Online-Journals.org

iJEP International Journal of Engineering Pedagogy

iJEP | eISSN: 2192-4880 | Vol. 15 No. 4 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijep.v15i4.54419

PAPER

Optimizing Cheating Detection in Online Exams with K-Shingling, MinHashing, and LSH: A Comparative Analysis with TF-IDF and BoW

Nabila El Rhezzali(✉), **Imane Hilal**, **Meriem Hnida**

ITQAN Team, LyRica Lab, School of Information Sciences (ESI), Rabat, Morocco

nabila.el-rhezzali@esi.ac.ma

ABSTRACT

Detecting cheating in online exams is a major challenge, not least to guarantee the originality and independence of answers. This paper presents a comparative analysis of three feature extraction methods for cheating detection based on similarity detection: Term frequency-inverse document frequency (TF-IDF), Bag of Words (BoW), and a new approach combining K-Shingling, MinHashing, and Locality Sensitive Hashing (LSH). We evaluate these methods in terms of their ability to accurately and efficiently identify similarities between student responses. Experimental results show that the K-Shingling, MinHashing, and LSH pipelines consistently outperform or match traditional approaches. Logistic regression and random forest classifiers with MinHashing + LSH achieve perfect scores of 1.00 in terms of precision, recall, F1 score, and accuracy, demonstrating the robustness and effectiveness of the method. In comparison, TF-IDF and BoW show mixed performance between classifiers, with notable limitations in terms of scalability and sensitivity to text variations. This study highlights the scalability and computational efficiency of the K-shingling, MinHashing, and LSH approaches, making them particularly suitable for large-scale online examination environments. By offering a detailed performance comparison, we demonstrate that K-shingling, MinHashing, and LSH provide a more reliable and efficient solution for detecting cheating in online exams, paving the way for greater academic integrity in digital education.

KEYWORDS

cheating detection, online exams, string-based similarity, term frequency-inverse document frequency (TF-IDF), Bag of Words (BoW), k-shingling, Minhashing, locality sensitive hashing (LSH), logistic regression, random forest, support vector machines (SVM)

1 INTRODUCTION

In 2019, as COVID-19 spread globally and caused widespread disruption, educational institutions quickly set up online courses and exams [1]. While this approach

El Rhezzali, N., Hilal, I., Hnida, M. (2025). Optimizing Cheating Detection in Online Exams with K-Shingling, MinHashing, and LSH: A Comparative Analysis with TF-IDF and BoW. *International Journal of Engineering Pedagogy (iJEP)*, 15(4), pp. 40–56. https://doi.org/10.3991/ijep.v15i4.54419

Article submitted 2025-01-01. Revision uploaded 2025-03-25. Final acceptance 2025-03-26.

has overcome many of the challenges posed by COVID-19, it has also brought several benefits related to online accessibility. Online testing enabled students to take tests from any location with Internet access, eliminating the need to travel to specific test centers [2]. In addition, supervisors found it easier to monitor students using the combined camera feeds to keep an eye on their behavior [3]. Despite the convenience of online exams, they have one major drawback: an increased risk of cheating [4]. The absence of direct supervision allows students to engage in dishonest practices, such as answer-seeking or cheating off-camera, making it difficult for instructors to detect faults [5]. Technologically, however, it is possible to activate several cameras and use facial recognition and behavioral recognition systems, or even ear recognition. However, online surveillance remains time-consuming and demanding [6]. Indeed, online supervisors cannot monitor all students at the same time, so some students inevitably cheat when they are not being directly observed. As online education becomes increasingly popular, it has become crucial to have robust methods for detecting cheating in online exams. Cheating not only calls into question the value of academic qualifications but also disrespects the hard work of honest students. To tackle this problem, educators and institutions are exploring new strategies, such as sentence similarity analysis, to effectively identify cheating in online exams.

Machine learning has established itself as a fundamental element of information technology over the last two decades, fundamentally reshaping various aspects of our lives. Applications in the healthcare field, in particular, concern classification and early diagnosis using medical ultrasound imaging and magnetic resonance imaging. Its integration into everyday applications often takes place in the shadows, but its impact is profound and widespread. As the volume of available data continues to grow exponentially, the potential for sophisticated data analysis increases accordingly. This evolution suggests that advanced data analysis will increasingly play the role of a key driver of technological innovation. The interaction between large datasets and machine learning algorithms enables us to extract meaningful information, identify patterns, and make predictions with unprecedented accuracy. From personalized recommendations on streaming platforms to fraud detection in financial systems, machine learning is becoming an indispensable tool in every industry. What's more, as organizations leverage this technology, the demand for skilled data science and machine learning professionals is rising sharply, highlighting a changing workforce requirement. This environment fosters interdisciplinary collaboration, bringing together experts in statistics, computer science, and specialized fields to tackle complex challenges. In essence, the future of machine learning is not just about processing data but transforming it into actionable intelligence that stimulates progress and innovation, making it an integral part of our journey of technological advancement.

