public dataset of notifications, which motivates our decision to collect and publish our own dataset of user interactions with notifications (described in Section 4.1).

2.2. Background Review. This section reviews the background machine-learning techniques used to conduct this research.

2.2.1. Preprocessing Text. The curse of dimensionality is a well-known effect in text mining [37] that occurs when there are so many dimensions (different words in our case) and that the distance between two documents is always too far. Text preprocessing is a suite of methods that facilitate text analysis by eliminating uninformative text and reducing the dimensionality of the features [38]. Among these techniques, we find

(1) *Tokenization.* Parsing the text to generate terms. Although they are really different concepts [39], traditionally, in text mining *term* and *word* have been used interchangeably, and we also do this in this paper.

(2) *Removing stop words.* A stop word is a commonly used word of a language (e.g., "the", "for", "a"). They are the glue or link for more semantic words, and so stop words give little information about the content of the text. There exist lists of stop words for most languages.

(3) *Removing multiple spaces, numbers, and punctuation symbols.* Symbols provide little information when we analyze tokenized words, but punctuation symbols are useful if, for instance, we analyze phrases.

(4) *Removing repeated text.* Frequently, a document has repeated texts such as headers, footers, or copyright information that are removed to avoid outweighing the importance of these words.

(5) *Removing sparse terms.* The words that appear just once or a few times increase the dimensionality, and their scarceness indicates a low relevance in the document, so they can be removed. In particular, removing words appearing only once in the whole document is an effective approach to remove misspelled words.

(6) *Removing equivalences.* To further reduce the dimension of the terms, terms with the same meaning can be grouped, with techniques such as (a) *lowercasing*; (b) accent removal; (c) *synonyms grouping*, in which a thesaurus is used to homogenize synonymous words; (d) *stemming*, which groups similar words by removing the plurals, and after that it reduces them to their root [40]; or (e) *lemmatization*, which identifies lexically or semantically similar words [41].

2.2.2. Relationships between Words. Text is unstructured data, so we have to capture its structure. The most common ways to represent the relationships between words are the following [42]:

(1) *Bag of word.* This is a simplified representation of a document, in which only the number of occurrences of each word is taken into account, but disregards the order or the words.

(2) *Word vector.* This representation extends the idea of bag of words by assigning a rating to each word in the document. A popular rating is the number of times each word occurs (the approach followed by the bag of word). Below, we will describe a more effective rating technique named TF-IDF.

(3) *Term Document Matrix* (TDM). This representation describes the frequency of each term by using a matrix (Figure 2). For this purpose, we define a *vocabulary* as all the words appearing in any of the document. In this matrix, each row corresponds to a vocabulary and each column to a document, that is, each column is a word vector. Note that this representation requires a corpus, that is, a collection of documents, one for each column. Some authors transpose the TDM so that the rows correspond to the vocabulary and columns correspond to the words appearing in each document. This representation is referred as *Document Term Matrix* (DTM).

(4) *n-grams.* This is a contiguous sequence of n terms for considering the relationship between consecutive words. The clustering of 2 or more words is named a *collocation.* For instance, [43] has used n-grams collocations and Markov Chains to determine the probability of a word being followed by another word and to identify common grammar mistakes.

2.2.3. Characteristic Words of a Document. A problem that has been widely studied is how to find the characteristic words of a document (e.g., [44–47]). These characteristic words can be used, for instance, to implement keyword-based document search.

A first approximation is the *Term Frequency* (*TF*) [48]. This score counts the relative frequency of each term t in the analyzed document d and selects as characteristic terms of the document those with a higher frequency. That is, this score merely selects those terms that maximize the following $TF(t, d)$ function, where $|t \in d|$ is the number of times the term appears in document d, and $|d|$ is the number of terms in the document d:

$$TF(t, d) = \frac{|t \in d|}{|d|}. \quad (1)$$

Note that the ratio in (1) compensates for the differences in length of the analyzed documents, so that the *TF* score only depends on the relative frequency of the term and not on the size of the document.

The major problem with the *TF* is that words with higher frequency tend to be the same in all documents, even if we remove those words in a list of stop words. This effect is an empirical law known as Zipf's law. The conclusion is that the frequency of words is not the best way to find the characteristic terms of a document.

An effective and popular approach to finding the characteristic words of a document within a corpus is the *TF-IDF* score. The *TF-IDF* score has been frequently used along with a clustering algorithm to classify text [49] or decide on the topics of a corpus [50]. This score combines the *TF* with the Inverse Document Frequency (*IDF*) to choose the characteristic words of the document. The *IDF* gives a higher score to rare terms using the following

Term document matrix (TDM)

	d1	d2	d3	d4	d5
Boat	1	0	1	0	0
Sand	0	0	1	1	0
See	1	2	1	1	2
Ship	2	1	0	0	0
Sun	0	0	3	1	0

Word vector

Figure 2: Relationship between word vector, TDM.

formula, by dividing the number of documents in the corpus $|D|$, by the number of documents in the corpus where the term t appears $|t \in D|$. This ratio is never less than 1, so the logarithm is always positive:

$$IDF(t, D) = \ln\left(\frac{|D|}{|t \in D|}\right). \quad (2)$$

Note that in (2), D is uppercased to indicate that the score is calculated with respect to the corpus of documents and not the currently evaluated document d. The *TF-IDF* score takes advantage of the fact that Zipf's law does not only apply to a document, but also to a corpus of documents. Then, we can calculate the frequency of the words in a corpus D and compare it with the frequency of the words in a certain document d.

The *TF-IDF* score is the product of both indices (3), that is, it is the product of the times that the term appears in our document $TF(t, d)$ and the infrequency of the term in the general corpus $IDF(t, D)$:

$$\text{TF} - \text{IDF}(t, d, D) = TF(t, d) \cdot IDF(t, D). \quad (3)$$

The main reasons for the popularity of this score are the following:

(1) It captures how specific the term t is for a given document d only; therefore, it effectively selects a small set of rare words as characteristic words of the document.

(2) It is not necessary to remove stop words. This is because a very frequent term t receives an $IDF(t) = 0$ and so $TF\text{-}IDF = TF(t) \cdot 0 = 0$.

(3) The accuracy of the selection can be improved with stemming, as stemming groups words with the same root.

2.2.4. Topic Modeling. Topic modeling is the area of machine learning that applies unsupervised clustering techniques for discovering the topic of a collection of documents. Latent Dirichlet Allocation (LDA) [51, 52] is a popular topic modeling method to summarize the meaning of the words and documents (the observed variables) in hidden variables (latent low-dimensional clusters). Its popularity is based on its fuzzy clustering approach, in contrast with other hard clustering methods such as Explicit Semantic Analysis [53].

LDA assumes that each document is a mixture of a small number of topics, and each topic is a mixture of a number of words. Therefore, this one-to-many mapping implies an overlap in both the topics of a document and the terms of a topic.

In particular, the LDA algorithm performs Bayesian inference on the observable variables (words and documents) to update the posterior probabilities of the initial belief on the hidden latent variables (topics). The algorithm produces a set of topics, the topic proportion for each topic, and two posterior probabilities:

(1) *Beta probabilities* are the estimates of the probability of a word belonging to each topic. The more often a word occurs in a topic, the higher the value of beta. The words with higher beta probabilities in each topic can be used to summarize the topics.

(2) *Gamma probabilities.* While learning topics, LDA also learns topic proportions per document. The gamma probabilities are the estimates of the proportion of words of a document that are generated by each topic. The more the words of a document are assigned to a topic, the higher the gamma value is.

2.2.5. Recommendation. A *recommender* is an automatic information-retrieval filter that builds a model from the user's profile and behavior to predict the rating or preference that a user would have for an item. One type of recommender is a *content-based filter* [54], which uses the description of the item and the user profile. That is, the content-based filter basically decides the best-matching between these two sets:

(1) *Content*, annotated with keywords that describe the content itself.

(2) *User profile*, in which keywords describe the user's preferences.

For instance, if the content is documents, the keywords that describe this content may be inferred from the *TF-IDF* score of their words. The keywords in a user's profile will correspond to the query to the search engine. Sometimes *tagging* is used to make the content and the user profile comparable (e.g., [55]). As the user visits content, the user's profile is annotated with tags form the content.

Another popular type of recommender is a *collaborative filter*. A collaborative filter [56] collects the preferences that a large group of people have assigned to a group of items and predicts the rating of a user for an item that has not been rated yet. Usually, the collaborative filter operates in two phases:

(1) *Training phase.* During this phase, we collect the user's preferences for particular items and create an Item User Matrix (IUM), like the one shown in Table 1. The rows of the IUM correspond to the items $\mathbf{i} = \{i_1, i_2, \ldots, i_k\}$, the columns correspond to the individual users $\mathbf{u} = \{u_1, u_2, \ldots, u_p\}$, and the entries correspond to the relevance that each user has assigned to each item. Note that usually the IUM is a

Table 1: Example of IUM with user ratings of some items created during the training state.

	u_1	u_2	...	...	u_p
i_1	**		*****		**
i_2		*	***		
⋮	*****	***		**	****
⋮		*	***		
i_k	****		***		**

sparse matrix, that is, most items are not voted by users. The basis of any collaborative filter is to find people with similar tastes. If user u_i and user u_j have assigned similar ratings to a set of items, we assume that they have similar tastes. If we now detect that u_i has assigned a rating to an item that u_j has not rated, we assume that the rating that u_i assigns to this item will be similar to that of u_j.

(2) *Exploitation phase.* During this phase, we receive a new item that a user has not rated, and the collaborative filter predicts the rating that this user will assign to that item.

An issue that collaborative recommenders have is the so-called *cold-start* problem. In particular, the recommender will not properly predict the rating of users when it has not yet collected enough information. This issue occurs in two cases: (a) when the user is new, so we do not have enough information about the topics of interest of the user, and (b) when the item is new, and we do not have enough ratings for this item.

A *hybrid recommender* is a combination of content-based and collaborative filtering. The hybrid recommender takes advantage of both the representation of the content as well as the similarities among users. One advantage of combining information is that this process can produce a more informed prediction. Another advantage is that it can reduce the cold-start problem by overweighing the content analysis when an item has not received enough ratings, and vice versa.

2.2.6. Evaluating Recommendations. Evaluating recommendation basically consists in measuring the ability of a recommender to generalize current user's rates for some items to other unrated items.

Regarding the recommendation tasks to evaluate, Shani and Gunawardana [57] describe different approaches for the evaluation: top-N recommendation, some good items, all good items, rating prediction, utility optimization, etc. In Sections 4.2 and 5.1, we have evaluated the classification performance of our proposal using the top-N recommendation approach. Top-N recommendation assumes that there are a large number of items, but the user does not have time to review all of them. Therefore, top-N recommendation aims to identify the N items that the user will most probably accept. To this end, top-N recommendation orders the items by predicted ratings and chooses the first N.

Regarding the performance metrics, in a different paper [58], Shani and Gunawardana have proposed a set of metrics to compare collaborative filters, such as accuracy, prediction, recall, sensibility, specificity, utility, diversity, coverage, and novelty. Section 4.2 justifies the selection of the metrics used in this research, which are introduced below. Table 2 summarizes the formulas for calculating these metrics:

(1) *Accuracy.* This is a measure of the closeness of agreement between a prediction and its actual value (whether the user has actually accepted it or not). It is computed as the number of items correctly classified (TP + TN) divided by the total number of items analyzed (TP + TP + FP + FN).

(2) *Precision.* This is a measure of how reliable a recommendation is. It measures the proportion of positive items correctly predicted as such (TP), among all items that have been classified as positive (TP + FP).

(3) *Recall.* This is a measure of how complete a result is. It measures the proportion of positive items predicted as such (TP) among all items that are truly positive (TP + FN), that is, among all recommendations that the user would have accepted.

Usually, a recommender aims to find items of the *class of interest*, that is, the items rated as positive, among all the available items. The problem is that usually the items of the class of interest are far less than the total number of items. This problem is known as the *class imbalance problem*. The accuracy alone is not enough to measure the performance of the recommender suffering from the class imbalance problem. This is because the recommender can reach a high accuracy by simply predicting every item as non-recommendable. When the class imbalance problem occurs, precision is a more reliable metric since it only ponders positive predictions. We can also detect that the recommender is recommending too little if the recall falls near to 0.0.

3. Proposed Solution

In this section, we describe the architecture of the service, as well as our proposal of implicit indicators to be used. Then, we describe how we have implemented the recommender, how to compute each of the intermediate matrices and their interpretation, and how to train and use the recommender.

3.1. Service Architecture. The architecture of a recommender has a great impact in the way pieces of feedback are gathered and how they are used. Therefore, this section summarizes the architecture of the recommender that we have implemented. In particular, our recommender is a software-as-a-service (SaaS) application that helps organizations be polite with their users by preventing the sending of notifications that are not likely to interest them. Figure 3 shows the roles involved in our service and the relationships between them.

In particular, there are three roles involved:

(i) *Client company.* This is the organization interested in sending notifications to its users without

Table 2: Formulas for the selected performance measures.

Accuracy = TP + TN/TP + TN + FP + FN	Precision = TP/TP + FP	Recall = TP/TP + FN

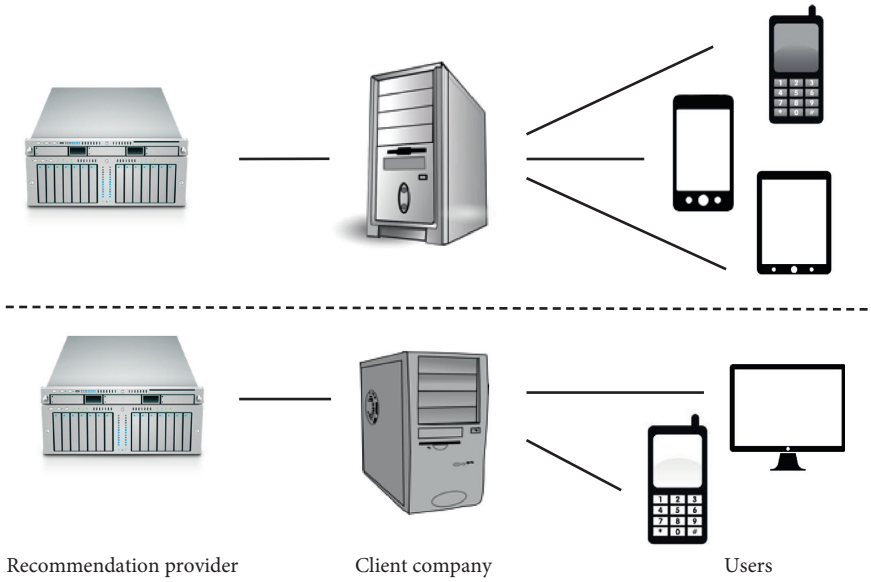


Figure 3: Service architecture (clipart source: pixabay.com).

disturbing them. Therefore, they do not want to send notifications if they are not of interest to their users. For example, a mobile application development company has a list of users. However, this is a polite company, and so does not want to bother its users by asking them to fill out surveys, or sending notifications that are not of their interest.

(ii) *Client users.* These are the users of the above client company. Obtaining implicit-feedback on these users can help send them only notifications of their interest. For example, the mobile application can collect information about the behavior of these users without disturbing them with explicit questions.

(iii) *Recommendation provider.* This is our classification service that this paper describes in more detail. The client company provides implicit-feedback to our recommendation provider. This feedback contains the interactions of its users during past notifications and corresponds to the training phase of our recommender. Given a new notification, our service returns both: (1) the top N notifications for each user and (2) a numerical prediction rating (from 0 to 10) estimating the interest in the N notifications for each user. The client company now has to decide the N number of notifications to select to each user and next from which threshold of numerical interest rating to send the notifications to its users.

Note that this architecture facilitates the service being implemented independently of the business model of the client company. For example, notifications for a cinema may be movies, and notifications for a repair shop may be promotional discounts. In this model, the client company subscribing to our SaaS forms a natural grouping of users, and information is not shared between client companies. The dashed line in Figure 3 indicates this lack of data sharing between companies.

3.2. Selecting Implicit Interest Indicators. As motivated in Section 1, we do not want to bother users asking for explicit-feedback (rating) about the content that they are interested in. Conversely, we want to represent the value of the notifications for the user by mapping the user's interactions to numerical ratings indicators. We use the 1 to 5 Likert scale to capture the level of interest-disinterest on a symmetric scale. We have identified 5 sources of useful interaction (summarized in Table 3):

(1) *Examination.* The first time the user opens a notification, we increase the estimation of the interest of the user in the content by increasing the rating of that user in this notification in +1.

(2) *Reopening.* If later on, the user reopens the notification, this indicates that the user found its content useful, and so we increase the rating by +1 for each new reopening.

(3) *Frequency.* We overweigh the interest from users with little examinations, with respect to a user who examine notifications frequency.

(4) *Fast reading.* We underweigh the interest if we detect a short reading time as an indicator of lower interest. To detect it, we study the time until the next access.

(5) *Printing.* A user who prints a document is showing interest in using it or reading it in more detail.

Table 3: Implicit interest and disinterest indicators.

N	Indicator	Value (rating)
1	Examination	Add +1 for first time it is open
2	Reopening	Add +1 for subsequent reopening
3	Frequency	Add +0.5 or −0.5 depending on whether the number of exploration is above or below the mean
4	Fast reading	Add −0.5 if we detect a reading time below 5 seconds
5	Printing	Add +1 as printing indicate interest in detailed reading
6	Sharing	Add +1 for each time the user shares it
7	Skipping	Rate 2 (dislike)
8	Deletion	Rate 1 (strongly dislike)

(6) *Sharing*. The notifications are annotated with a "share it" button. If the user shares a notification, this is an indicator that the notification has value for the person with whom the user is sharing it. Therefore, we increase the notification rating in +1 each time the user presses the "share it" button.

(7) *Skipping*. If the users go through the summary of a notification without opening it, this is an indicator that this content does not particularly interest them. Therefore, we assign a rating of 2 (dislike) to the notifications that the user has not opened.

(8) *Deletion*. Notifications are annotated with a "delete" button. If the user makes the effort of explicitly deleting a notification, this indicates that its content not only is not of their interest, but somehow it is disturbing or even offending. Therefore, we assign a rating of 1 (strongly dislike) to the notifications that the user deletes.

The maximum rating that a notification can receive is +5, as this value is an enough indicator of high interest of the user in this notification.

Note that these indicators have been chosen to ease their nondisturbing collection in different web applications. Of course, other more accurate indicators can and have been used (e.g., reading time, mouse movements, and highlighting content [2, 3]). However, the additional effort of their collection in a web page (i.e., using JavaScript client scripting) causes the websites to refuse the integration of these gathering protocols, because this may slow down the page rendering. In fact, this is what happened when we asked website administrators of our University, UNIR, to integrate these scripts to collect more advanced implicit indicators for our experiments. Therefore, we opted to develop a solution that can be integrated into websites easily and effortlessly. Nonetheless, the reader can add to the model more elaborated indicators if they find their collection feasible in their website.

Note that different website owners will be willing or able to collect different indicators. Therefore, you can optimize the initial values of the indicators proposed in Table 3 by adjusting the model parameters during the training phase. Note also that Table 3 contains variables, and this adjustment may not be necessary if the classifier automatically scales their values during the training phase.

3.3. Recommendation Approach. This section describes how our recommender predicts the most interesting notifications for each user. The process described herein performs a dimensionality reduction, so that we start with the terms of the documents and end with the topics of interest for each user.

This section describes our recommender and so uses the term "notifications," although traditionally the literature has used the term "documents." Therefore, in the rest of this document, the terms document and notification are used as interchangeable synonyms.

Our recommender is a hybrid recommender combining the two filtering approaches described in Section 2.2.5. In particular, our hybrid recommender operates in two major phases:

(1) *Content-based filtering*. The recommender uses text mining techniques to reduce the dimensionality of the words in the documents to topics in the following two steps. First, we use the *Term Document Matrix* (TDM) to identify the characteristic words of the document, that is, those words with the higher *TF-IDF* score. Second, the recommender applies a LDA algorithm (Section 2.2.4) to compute the *Document Topic Matrix* (DTM), which represents the topic proportion for each document.

(2) *Collaborative filtering*. The recommender applies collaborative filtering on the dataset to further reduce dimensionality and predict the interest of the user in an unseen notification. In particular, the recommender first creates a *Document User Matrix* (DUM) gathering the interest of the user in each document. Second, the recommender utilizes the DUM and the DTM calculated above to produce the (*Topic User Matrix*) TUM, which contains the interest of each user in each topic.

When a new unseen notification arrives, we apply the first step to identify the topics of the notification. Then, we use the second step to predict the interest of each user in the topics of the notification.

Figure 4 summarizes the flowchart and concepts that our hybrid recommender uses. In particular:

(1) *Flowchart*. The dashed arrows indicate the matrices computation steps. First, we use the indicators dataset to generate the DUM (Section 3.3.3). Second, we retrieve the documents and compute the TDM and DTM. Third, we combine the DTM and DUM in the TUM (Section 3.3.4).

(2) *Relationships*. Double arrows indicate the matrices relating these concepts. The following sections describe in more details the relationships in Figure 4 and how we compute each matrix.

3.3.1. Computing the TDM. The first task that the recommender has to do is to analyze the text of the notification and obtain a concise representation by means of the TDM, which eases the selection of the characteristic words of the notifications. This TDM performs a dimensionality

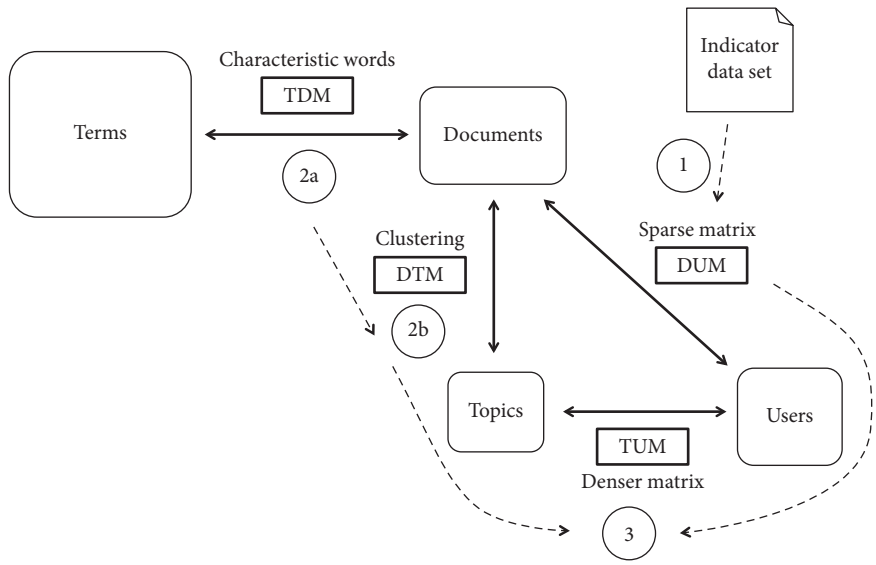


Figure 4: Flowchart and matrices relating the concepts in our hybrid recommender. The TDM (Term Document Matrix) counts the terms in each document. The DTM (Document Term Matrix) contains the estimated topic to document imputation. The DUM (Document User Matrix) indicates the user-document interest. The TUM (Topic User Matrix) contains the estimated interest of the users in each topic.

reduction by discarding uninformative words; for the remaining words, the TDM contains the *TF-IDF* scores. We compute the TDM in the following steps:

(1) *Preprocessing.* In this step, the notifications are tokenized, and then we remove stop words, white-spaces, numbers, and punctuation symbols (Section 2.2.1). We also group equivalent terms using lowercasing, accents removal, synonym grouping, and stemming. Finally, we remove words appearing only once in the whole corpus because by inspection, we have found that it is a useful way to eliminate misspelled words. Note that it is not necessary to remove repeated terms (such as headers, footers, or copyright information) because according to (2), those terms will have an $IDF(t, D) = 0$, and therefore, they will never be selected as characteristic words in the next step.

(2) *Feature selection.* In this step, we identify and ponder the characteristic words of a document. We have observed that merely counting the occurrence of terms in the notifications implies that the words more frequently used in the language are over-pondered during classification. To remove this bias, we have used the *TF-IDF* score to decide whether a word is a characteristic word of a document. In particular, as described in Section 2.2.3, this score considers the number of times that a term t appears in a document $|t \in d|$ as well as the document length $|d|$, and compares it with the number of times that this term appears in the corpus $|t \in D|$, to provide more weight to uncommon words that are appearing relatively often in the document d. These words will be the characteristic words of the document.