This paper presents an approach to automatically detect cheating in final exams based on sentence similarity analysis. The rest of this paper is organized as follows. Section 2 presents the context of this study and the problem statement. Section 3 is dedicated to an exhaustive literature review. Our proposed approach is detailed in section 4. Section 5 is dedicated to the main results and a discussion. Finally, section 6 presents the main conclusions and future work.

2 CONTEXT AND PROBLEM STATEMENT

With e-learning, students can study whenever and wherever they want, making e-learning an essential tool for teachers worldwide. In the past, geographical barriers limited access to education, making it difficult for teachers and students to travel. Today, online teaching enables consistent learning, better collaboration, and global access to education for all. With the rise of online learning, particularly accelerated by the COVID-19 pandemic, the shift from traditional classroom exams to online assessments has made cheating a more common and harder-to-detect problem.

Educational institutions face a major challenge in identifying cheaters in distance learning exams. Students may cheat individually or collaborate with others, and it is difficult to detect this type of behavior in an online environment, especially in distance learning courses. Some students may gradually collaborate during online exams, whether they are in the same or different locations, using the internet or social networks to communicate. This study aims to solve this problem by using similarity techniques to measure the similarity between students' answers and identify potential cheating.

3 RELATED WORK

3.1 Cheating and cheating detection methods

Cheating on exams is common throughout the world, despite advances in detection methods. Over the past decade, numerous studies have examined how students cheat and how universities can try to combat the problem [7]. Previous studies have reported that academic cheating is prevalent among students, with many students admitting to dishonest behaviors and perceiving a relatively low risk of detection [8]. The integration of digital platforms into higher education has expanded the vulnerabilities of modern assessment methods. Student academic misconduct has evolved from classical analog strategies, such as passing handwritten notes, to more technology-enabled forms of cheating in electronic learning environments [9].

There are many reasons why students cheat, including academic pressure, the belief that they will not get caught, and easy access to online resources [10]. Difficult exams, strict grading, and a competitive academic environment can drive students to adopt dishonest practices to meet performance expectations.

Online exams, in particular, can seem easier to manipulate, especially when proctoring is weak or inconsistent. The lack of face-to-face proctoring and anonymity of remote exams can make students feel less accountable, reducing their reluctance to cheat compared to in-person exams.

The way assessments are designed and the learning environment as a whole can influence cheating behavior, as situational factors such as contextual pressures and perceived opportunity may affect students' likelihood of engaging in academic dishonesty [11]. Cultural attitudes also influence how students perceive cheating. In some educational systems, working together on homework is encouraged, which can blur the boundaries between collaboration and dishonesty. On the other hand, schools with strict academic integrity policies create environments where cheating is more strongly discouraged and carries more serious consequences.

Professors play a key role in preventing cheating. Clear academic integrity policies, honor codes, and strong monitoring, both through technology and direct supervision have been shown to reduce dishonest behavior [12]. Research suggests that when instructors promote ethical academic habits, teach proper citation methods, and openly discuss expectations, students are less likely to cheat. In addition, strict deterrents such as plagiarism detection software, artificial intelligence (AI)-based exams, and severe consequences for misconduct make cheating riskier and less attractive.

Because cheating is a complex problem, it needs to be tackled from a number of different angles. Research in the fields of education, psychology, and academic integrity can help us better understand why students cheat and how to prevent it. By applying psychological theories on ethical decision-making, considering the influence of culture on student behavior, and using the latest technologies for online monitoring and cheating detection, schools can create stronger strategies for maintaining academic honesty in digital learning.

Typical methods of cheating involve using cheat sheets that are prewritten and often written in small fonts. These sheets can be concealed in clothing, under a wristwatch, on the floor, inside books, or under folders placed under the desk [13]. Some students use mobile phones to send text messages containing question numbers or correct answers as a means of communication [14]. Moreover, another method involves using iPods with recording capabilities, with the earphone wires hidden behind long hair [13].

Online cheating encompasses various methods, including using digital devices for communication and accessing information [7]. For example, students may use cell phones or other Bluetooth-enabled devices to share answers [14]. Similar to traditional settings, students can access prewritten notes using digital devices or hide them within online platforms. The main distinction is that online cheating heavily depends on technology to enable dishonest behavior. Online exams provide an effective way to administer tests, allowing students to study from any convenient location without the need for physical travel. However, they face challenges related to cheating since there are no physical proctors to supervise and monitor the exam process. This is known as distance cheating, which encompasses various forms of dishonest behavior. Examples of distance cheating include taking exams on behalf of another student, using applications to solve exam questions, sharing test questions with experts to obtain answers, and downloading resources from the internet, such as using e-books. The following paragraph will examine the current literature on techniques for identifying cheating.