(3) *TDM matrix generation.* As described in Section 2.2.2, the TDM is a sparse matrix representation of the words in each document. In particular, in the TDM, a *word vector* corresponds to a column in the TDM, a row corresponds to the vocabulary, and the *TF-IDF* score indicates the level to which a word is important (i.e., a characteristic word) for a document. Therefore, we can reduce the dimensionality of the TDM by selecting the words with a higher *TF-IDF* score, that is, the less frequent words correspond to the characteristic words.

3.3.2. Computing the DTM. When two notifications have a relatively similar *TF-IDF* score for their word vectors, this is an indicator that they are dealing with the same topics. To find related notifications, we have to project them into groups with related topics by applying a standard clustering algorithm. In this way, once we determine that a notification is related to a specific topic, we assume that the notifications of its cluster deal with relatively similar topics.

As we do not want to have our customer manually labeling the topics of their notifications, we have opted to use an unsupervised clustering algorithm. In particular, we use the collapsed Gibbs sampling method as described in [59]. To determine an appropriate number of topics, we used the *binary logarithm rule* (4), where the number of nodes of the tree corresponds to the number of documents in the corpus $|D|$, and the depth of the tree corresponds to the number of topics k, that is,

$$k = \lceil \log_2 |D| \rceil. \tag{4}$$

After executing the clustering algorithm with k topics, each row in the DTM corresponds to a document, each column corresponds to a topic, and each entry is an integer

indicating the number of times words in each document were assigned to each topic.

Finally, we normalize the DTM so that all rows sum up to 1.0. In this way, an entry in the DTM will contain the levels of belonging of each document to each topic.

3.3.3. Computing the DUM. The DUM corresponds to the IUM traditionally used in collaborative filters (as described in Section 2.2.5), but where items correspond to documents, and the user's ratings are implicit. We compute the DUM by converting the collected implicit-feedback into the user's ratings according to Table 3. In particular, the rows in the DUM represent documents, and the columns represent users.

Note that we will use the DUM in two ways:

(1) *Normalized.* For content filtering, we do the normalization because, in this case, the DUM is merely an intermediate step to calculate the TUM (computed in the next step), which represents the prediction of topics of interest for each user. In particular, we normalize the DUM so that the column of each user sum to 1.0. In this way, the votes of all users weigh the same. That is, if the user has accessed several documents, the user has shown more interest in more documents, but the opinions of all users are equally important to determine the topics in the next step.

(2) *Unnormalized.* For collaborative filtering, we are searching for documents that are of interest to other similar users. Therefore, we use the DUM as is to retain information on all documents evaluated; that is, without lessening the relevance of entries of users who have interacted with more documents.

3.3.4. Computing the TUM. The above DUM has two main difficulties:

(1) There is no direct mechanism to use the DUM to predict the level of interest that a user will have when new unseen notifications arrive.

(2) The number of notifications grows rapidly, and the users do not provide any implicit-feedback for most of the notifications. As a consequence, the DUM will be a sparse matrix.

The TUM addresses both problems:

(1) The TUM relates the users to their topics of interest. Therefore, on the arrival of a new notification, we use the DTM to determine the topics of the notification, and therefore, we map these topics to the level of interest of each user.

(2) The number of topics tends to grow more slowly than the number of notifications; therefore, the TUM is denser than the DUM.

The TUM is computed as follows:

(1) The recommender normalizes the DUM so that the documents of interest for each user sum 1.0. This normalization aims to represent the relative importance of topics for each user, irrespective of how active the users are individually.

(2) The recommender multiplies the transposed DUM by the DTM to obtain the TUM. In the TUM, rows represent users, columns represent topics, and entries represent the levels of interest of each user in each topic.

Note that we use the DUM colwise normalized and the DTM rowwise normalized. Therefore, the TUM will be rowwise normalized, which means that the interest of each user for the topics will sum 1.0.

3.3.5. Recommendation Phases. As usually in automatic recommendation, our recommender also operates in two major phases described here: the *training* and the *exploitation phases*.

(1) Training phase. During this phase, we analyze the text of the training notifications and use the implicit *indicator dataset* collected in Section 4.1, to generate the TDM, DTM, DUM, and TUM. In particular, during this phase, the following tasks are executed:

(1) The recommender uses the indicator dataset to retrieve the text of the notifications.

(2) The recommender analyzes the text of the notifications and computes the TDM (Section 3.3.1).

(3) The recommender applies the topic-clustering algorithm to the TDM in order to obtain the DTM, which indicates the proportion of topics of interest in each document (Section 3.3.2).

(4) The recommender uses this indicator dataset and indicators described in Section 3.2 to create the DUM, which contains the user's interest in each document (Section 3.3.3).

(5) The recommender computes the TUM, which indicates the interest of the user in each topic (Section 3.3.4). In particular, the TUM is computed with formula (5), where DUM'_n represents the normalized and transposed DUM:

$$\mathrm{TUM} = \mathrm{DUM}'_{\mathrm{norm}} \cdot \mathrm{DTM}. \tag{5}$$

(2) Exploitation phase. During this phase, we receive a new *user*, and we have to predict a top-N recommendation list for that user with notifications that this user has not rated. This implies the following task:

(1) The recommender uses formula (6) to obtain the $\mathrm{DUM}_{\mathrm{predict}}$, which predicts the level of interest of the user for each unseen document.

$$\mathrm{DUM}_{\mathrm{precict}} = \mathrm{TUM} \cdot \mathrm{DTM}'. \tag{6}$$

Then, we normalize the $\mathrm{DUM}_{\mathrm{precict}}$ so that all rows sum up to 1.0. Lastly, we select the higher predicted

scores up to $1-1/k$ for the user. Note that formula (4) defines k clustering topics for 2^k documents. Thus, $1-1/k$ approaches 1.0 as the number of topics (and also documents) increases. This means that the more the documents there are, the higher the threshold will be for selecting a document.

(2) We create a top-N recommendation list in the following way:

(a) *Content filtering phase.* The recommender selects the documents for which the predicted user interest is above $1-1/k$ threshold, where k is the number of topics (4). This phase uses the content similarity criterion to recommend content.
If the above phase is not enough to obtain N unseen documents, this means that there is no more content on the topics that are of interest to the user. Then, we activate the collaborative filter.

(b) *Collaborative filtering phase.* We compute the user-user similarity to find the documents that have been of interest to similar users. In particular, we compute the *User User Matrix* (UUM) using the cosine similarity formula (8). Then, we use the higher rated documents by the most similar users to complete the top-N recommendation list. This phase uses the diversity criterion to recommend content.

4. Methods

This section summarizes how we have created the dataset for evaluating our proposal, evaluation criteria, and protocol for evaluating the classification performance. We also describe how we have validated dimensionality reduction; that is, the selection of the characteristic words of each document and the unsupervised model to cluster by topics.

4.1. Experiments Setup. Before initiating this collection, we have reviewed different datasets (Section 2.1.1). However, we found these datasets inappropriate for our research, because we need implicit indicators, such as the ones defined in Table 3 for notifications of a company. Therefore, we have accomplished the collection of our implicit indicators in the *indicator dataset*.

4.1.1. Collecting the Notifications. As described in Section 4, our ultimate goal is to create a recommendation provider that helps client companies customize the sending of notifications to their users. As we have not been able to find a standard dataset containing these notifications from a company, we initiated the construction of our own dataset with the resources we have in UNIR. In particular,

(i) The notifications we have used in our experiments are blog posts from a list of RSS URLs in Spanish at UNIR Revista (http://www.unir.net/vive-unir/). UNIR shows the students these blog posts in the front page of a number of virtual courses (4 graduate courses and 20 postgraduate courses, all of them on different topics related to technology and engineering). Therefore, these blog posts resemble the notifications that we want to simulate. RSS and Atom are standard protocols for publishing blog posts. As they use a well-defined XML format, this content can be easily collected.

(ii) With this dataset, we are assuming that the blog posts are equivalent to the notifications that we intend to evaluate. However, the performance of blog posts vs. notifications recommenders is not always directly comparable. For instance, notifications are often extremely short texts, compared to blog posts.

(iii) The indicators dataset we currently have is a dataset in progress. Though currently it is a small dataset, we have published it.

4.1.2. Collecting the User Interactions. To collect the implicit indicators for our experiments, we have published the abovementioned blog posts in the front page of the virtual courses of various subjects that have taken place in the academic year 2018-2019 at our university (UNIR).

The protocol to show the blog post to the student has been as follows:

(1) When student enters the classroom, the latter 5 blog posts are shown.

(2) Merely clicking on its title redirects the user to the blog post, and we record the date, user ID, and visited URL.

(3) We have been registering this activity for 1 month.

Although the collection of this dataset is a work in progress, at the time of writing this article, we have obtained the interactions of more than 100 students. With the logs of this activity, we are able to collect indicators 1–4 of Table 3.

4.2. Evaluation Criteria and Protocol. Since our approach aims to be nondisturbing, we are interested in selecting the best notifications from a large number of notifications. Therefore, we measured the classification performance using the top-N recommendation task (Section 2.2.6).

To measure the classification performance of our implicit-feedback recommender, we follow a leave-one-out approach. In particular:

(1) The recommender iterates the users in the DUM in which each nonempty entry indicates the actual interest of the user in this notification. We remove a nonempty entry from the DUM for each user with 2 or more entries. That is, we need at least one remaining rating in the DUM to know something about the user. The *removed document* will be the document to be tested, the corresponding user is the *test user*, and the new matrix will be the *test DUM*.

(2) The recommender uses test DUM to regenerate the TUM as described in Section 3.3.4.

(3) The recommender obtains the top-N recommendation list, executing the prediction phase described in Section 3.3.5. In this top-N recommendation list, N is the number of documents in the *user list*, i.e., the number of elements for which the user has shown interest according to the original DUM.

(4) We contrast the user list with the top-N recommendation list measuring the accuracy, precision, and recall. In particular, for each recommendation in the top-N recommendation list, the confusion matrix is generated according to the imputation rules described in Table 4.

4.3. Dimensionality Reduction Model Validation. In addition to evaluating the classification performance, we validate the consistency of the dimensionality reduction implemented in the content-based filter. In particular, the content-based filter implements two reductions of dimensionality:

(1) *Characteristic words of a document.* We aim to validate the semantic coherence between the characteristic words chosen to represent the documents and the latent topics. The results are reported in Section 5.2.

(2) *Unsupervised topic clustering.* As LDA topic clustering is unsupervised, we wonder whether the obtained clusters are semantically adequate. For this purpose, we have studied the coherence and convergence, as described below. The results are reported in Sections 5.3 and 5.4.

4.3.1. Coherence. To determine to what extent documents classified by topic are coherent with the documents classified by characteristic words, we have calculated the *Document Distance Matrix* (DDM) using two features:

(i) The distance among documents according to the *TF-IDF* score: $\mathrm{DDM}_{TF\text{-}IDF}$

(ii) The distance among documents according to the proportion of topic imputation: $\mathrm{DDM}_{\mathrm{topic}}$

If these features are coherent, the difference between both will be low:

$$\text{heatmap} = \left|\mathrm{DDM}_{TF-IDF} - \mathrm{DDM}_{\mathrm{topics}}\right|. \tag{7}$$

To calculate the DDM, we first measure the similarity between documents as the degree to which the features (either *TF-IDF* or topics) overlap. For this purpose, we use the cosine similarity between the features vectors of each pair of documents. The cosine similarity takes the sum of the n features product normalized by the product of their Euclidean lengths. In particular, for the documents with word vectors $\mathbf{u}$, $\mathbf{v}$, the cosine similarity is defined as

$$\text{similarity}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{|\mathbf{u}|\cdot|\mathbf{v}|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}. \tag{8}$$

Table 4: Confusion matrix classification rules.

Imputation	Description
TP	The recommended document is in the user list
TN	A nonrecommended document is not in the user list
FP	A recommended document is not in the user list
FN	A nonrecommended document is in the user list

In general, the cosine similarity ranges from −1.0, meaning exactly opposite vectors, to 1.0, meaning exactly the same vectors. However, as the vector values are all positives, (8) will range from 0.0 (completely disjointed documents) to 1.0 (the same document).

To use the similarity measure in (8) as a distance metric, we use the following formula:

$$\text{distance}(\mathbf{u},\mathbf{v}) = 1.0 - \text{similarity}(\mathbf{u},\mathbf{v}). \tag{9}$$

The resulting DDM is a squared symmetric matrix where the entry on row i and column j represents the distance between documents d_i and d_j.

4.3.2. Convergence. The collapsed Gibbs sampling method [59] repetitively iterates all words in all documents updating prior and posterior probabilities of the hidden variables (topics). After a number of iterations, the model tends to converge to a stable topic assignment state. The *perplexity index* [60] has been proposed to determine when the model is fitted, and we can stop the iterations. Basically, this index computes the likelihood of the parameters given the observations. The perplexity is defined as the natural log of two likelihood values:

(i) *Full likelihood.* The log-likelihood including the prior.

(ii) *Assignments likelihood.* The log-likelihood of the observations conditioned to the assignments.

A lower likelihood score indicates better generalization performance. Section 5.4 studies this convergence.

5. Result and Discussion

This section provides the results of the classification performance as well as a validation for the coherence and convergence of our dimensionally reduction approach.

5.1. Classification Performance. Figure 5 shows the classification performance of executing the above evaluation protocol. You can obtain the numerical values of this figure in the file *evaluation.R*. The horizontal axis shows the time evolution, and the vertical axis shows the classification performance using the indicators accumulated up to the day of the evaluation.

Inspecting the collected indicator dataset, we found that notification recommendation suffers from the class imbalance problem (Section 2.2.6), that is, the user does not show interest in most of the notifications that were presented.

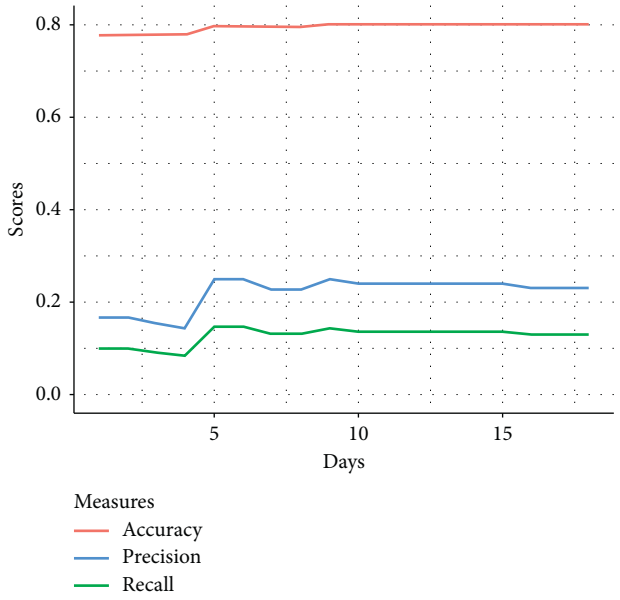


Figure 5: Performance of the top-N recommendation using users with 1 or more interactions.

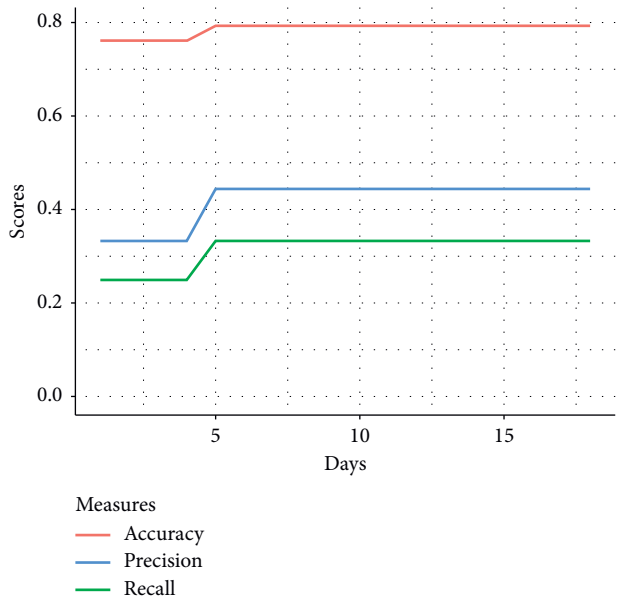


Figure 6: Performance of the top-N recommendation using users with 2 or more interactions.

Similarly, the top-N recommendation list also suffers this effect, as most of the notifications are not in this list. Therefore, accuracy overestimates the classification performance of the recommender (Figure 5). Nonetheless, we have included accuracy in our analysis to ease the comparison, as it is a standard metric in most of the state-of-the-art recommenders.

Limiting false positives is essential to avoid the lack of trust that occurs whenever the recommender returns a noninteresting notification. Precision indicates the ability of the recommender to create a top-N recommendation list that resembles the user list, that is, without FP. This is an ambitious goal, because the precision formula (Table 2) compares how many times we succeed against how many times we make a wrong recommendation, but disregards all the documents that were correctly filtered (i.e., TN). This fact justifies the high difference between accuracy and precision in Figure 5. That is, approximately, only in 13% of the cases, the recommender is able to correctly guess which is the leave-one-out element.

To best estimate the classification performance, we have also added recall to Figure 5, which indicates the completeness of the actual top-N notifications for the user.

The most obvious way to increase precision and recall is to increase the number of user interactions. Figure 6 shows how precision and recall improve if we repeat the above evaluation using only those users who have interacted with 2 or more documents.

Other authors obtain the same effect increasing the number of user interactions. For instance, [26] follows our top-N performance evaluation approach; when they use 10 elements in the list, they obtain a precision of 0.22 and a recall of 0.25. If they increase these elements to 30, they achieve a precision of 0.32 and a recall of 0.42. Similar figures are obtained by the system presented in [28], in which the recall increases from 0.25 to 0.41 when the number of elements increases from 10 to 30. Also, the system presented in [27] gets higher f-measure values as the number of elements in the list increases. Unfortunately, we have not been able to reproduce their experiments with our indicators dataset because our dataset is sparser and we barely have users with more than 3 interactions. Notice that although the results are similar, datasets used in the experiments are not the same, and so the results cannot be directly compared.

The authors of [19] also combine content-based and collaborative filtering, by targeting sports news, which is a further controlled domain than ours. In addition, they combine explicit rating for collaborative filtering, with implicit-feedback for content-based filtering by counting the number of times the user accesses the news. To evaluate the system, they analyze the user clicks (of around 5000 users during 10 days) and they find that 27% of the recommended articles are viewed, being 50% of recommended articles removed. Although the evaluation method is user based (i.e., it is not offline), we find these results to be in concordance with the precision obtained in the other studies mentioned above.

Finally, it is worth to mention that the design of all these experiment assumes that the users always choose the documents that are of maximum interest for them. However, it is known that the behavior of users on the Internet is impulsive and explorative [61]. Therefore, we hypothesize that part of these relatively low precision and recall scores are due to the fact that users access the documents without analyzing in details which are the most interesting for them. To further analyze this hypothesis, we would have to ask the user, which would involve comprehensive fieldwork for future work. An argument in favor of this hypothesis is that the user's choices are only based on the title of the post, while the recommender analyzes the entire text thoroughly.

Table 5: The top words and representative document for each cluster.

Doc ID	Top characteristic words		Title of the most representative document
2	Eficiencia	Mejora	Luis lizasoain: "Cualquier centro de cualquier tipo puede ser de alta eficacia"
3	Colombia	Empresa	Los nuevos graduados se suman a la creciente familia colombiana de UNIR
4	Derecho	Seguridad	Nace la escuela sagardoy de derecho del trabajo

Note that top characteristic words have a high semantic relationship with the document that best represents each topic.

Although implicit-feedback is easier to obtain, it is a challenge to convert raw data into user ratings because implicit-feedback is inherently noisy. Given that the ratings are somehow artificially created from implicit data, a confidence level may be considered to gauge the confidence in the estimated ratings. Particularly, there are some studies finding that reducing confidence in the preferences of those users with more intense activity improves the performance [25].

5.2. Representative of Each Topic. To determine the representativeness of the characteristic words and topics, we have used our dataset to generate the 2 top words in each topic, as well as the document that best represents each topic. The rows in Table 5 correspond to the topics. For each topic, we show the two most representative words along with the title of the most representative post for this topic.

The DTM represents the assignments of the document to the topics. Figure 7 shows the distribution of topics across all the documents. You can obtain the numerical values of this figure in the file *validation.R*. Note that the representative documents in Table 5 match the documents with the highest proportion of the corresponding topic in Figure 7.

5.3. Coherence. This section studies the coherence of the dimensionality reduction approaches by measuring the difference among the documents classification according to the $\mathrm{DDM}_{TF\text{-}IDF}$ and the $\mathrm{DDM}_{\mathrm{topic}}$ (Section 4.3.1 for further details).

Remember that formula (9) calculates this distance between these matrices, where 0.0 means the same document and 1.0 means completely disjointed documents. Figure 8 shows the heatmap of this difference (7). You can obtain the numerical values of this figure in the file *validation.R*. Light colors indicate high coherence; that is, low difference between both approaches to measuring distances. Note that the heatmap is symmetric, and both metrics reach maximum coherence when both documents are equal (the main diagonal). In general, both metrics give similar distances, and so the heatmap is light. The darkest squares correspond to a lower coherence; that is, the documents are not receiving the same distance with both approaches.

5.4. Convergence. Our LDA clustering algorithm uses four parameters:

(i) *Number of topics.* We use the binary logarithm rule (4). Sections 5.2 and 5.3 discuss the suitability of this parameter.

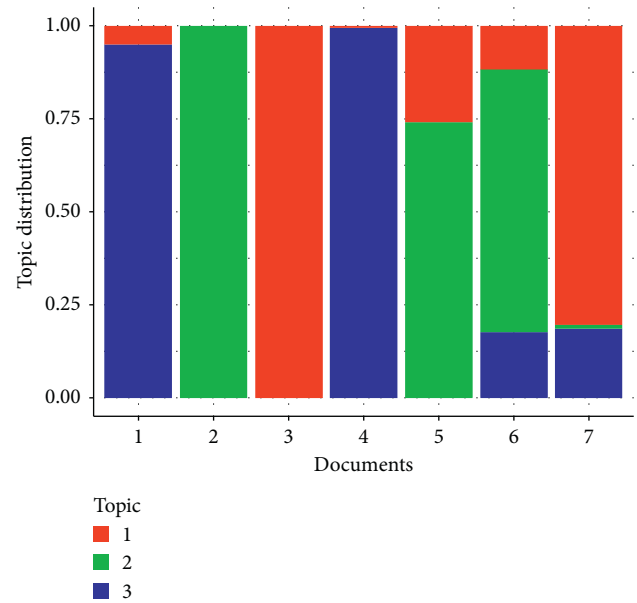


Figure 7: Distribution of topics in the documents.

(ii) *Iterations and learning rates.* We execute $i = 100$ iterations. The $\alpha = 0.1$, $\beta = 0.1$ probabilities can be interpreted as learning rates. Section 5.4 discusses the suitability of these parameters.

Figure 9 shows the convergence of our dataset with the learning rates $\alpha = 0.1$, $\beta = 0.1$, and $i = 100$ iterations. You can obtain the numerical values of this figure in the file *validation.R*. A greater log-likelihood is considered more adequate for the parameters. We can observe that after 15 iterations, the model has stabilized.

6. Conclusions

This paper shows how the gathering of pieces of implicit-feedback is enough to personalize content delivery, that is, without the need to disturb the user by asking them to fill in additional personal information. The recommender operates autonomously and automatically with standard data mining techniques, so its use does not imply an additional cost of adding to the notifications metadata (as is usually the case with other content-based and knowledge-based recommenders). The recommender is able to select content for users with a similar profile using standard collaborative-filtering techniques. The adding of content-based filtering allows us to effectively address the cold-start problem (a limitation of pure collaborative filters). In particular, it is

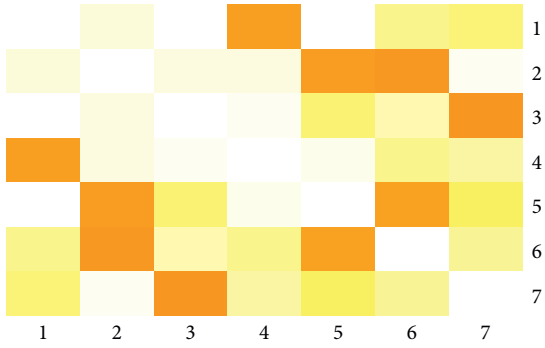


Figure 8: Heatmap of the difference between $DDM_{TF\text{-}IDF}$ and DDM_{topics}.

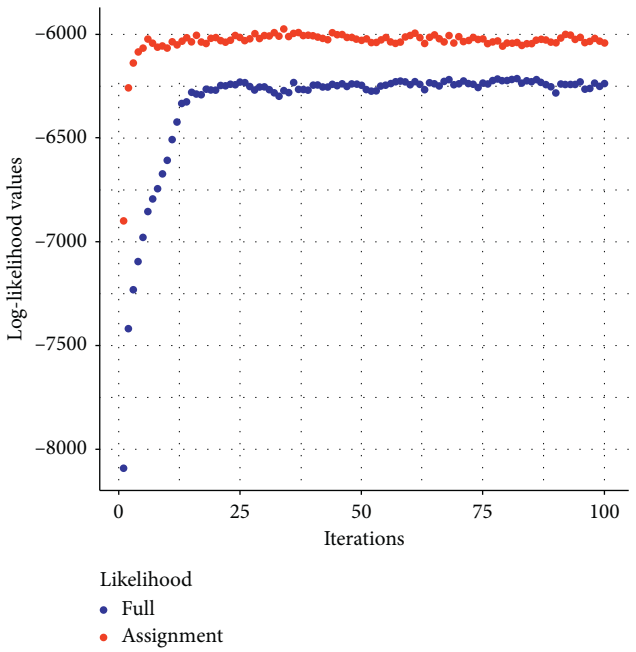


Figure 9: Convergence.

enough that a user has chosen a single document to determine the topics of interest and initiate recommendations on similar topics.

6.1. Future Work. We have identified three areas of future work:

(1) Our recommender has been evaluated offline, without the explicit participation of the user in the evaluation. However, according to some studies, great offline performance does not necessarily mean online success [62]. Therefore, it is important to also consider the perceived utility of recommendations by the user in future work. This work would also allow us to analyze the hypothesis laid out in Section 5.1: to what extent the user exhaustively analyzes or impulsively chooses the documents [61].

(2) The user interests change over time [26]. Therefore, as future work, we may introduce some temporal mechanism that model the gradual decay of the relevance of past readings [26, 29], or the user trends [33].

(3) Finally, the evaluation of the classification performance has been implemented with a relatively small dataset, which also does not include all the indicators defined in Table 3. For this reason, we want to increase the volume and type of implicit indicators and update our published dataset.

Data Availability

The anonymized dataset and source code needed to reproduce these results are publicly available at https://github.com/ifernandolopez/hybrid-filter.

Disclosure

The authors have not published the raw dataset, but an anonymized version of it.

Conflicts of Interest

The authors declare that there are no conflicts of interests.

Acknowledgments

The authors thank the technical department of UNIR for creating the tool that collects the user interaction of several virtual courses.

References

[1] J. S. Keem and S. Lee, "A study on consumers' experiences and avoidances of mobile shopping application advertisements," *Journal of Global Fashion Marketing*, vol. 9, no. 2, pp. 148–160, 2018.

[2] E. R. Núñez-Valdez, J. M. C. Lovelle, G. I. Hernández, A. J. Fuente, and J. E. Labra-Gayo, "Creating recommendations on electronic books: a collaborative learning implicit approach," *Computers in Human Behavior*, vol. 51, pp. 1320–1330, 2015.

[3] M. Claypool, P. Le, M. Wased, and D. Brown, "Implicit interest indicators," in *Proceedings of the 6th International Conference on Intelligent User Interfaces-IUI '01*, pp. 33–40, Santa Fe, NM, USA, January 2001.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[5] R. Katarya, "Movie recommender system with metaheuristic artificial bee," *Neural Computing and Applications*, vol. 30, no. 6, pp. 1983–1990, 2018.

[6] R. Katarya and O. P. Verma, "Efficient music recommender system using context graph and particle swarm," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2673–2687, 2018.

[7] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.

[8] X. Hu, W. Zhu, and Q. Li, "HCRS: a hybrid clothes recommender system based on user ratings and product features," 2014, http://arxiv.org/abs/1411.6754.

[9] J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A. Almeida Figueiredo, and C. Martins, "A hybrid recommendation approach for a tourism system," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3532–3550, 2013.

[10] D. Horowitz, D. Contreras, and M. Salamó, "EventAware: a mobile recommender system for events," *Pattern Recognition Letters*, vol. 105, pp. 121–134, 2018.

[11] M. R. Ghorab, D. Zhou, A. O'Connor, and V. Wade, "Personalised information retrieval: survey and classification," *User Modeling and User-Adapted Interaction*, vol. 23, no. 4, pp. 381–443, 2013.

[12] M. Kunaver and T. Požrl, "Diversity in recommender systems-a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017.

[13] M. Karimi, D. Jannach, and M. Jugovac, "News recommender systems-survey and roads ahead," *Information Processing & Management*, vol. 54, no. 6, pp. 1203–1227, 2018.

[14] M. H. Aghdam, M. Analoui, and P. Kabiri, "Analiysis of self-similarity in recommender systems," in *Proceedings of 2014 Iranian Conference on Intelligent Systems (ICIS)*, pp. 1–4, Bam, Iran, February 2014.

[15] A. Montes-García, J. M. Álvarez-Rodríguez, J. E. Labra-Gayo, and M. Martínez-Merino, "Towards a journalist-based news recommendation system: the Wesomender approach," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6735–6741, 2013.

[16] O. Sanjuán, E. Torres, H. Castán, R. Gonzalez, C. Pelayo, and L. Rodriguez, *Viabilidad de la Aplicación de Sistemas de Recomendación a Entornos de e-Learning*, University of Oviedo, Oviedo, Spain, 2009.

[17] E. Mannens, S. Coppens, T. De Pessemier et al., "Automatic news recommendations via aggregated profiling," *Multimedia Tools and Applications*, vol. 63, no. 2, pp. 407–425, 2013.

[18] N. Jonnalagedda, S. Gauch, K. Labille, and S. Alfarhood, "Incorporating popularity in a personalized news recommender system," *PeerJ Computer Science*, vol. 2, p. e63, 2016.

[19] P. Lenhart and D. Herzog, "Combining content-based and collaborative filtering for personalized sports news recommendations," in *Proceedings of ACM Conference on Recommender Systems CBRecSys@ RecSys*, pp. 3–10, Boston, MA, USA, 2016.

[20] K. Bagherifard, M. Rahmani, M. Nilashi, and V. Rafe, "Performance improvement for recommender systems using ontology," *Telematics and Informatics*, vol. 34, no. 8, pp. 1772–1792, 2017.

[21] S. Agarwal and A. Singhal, "Handling skewed results in news recommendations by focused analysis of semantic user profiles," in *Proceedings of 2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, pp. 74–79, USA, February 2014.

[22] M. Y. Hsieh, T. H. Weng, and K. C. Li, "A keyword-aware recommender system using implicit feedback on Hadoop," *Journal of Parallel and Distributed Computing*, vol. 116, pp. 63–73, 2018.

[23] N. Jiang, "Implicit feedback recommender system based on matrix factorization," in *Proceedings of the 12th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, pp. 77–85, Wuhan, China, September 2018.

[24] E. R. Núñez-Valdez, D. Quintana, R. González Crespo, P. Isasi, and E. Herrera-Viedma, "A recommender system based on implicit feedback for selective dissemination of ebooks," *Information Sciences*, vol. 467, pp. 87–98, 2018.

[25] J. Hu, J. Liang, Y. Kuang, and V. Honavar, "A user similarity-based Top-N recommendation approach for mobile in-application advertising," *Expert Systems with Applications*, vol. 111, pp. 51–60, 2018.

[26] L. Li, L. Zheng, F. Yang, and T. Li, "Modeling and broadening temporal user interest in personalized news recommendation," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3168–3177, 2014.

[27] L. Li and T. Li, "News recommendation via hypergraph learning: encapsulation of user behavior and news content," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 305–314, Rome, Italy, February 2013.

[28] L. Zheng, L. Li, W. Hong, and T. Li, "PENETRATE: personalized news recommendation using ensemble hierarchical clustering," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2127–2136, 2013.

[29] M. Lu, Z. Qin, Y. Cao, Z. Liu, and M. Wang, "Scalable news recommendation using multi-dimensional similarity and Jaccard-Kmeans clustering," *Journal of Systems and Software*, vol. 95, pp. 242–251, 2014.

[30] W. J. Lee, K. J. Oh, C. G. Lim, and H. J. Choi, "User profile extraction from Twitter for personalized news recommendation," in *Proceedings of 16th International Conference on Advanced Communication Technology*, pp. 779–783, Pyeongchang, South Korea, February 2014.

[31] W. Gu, S. Dong, Z. Zeng, and J. He, "An effective news recommendation method for microblog user," *Scientific World Journal*, vol. 2014, Article ID 907515, 14 pages, 2014.

[32] J. Zheng and Y. Wang, "Personalized recommendations based on sentimental interest community detection," *Scientific Programming*, vol. 2018, Article ID 8503452, 14 pages, 2018.

[33] R. C. Bagher, H. Hassanpour, and H. Mashayekhi, "User trends modeling for a content-based recommender system," *Expert Systems with Applications*, vol. 87, pp. 209–219, 2017.

[34] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web-WWW '05*, p. 22, Chiba, Japan, May 2005.

[35] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[36] H. J. Lee and S. J. Park, "MONERS: a news recommender for the mobile web," *Expert Systems with Applications*, vol. 32, no. 1, pp. 143–150, 2007.

[37] E. Keogh and A. Mueen, "Curse of dimensionality," in *Encyclopedia of Machine Learning and Data Mining*, pp. 314-315, Springer, Berlin, Germany, 2017.

[38] G. C. Banks, H. M. Woznyj, R. S. Wesslen, and R. L. Ross, "A review of best practice recommendations for text analysis in R (and a user-friendly app)," *Journal of Business and Psychology*, vol. 33, no. 4, pp. 445–459, 2018.

[39] L. Wetzel, *Types and Tokens*, Stanford Encyclopedia of Philosophy, USA, 2018.

[40] J. Singh and V. Gupta, "A systematic review of text stemming techniques," *Artificial Intelligence Review*, vol. 48, no. 2, pp. 157–217, 2017.

[41] B. Banerjee, T. Sarkar, P. Chakraborty, and A. R. Pal, "A comparison between extrinsic and intrinsic technique for multi-document text summarization," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 768–772, Bengaluru, India, May 2017.

[42] C. Y. J. Wen, "Text categorization based on a similarity approach," in *Proceedings of International Conference on*

Intelligent System and Knowledge Engineering, Chengdu, China, October 2007.

[43] D. Forsyth, "Markov chains and hidden Markov models," in *Proceedings of Probability and Statistics for Computer Science*, pp. 331–351, Springer International Publishing, Cham, Switzerland, January 2018.

[44] J. Sun, X. Zhang, D. Liao, and V. Chang, "Efficient method for feature selection in text classification," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Antalya, Turkey, August 2017.

[45] A. Moreno and T. Redondo, "Text analytics: the convergence of big data and artificial intelligence," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, no. 6, p. 57, 2016.

[46] H. Cordobés, A. F. Anta, L. F. Chiroque, F. Pérez, T. Redondo, and A. Santos, "Graph-based techniques for topic classification of tweets in Spanish," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, no. 5, p. 31, 2014.

[47] Y. Boulid, A. Souhar, and M. Ouagague, "Spatial and textural aspects for Arabic handwritten characters recognition," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 1, p. 86, 2018.

[48] Y. Chen and X. Wang, "Text feature extraction based on joint conditional entropy," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, pp. 2055–2058, Changchun, China, December 2012.

[49] Z. Yun-tao, G. Ling, and W. Yong-cheng, "An improved TF-IDF approach for text classification," *Journal of Zhejiang University SCIENCE*, vol. 6, no. 1, pp. 49–55, 2005.

[50] F. Horn, L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Discovering topics in text datasets by visualizing relevant words," 2017, http://arxiv.org/abs/1707.06100.

[51] H. Jelodar, Y. Wang, C. Yuan et al., "Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, pp. 1–43, 2018, In press.

[52] Á. M. Navarro and P. Moreno-Ger, "Comparison of clustering algorithms for learning analytics with educational datasets," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 2, p. 9, 2018.

[53] M. Anderka and B. Stein, "The ESA retrieval model revisited," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval-SIGIR '09*, pp. 670-671, Boston, MA, USA, July 2009.

[54] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, pp. 325–341, Springer-Verlag, Berlin, Germany, 2007.

[55] R. Jacoby and B. O'Kane, "System and method for tagging content and delivering the tag to buddies of a given user," US9848246B2 Patent, 2017.

[56] M. Jalili, "A survey of collaborative filtering recommender algorithms and their evaluation metrics," *International Journal of System Modeling and Simulation*, vol. 2, no. 2, pp. 14–17, 2017.

[57] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, pp. 257–297, Springer, Berlin, Germany, 2011.

[58] G. Shani and A. Gunawardana, "Tutorial on application-oriented evaluation of recommendation systems," *AI Communications*, vol. 26, no. 2, pp. 225–236, 2013.

[59] Y. Zhao and Y. Cen, *Data Mining Applications with R*, Elsevier, Amsterdam, Netherlands, 2013.

[60] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[61] Y. Chen, Y. Lu, B. Wang, and Z. Pan, "How do product recommendations affect impulse buying? An empirical study on WeChat social commerce," *Information & Management*, vol. 56, no. 2, pp. 236–248, 2018.

[62] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber, "Offline and online evaluation of news recommender systems at swissinfo.ch," in *Proceedings of the 8th ACM Conference on Recommender systems-RecSys '14*, pp. 169–176, Silicon Valley, CA, USA, October 2014.

Hindawi
Scientific Programming
Volume 2019, Article ID 9431073, 14 pages
https://doi.org/10.1155/2019/9431073

Research Article

Design and Implementation of a Machine Learning-Based Authorship Identification Model

Waheed Anwar[1], **Imran Sarwar Bajwa**[1] **and Shabana Ramzan**[2]

[1]*Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Pakistan*
[2]*Department of Computer Science, Govt. Sadiq College Women University, Bahawalpur 63100, Pakistan*

Correspondence should be addressed to Waheed Anwar; waheed@iub.edu.pk

Received 29 October 2018; Accepted 18 December 2018; Published 16 January 2019

Guest Editor: Vicente García-Díaz

In this paper, a novel approach is presented for authorship identification in English and Urdu text using the LDA model with *n*-grams texts of authors and cosine similarity. The proposed approach uses similarity metrics to identify various learned representations of stylometric features and uses them to identify the writing style of a particular author. The proposed LDA-based approach emphasizes instance-based and profile-based classifications of an author's text. Here, LDA suitably handles high-dimensional and sparse data by allowing more expressive representation of text. The presented approach is an unsupervised computational methodology that can handle the heterogeneity of the dataset, diversity in writing, and the inherent ambiguity of the Urdu language. A large corpus has been used for performance testing of the presented approach. The results of experiments show superiority of the proposed approach over the state-of-the-art representations and other algorithms used for authorship identification. The contributions of the presented work are the use of cosine similarity with *n*-gram-based LDA topics to measure similarity in vectors of text documents. Achievement of overall 84.52% accuracy on PAN12 datasets and 93.17% accuracy on Urdu news articles without using any labels for authorship identification task is done.

1. Introduction

Stylometry is the study of distinct linguistic styles and individual writing practices with the purpose of determining the authorship of a written piece of text [1]. A writing style represents the linguistic choices of a writer that persist throughout one's work. The stylometric research is inspired by the hypothesis that every person has a unique and distinct writing style, referred to as "stylistic fingerprint" [2] that can be measured and learned. Here, a stylistic fingerprint of a writer means a set of features frequently used by that author such as word length, sentence length, choice of certain words, and syntactic structure of a sentence. The stat-of-the-art perspective of stylometry research is authorship analysis [2–7]. In the recent past, the domain of authorship analysis has embraced new dimensions of research typically with the emergence of machine learning techniques for text mining. One of the recent and emerging trends in authorship analysis is computational extraction of stylometric features from the text of an author instead of engineering the stylometric features manually [8–10]. The main focus of authorship identification is deciding the most probable author of a target document among a list of known author's [3]. From a machine learning aspect, authorship identification can be perceived as one label multiclass text classification problem where the role of classes are played by contestant authors [11].

The detailed literature review in the domain of authorship identification for the last two decades revealed that it is a field of great interest and has been mainly applied on the English language [4, 6, 12–14]. Additionally, few solitary efforts were under taken for other languages: Arabic [6, 15], Dutch [16–18], Greek [7, 19], and Portuguese [20, 21]. However, there is no major contribution in the field of authorship identification of Urdu text except for Urdu poetry [22]. To the best of our knowledge, there is neither a

theoretical model nor a subsequent accurate tool available for authorship identification of Urdu newspaper columns. Therefore, such authorship identification application for Urdu language is timely as discussed in [23]. In this paper, an improved approach is discussed that uses similarity measures as tf-idf along cosine similarity and a KNN-based classification module for more accurate results. This paper also compares the results of our approach with PAN12 dataset.

Latent Dirichlet allocation (LDA) [24] was found to be a flexible generative probabilistic unsupervised topic model typically used for the authorship identification for text documents [8, 9, 25, 26]. LDA was used with similarity measuring techniques such as Hellinger [9]. During the literature review, it has been found that the results of the previously used similarity measuring techniques provide low accuracy and there is a need in improvement in topic matching process of LDA-based author identification. In this paper, we propose a methodology for the use of *n*-grams with LDA to find similarity in vectors of text documents by using cosine distance metric. In the literature review, it was identified that the cosine similarity [27] has not been previously employed with LDA for authorship identification of the text documents. One of the objectives of the research presented in this paper was to investigate the behaviour of cosine similarity with LDA in comparison with other similar previously used techniques for authorship identification.

The presented approach builds the LDA model on *n*-grams texts instead of simple text. *N*-grams have been used to keep personal stylistic attributes of the text writer. The LDA model generates topical representation of text documents. These topical representations have been used to build cosine similarity metric for KNN classifier. Here, LDA's application on *n*-grams words not only keep various stylistic fingerprints to identify the writing style of a particular author but permits us to analyse a large dataset of Urdu newspaper articles and can identify the potential author for testing dataset. The presented approach emphasizes on author instance-based and profile-based classifications of text. We used LDA which can handle high-dimensional and sparse data, allowing more expressive representation of texts. LDA is also suitable considering the heterogeneity of the dataset, inherit ambiguity of Urdu language text, and diversity in writing styles of authors. A large dataset was collected for performance testing of the presented approach. The results of experiments show superiority of the proposed approach over the state-of-the-art representations and other algorithms used for authorship identification. The contributions of the presented work are the use of KNN classifier with cosine similarity metric extracted from LDA topics to measure similarity in vectors of text documents and achievement of satisfactory results on English and Urdu news articles without using any labels for authorship identification task.

The rest of the paper is structured as follows: Section 2 discusses the outcomes of the detailed literature. Section 3 describes the materials and methods of the presented research and the LDA-based used approach for authorship identification in PAN12 authorship identification task and Urdu newspaper articles; Section 4 provides details of the experiments, their results, and discussions; at last in Section 5, conclusion are drawn.

2. Related Work

In the literature, a large number of works in the past had been focused on computational linguistics-based methods for identification of stylometric features from text and their application to attribute possible author of the text. The focus of these approaches was to improve various tasks of authorship analysis of a piece of text such as authorship identification, author verification, and author profiling.

The first approach to authorship identification is the use of univariate or multivariate measures that can reflect the style of a particular author. Individual measures were proposed such as word occurrence or frequencies of specific word [28], mean sentence length or wge word length [29], and word richness [11]; however, none of these univariate measures prove to be adequate [30]. The idea behind the multivariate approach is to take documents as points in vector space, and by using some suitable similarity measures, assign the query document to the author, whose documents are closest to the query document [31]; furthermore, other distance-based similarity metrics such as Euclidean distance, Kullback–Leibler, and Hellinger distance were applied to various feature sets for authorship identification [4, 19, 22].

The second approach is statistical machine learning techniques. Individual author is a category value, and a classification model is built. Machine learning techniques are further separated into two subgroups, one is supervised and other is unsupervised. In supervised learning, a classifier is built using both features and the categorical value. However, unsupervised models work on unlabelled data [24]. For authorship identification, supervised techniques include support vector machine (SVM) [13, 32, 33], decision trees [6], linear discriminant analysis [34], and neural networks [35, 36]. The support vector machine outperformed other supervised techniques such as linear discriminant analysis and neural networks in terms of accuracy. Unsupervised classification techniques include principal component analysis (PCA) [37, 38], cluster analysis [39], word2vec [40], doc2vec or distributed document representation [41], and LDA [8, 9, 25, 26]. The work discussed in [23] is the first attempt to address author identification in Urdu text and that approach is improved in this paper by using tf-idf along cosine similarity and a KNN-based classification module for more accurate results.

The first systematic study of authorship identification by using enhanced version of LDA was presented by Michal Rosen-Zvi et al. [8]. The LDA model has the ability to identify all hidden topics from large numbers of features and present them as LDA topics, thus, serving for dimensionality reduction and making it attractive for text analysis problems.

We collected articles from Web of Science by applying the search query "authorship identification" in titles. The query provided 714 articles with default settings in the span of 2007 to 2018 as now we cannot get articles beyond 2007 from Web of Science. We used CiteSpace tool

(URL http://cluster.ischool.drexel.edu/~cchen/citespace/download/) to visualize patterns and trends in the authorship attribution domain. Figure 1 shows most influential authors with cited reference network.

3. Materials and Methods

In order to make the result of the present study reproducible, in this section, the main steps of our proposed framework for authorship identification are discussed; that is, English and Urdu corpus, their datasets, text preprocessing, models with their parameter settings, and experiments. The materials used are corpora in English (Table 1) and Urdu (Table 2), and datasets (Tables 3 and 4) have been generated from these corpora and the most important have been *n*-grams-based inferred topics from LDA. The relevant methods include the methods of preprocessing, various features extraction and selection, document term matrix preparation, topics extraction using latent Dirichlet allocation, and the cosine metric for KNN classifier-based methodology for classification.

We used a publicly available dataset in English from the authorship identification competition of the PAN 2012 [42]. The competition included six tasks for authorship identification for both close and open classes and two tasks for intrinsic plagiarism. Close class means the author of a test or an anonymous text is among the closed set of candidate authors of training data, and open class means the author of a test document might be none of the closed set of candidates. The task notation and description are listed in Table 1. The PAN12 dataset has training and testing parts for closed-class and open-class authorship identification problems.

In the PAN 2012 competition, the training data were extremely small with only two documents per author provided for training. The length of documents varies: in tasks A to D, short documents were given having words in range 2,000 to 13,000 words, while tasks I and J dealt with long documents containing approximately 30,000 to 160,000 words. Tasks E and F were related to plagiarism detection and are out of scope for the present study. From the PAN12 competition, there were two types of tasks on the bases of testing documents: the first one is author-dependent recognition where all the authors of the testing documents were among the training documents authors and the second one is author-independent recognition where some testing documents were from unknown authors which were not part of training documents authors. We only selected author-dependent tasks (A, C, and I); however, author-independent tasks were not in scope of the present study.

The UrduCorpus has 4,800 documents written by twelve well-known Urdu newspaper columnists with 400 articles per author. It contains 5,631,850 words (tokens) in total; at the document level, the mean length was 1174 words. The longest document was written by Dr Muhammad Ajmal Niazi (2,170 words) and the shortest by Irshad Ansari (86 words). When considering the mean length per author, Irshad Ansari wrote the shortest document (396 words per document), while Javed Chaudhary is the author of the longest document (1,690 words per document).