Javed and Aslam [15] developed a method to detect faces, eyes, and human presence using an age detection and Kalman filtration algorithm. Their system could track eye movements and pupil behavior to identify if a student was cheating. However, this system had a limitation, as it could not detect objects such as smartphones, notes, or other cheating devices. Additionally, it did not include voice analysis to catch someone else providing answers to the student. In 2017, Atoum et al. [5] used a multimedia analytics system to monitor online reviews, involving audiovisual observation and an support vector machines (SVM) classifier. The system had a strong segment-based detection rate. However, a limitation was that it needed two cameras to effectively detect cheating behavior. Bawarith et al. [16] proposed an automated e-exam cheating detection system that monitors student behavior during online examinations to identify suspicious activities and potential academic dishonesty. On the other hand, Ghizlane et al. [17] proposed an online exam monitoring system aimed at reducing academic cheating through automated supervision. The system focuses on visual monitoring and does not describe the use of additional modalities such as browser activity tracking or audio-based analysis. In addition, Ozgen et al. [18] created an online interview anti-cheating system that uses object and face detection with a HOG-based SVM detector. This system could identify cheating without needing two cameras. However, it lacked some features, such as voice analysis, browser detection, and the ability to recognize cheating without just relying on facial expressions.

Tiong and Lee [19] proposed a deep learning-based approach for detecting cheating behavior in online examinations. Their method relies on behavioral data analysis using recurrent neural network architectures, including LSTM and DenseLSTM models, to identify suspicious activity patterns. The study evaluates multiple models and reports that DenseLSTM achieves the highest detection accuracy. The proposed framework also considers basic security aspects such as system-level monitoring. However, it does not provide an in-depth discussion of limitations related to specific monitoring modalities. On the other hand, Jadi [20] used facial expression detection and browser monitoring, utilizing CNN to identify cheating. This system only needed one camera to detect facial expressions and prevent the opening of other applications. However, it did not include voice analysis, which could be seen as a limitation. Dilini et al. [21] developed a browser extension that uses eye tracking with face detection and OCSVM to spot cheating. The system works well with web browser-based testing platforms, improving exam quality. However, it might not catch cheating if a student uses notes stuck to their screen. In addition, Barrientos et al. [22] aimed to use Amazon Web Services (AWS) to detect dishonest behavior among students, such as using smartphones to search for answers or committing plagiarism. They implemented face detection and TensorFlow for this purpose. The system is powerful because it uses open-source tools and can recognize human faces and objects. However, it does require purchasing AWS services. Soltane and Laouar [23] developed a smart detection and recognition system that uses face and sound detection with CNN technology. The system's advantage is that it only requires one camera and microphone. However, it does not monitor the examiner's browser activity.

In recent years, researchers have concentrated on developing effective methods for detecting cheating in electronic exams. Ongoing studies aim to enhance monitoring techniques for cheating incidents. Over the past few decades, numerous studies have explored various approaches to this challenge. For example, the authors in [24] created two classification models using decision trees for cheating detection: one based on cosine similarity and the other on the overlap coefficient. Their findings indicated that the model utilizing the overlap coefficient outperformed the cosine similarity model when compared to results from expert evaluations. In another study [25] Ruiperez-Valiente et al. proposed a machine learning-based approach for detecting multiple-account cheating in online learning environments. Their method uses supervised learning techniques to analyze student behavior and problem features in order to identify suspicious patterns associated with cheating. Additionally, in [26] the authors conducted an empirical study comparing proctored and unproctored online examinations to assess differences in student performance outcomes. Using regression-based analysis, they examined how the relationship between prior academic achievement and final exam scores varied across testing conditions. Their results indicate that unproctored exams tend to distort expected performance patterns, suggesting reduced assessment validity and raising concerns about academic integrity in online environments.

Many existing methods for detecting cheating in online exams rely on physical monitoring or behavioral analysis, such as eye tracking or facial expression detection, which often overlook critical aspects like unauthorized materials or external assistance. While some approaches have incorporated machine learning models to enhance detection, they still primarily focus on observable behaviors rather than the content of student responses. In contrast, our method analyzes sentence similarities in student answers, enabling direct identification of copying or collaboration. This content-centric approach not only targets the actual outputs of cheating but also adapts to modern cheating techniques, providing a more comprehensive and effective solution. By leveraging advanced feature extraction techniques, we enhance detection accuracy without the need for complex hardware setups, ensuring robust monitoring of academic integrity in increasingly digital educational environments.

Recent advances in cloud computing and AI have transformed education, especially in areas such as online learning, computer simulations, and augmented reality. According to [27] computer simulations and cloud-based technology make education more open and accessible by improving scalability, automation, and flexibility. Their study highlights how these technologies enhance student engagement and create more personalized learning experiences.

Similarly, another study by [28] explores how cloud-based technology and augmented reality (AR) work together to enhance education. It shows that the combination of AR and cloud-based platforms makes learning more interactive and immersive, benefiting students in a variety of subjects. Meanwhile, research in [29] examines why students choose to use AI tools for their academic work. Focusing on the humanities and social sciences, the study identifies key factors influencing AI adoption, including the usefulness and ease of use of the technology, as well as ethical concerns.

To conclude, such studies help set the stage for our research by showing how technology-enhanced learning is linked to academic integrity, cheating detection, and plagiarism prevention.