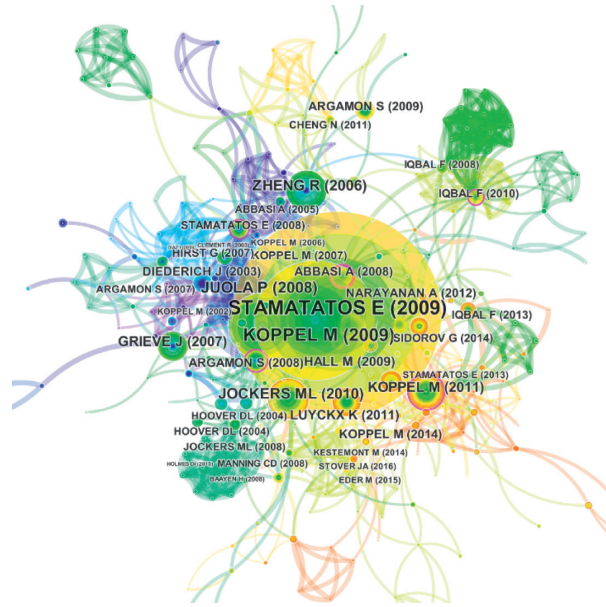


Figure 1: Network of most influential authors in the authorship identification domain.

3.1. Datasets. In the preparation of the datasets from PAN12 and UrduCorpus, we used two representations of author-specific documents.

3.1.1. Instance-Based Dataset. In this type of representation, all documents were treated individually.

3.1.2. Profile-Based Dataset. All the author-specific documents were concatenated into a single file. This single document represents an individual author.

We prepared 12 datasets from PAN12 as shown in Tables 2 and 3; among these datasets, six were instance-based as original text and *n*-grams representation of text and six were profile-based as original text and *n*-grams representation of text.

The number of training and testing documents in the profile-based datasets are equal, whereas in the instance-based datasets, training documents are double than testing for model evaluation.

Similarly, we prepared four datasets from UrduCorpus as shown in Table 3. In Table 3, among these datasets, two were instance-based as original text and *n*-grams representation of text and two were profile-based as original text and *n*-grams representation of text. We randomly used 75% and 25% of data by each author for training and testing, respectively.

Two profile-based datasets of UrduCorpus have only twelve lengthy documents for training, and each dataset has equally twelve hundred test documents for model evaluation.

Figure 2 depicts the proposed framework for authorship attribution using the topic modelled with LDA with cosine metric for the KNN classifier.

Table 1: PAN 2012 authorship identification competition tasks.

Task name	Type	Authors	Descriptions
A	Authorship identification	3	Closed class, short text
B	Authorship identification	3	Open class (of task A), short texts
C	Authorship identification	8	Closed class, short texts
D	Authorship identification	8	Open class (of task C), short texts
I	Authorship identification	14	Closed class, novel length texts
J	Authorship identification	14	Open class (of task I), novel length texts
E	Intrinsic plagiarism	2–4	Mixed set of paragraphs by individual author
F	Intrinsic plagiarism	2	Consecutive intrusive paragraphs by individual author

Table 2: PAN12 datasets used in the experiments.

Dataset	Description	Training documents	Testing documents
A_1	Instance-based (original text)	6	6
A_2	Instance-based (*n*-grams)	6	6
A_3	Profile-based (original text)	3	6
A_4	Profile-based (*n*-grams)	3	6
C_1	Instance-based (original text)	16	8
C_2	Instance-based (*n*-grams)	16	8
C_3	Profile-based (original text)	8	8
C_4	Profile-based (*n*-grams)	8	8
I_1	Instance-based (original text)	28	14
I_2	Instance-based (*n*-grams)	28	14
I_3	Profile-based (original text)	14	14
I_4	Profile-based (*n*-grams)	14	14

Table 3: Urdu datasets used in the experiments.

Dataset	Description	Training documents	Testing documents
D_1	Instance-based (original text)	3600	1200
D_2	Instance-based (*n*-grams)	3600	1200
D_3	Profile-based (original text)	12	1200
D_4	Profile-based (*n*-grams)	12	1200

3.2. Document Preprocessing. It is observed from the literature review that it is not needed for vigorous preprocessing in authorship attribution. As writer's grammatical mistakes, their preferences of letter abbreviation, letter capitalization, and word prefixes and suffixes all are essential part of one's writing style. In this case, it is not feasible to correct grammatical mistakes or stem words, such actions may reduce the number of features specific to writer.

3.2.1. Tokenization. Tokenizing means to change sentences into small units like words and characters. We used Natural Language Toolkit (NLTK) [43] for tokenizing at word-level after ignoring all whitespaces.

3.2.2. Lowercasing. Languages such as English have uppercase and lowercase texts. It is recommended to lowercase them before any further preprocessing. We applied this process on the PAN12 dataset. In the Urdu language, we only have one case, so no need to change it into any other.

3.2.3. N-Grams Generation. *N*-gram is a grouping of adjacent words or characters of length *n*. We can generate these *n*-grams for any language. In other words, *n*-grams features are language independent. They can capture the language structure of a writer; for instance, what character or word was anticipated to follow the given one. The choice of *n* is very vital in *n*-grams generation. If the value of *n* is small which produces short *n*-grams, we may fail to capture important differences. On the contrary, if the value of *n* is large, it produces long *n*-grams; as a result, we may only stick to specific cases. Ideal *n*-grams length really depends on the application, a good rule of thumb in word level *n*-grams is to use *n*-grams where $n \in \{1, \ldots, 5\}$. To overcome the limitation of the bag of the word model where the contextual information is lost, we can capture more semantically meaningful information from text with *n*-grams. Lexical *n*-grams are popular, as they have shown to be more effective than character *n*-grams [44] and syntactic *n*-grams when all the possible *n*-grams are used as features [45]. Moreover, it has been shown to be effective in identifying the gender of tweeters [46]. For ease of understanding, we used underscores (_) to replace spaces in

Table 4: *N*-grams (1–5) for sentence "writing is footprint of a writer."

N-grams types	Sentence representation
Word unigrams	Writing, is, footprint, of, a, writer
Word bigrams	Writing_is, is_footprint, footprint_of, of_a, a_writer
Word trigrams	Writing_is_footprint, is_footprint_of, footprint_of_a, of_a_writer
Word fourgrams	Writing_is_footprint_of, is_footprint_of_a, footprint_of_a_writer
Word fivegrams	Writing_is_footprint_of_a, is_footprint_of_a_writer

word *n*-grams and represent them as a single word in the vocabulary and subsequently in the bag of the word model. Table 4 shows a simple sentence and its complete lists of unigrams, bigrams, trigrams, fourgrams, and fivegrams words generated from it.

For word-level *n*-grams feature vector length varies as choice of *n* varies, it can increase rapidly almost *n*-times with *n*-grams.

3.2.4. Stop Word Removal. Stop words are common words in a given language which has high-frequency in the text of language. For instance, in English words like a, an, the, this, and for are stop words. Stop words generally have minor lexical content, and they have enormous presence in the text document. However, they fail to distinguish it from other texts. Sometimes, we want to remove these words from the document before further processing. In languages such as English, we have stop words list; however, in the Urdu language, we do not have such list. We add constraint that each selected word should not appear in every document. Thus, we want to ignore stop words appearing in almost every document. We ignore all words occurring in 70 percent or more documents. Taking into account this constrain, we ignore 666 most frequent words.

3.2.5. Stemming. Stemming is the process of extracting the base word from the given word. This base word is called the stem or root word. We used a rule-based stemmer with the help of Stanford coreNLP tools to stem datasets words.

3.3. Syntax Analyzer and Feature Extraction. Extracting numerical information from raw text documents is normally termed as feature extraction process. Among extracted features, only those features are selected that best fit the training model. After this process, if the features set dimensionality is huge and difficult for computation, then it requires dimensionality reduction algorithms for appropriate performance. The following feature extracting techniques were used for the proposed model.

3.3.1. TF-IDF. We can produce distinct feature vectors based on information captured from the texts. This could be simply raw frequency of each word or term frequency and inverse document frequency (tf-idf). We can use tf-idf to

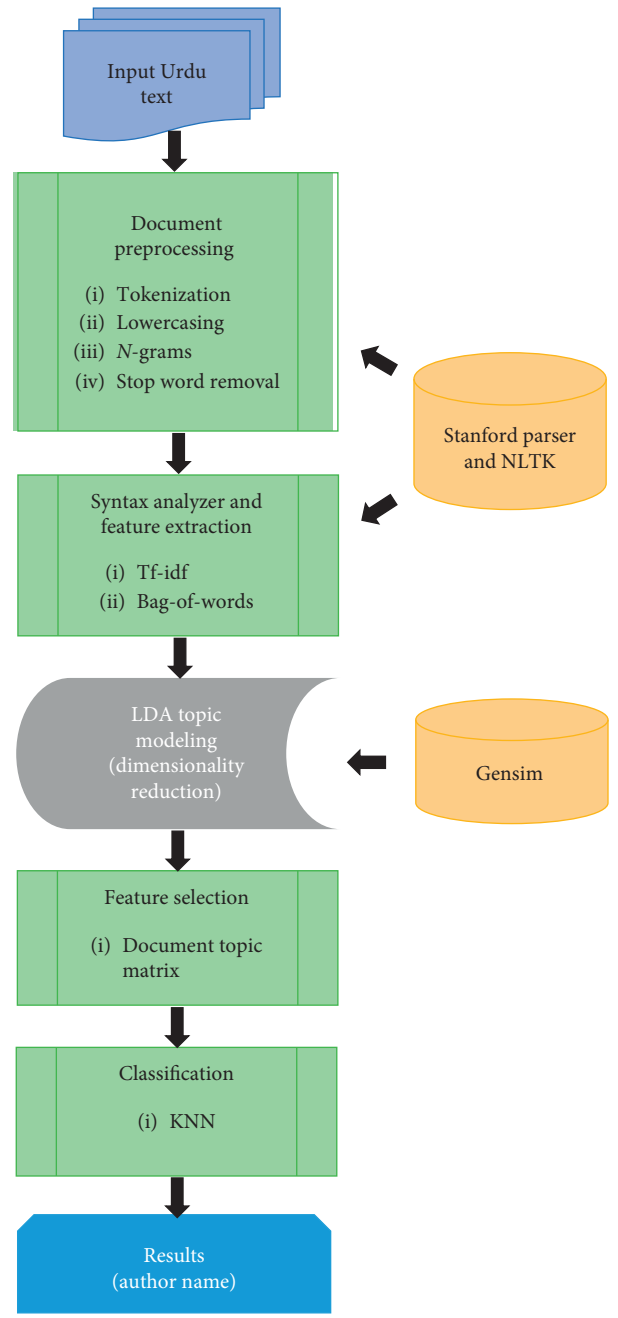


Figure 2: Architecture of text-based forensic analysis approach.

assess how significant a word is in identifying the actual author for a given document. This tf-idf value can be obtained by multiplying the ratio of the word in a document to the reciprocal of the ratio used in the all documents.

3.3.2. Bag of Words Extraction. In natural language processing, bag-of-words is a classic model. In this model, text is considered as a set of words each one having a frequency of occurrence in the corpus; however, their contextual information is lost. In other words, it is order less document features representation in the form of frequencies that occurs in the document to form a dictionary, and this

dictionary may consist of character, character n-grams, words, words n-grams, or some other features extracted from text. If we use all distinct words for our vocabulary, it can increase overall corpus dimensionality which is difficult for computation. For feature selection, we have applied two schemes.

(1) Term Document Frequency. We have been considering only those terms having occurrence in 10 or more documents (tdf $\geq$ 10), and it reduced vocabulary size to 104,867 terms in instance-based n-grams dataset of UrduCorpus.

(2) Percentage of Documents for Term. As a second constrain, we wanted to remove stop words or other most frequent words from corpus. We ignore all words occurring in 90 percent or more documents. Taking into account this second constrain, we ignore 666 most frequent words.

Finally, we obtain a vocabulary of size 104,201 terms in instance-based n-grams dataset D_2, similar feature selection schemes were applied on simple instance and profile-based datasets we obtained vocabulary of size 44,634 terms and for profile-based n-grams dataset applying slightly different feature selection schemes as there has been only twelve long concatenated documents. We have selected only those terms which occurred in two or more documents, however, not more than ten documents. We capture 55,423 terms for vocabulary.

For PAN12 English datasets, training documents words and distinct words for each dataset are given in Table 5.

We applied different feature selection techniques as the training documents for each author were only two in each dataset. First, we selected topmost frequent 2,500 words for each author, and then ignoring the common words which other authors also used, we only selected author specific words. In this way, stop words were ignored and only those words were captured with which were author specific. We also add the second constraint that, among these author specific words, only top 300 most frequent words for each author were selected to build balance vocabularies for A1, A_3, C_1, C_3, I_1, and I_3 datasets. For datasets A_2, A_4, C_2, C_4, I_2, and I_4 having n-grams words, we selected top 500 most frequent words for each author to build vocabularies of equal size with respect to authors.

3.4. Document Term Matrix. Text documents are generally represented as a vector, where each attribute represents particular term frequency occurrence. This vector form representation can be used to find the similarity between the two corresponding documents. We prepared document term matrix (Figure 3) from training dataset based on the selected features which were saved in the form of vocabulary by using Gensim dictionary class. The LDA model looks for repeating term patterns in the entire document term matrix.

3.5. Feature Selection Using LDA. We can use topic models for the purpose of information retrieval and feature selection from unstructured text. A topic modelling algorithm, for example, latent Dirichlet allocation [24], is useful for organizing large volume of textual data into overlapping clustering of documents [24, 47], which differ from other text mining approaches, which are rule-based and use dictionary or regular expressions-based keyword searching. LDA is a flexible generative probabilistic topic model for collection of discrete data, which expresses the documents as a collection of topics mixture with different probabilities for these topics in documents, and each topic is expressed as a list of words with probabilities for them to belong to that topic.

Table 5: PAN12 datasets used in the experiments.

Dataset	Training documents words	Distinct words	Vocabulary size
A_1	25,771	3,252	900
A_2	128,845	48,012	1,500
A_3	25,771	3,252	900
A_4	128,845	48,012	1,500
C_1	96,052	26,654	2,400
C_2	480,250	133,256	4,000
C_3	96,052	26,654	2,400
C_4	480,250	133,256	4,000
I_1	2,353,267	137,315	4,200
I_2	11,766,325	7,839,471	7,000
I_3	2,353,267	137,315	4,200
I_4	11,766,325	7,839,471	7,000

We have used LDA to reduce the dimension of document term matrix into a new matrix (Figure 3). We named new matrix as document topic matrix, as each cell represents specific topic weight in that document and each matrix row now have topical representation of whole document in the normalized form.

This representation achieved dimensionality reduction of the document term matrix. For example, for Urdu dataset, reduction of D_2 document term matrix from $3600 \times 104{,}201$ to document topic matrix 3600×120 is achieved, which is almost 91% less of the vocabulary of the corpus. This result is extremely helpful in features selection and classification of documents.

3.6. Hyper Parameters and Parameters of LDA. LDA has corpus-level parameters named hyperparameters α and β sampled only once, and these parameters are from the Dirichlet distribution. The first parameter α controls the distribution of document topics, and β is accountable for distribution of topic words. The high value of α means each document possibly contains mixture of almost every topic not a particular one, while low value of α means that the document contains some topics. Similarly, high β means that each topic contains a mixture of most of the words not just specific words. The low value of β means the topic may represent a blend of just some of the words. In a nutshell, high α will produce documents more identical to each other and high β will produce topics more identical to each other. However, the number of topics k is user defined, and we need to figure out the number of topics based on the data. Thus, each document d_i, for $i = 1, \ldots, n$, is generated based

	Term1	Term2	Term3	…	Term104201
Doc1	2	0	1	…	0
Doc2	3	3	0	…	1
Doc3	2	1	4	…	1
⋮	⋮	⋮	⋮	⋱	⋮
Doc3600	1	0	2	…	0

Dimensionality reduction using LDA

	Topic1	Topic2	Topic3	…	Topic120
Doc1	0.0630	0.0318	0.0930	…	0.0630
Doc2	0.1392	0.0000	0.0426	…	0.1392
Doc3	0.0000	0.0165	0.0515	…	0.0010
⋮	⋮	⋮	⋮	⋱	⋮
Doc3600	0.0007	0.0005	0.0000	…	0.0515

Figure 3: Conversion of the document term matrix to the document topic matrix.

on a distribution over the k topics, where k defines the maximum number of topics. This value is fixed and defined a priori; a lower value for the number of topics may result in border topics such as education, sports, and fashion, and a larger value for the number of topics may result in more focused topics such as science, football, and hairstyle. A large k value for topics means that the algorithm requires lengthier passes to estimate the word distribution for all the topics, and a good rule of thumb is to choose a value that makes sense for a particular case; in the present context, we may consider $k = 12$ at least assuming that each k value corresponds to the individual author writing style and thus choosing $k = 12$ was a sensible choice, and a larger k value than 12 was required to imitate the versatility of the writing style of a particular author as two or more topic distributions were more helpful in this regard. However a lower k value than 12 does not make any sense. We used k between 12 and 120 with the interval of 10.

3.7. LDA + Cosine Similarity. This method is our main contribution, as it achieves state-of-the-art performance in authorship identification with many candidate authors. The main idea of our approach is to use the LDA model in such a way that it provides us dimensionality reduction along with maintaining the author specific writer style and then use cosine similarity in LDA model topic space, to determine the most likely author of the test document. We used n-grams to capture the author writing style. Documents were represented as bag-of-words, so each document from both training and test sets converted into sparse vector and were mapped into LDA topic space to generate a vector representation for each one, which can be represented as u_i and v_i as outcomes, respectively.

In text similarity measures, cosine similarity is one of the most popular one. It is a distance metric from computational linguistics to measure similarity between document vectors. In order to find cosine similarity between two documents u and v, first we need to normalize them to one in L_2 norm:

$$\sum_{i=1}^{k} u_i^2 = 1. \tag{1}$$

Now, cosine similarity between these two normalized vectors u and v will be the dot product of them:

$$\cos(u, v) = \frac{\sum_{i=1}^{k} u_i v_i}{\sqrt{\left(\sum_{i=1}^{k} u_i^2\right)}\sqrt{\left(\sum_{i=1}^{k} v_i^2\right)}}, \tag{2}$$

where u_i and v_i are the vectors of n dimensions over the document sets $\mathbf{u}$ and $\mathbf{v}$ where $i = 1, 2, 3, \ldots, k$.

Cosine similarity is considered as the one of the best in similarity measurement. Cosine similarity is very simple in implementation complexity as in Gensim [48]. We also used different evaluation metrics in order to validate and compare our results.

3.8. Classification. Text documents are generally represented as a vector where in a document each attribute represents particular term frequency occurrence. This term vector form representation is used to find the similarity between the two corresponding documents. We can apply KNN to our data that will learn to classify new articles based on their distance to our known articles (and their labels). The algorithm needs a distance metric such as Euclidian distance or cosine similarity to determine which of the known articles are closest to the new one. We used cosine similarity.

4. Results

In our experiments, we validated the proposed authorship identification scheme by performing tests on twelve datasets of PAN12 and four datasets of UrduCorpus. In order to build the low-dimensionality topical representation, the LDA model receives tokenized text documents with n-grams of the training set without any label (without the author to which they belong) as input data type and for evaluation the unlabeled text documents from the testing set. The experiments comprised in testing a cosine base classifier with the

output of LDA k-topics in the corpus, and these topics form a lower-dimensional representation of the corresponding training set based on vocabulary and then evaluating the classifier with the testing set using the same lower-dimensional representation. The overall authorship identification accuracy rate (AR) is computed by the following equation:

$$\text{accuracy rate (AR)} = \frac{\text{number of correctly identified articles}}{\text{total number of test set articles}} \times 100. \quad (3)$$

4.1. Experimental Setup. All the experiments were performed to test the performance and accuracy of the proposed approach using Intel i7 @ 2.8 GHz, operating on windows 10 pro 64-bit with 6 GB memory. Python 3 (Python Software Foundation, Wilmington, DE, USA), NLTK [43], and LDA implementation in Gensim [48] library have been used for the development of the system. LDA implementation in Gensim allows both estimation of topics distribution on training data and inference of these topics on the test data. We used UrduCorpus dataset (Table 3) that belongs to the news domain. Note that change of newspaper may affect the writing style of an author, and similarly over the passage of time, the individual writing style may also change. The nature of articles (their topics) also influences the choice of words; however, every individual has his/her own vocabulary, and he/she may like to use specific words unintentionally which can be used for his/her writing style identification.

To evaluate and compare LDA for authorship identification, we used PAN12 datasets from small datasets having 3 authors and 12 documents, medium datasets having 8 authors and 24 documents, and large datasets of novel length documents with 14 authors and 42 documents and for Urdu dataset, having 4,800 documents written by twelve well-known Urdu newspaper columnists. We used various performance metrics (precision, recall, and F_1-measure) along with accuracy to demonstrate the quality of autodecision-making of cosine-based KNN classifier on PAN12 and UrduCorpus.

4.2. Results and Discussion. In order to validate the results, we evaluated LDA-based authorship identification approach on instance and profile-based datasets with and without n-grams, and we carried out a series of experiments on each dataset with several filters on the term frequency and frequent words removal to generate vocabulary with most appropriate features and different number of LDA topics (12, 24, 36, . . ., 120) for UrduCorpus and LDA topics 3 to 54 for different datasets of PAN12. We presented each experiment with best performance parameter setting for PAN12 as shown in Table 6.

Our results on PAN12 datasets depicts that LDA with n-grams performed better as compare to simple text. When we compared instance-based n-grams with profile-based n-grams, the results were the same, as we have used identical vocabulary in each comparative dataset. Secondly, here we have used balanced number of documents and also the features extracted from these documents were also equal, therefore, in most of the cases, instance-based results were the same as compared to relevant profile-based results. Overall best performance on A, C, and I datasets was 84.53%.

For Urdu datasets, we reported experiments with best performance parameter setting in Table 7.

Our results clearly show that LDA instance-based n-grams approach outperforms LDA profile-based approach significantly, although we were hoping vice versa as mentioned in the literature [9]. In profile-based approach when documents were concatenated into single file to form the author profile, some important authorship features lose their prominence in the profile, and these features have significant discriminating power that sharply contrasts documents between the authors. Secondly, although we have used balanced corpus in terms of the number of documents, the average document length per author varies, so when concatenating documents into the author profile, some profiles have a smaller number of words in total as compared to those of others resulting in unbalanced feature extraction, whereas in the instance-based approach, some documents of an author were long enough to become strong candidate of attributed document. Thirdly, in instance-based approach, different features can be combined easily, whereas in profile-based approach, it is difficult to do so.

We have reported the accuracy percentage yielded by the LDA + cosine similarity approach, in LDA model, setting the number of topics k between (12, 24, 36, . . . , 120) with various vocabulary sizes. Our result shows that varying the number of topics in the LDA model is critical and it has a huge impact on performance. Usually, accuracy increases with the number of topics in a certain range and then begins to decrease. A clear and precise prescription for this parameter is not possible, even in the same dataset with different vocabulary sizes.

In order to evaluate the proposed LDA-based approach on four datasets, we used the same number of topics with identical vocabulary size initially; however, the results were not satisfactory for couple of datasets, as with combination of n-grams document size increases in terms of tokens and length in the dataset, and thus in these datasets, we cannot use the same vocabulary size for each LDA model. We tuned LDA models with different vocabulary sizes keeping the same k topics. We have reported the best performance of each dataset with different vocabulary sizes but the same number of topics between 12 and 120 in the current context, and we may assume that each topic at least matches to the writing style of an author, and thus, fixing $k = 12$ is a reasonable choice. However, the value of k could be larger than 12 representing the fact that each author may require two or more topical representations to well describe the style of a given author. When applying the LDA model on instance-based n-grams dataset with a vocabulary of 104,201 terms and LDA 60 topics, we achieved an accuracy of 93.17% with KNN classifier setting of $k = 7$. Hence, evaluations reported in this graph indicate that the LDA-based authorship

Table 6: Unsupervised classification of documents based on LDA topics with cosine similarity on twelve datasets of pan12.