3.2 Feature extraction and classifiers

Feature extraction is the process of transforming raw data into structured numerical representations, enabling machine learning models to process and analyze it effectively. In text analysis, methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) are commonly used. TF-IDF highlights unique terms by balancing their frequency across documents, while BoW represents text as unordered word collections, offering simplicity but limited contextual understanding. These features are then processed by classifier algorithms such as SVM, logistic regression, and random forest that assign labels or categories to data based on learned patterns. Together, feature extraction and classification form the foundation of supervised learning tasks such as text classification, similarity detection, and fraud detection.

Feature extraction: Bag of Words and TF-IDF. The BoW model is one of the simplest and most widely used methods for feature extraction in text processing. It represents each document as a vector of word frequencies based on a predefined vocabulary. While this method is intuitive and easy to implement, it suffers from notable limitations: it disregards word order and semantic relationships, and it often produces high-dimensional, sparse vectors. Despite these drawbacks, BoW remains relevant for straightforward tasks such as text classification and spam filtering, where interpretability and computational simplicity are prioritized. Early foundational works such as the authors in [30] introduced distributional representations of text, which evolved into practical applications through contributions like the authors in [31], who formalized the use of word frequency in information retrieval systems.

In contrast, TF-IDF improves upon BoW by weighting terms according to their importance. This is achieved by combining the term frequency in a document with the inverse document frequency across a corpus, reducing the influence of common but non-discriminative words. TF-IDF is particularly useful for distinguishing documents in tasks such as information retrieval and text summarization. However, it shares some limitations with BoW, such as ignoring word order and semantics, and it can be computationally more demanding. Seminal contributions from Jones [32], who introduced term specificity, and Salton and McGill [33], who formalized vector space models, combined to establish TF-IDF as a cornerstone of text processing, emphasizing its ability to enhance the relevance of retrieved information.

Classifiers. Machine learning has become a foundational element of information technology, playing a vital role in our lives over the past two decades. As the volume of available data continues to grow exponentially, we can anticipate that intelligent data analysis will become even more prominent, serving as a key driver of technological advancement. In this subsection, we provide a brief description of three machine learning algorithms selected for their effectiveness in various tasks, along with justifications for their choice.

Logistic regression is a widely used algorithm for binary classification problems. Its simplicity and interpretability make it an attractive choice, as it provides insights into the relationship between input features and the probability of an outcome. This is particularly important in fields such as healthcare and social sciences, where understanding the impact of features is crucial [34]. Random forest is an ensemble learning method that combines multiple decision trees to improve accuracy and robustness. It is particularly effective in handling complex datasets with high dimensionality and reducing the risk of overfitting. Its versatility makes it suitable for both classification and regression tasks across various domains [35]. SVM is known for its effectiveness in high-dimensional spaces and for problems where classes are not linearly separable. It works by identifying the optimal hyperplane that maximizes the margin between different classes. The use of kernel functions allows SVM to be applied in a wide range of applications, including text classification and image recognition [36].

4 METHODOLOGY

4.1 Overview of cheating detection using similarity analysis

In our approach (see Figure 1), we propose an automatic final exam cheating detection model, which consists of two main phases. The first phase consists of creating a specialized dataset specifically designed for training our model. This dataset is built using textual data from PubMed articles, which provides rich and diverse content that improves our feature extraction process. We use k-shingling hashing to decompose documents into sets of k-word sequences, capturing local similarities within the text. Min-hashing is then applied to generate signatures for each document, effectively reducing dimensionality while preserving similarity information. Finally, we use locality-sensitive hashing (LSH) to organize these min-hash signatures into buckets, facilitating the identification of similar submissions and improving our ability to detect potential fraud.

The second phase focuses on classification, where we apply three different algorithms: logistic regression, random forest, and SVM. Each algorithm is used to classify students as cheaters or non-cheaters based on features extracted from their assignments. By comparing the performance of these classifiers, we aim to determine the most effective method to accurately identify cheating behaviors. This two-phase approach not only leverages advanced feature extraction techniques but also harnesses the power of various machine learning models to provide a robust solution for detecting academic dishonesty in online assessments. More details are presented in the next section.

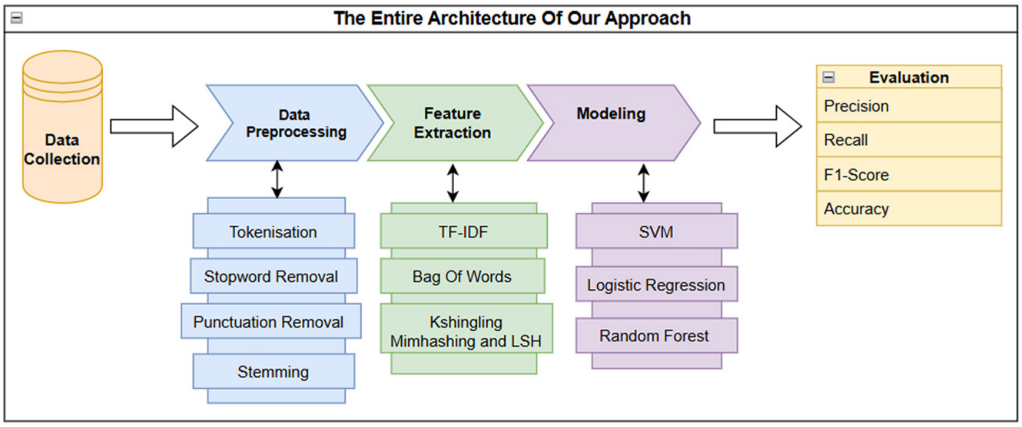

Fig. 1. Block diagram of the proposed approach

4.2 Proposed feature extraction technique

The need for effective similarity measures is increasingly critical in the context of widely used online education. In data mining, one of the main challenges is to identify near-duplicate documents. This task is essential for detecting cheating in online exams and can be addressed using natural language processing techniques, such as K-shingling, Minhashing, and LSH. These strategies have the potential to improve our understanding of identifying similarity between near-duplicate pairs in exam submissions [37].

K-Shingling method. To efficiently represent documents as datasets, you can break them down into sentences made up of chosen strings. This method helps identify shared components in documents, even if they appear in different sequences. By using metrics such as Jaccard similarity, you can measure how similar these sets are [38].

Documents are collections of text, where k-shingles are important components, this character-level decomposition serves as a foundational string-based representation for calculating lexical similarity across document corpora [39]. Consider document D "Machine learning is a field focused on figuring out how to teach machines to learn," with k set to 2. Consequently, D's 2 shingles include {"Machine learning," "learning is," "is a," "a field," "field focused," "focused on," "on figuring," "figuring out," "out how," "how to," "to teach," "teach machines," "machines to," "to learn"}.

Choosing a very small value for k can result in numerous common short strings (such as "is a" or "how to") appearing across entirely unrelated texts. This leads to a high, artificially inflated shingle similarity that does not necessarily correspond to a true semantic or structural relationship between the documents. To accurately evaluate these overlapping sets at scale, Broder [40] introduced the formal mathematical metrics of resemblance and containment, which allow researchers to quantify these exact similarities through compact document sketches.

Minhashing. Shingle-based datasets are naturally very large, making it difficult to manage massive collections like millions of documents. Since these datasets can exceed the capacity of main memory, a practical solution is to replace them with smaller, more manageable representations known as "signatures." The key requirement for these signatures is that they must enable comparison between sets and help estimate the Jaccard similarity, which is an essential measure of how similar the sets are [41]. To create these signatures, you need a "characteristic matrix," which is crucial as it displays all sets and their shingles. First, you must understand how to calculate Minhash values. After that, the next step involves creating a signature matrix using several Minhash functions, which requires multiple steps. Initially, you randomly generate a set of n permutations of the rows of the characteristic matrix. The signature of a column, labeled S, is created by applying a series of hash function (h_1, h_2, … h_n) to each element in that column, resulting in a signature vector [$h_1(S)$, $h_2(S)$, … $h_n(S)$]. This signature matrix has n columns, significantly reducing the size of the matrix compared to the original characteristic matrix. However, in practice, applying permutations to large feature matrices is impractical. Randomly selecting and sorting rows by permutation would be too time-consuming, so we use a hash function to permute the matrix instead [42]. Instead of selecting n permutations from the row at random, we randomly choose n hash functions, h_1, h_2, … h_n. To calculate the signature matrix, we follow these steps for each row r:

1. Compute $h_1(r)$, $h_2(r)$, … and $h_n(r)$.
2. If the value in column c for row r is "0," we do nothing.
3. If the value in column c for row r is "1," we extract the values of $h_i(r)$ for $i = 1, 2, \ldots n$

To construct the Minhash matrix, we first select n Minhash functions. In this example, we choose n to be 2, and we use the hash functions $h_1(x) = (x + 1) \bmod 17$ and $h_2(x) = (3x + 1) \bmod 17$. For simplicity, we label the rows numerically from 0 to 16 instead of using letters.

Locality-sensitive hashing. One of the major challenges of similarity identification methods is understanding that their goal often goes beyond simply calculating the degree of similarity of two documents. Instead, the primary goal may be to thoroughly examine all documents to identify those that share significant similarities. However, this comprehensive evaluation of all document combinations is inherently time-consuming [43]. The current task is to create strategies that focus on the most likely combinations that are similar to each other, rather than evaluating every possible pair. In this context, LSH provides a comprehensive solution [44]. The basic concept behind LSH suggests that when hashing items, the process should be performed in stages. This groups identical or highly similar items into common hash buckets. If pairs end up in the same bucket across multiple hashes, they are considered potential matches for similarity checks. This targeted approach reduces the computational burden compared to evaluating all possible document combinations [45]. Using Minhash signatures for LSH, a practical approach is to divide the signature matrix into "b" bands. Each band contains "r" rows, making the total number of "n" rows equal to "br." A separate hash function is selected for each band. Using "r" integers, this function calculates the number of buckets and the number of hashes based on the vectors. While it is possible to use a single hash function for all bands, using separate tables for hashing in each band ensures that columns with similar vectors are not accidentally grouped into the same bucket across different bands.