Dataset with description	Parameters	Accuracy rate (%)
A_1 instance-based (original text)	Vocabulary 900, $k = 6$	83.3
A_2 instance-based (n-grams)	Vocabulary 1500, $k = 6$	**100**
A_3 profile-based (original text)	Vocabulary 900, $k = 3$	83.3
A_4 profile-based (n-grams)	Vocabulary 1500, $k = 3$	**100**
C_1 instance-based (original text)	Vocabulary 2400, $k = 16$	50.0
C_2 instance-based (n-grams)	Vocabulary 4000, $k = 16$	**75.0**
C_3 profile-based (original text)	Vocabulary 2400, $k = 8$	62.5
C_4 profile-based (n-grams)	Vocabulary 4000, $k = 8$	**75.0**
I_1 instance-based (original text)	Vocabulary 4200, $k = 28$	64.3
I_2 instance-based (n-grams)	Vocabulary 7000, $k = 28$	**78.6**
I_3 profile-based (original text)	Vocabulary 4200, $k = 14$	64.3
I_4 profile-based (n-grams)	Vocabulary 7000, $k = 14$	**78.6**

Table 7: Unsupervised classification of documents based on LDA topics with cosine similarity on four datasets of UrduCorpus.

Method and dataset	Parameters	Accuracy rate (%)
LDA instance-based (original text)	Vocabulary 44,634, $k = 24$	91.42
LDA instance-based (n-grams)	Vocabulary 104,201, $k = 60$	**93.17**
LDA profile-based (original text)	Vocabulary 44,634, $k = 60$	91.83
LDA profile-based (n-grams)	Vocabulary 55,423, $k = 72$	91.75

attribution model performs significantly better on instance-based n-grams dataset than other datasets almost on each k topics selection. Note that to further elaborate the results in the following, we used proposed model with instance-based n-grams dataset.

Figure 4 depicts the result of multiple experiments that compare the unsupervised classification of documents based on LDA topics with the KNN classifier on four datasets.

Figure 5 depicts the result of multiple experiments that compare the unsupervised classification of documents based on LDA topics with cosine similarity on PAN12 datasets.

In order to evaluate the proposed LDA-based approach on PAN12 datasets, we used the number of topics k between 3 and 70 depending upon the number of authors and theirs documents with various vocabulary sizes (Table 5). We cannot use the same vocabulary size for each LDA model. We tuned LDA models with different vocabulary sizes keeping dataset specific k topics (Figure 6). We reported the best performance of each dataset with different vocabulary sizes and number of topics between 3 and 70; in the current context, we may assume that each topic at least matches to the writing style of an author thus fixing k equal to 3 for dataset A, 8 for dataset C, and 14 for dataset I is a reasonable choice. However, the value of k could be larger than 3, 8, and 14, respectively, representing the fact that each author may require two or more topical representation to well describe the style of a given author.

When applying the LDA model on A_2 and A_4 datasets with the vocabulary of 1,500 terms and k values of 6 and 3 topics, respectively, we achieved an accuracy of 100% with cosine similarity. On dataset C, we found a best accuracy of 75% with datasets C_2 and C_4 with the vocabulary of 4,000 terms and k values 16 and 8, respectively. Similarly, for I dataset, we found the best accuracy of 78.57% with I_2 and I_4 with vocabulary of 7,000 terms and k values of 28 and 14, respectively. These results clearly indicate that our approach works well both on instance-based and profile-based approaches and on instance-based approach model, and it required a greater number of k topics as compared to the profile-based approach because in profile-based approach, all author specific documents were concatenated and that is why, it required less number of topics. For instance-based approach, the LDA model achieved best results when the k topics were equal to training documents because in PAN12, training documents were limited to only two per author as compared to UrduCorpus where the training documents were 300 per author; therefore, here we assumed that each document represents only one topic. Hence, evaluations reported in these graphs indicate that the LDA-based model performs almost the same on instance-based n-grams dataset and profile-based n-grams dataset however with different k topics. The same Urdu dataset was used to further elaborate the results of the used model with instance-based n-grams datasets.

Figure 6 shows the confusion matrix obtained with proposed methodology on 1200 test documents.

This confusion matrix can be used for various performance measures which can evaluate our results in different ways. As we can see, there is a clear diagonal heatmap which represents the accuracy with respect to the author; however, there were some documents which were misclassified. Three out of twelve authors have at least six misclassified documents towards single author; for instance, ten documents for actual author number eight were misclassified towards predicated author number five which shows some resemblance of one's writing styles. One notable result was that authors with maximum accuracy also did not have any misclassified document in their favor, which shows their unique writing style.

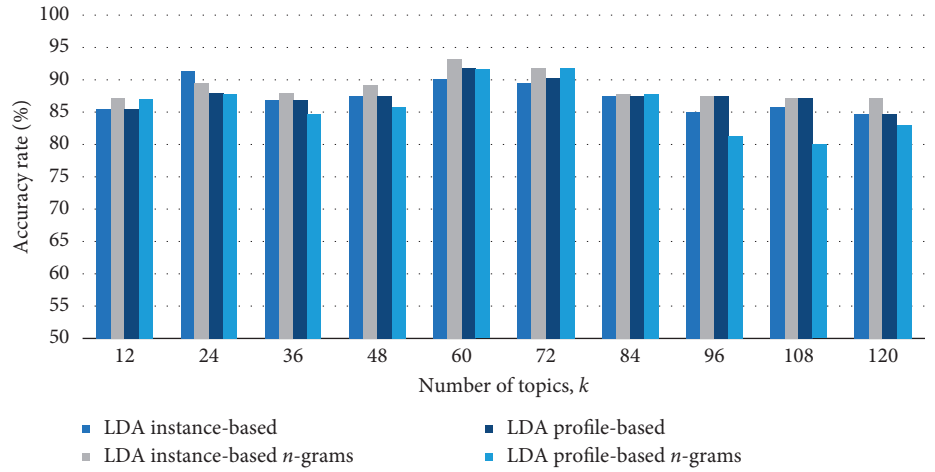


Figure 4: Classification of documents based on LDA topics with KNN classifier on four datasets.

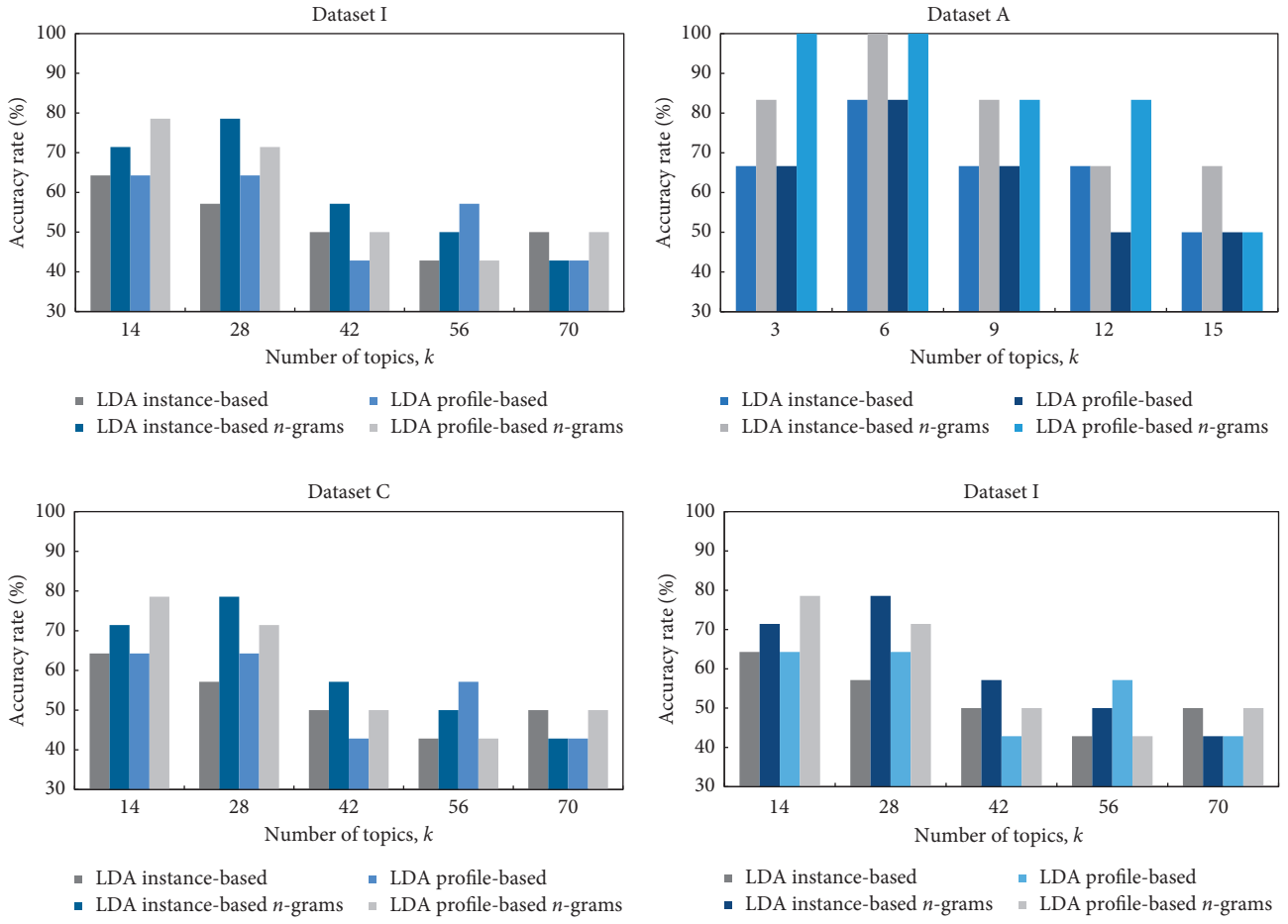


Figure 5: Evaluation of LDA authorship identification using KNN on PAN12 datasets.

Figures 7–9 show the confusion matrix obtained with proposed methodology on PAN12 datasets A, C, and I test documents.

As we can see, from these confusion matrixes of instance-based n-grams datasets A, C, and I of PAN12, there is a clear diagonal heatmap which represents the accuracy with respect to the author; however, there were some documents which were misclassified in C and I datasets. Documents of two out of eight authors were misclassified.

4.3. Interpretation of Misclassified Articles. There can be a number of reasons for misclassification of articles. Firstly, we

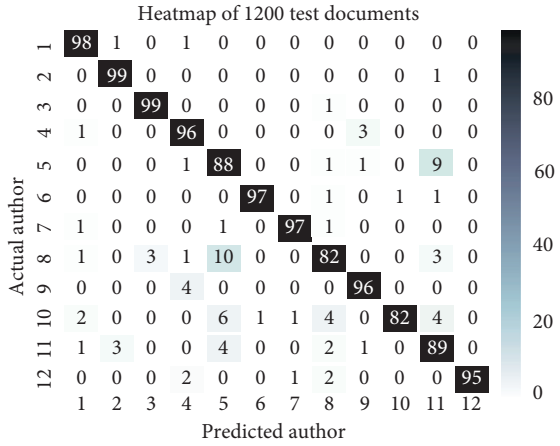


Figure 6: Confusion matrix for test documents of UrduCorpus using instance-based *n*-grams approach.

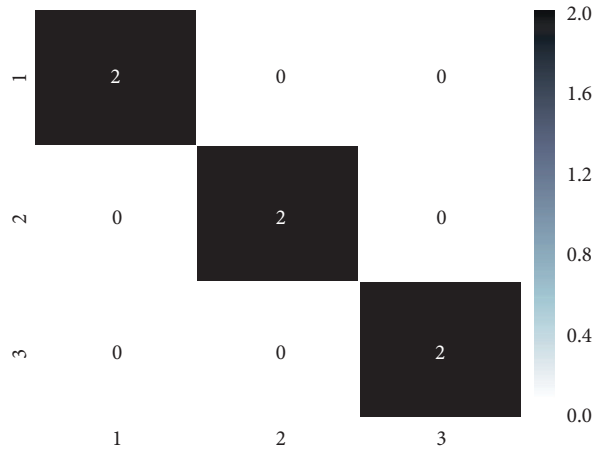


Figure 7: Confusion matrix for test documents of A dataset using instance-based *n*-grams approach.

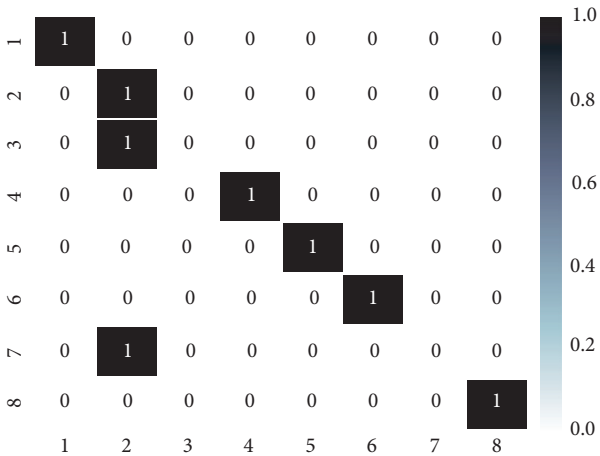


Figure 8: Confusion matrix for test documents of C dataset using instance-based *n*-grams approach.

found that few authors have the writing style such that, in their articles, first they gave quoted paragraphs of other authors and then discussed their point of view on that topic.

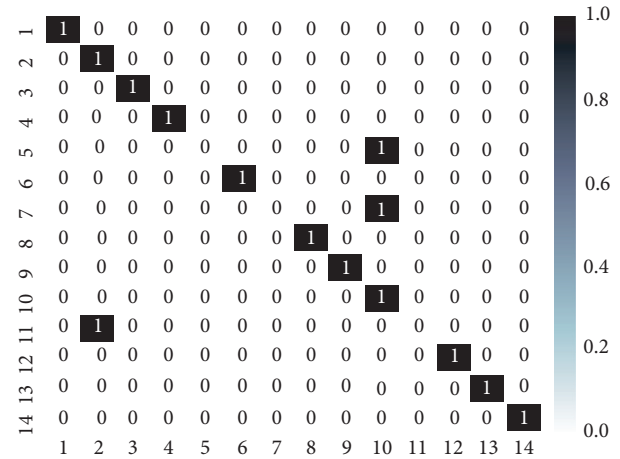


Figure 9: Confusion matrix for test documents of I dataset using instance-based *n*-grams approach.

In this way, they intermingle their writing style with other authors. Secondly, authors wrote on various domains like politics, religion, sports, and entertainment as the corpus was not domain specific.Our proposed scheme may model an author in consequence of a document in respect to any other author in the specific domain that may result in misclassification. Thirdly, short size of the testing article may be the cause of misclassification.

In Table 8, we reported individual class results in terms of precision, recall, F_1-measure, and accuracy rate obtained on instance-based *n*-grams dataset by applying the proposed scheme for authorship identification.

The experiment shows our approach models the authors more accurately on *n*-grams instance-based dataset. We achieved 93.17% accuracy rate on this dataset, and other performance measures were also satisfactory, as precision measure was fluctuating from 81% to 100% and recall measure was between 82% and 99% on individual basis of this dataset. As there is a tradeoff between precision and recall, we attained 93.1% precision and 92.9% recall on 1200 test documents of the instance-based *n*-grams dataset.

In Tables 9–11, we reported individual class results in terms of precision, recall, F_1-measure, and accuracy rate (percentage of correct answers) obtained on instance-based *n*-grams datasets A, C, and I of PAN12 by applying the proposed scheme for authorship identification.

We achieved 84.52% of an average accuracy rate on PAN12 instance-based and *n*-grams datasets A_2, C_2, and I_2, and other performance measures were also satisfactory, as average precision measure was 80%, recall measure was 84.67%, and F_1-measure was 80.67% on these datasets.

5. Conclusions

In this paper, we designated the authorship identification problem in Urdu news articles and English PAN12 tasks in the context of the closed-class problem. As a new authorship identification scheme, we proposed an approach using latent Dirichlet allocation (LDA) paradigm in conjunction with *n*-

Table 8: Unsupervised classification of author documents on instance-based n-grams Urdu dataset.

Authors	Precision	Recall	F_1 measure
Asad Ullah Ghalib	0.94	0.98	0.96
Dr. Muhammad Ajmal Niazi	0.96	0.99	0.98
Dr. Tauseef Ahmad Khan	0.97	0.99	0.98
Haroon Ur Rashid	0.91	0.96	0.94
Irshad Ahmad Arif	0.81	0.88	0.84
Irshad Ansari	0.99	0.97	0.98
Javed Chaudhary	0.98	0.97	0.97
Karnal R Ikram Ullah	0.87	0.82	0.85
Khursheed Nadeem	0.95	0.96	0.96
Nawaz Raza	0.99	0.82	0.90
Nazeer Naji	0.83	0.89	0.86
Qayyum Nizami	1.00	0.95	0.97
Average	0.931	0.929	0.930

Table 9: Unsupervised classification of author documents on instance-based n-grams PAN12 dataset A.

Authors	Precision	Recall	F_1 measure
candidate00001	1	1	1
candidate00002	1	1	1
candidate00003	1	1	1
Average	1.00	1.00	1.00

Table 10: Unsupervised classification of author documents on instance-based n-grams PAN12 dataset C.

Authors	Precision	Recall	F_1 measure
candidate00001	1	1	1
candidate00002	0.33	1	0.5
candidate00003	0	0	0
candidate00004	1	1	1
candidate00005	1	1	1
candidate00006	1	1	1
candidate00007	0	0	0
candidate00008	1	1	1
Average	0.67	0.75	0.69

grams to produce reduced dimension topical representation of documents. We explained how the topical representations of LDA could be used with cosine distance metric for classification of test documents. Our approach yields satisfactory performance. The best result in terms of accuracy and F_1-measure were achieved with n-grams introduction in the model which captures more stylistic features of an author. The lessons learned were that each language required different configurations at each stage, appropriate selection of the dimensionality of the representation is crucial for authorship identification, and it is possible to significantly improve the accuracy results by fine tuning the size of vocabulary and k topics in LDA.

One possible improvement to the study would be the implementation of the supervised learning model to get good accuracy. This would increase the effort of annotating the corpus. Secondly, we could train the model developed in the study, on a larger set of columnists. One could aim to design and deploy an automated website scraper incorporated with the proposed LDA model to collect other such online articles and create a comprehensive database of all such columnists. By doing so, it could probably help authorship identification on a larger scale.

Table 11: Unsupervised classification of author documents on instance-based n-grams PAN12 dataset I.

Authors	Precision	Recall	F_1 measure
candidate00001	1	1	1
candidate00002	0.5	1	0.67
candidate00003	1	1	1
candidate00004	1	1	1
candidate00005	0	0	0
candidate00006	1	1	1
candidate00007	0	0	0
candidate00008	1	1	1
candidate00009	1	1	1
candidate00010	0.33	1	0.5
candidate00011	0	0	0
candidate00012	1	1	1
candidate00013	1	1	1
candidate00014	1	1	1
Average	0.70	0.79	0.73

Data Availability

The implementation and datasets used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors thank Muhammad Omer for his technical guidelines.

References

[1] D. I. Holmes, "The evolution of stylometry in humanities scholarship," *Literary and Linguistic Computing*, vol. 13, no. 3, pp. 111–117, 1998.

[2] D. I. Holmes, "Authorship attribution," *Computers and the Humanities*, vol. 28, no. 2, pp. 87–106, 1994.

[3] P. Juola, "Authorship attribution," *Foundations and Trends in Information Retrieval*, vol. 1, no. 3, pp. 233–334, 2006.

[4] C. E. Chaski, "Empirical evaluations of language-based author identification techniques," *Forensic Linguistics*, vol. 8, no. 1, pp. 1–65, 2001.

[5] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 83–94, 2010.

[6] A. Abbasi and H. Hsinchun Chen, "Applying authorship analysis to extremist-group Web forum messages," *IEEE Intelligent Systems*, vol. 20, no. 5, pp. 67–75, 2005.

[7] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Text genre detection using common word frequencies," in *Proceedings of the 18th Conference on Computational Linguistics*, vol. 2, p. 808, Saarbrücken, Germany, August 2000.

[8] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence*, pp. 487–494, Banff, Canada, 2004.

[9] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with latent dirichlet allocation," in *Proceedings of Fifteenth Conference on Computational Natural Language Learning*, pp. 181–189, Portland, OR, USA, 2011.

[10] A. Caliskan-Islam, "Stylometric fingerprints and privacy behavior in textual data," , ProQuest Diss. Thesis, 2015.

[11] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[12] D. I. Holmes, M. Robertson, and R. Paez, "Stephen crane and the New-York tribune: a case study in traditional and non-traditional authorship attribution," *Computers and the Humanities*, vol. 35, no. 3, pp. 315–331, 2001.

[13] M. Koppel, J. Schler, and S. Argamon, "Computational methods in authorship attribution," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 1, pp. 9–26, 2009.

[14] M. Kestemont, "Overview of the author identification task at PAN-2018: cross-domain authorship attribution and style change detection," *CEUR Workshop Proceedings*, vol. 2125, 2018.

[15] A. S. Altheneyan and M. E. B. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 4, pp. 473–484, 2014.

[16] P. Juola and R. H. Baayen, "A controlled-corpus experiment in authorship identification by cross-entropy," *Literary and Linguistic Computing*, vol. 20, no. 1, pp. 59–67, 2005.

[17] J. Hoorn, S. Frank, W. Kowalczyk, and F. van der Ham, "Neural network identification of poets using letter sequences," *Literary and Linguistic Computing*, vol. 14, no. 3, pp. 311–338, 1999.

[18] P. Maitra, S. Ghosh, and D. Das, "Authorship verification – an approach based on random forest notebook for PAN at CLEF 2015," in *Proceedings of Working Notes for CLEF 2015 Conference*, pp. 1–9, Toulouse, France, September 2015.

[19] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "*N*-Gram-based Author profiles for authorship attribution," in *Proceedings of the conference Pacific Association for Computational Linguistics*, pp. 255–264, Halifax, NS, Canada, 2003.

[20] D. Pavelec, L. Oliveira, E. Justino, and L. Batista, "Using conjunctions and adverbs for author verification," *Journal of Universal Computer Science*, vol. 14, no. 18, pp. 2967–2981, 2008.

[21] R. Sousa Silva, G. Laboreiro, L. Sarmento, T. Grant, E. Oliveira, and B. Maia, "'twazn me!!! ;(' automatic authorship analysis of micro-blogging messages," in *Proceedings of International Conference on Application of Natural Language to Information Systems*, vol. 6716, pp. 161–168, Salford, UK, 2011.

[22] A. A. Raza, A. Athar, and S. Nadeem, "*N*-gram based authorship attribution in Urdu poetry," in *Proceedings of the Conference on Language & Technology*, pp. 88–93, Poznań, Poland, 2009.

[23] W. Anwar, I. Sarwar Bajwa, M. A. Choudhary, and S. Ramzan, "An empirical study on forensic analysis of Urdu text using LDA based authorship attribution," *IEEE Access*, vol. 6, p. 6600, 2018.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 993–1022, 2003.

[25] R. Arun, R. Saradha, and V. Suresh, "Stopwords and stylometry: a latent Dirichlet allocation approach," *NIPS Work*, pp. 1–4, 2009.

[26] J. Savoy, "Authorship attribution based on a probabilistic topic model," *Information Processing & Management*, vol. 49, pp. 341–354, 2013.

[27] S. Sohangir and D. Wang, "Document understanding using improved sqrt-cosine similarity," in *Proceedings-IEEE 11th International Conference on Semantic Computing, ICSC*, pp. 278-279, San Diego, CA, USA, 2017.

[28] T. C. Mendenhall, "The characteristic curves of composition," *Science*, vol. 9, pp. 237–246, 1887.

[29] G. Yule, "The statistical study of literary vocabulary," *Modern Language Review*, vol. 39, no. 3, pp. 291–293, 1944.

[30] J. Grieve, "Quantitative authorship attribution: an evaluation of techniques," *Literary and Linguistic Computing*, vol. 22, no. 3, pp. 251–270, 2007.

[31] J. Burrows, "'Delta': a measure of stylistic difference and a guide to likely authorship," *Literary and Linguistic Computing*, vol. 17, no. 3, pp. 267–287, 2002.

[32] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler, "Automatically profiling the author of an anonymous text," *Communications of the ACM*, vol. 52, no. 2, pp. 119–123, 2009.

[33] E. Stamatatos, "Author identification: using text sampling to handle the class imbalance problem," *Information Processing & Management*, vol. 44, no. 2, pp. 790–799, 2008.

[34] C. E. Chaski, "Who' s at the keyboard? Authorship attribution in digital evidence investigations," *International Journal of Digital Evidence*, vol. 4, no. 1, pp. 1–13, 2005.