4.3 Application of the proposed feature extraction to cheating detection

The proposed approach (see Figure 1) involves fetching random articles from PubMed to extract genuine answers, which are then paired with simulated cheating answers generated through strategies such as copying, paraphrasing, and collaborating. These answers are preprocessed through tokenization, stopword removal, punctuation removal, and stemming. Feature extraction is performed using three methods: TF-IDF, BoW, and a proposed technique based on K-shingling, MinHashing, and LSH to capture text similarity. Machine learning models such as SVM, logistic regression, and random forest are trained on the extracted features to classify answers as genuine or cheating. The performance of each feature extraction method is evaluated and compared, with the goal of identifying the most effective approach for detecting cheating based on response similarity. Our approach follows the following steps:

- **Step 1:** Data Collection
  - Fetch random articles from PubMed using the `PubMed` API. These articles are used as a source of both genuine answers and potential cheating answers.
- **Step 2:** Answer Extraction
  - From the fetched article texts, extract sentences that serve as "genuine answers." A subset of these sentences is selected to simulate realistic responses to potential cheating detection scenarios.
  - In parallel, generate "cheating answers" using various strategies, such as:
    - Copying: Directly copying the original text.
    - Paraphrasing: Reversing or slightly altering the text to simulate paraphrasing.
    - Collaborating: Combining elements from multiple sources to simulate collaborative cheating.
- **Step 3:** Preprocessing
  - Tokenization: Split text into individual words (tokens).
  - Stopwords Removal: Filter out common, unimportant words (e.g., "the," "is").
  - Punctuation Removal: Remove punctuation marks that don't contribute to meaning.
  - Stemming: Reduce words to their root forms using stemming (e.g., "running" becomes "run").
  - The result is a set of processed answers ready for feature extraction.
- **Step 4:** Feature Extraction
  - Term Frequency-Inverse Document Frequency: Compute TF-IDF features for each answer, which capture the importance of each word in the context of the entire corpus.
  - Bag of Words: Convert the text into a vector of word frequencies, where each word corresponds to a feature.
  - MinHash, LSH with k-Shingling.
  - K-Shingling: Convert the answers into overlapping shingles (small segments of text) of size 'k'.
  - MinHashing: Generate signatures for each shingle using the MinHash algorithm, which provides a compact representation of the text.
  - Locality Sensitive Hashing: Group similar documents together by applying LSH to the MinHash signatures. This helps identify similar answers that may indicate cheating.
- **Step 5:** Modeling and Classification
  - The preprocessed features are used to train classification models that can distinguish between genuine and cheating answers. Several machine learning algorithms are tested:
    - Support Vector Machine.
    - Logistic Regression.
    - Random Forest Classifier.

- Models are trained using the features derived from TF-IDF, BoW, and MinHash + locality-sensitive hashing.
- Evaluate the models using various performance metrics such as accuracy, precision, recall, and F1-score.

- **Step 6:** Evaluation and Comparison:
  - Evaluate the performance of the different feature extraction methods (TF-IDF, BoW, and the proposed MinHash + LSH with k-Shingling) by comparing the classification results for each method.
  - The goal is to assess which feature extraction technique provides the best accuracy and detection capability for identifying cheating behavior based on response similarity.
  - These steps outline the entire process, from collecting data and extracting answers to training models and evaluating their performance.

5 MAIN RESULTS AND DISCUSSION

Table 1 allows a comparison between the different approaches to identify which is potentially the most effective.

Table 1. Comparison of different approaches to effectiveness evaluation

Methods	Precision	Recall	F1-Score	Accuracy
SVM with TF-IDF	0.84	0.80	0.73	0.80
Logistic Regression with TF-IDF	0.85	0.81	0.76	0.81
Random Forest with TF-IDF	1.00	1.00	1.00	1.00
SVM with BoW	0.98	0.97	0.97	0.97
Logistic Regression with BoW	1.00	1.00	1.00	1.00
Random Forest with BoW	1.00	1.00	1.00	1.00
SVM with MinHash + LSH	0.84	0.80	0.73	0.80
Logistic Regression with MinHash + LSH	1.00	1.00	1.00	1.00
Random Forest with MinHash + LSH	1.00	1.00	1.00	1.00

The results demonstrate that the k-shingling, MinHashing, and LSH approaches perform comparably to or better than traditional TF-IDF and BoW methods in detecting similarity between texts. Specifically, the MinHash + LSH approach with logistic regression and random forest classifiers achieved perfect scores of 1.00 for precision, recall, F1-score, and accuracy. These values indicate exceptional performance in detecting similarities, highlighting the robustness and effectiveness of this approach.

In contrast, TF-IDF shows varying performance across different models. For instance, with Random Forest, TF-IDF achieved perfect scores of 1.00 across all metrics, similar to MinHash + LSH. However, TF-IDF combined with SVM or logistic regression performed less effectively, achieving precision scores of 0.84 and 0.85, recall scores of 0.80 and 0.81, F1-scores of 0.73 and 0.76, and overall accuracies of only 0.80 and 0.81, respectively.

Bag of words demonstrates strong performance with logistic regression and random forest, achieving perfect scores of 1.00 for precision, recall, F1-score, and accuracy. However, these results are not consistent across all models. For example, SVM with BoW scored 0.98 for precision and 0.97 for recall, F1-score, and accuracy, slightly below the performance of the MinHash + LSH approach. Furthermore, BoW's reliance on exact word matches and its large feature space make it less scalable for handling extensive datasets.

Overall, the MinHash + LSH approach consistently maintains high accuracy (1.00) across models while being computationally efficient and more scalable for large datasets than TF-IDF or BoW. This adaptability, combined with its exceptional similarity detection performance, demonstrates that the k-shingling, MinHashing, and LSH pipeline is a robust, efficient choice for large-scale similarity detection tasks.

Figure 2 allows a direct comparison of precision, recall, F1 score, and accuracy scores for each combination of model and feature extraction method, facilitating visual analysis of the performance of MinHashing and LSH compared to traditional approaches such as TF-IDF and bag of words.

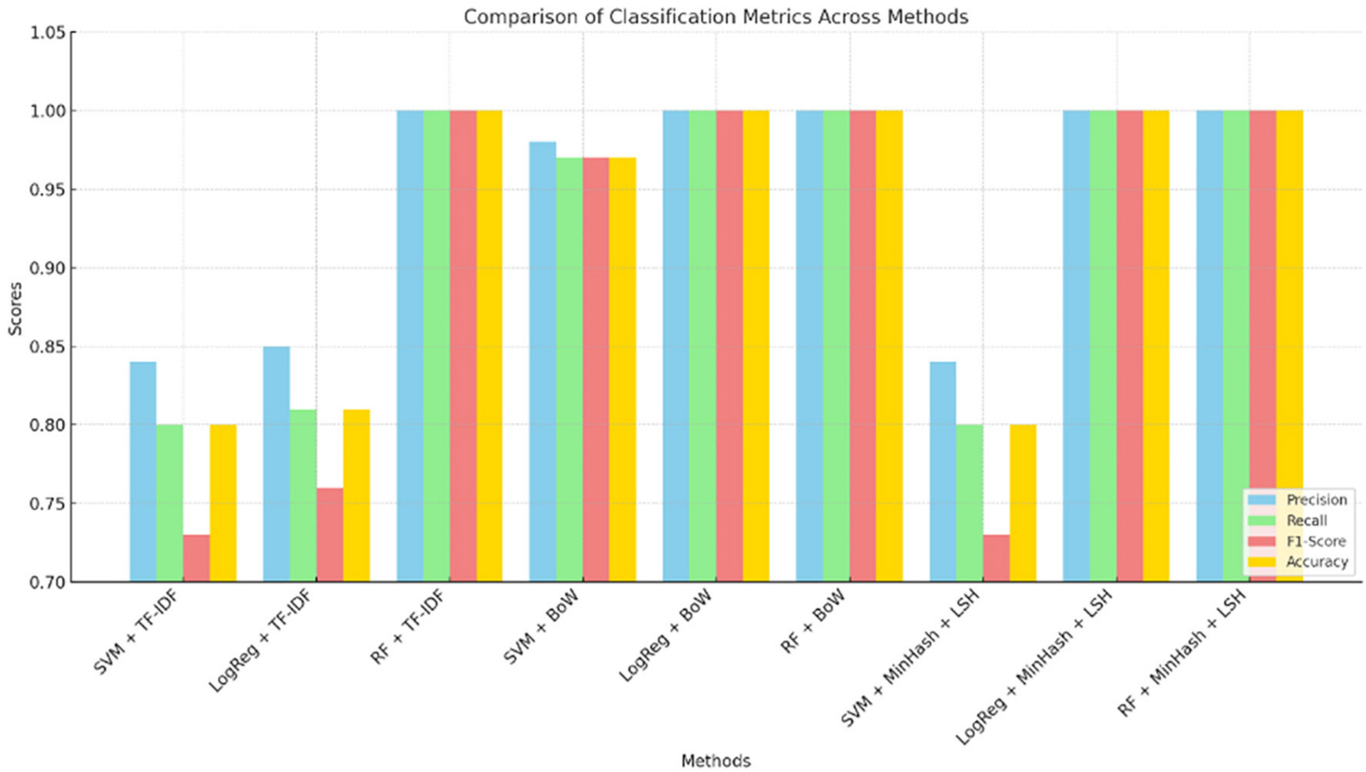

Fig. 2. Performance metrics of different models

The perfect scores (100% accuracy, F1-score, precision, and recall) in this study are due to several key factors. First, the preprocessing steps, such as removing stopwords and punctuation and applying stemming help clean the data and focus on the most relevant features. This reduces noise, lowers false positives, and ensures that only truly similar responses are detected. Second, the methods used in this study are highly effective at capturing textual similarity. By optimizing similarity functions specifically for this dataset, the system enhances feature representation and improves accuracy in detecting similarities. As a result, plagiarized responses are identified with minimal errors. However, it's important to note that these results were achieved under controlled conditions, where the dataset contained clearly similar responses with little variation or noise. In real-world educational settings, factors like differences in how students phrase their answers, attempts to evade detection, and subtle semantic differences could introduce challenges that may impact overall performance.