[35] F. J. Tweedie, S. Singh, and D. I. Holmes, "Neural network applications in stylometry: the Federalist Papers," *Computers and the Humanities*, vol. 30, no. 1, pp. 1–10, 1996.

[36] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: writing-style features and classification techniques," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378–393, 2006.

[37] J. F. Burrows, "Word-patterns and story-shapes: the statistical analysis of narrative style," *Literary and Linguistic Computing*, vol. 2, no. 2, pp. 61–70, 1987.

[38] A. Jamak, A. Savatić, and M. Can, "Principal component analysis for authorship attribution," *Business Systems Research*, vol. 3, no. 2, pp. 49–56, 2012.

[39] D. I. Holmes, "A stylometric analysis of mormon scripture and related texts," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 155, no. 1, pp. 91–120, 1992.

[40] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of International Conference on Machine Learning*, pp. 1–12, Atlanta, GA, USA, 2013.

[41] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of International Conference on Machine Learning*, pp. 1188–1196, Beijing, China, June 2014.

[42] P. Juola, "An overview of the traditional authorship attribution subtask," CLEF (Online work. Notes/Labs/Workshop), pp. 37–41, 2012, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.636.449&rep=rep1&type=pdf.

[43] S. Bird and E. Loper, "NLTK: the natural language toolkit," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 1–4, Barcelona, Spain, July 2004.

[44] I. Markov, E. Stamatatos, and G. Sidorov, "Improving cross-topic authorship attribution: the role of pre-processing," in *Proceedings of 18th Computational Linguistics and Intelligent Text Processing*, Budapest, Hungary, April 2017.

[45] H. Ding, I. Takigawa, H. Mamitsuka, and S. Zhu, "Similarity-based machine learning methods for predicting drug-target interactions: a brief review," *Briefings in Bioinformatics*, vol. 15, no. 5, pp. 734–747, 2013.

[46] J. D. Burger, J. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on twitter," *Association for Computational Linguistics*, vol. 146, pp. 1301–1309, 2011.

[47] M. Omar, B.-W. On, I. Lee, and G. S. Choi, "LDA Topics : representation and evaluation," *Journal of Information Science*, vol. 41, no. 5, pp. 1–14, 2015.

[48] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the Workshop New Challenges for NLP Frameworks*, pp. 45–50, Valletta, Malta, May 2010.

Hindawi
Scientific Programming
Volume 2018, Article ID 1890683, 10 pages
https://doi.org/10.1155/2018/1890683

Research Article

A Low-Cost Named Entity Recognition Research Based on Active Learning

Han Huang,[1] **Hongyu Wang**,[2] **and Dawei Jin**[1]

[1]*School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China*
[2]*School of Information Management, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Dawei Jin; jdw@zuel.edu.cn

Received 10 August 2018; Accepted 28 November 2018; Published 18 December 2018

Guest Editor: Vicente García-Díaz

Named entity recognition (NER) is an indispensable and very important part of many natural language processing technologies, such as information extraction, information retrieval, and intelligent Q & A. This paper describes the development of the AL-CRF model, which is a NER approach based on active learning (AL). The algorithmic sequence of the processes performed by the AL-CRF model is the following: first, the samples are clustered using the *k*-means approach. Then, stratified sampling is performed on the produced clusters in order to obtain initial samples, which are used to train the basic conditional random field (CRF) classifier. The next step includes the initiation of the selection process which uses the criterion of entropy. More specifically, samples having the highest entropy values are added to the training set. Afterwards, the learning process is repeated, and the CRF classifier is retrained based on the obtained training set. The learning and the selection process of the AL is running iteratively until the harmonic mean F stabilizes and the final NER model is obtained. Several NER experiments are performed on legislative and medical cases in order to validate the AL-CRF performance. The testing data include Chinese judicial documents and Chinese electronic medical records (EMRs). Testing indicates that our proposed algorithm has better recognition accuracy and recall rate compared to the conventional CRF model. Moreover, the main advantage of our approach is that it requires fewer manually labelled training samples, and at the same time, it is more effective. This can result in a more cost effective and more reliable process.

1. Introduction

With the continuous popularization of Internet and mobile Internet and the continuous improvement of information infrastructure in various domains, the available digital resources have grown explosively in our metaindustrial societies [1]. On one hand, the sources and the volume of information have become more abundant. The density of useful data is decreasing, which makes it more difficult to mine valuable information. In the era of big data, it is difficult for people to manually analyze and filter information, due to their high volume and variety. Automatic or semiautomatic effective extraction from a large number of digital resources could lead to the mining of hidden knowledge. This could be achieved with the help of big data and artificial intelligence technologies like NER [2, 3]. NER is a process of recognizing and classifying words or phrases with special characteristics or meanings in a text. It belongs to the category of unsigned word recognition in lexical analysis, and it is an indispensable part of information extraction and retrieval, intelligent Q & A, and other natural language processing technologies [4].

The important role of NER in natural language processing has motivated a lot of research in the domains of library information and computer science, and it has resulted in the proposal of several methods. However, the categories and extensions of named entities significantly vary under different research scenarios and domains [5]. More specifically, the named entities (NAE) mainly refer to names of persons, places, and time. In the biomedical field, they can include medical terms such as protein, genome, or labels of diseases. In legislative domains, the NAE may include legal concepts, terms, or provisions. Obviously, the specification of NER depends on the field of study, and it is difficult to migrate directly [6]. Moreover, in the case of the domain-specific NER, such as legislative, ancient Chinese poetry and so on, the domain entities are relatively scarce, that is, the directly available training data are a minority. At

the same time, due to the high specialization of the data, the domain knowledge must be rich when annotating the domain texts manually. The necessary professional talents and the heavy workload require a lot of manpower and available resources. Therefore, the use of less annotated data to train an effective NER model has become an important goal of the domain NER research [7].

Active learning algorithms can be used to solve the problem of sparse data annotating, in machine learning. AL consists of the learning and the selection modules. The learning module is the process of training high-quality classifiers, while the selection module is the process of generating training sets from large amounts of data [8]. The core of AL is to use the learning algorithm in order to get the most useful data from the training set and then to add this data to the manually developed annotation set. In this way, we can get a classifier with a strong generalization ability, without requiring a high volume of annotated data [9]. This paper applies active learning to the field of NER. AL is the main algorithmic framework, and CRF is the corresponding classifier. We propose a new approach called AL-CRF, aiming not only to improve the efficiency recognition of the CRF model, but also to decrease the number of annotated training samples as well. The testing experiments on medical and legislative fields have proven that our proposed method can produce a more efficient NER model with fewer training samples, which can effectively cut the cost of manual annotation and improve the overall efficiency.

2. Background

2.1. Named Entity Recognition. NER has been defined as an important subtask of information extraction in the MUC-6 conference [10]. The most commonly used relative approaches can be divided into three categories: rule and dictionary-based methods, statistical, and mixed ones. Early NER mainly used rule and dictionary-based methodologies, which require the design and development of the rule sets by domain experts and the use of proper linguists. Nevertheless, the rules fail to cover all linguistic phenomena, the construction period is too long, and moreover, this approach has a lower likelihood of portability [7]. Statistical methods mainly include hidden Markov models (HMM) [11], maximum entropy (ME) [12], support vector machines (SVM) [13], and CRF [14]. These kinds of methods use the labelled corpus data to train the model combined with statistical probability. They are easily transplantable and they have comparatively short construction periods although they have more strict requirements for feature selection and much more dependence on the corpus. The mixed methods model is a combination of rules, dictionary-based approaches, and statistical ones, and they combine the advantages of both. They employ rules to filter the target text in advance, and they are reducing the state of the search space, based on statistical methods. Recently, some hybrid methods have been introduced, based on deep learning (DL), which combines DL with rules or statistical approaches [15]. At the same time, driven by the demand of natural language processing in various fields, the recognition objective of NER has also evolved from the initial person name, location name, and time to the words or phrases with special meanings in the recognition text. Researchers have also carried out NER research mainly for specific domain entities, such as fishery data [16], dietary data [17], and Chinese legal documents data [18]. CRF model is one of the most popular ones.

2.2. Conditional Random Field. CRF is an undirected graph model proposed by Lafferty in 2001, which combines the characteristics of the ME and the HMM, and it considers the transition probabilities between contextual markers at the same time. The transition probability between tags is optimized and decoded in the serialization form, and the sequence data annotation is carried out by establishing the probability model [19]. CRF has strong reasoning ability and is widely used in sequential tagging tasks, such as part of speech tagging [20], significance testing [21], and new word discovery [22]. NER is also a special kind of sequence tagging problem, and the CRF has innate advantages in solving it. When applied to NER, the CRF has good stability, accuracy, and ease of use [23]. However, as a typical supervised model, it requires a lot of training data, and the convergence speed is slow. To solve these problems, many researchers combine other machine learning algorithms with CRF in an effort to improve its performance. Such efforts have been made by Deng et al. [24] who have proposed a short-term traffic flow forecasting model (MCRF) which is based on multiconditional random fields. It uses four kinds of feature functions to build multiple CRF feature subsets to reflect the multicumulative characteristics of traffic data. Xia et al. [25] have combined convolutional neural networks (CNN) with CRF, and they have proposed a hybrid classification approach for remote sensing images. These methods greatly improve the availability and effectiveness of the CRF model, but the training process is still inseparable from a large number of annotated data.

2.3. Active Learning. Active learning is a branch of machine learning (ML) that belongs to the area of artificial intelligence. It was originally proposed by Angluin of Yale University [26]. The learning module needs to continuously improve the classification accuracy and robustness of the classifier, and the purpose of the selection modules is to find out the most representative and extensive training data. Current research on active learning can be summarized in two aspects. On one hand, researchers have applied it to many fields. Wu et al. [27], Zhu et al. [28], Pohl et al. [29], have introduced the application of AL algorithm in social media data, spatial data annotation and image classification respectively. On the other hand, many researchers have put forward the idea of improving it. Wang et al. [30] have proposed a new multi-instance AL algorithm by combining diversity criterion with existing information measure. Patra et al. [31] have proposed the LAAL-ELM which is an online continuous learning method. Through the confidence meter of newly added data, this method selects the tagging set actively to update the classifier, and it reduces the

computational complexity. Active learning provides an algorithmic framework to solve the problem of sparse annotating data in the training process of ML. In practical research, machine learning algorithms are used as a classifier of active learning. Through the iteration of the learning and selection processes, the performance of the classifier keeps improving continuously.

To sum up, although NER has been developed and used for more than 20 years, the problem of this field has not been completely solved, due to the continuous diffusion of the named entity denotations in different scenarios and domains. In previous research efforts, CRF has been one of the most widely used approaches. However, its training requires a lot of annotated data. Though it is difficult to obtain annotated data in a specific field, AL can effectively solve this problem, as it is capable to find high-value data in order to train high-performance classifiers. Therefore, this paper combines active learning with the CRF model. It uses the CRF classifier, and it proposes a method to recognize the NAE which enriches the method of named entity recognition. At the same time, the training process requires a small amount of annotated data, which is very significant to the application areas where the annotated data is rare and the annotation is a hard task, e.g., in medical and legal cases.

3. Materials and Methods

3.1. CRF Model. The CRF model is a model that outputs the conditional probability of a random variable Y with a given random variable X. This model has various forms including the linear chain form, the matrix form, and so on. In the NER process, the CRF model is usually further simplified, that is, the random variables X, Y have the same graph structure, which is shown in Figure 1. X is the input text to be recognized, and $x_1, x_2, \ldots, x_{n-1}, x_n$ are sequences after word segmentation and feature tagging. The task of the CRF model is to predict the conditional probability of Y by training the model parameters, and the calculation method is shown in equation (1):

$$P(y \mid x) = \frac{1}{Z(x)} \exp \left(\sum_{i,k} \lambda_k t_k (y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l (y_i, x, i) \right), \quad (1)$$

where $Z(x)$ is a normalization factor, whereas t_k is an eigenfunction defined on the edge k, which is called transfer feature. It depends on the current position and on the previous position. Also, s_l is an eigenfunction defined on the node l which is called state feature, and it depends on the current position. The parameters λ_k and μ_l are the weights corresponding to t_k and s_l. The value of t_k and s_l is either 1 or 0. When the characteristic condition is satisfied, the value is 1, otherwise it is 0.

In this research, the observation set x is the sequence set that comprises a text corpus after the word segmentation and feature automatic annotation, and y is the annotation type corresponding to the observation set x. In the feature

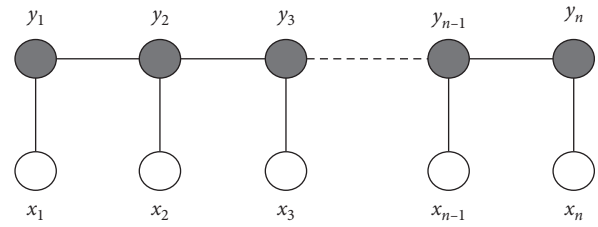


Figure 1: The structure of CRF model.

model construction, we use a 5-word tagging set that is expressed as $P = \{B, I, E, S, O\}$. Tag B is an entity starting word, and E represents the end of the entity. All of the entity is tagged as I except from the beginning and the final words. S represents the entity as a single word, and O is expressed as a word outside the entity.

3.2. Active Learning Algorithm. In this paper, the algorithmic process of the AL is shown in Figure 2. Firstly, the initial samples for training the basic CRF model are selected by following a certain strategy, and they are annotated by domain workers. Thereafter, according to the information, the CRF model is trained, the unlabelled samples are sorted according to certain ranking rules, and the top N samples are selected for manual annotating. Then the annotated data are added to the training set to retrain the CRF model. The learning process and selecting process are carried out iteratively until the exit condition is satisfied. Obviously, three key problems have to be solved in the AL process. First, the construction of the initial training set; second, the choice of the proper strategy for sample selection; and finally, the effective setting up of the iterative process and the quit condition.

The initial training set is used to train the benchmark classifier in the active learning algorithm. Therefore, selecting the representative initial training set can train the benchmark classifier with good recognition result, which would reduce the number of iterations and could accelerate the convergence process. Random Sampling is the basic algorithm for the construction of the initial training set. However, due to its limited size, the samples selected by this approach are considered less representative. On the other hand, the clustering method can aggregate samples with similar characteristics, so that the stratified sampling method based on the clustering results is more likely to choose the most representative samples.

In active learning, sample selection strategies (SSS) can be divided in two types, namely, the stream-based SSS and the pool-based one. The learning process of the stream-based SSS requires the processing of all unlabelled samples, which increases the query cost. In addition, since it requires presenting the sample annotation conditions in advance, it does not have good applicability [32]. The pool-based strategy is that selecting the sample with the highest contribution from the sample pool at a time, which reduces the query cost of the sample, so it is more widely used.

AL is a process which iteratively selects high-value samples for model training, in order to improve the efficiency of the classifiers. Although the increase in the number

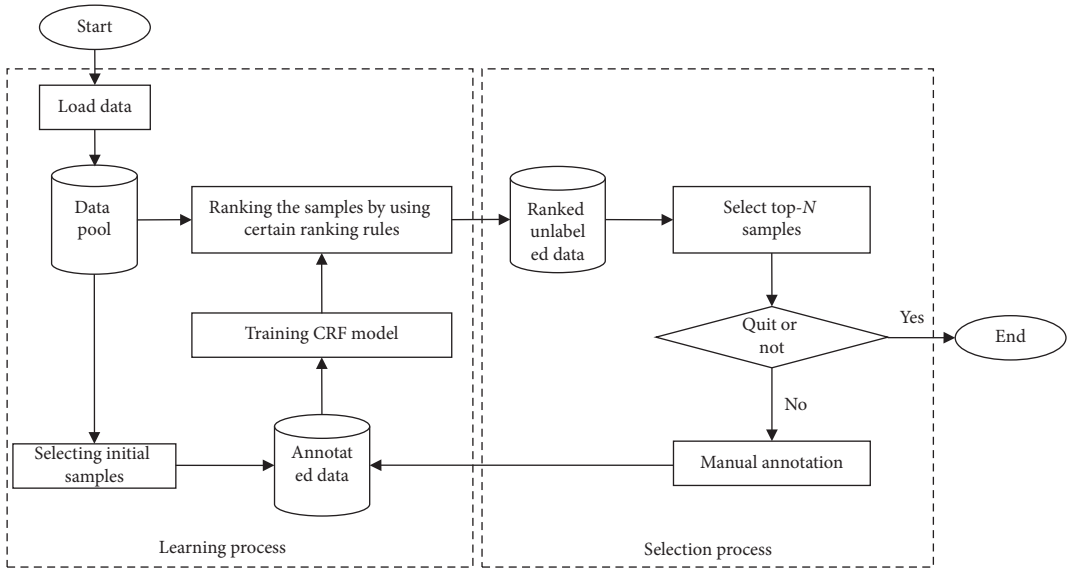


Figure 2: The process of active learning.

of iterations can improve the efficiency of the classifier, it also increases the workload of both the sample selection and the sample annotation. The training process strives to achieve the balance between sample labelling cost and classifier efficiency. Therefore, in general, the active learning algorithm will terminate the iteration when the model efficiency reaches the present precision or the number of samples reaches the given threshold value.

3.3. AL-CRF Model. The AL-CRF model takes active learning as the basic framework and the CRF as the basic classifier of AL. A hierarchical sampling method based on k-means clustering is used to select the training set for the initial learning. An SSS based on information entropy is adopted to select the samples for the iteration process. The quit condition of the iteration is based on a defined change rate of the F-value. The algorithm framework is shown in Figure 3.

In order to construct a more representative initial training set, the AL-CRF model uses the TF-IDF algorithm to vectorize the text data. It also employs the k-means algorithm to cluster the data, and it stratifies the data after clustering. The whole algorithmic process is described as follows:

(1) Vectorizing and normalizing data using TF-IDF to get the dataset X after loading the corpus.

(2) Choosing the number of K samples from the data set X randomly as C.

(3) Calculating the Euclidean distance between the remaining samples in X and C and classifying (assigning) the remaining samples in X, to the nearest cluster according to the distance.

(4) Calculating the mean value of each cluster and updating the original clustering center after all samples are divided.

(5) Comparing the new center with the previous clustering center. If there is no change, it will terminate; otherwise, go to step 2.

(6) Outputting the final clustering results.

(7) The initial sample set T was selected by stratified sampling according to the clustered results.

The AL-CRF model chooses a pool-based sample selection strategy, and it uses information entropy (IE) as a measure to evaluate the sample value based on uncertainty criteria. IE is a measurement unit used to measure the amount of information. The higher the value of the IE is, the more information is contained in the sample. This indicates that the classifier has not determined the proper category. Through the iterative process, the model predicts the sequence of IE values of the remaining samples, by employing the existing classifier. In this paper, the IE value of the sample is the sum of the IE value of each word in the sample, and the calculation method is shown in equations (2), (3), and (4):

$$H(x_j) = -\sum_i p(y_i \mid x_j) \log p(y_i \mid x_j), \quad (2)$$

$$H(s_t) = \sum_j H(x_j), \quad (3)$$

$$H(d_k) = \sum_t H(s_t). \quad (4)$$

where $H(x_j)$ represents the entropy value of word x_j, $p(y_i \mid x_j)$ indicates that the label belongs to the possibility of y_i under the given word x_j, $H(s_t)$ represents the entropy value of sentence s_t, and $H(d_k)$ represents the entropy value of the document d_k.

The AL-CRF model sets the change rate of F-value less than or equal than 0.1% as the iterative termination condition of the active learning. This means that $F_t - F_{t-1} \leq 0.1\%$, where F_t represents the F-value of the model in the t iteration, F_{t-1} represents the F-value of the model in the $t-1$ iteration, and F_0 represents the initial expression. The default value is zero. This is done in order to control the sample selection and to mark the cost of the training process.

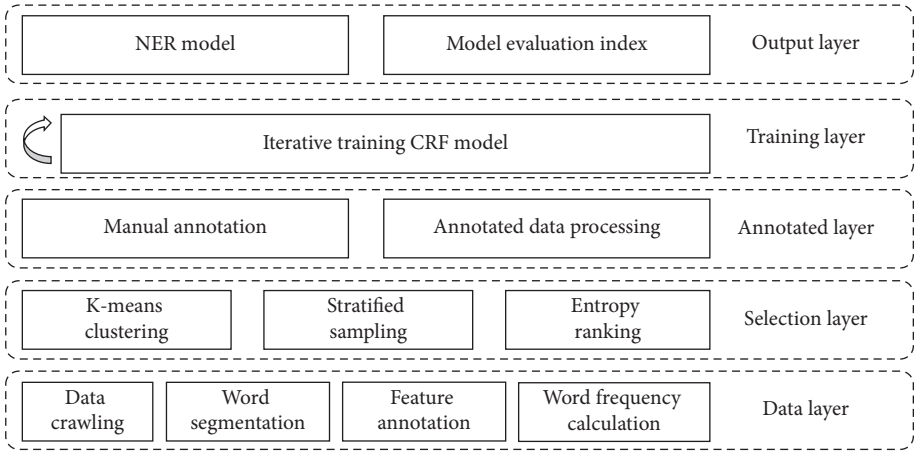


Figure 3: The algorithm framework of AL-CRF.

In summary, the AL-CRF model can be represented by the use of pseudocode as shown in the following Algorithm 1. The core of the algorithmic process is shown in Figure 4.

4. Experiment

In order to verify the effectiveness of the NER AL-CRF model and its performance in different domains, this paper selects two different datasets in the medical and the legislative fields. This completes the NER experiment.

4.1. Dataset. In the process of NER in the medical field, this paper uses the EMRs dataset which was released by the China Conference on Knowledge Graph and Semantic Computing (CCKS) 2017. To ensure the correctness of this experiment, we have selected only 300 EMRs that have been correctly annotated, whereas each EMR is divided into four parts, namely, "history characteristics," "hospital discharge," "general items," and "diagnosis and treatment." The total volume of the experimental data is 1,200 text numbers and 5 categories of entities. Overall, the total number is 23,719, and the distribution of categories is shown in Table 1.

In the legislative domain experiment, this paper obtained 61,515 copies of the judicial documents stored in the "Chinese justice document network." After clearing up the duplicates, blanks, and noncontent data, we got 59,788 valid data, containing 52,995 first-instance documents, 5,632 second-instance documents, 325 retrial documents, 37 penalty changes, and 799 documents in other categories. In this paper, 1,000 pieces of judicial documents have been extracted and manually annotated in the form of stratified sampling, which can be used as the corpus of the legislative NER experiment to reduce the cost of manual labelling. There are 73,217 legal entities in the corpus, including 5 categories of crime, penalty, legal principle, legal concept, and legal provision. The distribution is shown in Table 2.

Since there is no delimiter in Chinese itself, Chinese word segmentation is the basis of the data analysis. In order to improve the accuracy of the word segmentation, this paper attempts to construct the professional dictionaries in both medical and legislative fields by using disease symptoms, treatment technologies, crimes, legal institutions, and legal words obtained from the Internet. Then, we import them into the NLPIR segmentation tool of the Chinese Academy of Sciences and cut the words of EMRs and of the judicial documents, respectively.

After the completion of the manual annotation of the corpus, the program has been used to tag the POS and the length of each word automatically and to record whether it is left or right boundary word. According to the annotation method of 5-word tagging sets, the format of the corpus is shown in Table 3.

4.2. Experimental Design. In this paper, CRF and AL-CRF models have been used to recognize entities in medical and legislative domain for EMRs and judicial documents, respectively. Ten crossover trials have been conducted in the specific experiment, where the training and the testing sets have been determined based on a ratio of 9 : 1. The CRF++ has been used as a tool for training and evaluating CRF models and the Spark platform has been selected for text quantization and text clustering.

In the AL-CRF experiment, the initial corpus of the active learning model has been set to 100 copies, the sample size has been increased to 50 per each round of iteration, and the growth rate of the harmonic mean F has a value less than 0.1% as the iterative termination condition. In the CRF experiment, random sampling has been used to select the equivalent of the AL-CRF documents of the training set. Then the test data have been used to evaluate the effects of the two models respectively.

5. Results and Discussion

5.1. The Evaluating Indicator. According to the common indicator system of NER, we selected the following evaluation indicator, involving precision rate (P), recall rate (R), and F-measure (F) [18]. The calculation method is shown in equations (5), (6), and (7):

```
Initialization: unlabelled dataset U, F_0 = 0, i = 0, t = 0, initialization data number n, additional number N in iteration
// k-means clustering
select cluster centers randomly as C_i
do
  for u in U do
    for c in C_i do
      if dis(u, c) is min_u then
        the cluster of u is c
      end
    end
  end
  update C_i to C_{i+1}
while C_i! = C_{i+1}
output the clustered dataset S
select n samples from S by stratified sampling
annotate n samples into T
train CRFs by T
t⟵t + 1
calculate the F-value of CRFs as F_t
while F_t − F_{t−1} > 0.1% do
  calculate entropy in {S − T}
  rank the {S − T} according to the entropy
  annotate top N samples into T
  train CRFs by T
  t⟵t + 1
  calculate the F-value of CRFs as F_t
end
```

Algorithm 1: The pseudocode of AL-CRF.

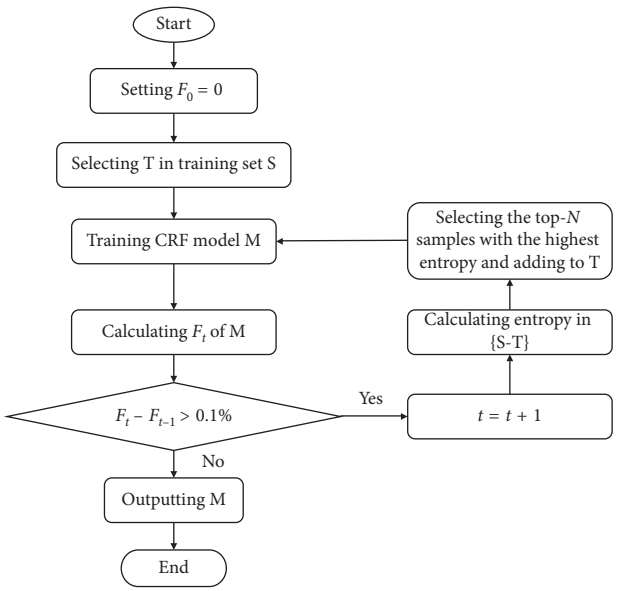


Figure 4: The flowchart of AL-CRF.

$$P = \frac{A}{A + W} * 100\%, \tag{5}$$

$$R = \frac{A}{A + U} * 100\%, \tag{6}$$

$$F = \frac{2 * P * R}{p + R} * 100\%, \tag{7}$$

where A represents the number of entities recognized correctly, W indicates the number of entities that are recognized by mistake, and U represents the number of entities that are not recognized at all.

5.2. Clustering Number Selection. In order to determine the number of k-means clustering, we have conducted 5 rounds of experiments on the EMRs data and on the judicial documents, respectively. Each experiment divided the training and testing sets according to the ratio of 4 : 1 to conduct clustering experiments.

Since the number of clusters is not too large, each round of experiments contains 14 clustering experiments, with the number of clusters ranging from 2 to 15. The sum of squared errors (SSE) of each round in the EMR experiments is shown in Figure 5.

According to the general principle of the elbow method, when the number of clusters is 5 and 8, the SSE value is lower. In order to further determine the number of clusters, we select the initial sample set which is selected by stratified sampling with 5 and 8 clusters to train the initial recognition model. Additionally, we use three alternatives as control groups. In the first one, we select the samples randomly; in the second mode, we employ stratified sampling with 2

Table 1: The entity distribution of the EMRs.

Category	Body(B)	Symptoms and signs (SS)	Examination and inspection (EI)	Disease and diagnosis (DD)	Treatment (T)
Number	8,282	6,941	6,903	657	936

Table 2: The entity distribution of the judicial documents.

Category	Charge(C)	Penalty(P)	Legal principle (LP)	Legal concept (LC)	Law(L)
Number	1,745	4,732	209	63,820	2,711

Table 3: The format of the corpus.

Word	POS	Length	Is left	Is right	Tag
无	V	1	N	N	O
发热	Vi	2	Y	Y	SS-S
,	Wd	1	N	N	O
时	Ng	1	N	N	O
有	Vyou	1	N	N	O
咳嗽	Vi	2	Y	Y	SS-S
、	Wn	1	N	N	O
咳	V	1	Y	N	SS-B
痰	n	1	N	Y	SS-E
,	Wd	1	N	N	O
无	V	1	N	N	O
胸	Ng	1	Y	N	SS-B
闷	V	1	N	Y	SS-E
、	Wn	1	N	N	O
气	N	1	Y	N	SS-B
短	a	1	N	Y	SS-E

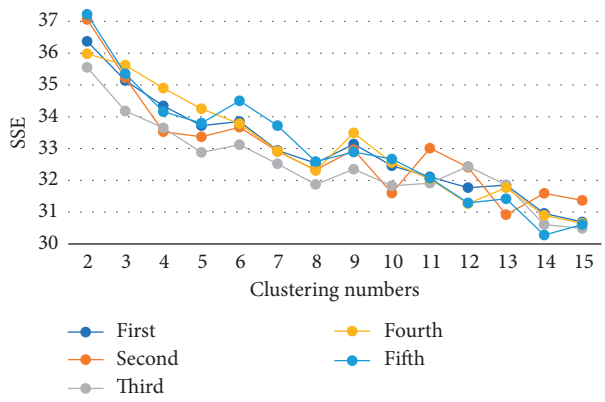


Figure 5: The SSE of each round in EMR experiments.

clusters; and in the third, we adopt again stratified sampling with 14 clusters. This is done in order to compare the influence of the number of clusters on the selection of the initial training set. The average recognition effect of the above experiments is shown in Table 4.

By analysing the results in Table 4, it can be seen that selecting the initial sample set after clustering the initial sample into the appropriate number of categories can improve the accuracy and the recall rate of the initial model. When the number of clusters is 8, the recognition efficiency of the model is the best. Therefore, for EMRs, the optimal number of clusters is 8. According to the same experimental procedure, the optimal number of clusters is 10 for the legislative judicial documents.

Table 4: The average recognition effect of initial model in EMRs.

Selection method	P	R	F
Random	0.8100	0.7658	0.7873
2 clusters	0.8115	0.7534	0.7813
5 clusters	0.8053	0.7707	0.7876
8 clusters	0.8174	0.7767	0.7965
14 clusters	0.8039	0.7682	0.7856

5.3. The Evaluation of AL-CRF and CRF

5.3.1. The Dataset of EMRs. In this paper, 1,200 EMRs have been divided to 10 equal parts. The training and the testing sets have been divided according to the ratio 9 : 1, and 10 comparative experiments have been carried out. The experimental results are shown in Table 5.

According to the results of Table 5, the recognition efficiency of the AL-CRF model tends to be stable when the number of iterations is 10, using 600 training samples. It is found that the CRF and the AL-CRF models have good recognition efficiency in the Chinese EMRs, with the recognition accuracy reaching over 90%, and most of the entities have been recognized. However, the recognition efficiency of the AL-CRF model is obviously better than the one of the CRF and the recognition accuracy can reach up to 95%. The F-value of the model increases almost by 3.65%. Specifically, the recognition effect of the five categories of entities in the medical field is shown in Table 6.

It can be seen that the AL-CRF model is superior to the CRF model in the recognition effect of various entities. In both models, the recognition effect is the best for symptoms and signs, while the entity recognition effect of treatment, examination, and inspection is not good, which may be related to the mixing of drug information into the entity of treatment and the unclear boundary between the entity of examination and inspection. Through analysis of experimental data, it is found that due to the imported custom dictionary, the word segmentation of symptoms and signs entity is more accurate, and its good recognition effect is related to the result of word segmentation.

5.3.2. The Dataset of Chinese Judicial Documents. At the same time, 10 comparative experiments have been conducted on the judicial documents. The experimental results are shown in Table 7.

Table 5: Comparison table of experimental results in EMRs.

Number of iterations		AL-CRF			CRF		
		P	*R*	*F*	*P*	*R*	*F*
1	11	0.9603	0.9314	0.9456	0.9223	0.9012	0.9116
2	10	0.9552	0.9328	0.9439	0.9252	0.8992	0.912
3	9	0.9516	0.9273	0.9393	0.8993	0.8632	0.8809
4	10	0.9501	0.9316	0.9408	0.9233	0.9014	0.9122
5	11	0.9627	0.9462	0.9543	0.9236	0.8979	0.9106
6	10	0.9513	0.932	0.9416	0.9124	0.9005	0.9064
7	10	0.9498	0.9297	0.9396	0.9157	0.8999	0.9077
8	11	0.9544	0.9406	0.9475	0.9208	0.9025	0.9136
9	10	0.9531	0.9201	0.9363	0.9082	0.8963	0.9022
10	10	0.9602	0.9224	0.9409	0.9144	0.9015	0.9079
Mean		0.9549	0.9314	0.9430	0.9165	0.8964	0.9065

Table 6: The recognition effect of 5 categories in medical field.

Model	Category	*P*	*R*	*F*
AL-CRF	Body (B)	0.9523	0.9197	0.9357
	Symptoms and signs (SS)	0.9791	0.9556	0.9672
	Examination and inspection (EI)	0.8042	0.8396	0.8215
	Disease and diagnosis (DD)	0.9381	0.9187	0.9283
	Treatment (T)	0.7938	0.7607	0.7769
CRF	Body (B)	0.9312	0.8899	0.9101
	Symptoms and signs (SS)	0.9746	0.9306	0.9521
	Examination and inspection (EI)	0.7659	0.8065	0.7857
	Disease and diagnosis (DD)	0.9133	0.8919	0.9025
	Treatment (T)	0.7624	0.7123	0.7365

Table 7: Comparison table of experimental results in judgement documents.

Number of iterations		AL-CRF			CRF		
		P	*R*	*F*	*P*	*R*	*F*
1	9	0.9314	0.9603	0.9456	0.8468	0.8824	0.8642
2	10	0.9328	0.9552	0.9439	0.8854	0.8992	0.8922
3	9	0.9273	0.9516	0.9393	0.8993	0.9132	0.9062
4	8	0.9316	0.9501	0.9408	0.8333	0.8714	0.8519
5	9	0.9462	0.9627	0.9543	0.8836	0.8979	0.8907
6	9	0.932	0.9513	0.9416	0.9024	0.9205	0.9114
7	10	0.9297	0.9498	0.9396	0.9057	0.9299	0.9176
8	8	0.9406	0.9544	0.9475	0.8708	0.9025	0.8864
9	9	0.9201	0.9531	0.9363	0.8982	0.9363	0.9169
10	9	0.9224	0.9502	0.9361	0.8844	0.9215	0.9026
Mean		0.9314	0.9539	0.9425	0.881	0.9075	0.894

According to the results of Table 7, the recognition efficiency of AL-CRF model tends to be stable when the number of iterations equals to 9, and 550 training samples are considered. Meanwhile, the AL-CRF model has also achieved promising results for the entity recognition in the case of the legislative domain. The recognition accuracy and recall have been improved, and the *F*-value has been increased by 4.85% compared with the *F*-value of the CRF model. Specifically, the effect of the 5 categories of entities in the recognition of the legislative field is shown in Table 8.

Trying to assess the effect of various entities to the recognition, in the case of the legislative data, we conclude that the AL-CRF model has a reliable performance and an obvious superiority, compared with the CRF, especially in legislative conceptual entities. Although, due to the wide range of legislative conceptual entities (e.g., "plaintiff" and "legal person)" and to the existence of relational concepts (e.g., "obligation of delivery" and "liability for compensation)" and finally due to the differences in the description of various legal documents, the overall recognition efficiency is still not as high as it should be. In addition, the large number of long entities in legislative principle entities has a negative effect.

6. Conclusions

In this paper, the active learning algorithm is applied to the domain of NER, and the hybrid AL-CRF model, which

Table 8: The recognition effect of 5 categories in legal field.

Model	Category	P	R	F
AL-CRF	Charge (C)	0.9713	0.9801	0.9757
	Penalty (P)	0.8727	0.8943	0.8834
	Legal principle (LP)	0.8062	0.8331	0.8194
	Legal concept (LC)	0.8898	0.9187	0.9041
	Law (L)	0.9765	0.9784	0.9774
CRF	Charge (C)	0.9672	0.9706	0.9689
	Penalty (P)	0.8583	0.8754	0.8668
	Legal principle (LP)	0.7926	0.8118	0.8021
	Legal concept (LC)	0.8547	0.8774	0.8659
	Law (L)	0.9718	0.9735	0.9726

employs the CRF classifier, is proposed. Through the recognition experiments of medical entities and legal entities, it is found that this method can train a good NER model with less annotated data, and it has a certain improvement over the CRF model in accuracy and recall. It can significantly reduce the labour cost for a large number of annotated data of the traditional methods, and it can speed up the convergence rate of the model. Thus, it can be more suitable for the recognition of domain entities with high annotation cost, and it lays the foundation for other natural language processing tasks in the specific domain.

Though this research effort is very promising, there are certain shortcomings. In terms of experimental data, although two datasets from different domains were adopted, their size was not big enough. Regarding the model itself, the adopted k-means approach can improve the initial sample quality; however, it is sensitive to noise and the outliers in the initial training set may affect the recognition efficiency of the model.

The extraction of relations between entities and knowledge fusion will be the key point in the next step of our research, in an effort to resolve the existing deficiencies.

Data Availability

This research has two different datasets. One of them is the open documents which is derived from "Chinese judgement document network" (http://wenshu.court.gov.cn/), and the other is the electronic medical records released by the China Conference on Knowledge Graph and Semantic Computing 2017.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Innovative Education Program for Graduate Students of Zhongnan University of Economics and Law (no. 201811409).

References

[1] Z. Liu and Q. Zhang, "Research overview of big data technology," *Journal of Zhejiang University (Engineering Science)*, vol. 48, no. 6, pp. 957–972, 2014.

[2] R. Grishman and B. Sundheim, "Message understanding conference-6: a brief history," in *Proceedings of 16th International Conference on Computational Linguistics COLING 1996 Volume 1*, Copenhagen, Denmark, August 1996.

[3] W. Zhao, L. Gao, and A. Liu, "Programming foundations for scientific big data analytics," *Scientific Programming*, vol. 2018, Article ID 2707604, 2 pages, 2018.

[4] Z. Sun and H. Wang, "Overview on the advance of the research on named entity recognition," *New Technology of Library and Information Service*, vol. 26, no. 6, pp. 42–47, 2010.

[5] A. Goyal, V. Gupta, and M. Kumar, "Recent named entity recognition and classification techniques: a systematic review," *Computer Science Review*, vol. 29, pp. 21–43, 2018.

[6] J. Zhao, "A survey on named entity recognition, disambiguation and cross-lingual coreference resolution," *Journal of Chinese Information Processing*, vol. 23, no. 2, pp. 3–17, 2009.

[7] S. Huang, X. Zheng, and D. Chen, "A semi-supervised learning method for product named entity recognition," *Journal of Beijing University of Posts and Telecommunications*, vol. 36, no. 2, pp. 20–23, 2013.

[8] W. Yang, X. Tian, S. Wang, and X. Zhang, "Recent advances in active learning algorithms," *Journal of Hebei University (Natural Science Edition)*, vol. 37, no. 2, pp. 216–224, 2017.

[9] M. A. Carbonneau, E. Granger, and G. Gagnon, "Bag-level aggregation for multiple-instance active learning in instance classification problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 1–11, 2018.

[10] N. Chinchor, "MUC-6 named entity task definition (version 2.1)," in *Proceedings of 6th Conference on Message Understanding*, Columbia, Maryland, November 1995.

[11] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-per-formance learning name-finder," in *Proceedings of Fifth Conference on Applied Natural Language Processing*, pp. 194–201, Washington, DC, USA, April 1997.

[12] A. E. Borthwick, "A maximum entropy approach to named entity recognition," Master's Dissertation, New York University, New York City, NY, USA, 1999.

[13] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proceedings of 19th International Conference on Computational Linguistics*, pp. 1–7, Taipei, Taiwan, September 2002.

[14] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of Seventh Conference on Natural Language Learning at HLT-NAACL*, vol. 4, pp. 188–191, Edmonton, Canada, May 2003.

[15] Y. Liu, "Named entity recognition in Chinese Micro-blog based on deep learning," *Advanced Engineering Sciences*, vol. 48, no. 2, pp. 142–146, 2016.

[16] J. Sun, H. Yu, and Y. Feng, "Recognition of nominated fishery domain entity based on deep learning architectures," *Journal of Dalian Ocean University*, vol. 33, no. 2, pp. 265–269, 2018.
[17] Q. Wei, T. Chen, R. Xu, Y. He, and L. Gui, "Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks," *Database*, vol. 2016, article baw140, 2016.
[18] L. Zhang, C. Qin, and W. Ye, "Research on legal field entity automatic recognition model based on conditional random fields," *New Technology of Library and Information Service*, vol. 11, pp. 46–52, 2017.
[19] J. Sun, J. Li, and G. Zhou, "An unsupervised Chinese part-of-speech tagging approach using conditional random fields," *Computer Applications and Software*, vol. 28, no. 4, pp. 21–23, 2011.
[20] S. Paisitkriangkrai, J. Sherrah, P. Janney, and V. D. Hengel, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 36–43, Boston, MA, USA, June 2015.
[21] S. Qian, Z. H. Chen, M. Q. Lin, and C. B. Zhang, "Saliency detection based on conditional random field and image segmentation," *Acta Automatica Sinica*, vol. 41, no. 4, pp. 711–724, 2015.
[22] F. Chen, Y. Liu, C. Wei, Y. Zhang, M. Zhang, and S. Ma, "Open field neologism discovery based on conditional random field," *Journal of Software*, vol. 5, pp. 1051–1060, 2013.
[23] H. Guo, "Accelerated continuous conditional random fields for load forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2023–2033, 2015.
[24] Z. Deng, J. Ren, and L. B. Liu, "Short-term traffic flow prediction algorithm based on multiple CRF model," *Computer Engineering and Design*, vol. 38, no. 10, pp. 2887–2891, 2017.
[25] M. Xia, G. Cao, G. Wang, and Y. Shang, "Remote sensing image classification based on deep learning and conditional random fields," *Journal of Image and Graphics*, vol. 22, no. 9, pp. 1289–1301, 2017.
[26] D. Angluin, "Queries and concept learning," *Machine Learning*, vol. 2, no. 4, pp. 319–342, 1988.
[27] J. Wu, A. Guo, V. S. Sheng, P. Zhao, and Z. Cui, "An active learning approach for multi-label image classification with sample noise," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 3, article 1850005, 2018.
[28] J. Zhu, H. Wang, B. K. Tsou, and M. Y. Ma, "Active learning with sampling by uncertainty and density for data annotations," *IEEE Transactions on Audio Speech and Language Processing*, vol. 18, no. 6, pp. 1323–1331, 2010.
[29] D. Pohl, A. Bouchachia, and H. Hellwagner, "Batch-based active learning: application to social media data for crisis management," *Expert Systems with Applications*, vol. 93, pp. 232–244, 2018.
[30] R. Wang, X. Z. Wang, S. Kwong, and C. Xu, "Incorporating diversity and informativeness in multiple-instance active learning," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1460–1475, 2017.
[31] S. Patra, K. Bhardwaj, and L. Bruzzone, "A spectral-spatial multicriteria active learning technique for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, pp. 5213–5227, 2017.
[32] J. Long, J. Yin, E. Zhu, and W. Zhao, "A summary of active learning research," *Journal of Computer Research and Development*, vol. 45, no. 1, pp. 300–304, 2008.

Hindawi
Scientific Programming
Volume 2018, Article ID 6385104, 7 pages
https://doi.org/10.1155/2018/6385104

Research Article

A 64-Line Lidar-Based Road Obstacle Sensing Algorithm for Intelligent Vehicles

Hai Wang,[1,2] **Xinyu Lou**,[1] **Yingfeng Cai**,[3] **and Long Chen**[3]

[1]*Institute of Automotive Engineering, Jiangsu University, Zhenjiang 212013, China*
[2]*Robotic and Automation Lab, The University of Hong Kong, Pok Fu Lam, Hong Kong*
[3]*School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Yingfeng Cai; caicaixiao0304@126.com

Received 16 August 2018; Accepted 4 November 2018; Published 21 November 2018

Guest Editor: Edward Rolando Núñez-Valdez

Based on the 64-line lidar sensor, an object detection and classification algorithm with both effectiveness and real time is proposed. Firstly, a multifeature and multilayer lidar points map is used to separate the road, obstacle, and suspension object. Then, obstacle grids are clustered by a grid-clustering algorithm with dynamic distance threshold. After that, by combining the motion state information of two adjacent frames, the clustering results are corrected. Finally, the SVM classifier is used to classify obstacles with clustered object position and attitude features. The good accuracy and real-time performance of the algorithm are proved by experiments, and it can meet the real-time requirements of the intelligent vehicles.

1. Introduction

Road target detection and classification is an important part of the safe driving of unmanned vehicles, especially in complex urban roads [1]. Among many kinds of sensors, due to the high resolution and precision of 64-line lidar, it is widely concerned by researchers and industrial developers. Unlike vision and millimeter radar, 64-line lidar has much bigger data amount to process which is more than 1 million 3D points per second at 10 Hz frequency, so that it has strict requirements on the real-time performance of the environment sensing algorithm.

There are two types of mainstream point cloud data processing methods. One is directly based on point cloud processing [2, 3], and the other is based on the grid map [4–6]. The former needs to process and classify every laser point which is extremely time consuming, while the latter one transfers 3D laser points to several 2D grid and then classify those new generated grids which is able to dramatically reduce classification calculation cost. The traditional grid map construction method contains several types such as mean height map [7], maximum height map, and minimum height map [8]. However, those methods that use single threshold are difficult to divide obstacles of different heights and shapes [9, 10]. For example, they cannot distinguish between slopes, low obstacles, and roadside, and they often consider hanging objects as obstacles such as twigs above the vehicle height.

In laser points clustering, the computation cost of existing clustering algorithm such as K-means clustering, density clustering [11, 12], and hierarchical clustering [13, 14] are $O(m)$, $O(m^2)$ and $O(m^2 \log m)$ which directly relate to the point number m. The above clustering algorithm will increase the computation time, and it is difficult to meet the real-time requirements of unmanned vehicles.

Classification of obstacle targets around unmanned vehicles in dynamic environments is also very important for path planning and behavior prediction of the unmanned vehicle. In [15], the point clouds acquired by the lidar were projected onto the grid, clustered by the global nearest neighbor (GNN) and for each candidate, its eigenvector was calculated and classified by using the support vector machine (SVM) based on the radical basis function (RBF) kernel. In [16], the vehicles and pedestrians were classified using the Gaussian hybrid model classifier (GMM classifier). In [17], the point cloud is projected onto a 2D grid, and the

features of the envelope rectangular block in the grid map are extracted and classified by RPN network. In [18], the support vector machine is combined with the reflection intensity probability distribution, the longitudinal height contour distribution, and the position and attitude correlation features to classify the lidar point cloud features. The literature [19] combines the basic features of point cloud and contextual semantic environment to construct the original and extended feature vector of point cloud and use the support vector machine for target recognition.

In this work, an object detection and classification algorithm with both effectiveness and real time is proposed. The algorithm separates the road, obstacle, and suspension objects by using a multifeature and multilayer elevation map. Then, obstacle grids are clustered by a grid-clustering algorithm with dynamic distance threshold. After that, by combining the motion state information of two adjacent frames, the clustering results are corrected. Finally, the SVM classifier is used to classify obstacles with clustered object position and attitude features.

2. Grid Map Construction

This paper uses a multifeature multilevel height map to abstract lidar point cloud data. The multifeature multilevel height map is a variant based on multiscale height maps which divide the surrounding space of the vehicle into three layers. The first layer is the pavement layer, indicating the road surface where the vehicle is able to travel. The second layer is the obstacle layer, including various obstacles such as vehicles, pedestrians, buildings, traffic signs, trees, and so on. The last layer is the suspension object layer, indicating the obstacle whose height is greater than the vehicle's safe height and will not affect the vehicle's travel but is detected by the lidar. The algorithm flow is shown in Figure 1.

2.1. Grid Point Cloud Segmentation. The laser points located in the same grid are sorted according to the height from small to large so that a list of data points is obtained. Set the two-point interval height threshold as H_t. When the interval between the upper point and the lower point is greater than the interval height threshold, these two points belong to different plane blocks. Repeat the process to traverse the entire grid map to form all the plane blocks P_k. For each plane block that has several laser points in the grid, it contains five features: maximum height $\mathrm{Max}H_k$, minimum height $\mathrm{Min}H_k$, height mean $\mathrm{Mean}H_k$, intensity mean $\mathrm{Mean}I_k$, and intensity variance σ_k. Here, maximum height, minimum height, and height mean characterize the geometric characteristics of point clouds in plane blocks, and the other two reflect the reflection intensity characteristics of point clouds.

2.2. Pavement Layer Detection. Unlike most algorithms which only use height features for pavement layer segmentation, height and intensity information of point cloud are both used in this work.

2.2.1. Height Information. The maximum and minimum height of the plane block is used as the pavement classification feature. Due to the error of the lidar calibration, a two-step approach is used to judge the ground floor.

For each plane block H, $\Delta H = \mathrm{Max}H - \mathrm{Min}H$. If $\Delta H > b$, this plane block is considered as an obstacle. If $\Delta H < a$, this plane block is considered as a pavement plane. If $a < \Delta H < b$, intensity characteristics will be introduced because it is difficult to determine whether the plane block is an obstacle or a pavement by relying solely on the height feature. Here, a and b are threshold in which a will be smaller and b will be larger.

2.2.2. Intensity Information. When $a < \Delta H < b$, the block may be a road with a steep slope or an object with a small vertical height, so intensity information is needed for further judgment.

The intensity value of the lidar return is between 0 and 255. Here, we take a big amount of samples and count their intensity values. In this work, the lidar intensity value probability distribution curve of 200 vehicles, 200 pedestrians, and asphalt, cement pavement is obtained as shown in Figure 2. It can be seen that no matter a vehicle or a pedestrian, its intensity variance is bigger since its surface material and color are usually not uniform. On the other hand, the pavement property is relatively uniform, so its intensity distribution is relatively regular and the variance is small. Based on this, if the intensity variance of a block is less than the variance threshold $\mathrm{Var}I_t$, the block will be taken as pavement plane otherwise it will be considered as obstacle.

2.3. Obstacle Layer and Suspension Layer Detection. After getting all the pavement layers, it is possible to obtain the average height H_G of all the pavement layers. Then, it is possible to set the suspension object layer height H_F as follows:

$$H_F = H_G + H_V + H_S, \tag{1}$$

where H_V is the height of the unmanned vehicle and H_S is the artificially set height of the obstacle from the roof of the vehicle while driving.

Hence, the plane block whose height is bigger than H_F will be considered as suspension layer while the rest is obstacle layer.

3. Obstacle Grid Clustering

Since we have projected the lidar point cloud into the grid map in the previous grid map construction step and obtained the obstacle grids, here the clustering time complexity reduced to $O(g)$, where g is the number of obstacle grids which is thousand times less than the number of raw lidar points.

Due to the fixed resolution of the lidar beam angle, its resolution will decrease as the distance increases which will lead to decompose a distant obstacle into multiple discrete parts and consider as multiple obstacles. To avoid this, we combine the obstacles motion state information of two adjacent frames to correct the spatial clustering results. The clustering algorithm flow chart is shown in Figure 3.

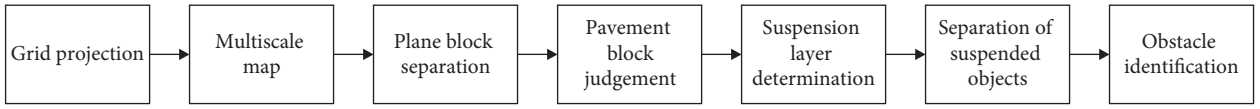


Figure 1: Flow diagram of multifeature multilayer height map construction.

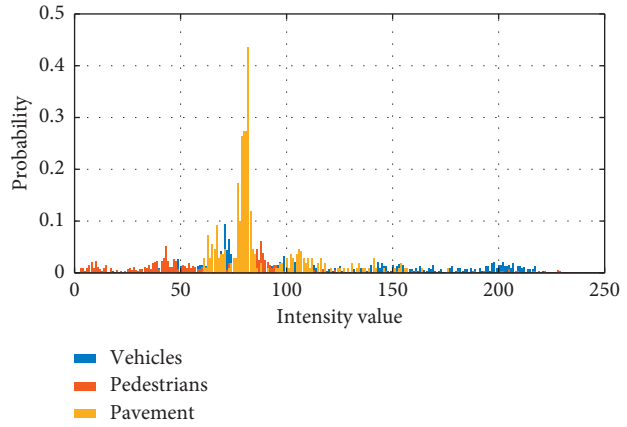


Figure 2: Probability distribution map of typical object return point intensity.

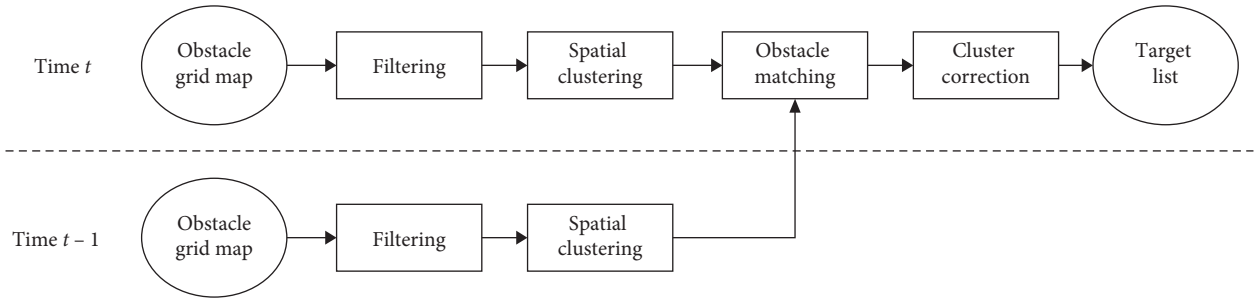


Figure 3: Clustering algorithm flow chart.

For different distance grid clustering in one frame, different distance thresholds are selected to cluster the discrete grids. The grid-clustering threshold is set as $N_T = D_T/G$, where G is the grid size and D_T is the radius parameter. For each grid, the adjacent grids in the area of $(2N_T - 1)\times(2N_T - 1)$ will be considered as one obstacle object.

In radius parameter setting, Borges proposed a method for calculating the radius parameter using distance values r_k [20], as shown in Figure 4.

Calculation formula of D_T is

$$D_T = r_{n-1}\frac{\sin(\Delta\varphi)}{\sin(20^\circ - \Delta\varphi)} + 3\sigma_r, \tag{2}$$

where r_{n-1} is the center coordinate distance of the obstacle grid, σ_r is lidar sensor measurement error, and $\Delta\varphi$ is horizontal angle resolution for lidar.

In order to further improve the accuracy of clustering, a target association matching clustering correction-based clustering is used. It pairs the closest obstacle blocks in the spatial clustering results at time $t-1$ and t by using four parameters: obstacle center coordinates, direction of motion, speed, and intensity mean.

4. Target Classification

In this work, dynamic obstacles in the road environment are separated into four categories: motor vehicles, nonmotor vehicles (bicycles), pedestrians, and others. Based on the motion characteristics and geometric contour characteristics of these four categories of obstacles, a SVM-based target classification method is proposed, as shown in Figure 5.

4.1. Target Feature. Since the information data of each obstacle are saved as a box model, the feature is also extracted from the box model. For each target box model, it has several groups of features listed as follows: (1) point X, point Y, point Z, and *alpha* which are the position and attitude features of the target; (2) length, width, height, and *delta* which are the contour features of the target. Here, *alpha* is the relative observation angle with the range of $(-\pi, \pi)$, as shown in Figure 6.

4.2. SVM Classifier. This work chooses the SVM classifier for obstacle classification which is good for small sample and nonlinear sample classification problems.

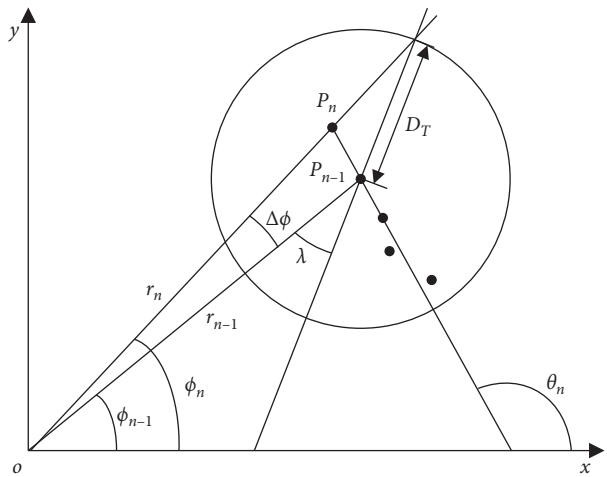


Figure 4: Radius parameter calculation method in [18].

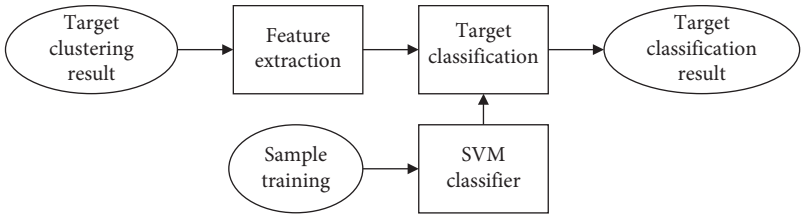


Figure 5: Classification process diagram.

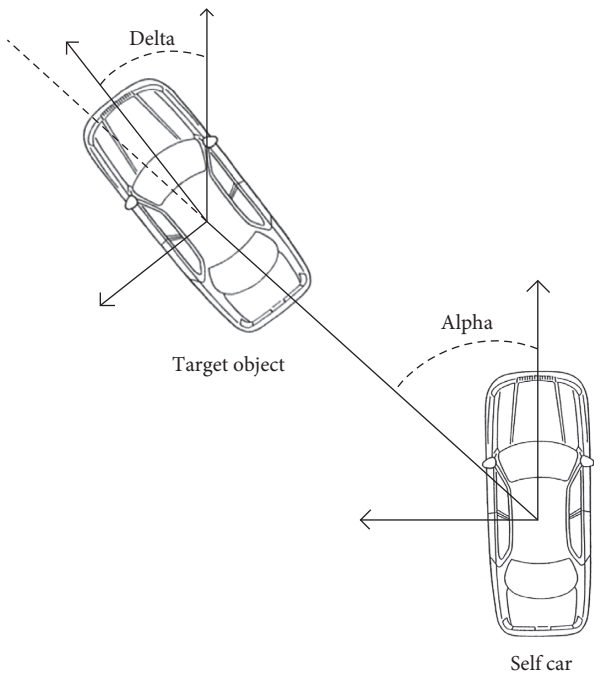


Figure 6: Schematic of *delta* and *alpha*.

In order to solve the nonlinear classification problem, the SVM classifier uses the kernel function to map the low-dimensional space classification problem to the high-dimensional feature space to construct a linear function for classification. The radial basis kernel function (RBF) is used:

$$K(x, y) = \exp\left(-\gamma|x - y|^2\right). \tag{3}$$

5. Experiment and Analysis

5.1. Clustering Experiment and Analysis. The clustering algorithm is compared with the eight-connected clustering algorithm under fixed distance threshold and density-based spatial clustering of applications with noise (DBSCAN) algorithm. We use these three methods to carry out experiments on 200 target vehicles in the road environment. Through the experimental analysis, the clustering accuracy rate of the obstacle targets is shown in Table 1. Partition-based methods, such as k-means, need to know the number of clusters in advance, so they are not suitable for unmanned vehicles and we do not include them in the comparison. The average time taken by this algorithm is about 15 ms.

A set of intuitive comparisons are shown in Figure 7. Figure 7(a) shows the effect of the proposed clustering algorithm, Figure 7(b) is that of eight-connected clustering algorithm, and Figure 7(c) is that of DBSCAN. It can be seen that, in far distance, the eight-connected clustering algorithm marks one object incorrectly as multiple objects and the DBSCAN is a little better than that while our method still works well.

Table 1: Clustering algorithms comparison.

Target distance (m)	Eight-connected clustering algorithm accuracy (%)	DBSCAN accuracy (%)	Our method accuracy (%)
0–20	73.4	78.3	92.6
20–40	64.9	70.1	86.7
40–80	41.5	52.9	69.3
80–150	18.6	21.4	36.3

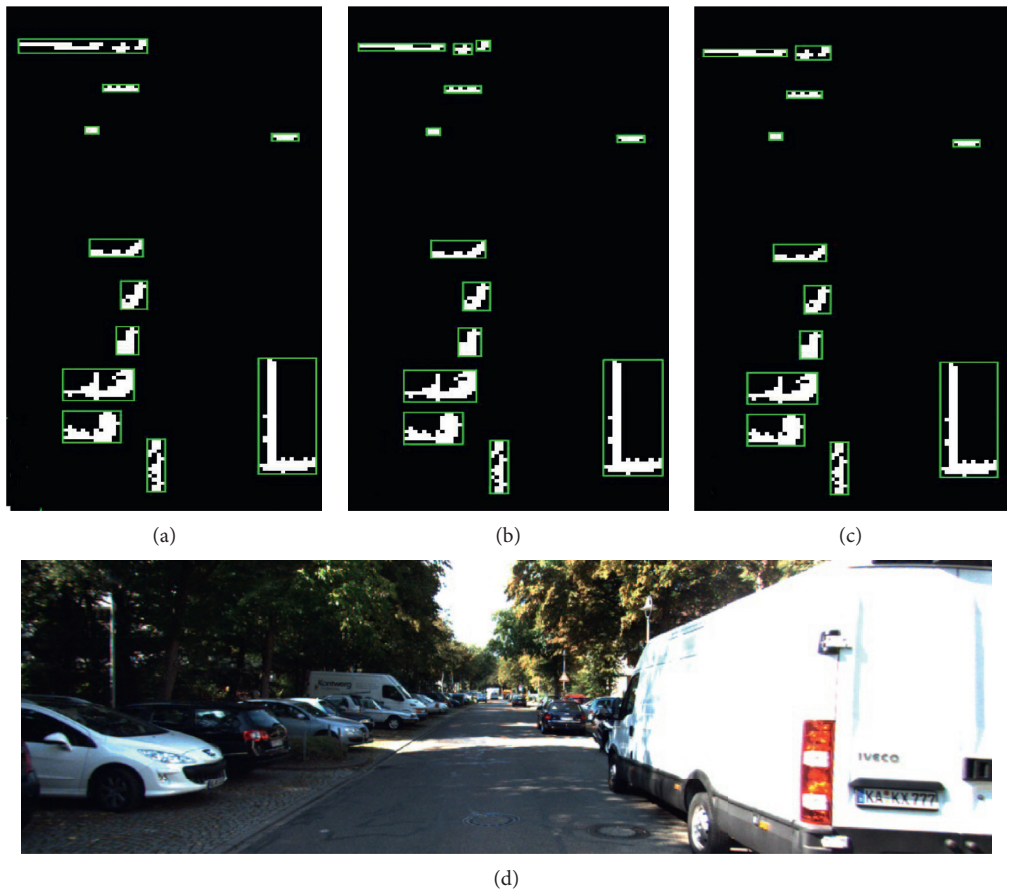


Figure 7: Clustering experiment. (a) Our method. (b) Eight-connected clustering. (c) DBSCAN. (d) Corresponding visual image.

5.2. Classification Experiment and Analysis. This experiment used the software developed by Professor Lin Chih-Jen of Taiwan University—LIBSVM [21]. In addition, the KITTI data set and the BDD100K are used for classification testing [22, 23]. Overall, there are 5217 samples containing 4091 vehicle samples, 417 bicycling samples, 573 pedestrian samples, and 136 other samples. Here, we take about 70% of the total number of samples as training samples and the rest as test samples. Through the grid optimization algorithm, the parameter penalty factor $C = 246$, the parameter $\gamma = 0.012065$, and the optimal recognition rate is 88.31%, as shown in Table 2.

A group of classification experiments is shown in Figure 8. Green, yellow, and red boxes mean vehicle, bicycle, and pedestrian, respectively, and the overall classification time is 10 ms per frame.

Table 2: Classification test result statistics.

	Overall	Vehicle	Bicycle	Pedestrians
Sample number	1566	1228	126	172
Correct classification number	1383	1172	71	138
Classification rate	88.31%	95.44%	56.35%	80.23%

6. Conclusion

Focusing on the difficulty of the large data of 64-line lidar which affect the real-time performance of unmanned vehicle, an object detection and classification algorithm with both effectiveness and real time is proposed. The algorithm separates the road, obstacle, and suspension by using a multifeature and multilayer elevation map. Then,

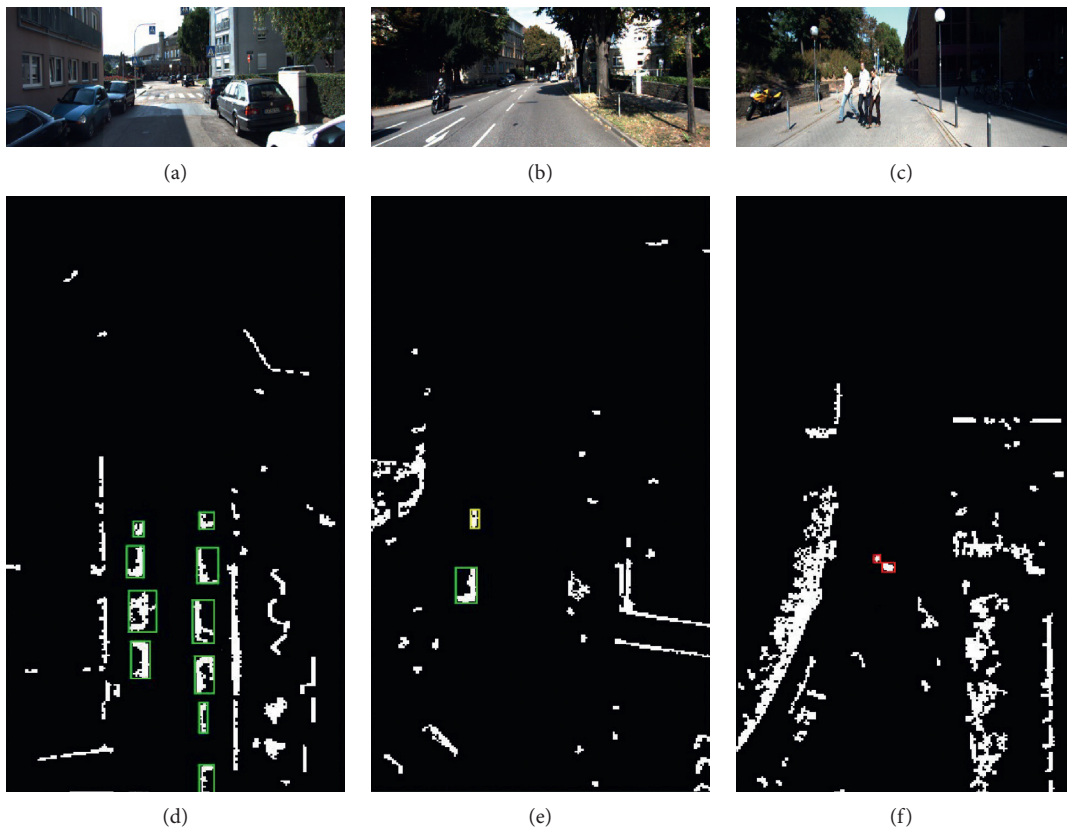


Figure 8: Target classification experiment result.

a grid-clustering algorithm based on dynamic distance threshold is used to cluster the obstacles, and the clustering results are corrected by combining the motion state information of two adjacent frames. Finally, SVM is used to classify obstacles. The experimental results show that the algorithm has good obstacle detection and classification accuracy and better real-time performance to meet the real-time requirements of unmanned autonomous vehicles while driving on the road. During the experiment, it was also found that the detection rate of bicycles and pedestrians is relatively low. This may be because the lidar can only scan a small part of pedestrians and bicycles far from the autonomous vehicle, and some of these parts are often filtered by the filtering algorithm or the features we use do not distinguish pedestrians and bicycles very well. So, in the future work, we will improve the filtering algorithm so that more obstacle information will be acquired and new features, such as speed, will be added to better distinguish pedestrians and bicycles.

Data Availability

The data sources in this work are from the public dataset KITTI.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Key Research and Development Program of China (2018YFB0105003), National Natural Science Foundation of China (U1764264, 61601203, U1664258, U1764257, and 61773184), Natural Science Foundation of Jiangsu Province (BK20180100), Key Research and Development Program of Jiangsu Province (BE2016149), Key Project for the Development of Strategic Emerging Industries of Jiangsu Province (2016-1094 and 2015-1084), and Key Research and Development Program of Zhenjiang City (GY2017006).

References

[1] Y. Cai, Z. Liu, H. Wang, and X. Sun, "Saliency-based pedestrian detection in far infrared images," *IEEE Access*, vol. 5, pp. 5013–5019, 2017.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77–85, Las Vegas, NV, USA, December 2016.

[3] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.

[4] S. Goga and S. Nedevschi, "An approach for segmenting 3D LiDAR data using multi-volume grid structures," in

Proceedings of Intelligent Computer Communication and Processing (ICCP), pp. 309–315, Cluj-Napoca, Romania, September 2017.

[5] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nuchter, and J. Wang, "3D LIDAR point cloud based intersection recognition for autonomous driving," in *Proceedings of 2012 IEEE Intelligent Vehicles Symposium (IV)*, pp. 456–461, IEEE, Madrid, Spain, June 2012.

[6] Y. Liu, S. T. Monteiro, and E. Saber, "Vehicle detection from aerial color imagery and airborne LiDAR data," in *Proceedings of 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1384–1387, IEEE, Beijing, China, July 2016.

[7] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, "Multivehicle cooperative local mapping: a methodology based on occupancy grid map merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2089–2100, 2014.

[8] S. Thrun, M. Montemerlo, H. Dahlkamp et al., "Stanley: the robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[9] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-d lidar," in *Proceedings of 2015 IEEE International Conference on Information and Automation*, pp. 1648–1653, Lijiang, China, August 2015.

[10] H. Wang, L. Dai, Y. Cai, X. Sun, and L. Chen, "Salient object detection based on multi-scale contrast," *Neural Networks*, vol. 101, pp. 47–56, 2018.

[11] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognition*, vol. 71, pp. 375–386, 2017.

[12] K. Yamazaki, "Effects of additional data on Bayesian clustering," *Neural Networks*, vol. 94, pp. 86–95, 2017.

[13] X. Zhang, H. Liu, and X. Zhang, "Novel density-based and hierarchical density-based clustering algorithms for uncertain data," *Neural Networks*, vol. 93, pp. 240–255, 2017.

[14] A. A. Liu, Y. T. Su, W. Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 102–114, 2017.

[15] M. Darms, P. Rybski, and C. Urmson, "Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments," in *Proceedings of 2008 IEEE Intelligent Vehicles Symposium*, pp. 1197–1202, Auburn Hills, MI, USA, February 2008.

[16] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *Proceedings of 2007 IEEE Intelligent Transportation Systems Conference*, pp. 1044–1049, Seattle, WA, USA, September 2007.

[17] J. Behley, V. Steinhage, and A. Cremers, "Laser-based segment classification using a mixture of bag-of-words," in *Proceedings of International Conference on Intelligent Robots & Systems*, pp. 4195–4200, Tokyo, Japan, November 2013.

[18] P. Babahajiani, L. Fan, and M. Gabbouj, "Object recognition in 3D point cloud of urban street scene," in *Proceedings of Asian Conference on Computer Vision*, pp. 177–190, Springer, Cham, Switzerland, November 2014.

[19] S. Wirges, T. Fischer, J. B. Frias, and C. Stiller, "Object detection and classification in occupancy grid maps using deep convolutional networks," 2018, http://arxiv.org/abs/1805.08689.

[20] P. Skrzypczynski, "Building geometrical map of environment using IR range finder data," *Intelligent Autonomous Systems*, pp. 408–412, 1995.

[21] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[23] F. Yu, W. Xian, Y. Chen et al., *BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling*, arXiv preprint arXiv:1805.04687, 2018.