While the proposed method outperforms TF-IDF and BoW in detecting cheating, it's important to recognize the limitations of MinHashing and LSH. One key challenge is the optimization of hash functions; if the hash functions aren't optimized properly, the system may misjudge similarities. Additionally, LSH can result in false positives, where harmless similarities are flagged as cheating because of random hash collisions. To mitigate this, we can add extra verification steps, like using multiple LSH bands, adjusting the similarity threshold, and having flagged responses reviewed manually by instructors. Instead of automatically marking responses as cheating, the system would flag them for review. The instructor could then assess whether the flagged responses are genuinely similar in a meaningful way, looking at factors like the topic or common academic phrasing before deciding whether it's a case of cheating. This approach helps cut down on false positives caused by accidental similarities. Furthermore, MinHashing focuses on lexical similarity, meaning it might miss cases where students paraphrase. To overcome this, we can combine semantic-aware techniques with MinHashing and LSH, which would improve the system's ability to detect paraphrased responses and make the detection process more reliable overall.

Our approach includes a thorough data preprocessing pipeline aimed at reducing the impact of cheating tactics. We start by removing stopwords and punctuation to reduce unnecessary differences in the text and apply stemming to make word forms consistent. This helps ensure that small changes, like adding irrelevant words or changing word endings, don't significantly affect the detection of similarities. However, while these steps make the system more robust, they may not fully address more advanced tactics, like completely changing sentence structure or rewording sentences to keep the meaning the same. Future research could explore additional techniques to better defend against these types of evasion strategies.

While our study mainly focuses on computational performance metrics, we understand that real-world educational impacts must also be considered. MinHash combined with LSH is good at detecting highly similar responses, but like any automated system, it can produce false positives, especially when students independently arrive at similar answers to factual questions. To reduce these risks, we recommend adding extra verification steps, like adjusting similarity thresholds, using semantic similarity analysis, and allowing instructors to review cases that are unclear. Ethical considerations are essential to make sure that the system does not wrongly accuse students. Instead of acting as an independent decision-maker, our approach works best as an initial filter within a larger cheating detection system that includes human oversight.

6 CONCLUSIONS

In this study, we explored the effectiveness of three feature extraction methods—TF-IDF, BoW, and k-shingling combined with MinHashing and LSH—to detect cheating in online exams through similarity analysis. The results demonstrate that the k-shingling, MinHashing, and LSH approaches provide superior or comparable performance to traditional methods while addressing key limitations such as scalability and sensitivity to textual variations.

Specifically, the MinHash+LSH pipeline achieved perfect precision, recall, F1 score, and accuracy of 1.00 with logistic regression and random forest classifiers, highlighting its robustness and reliability. Although TF-IDF and BoW achieved high accuracy in some cases, their performances were inconsistent across classifiers and datasets. Moreover, the computational efficiency and scalability of MinHash+LSH make it particularly well-suited for large-scale online examination environments where traditional methods often struggle.

In conclusion, the k-shingling, MinHashing, and LSH methods stand out as highly effective and efficient solutions for similarity detection in cheating scenarios. Its ability to maintain high performance while handling large datasets and text variations makes it a valuable tool for improving academic integrity in online education systems. This approach paves the way for more robust, scalable, and fair cheating detection mechanisms in digital learning environments.

7 REFERENCES

[1] U. Vellappan, L. Lim, and S. Y. Lim, "Engaging learning experience: Enhancing productivity software lessons with screencast videos," *J. Inform. Web Eng.*, vol. 2, no. 2, pp. 189–200, 2023. https://doi.org/10.33093/jiwe.2023.2.2.14

[2] M. Labayen, R. Vea, J. Flórez, N. Aginako, and B. Sierra, "Online student authentication and proctoring system based on multimodal biometrics technology," *IEEE Access*, vol. 9, pp. 72398–72411, 2021. https://doi.org/10.1109/ACCESS.2021.3079375

[3] A. W. Muzaffar, M. Tahir, M. W. Anwar, Q. Chaudry, S. R. Mir, and Y. Rasheed, "A systematic review of online exams solutions in e-learning: Techniques, tools, and global adoption," *IEEE Access*, vol. 9, pp. 32689–32712, 2021. https://doi.org/10.1109/ACCESS.2021.3060192

[4] S. Dendir and R. S. Maxwell, "Cheating in online courses: Evidence from online proctoring," *Comput. Hum. Behav. Rep.*, vol. 2, p. 100033, 2020. https://doi.org/10.1016/j.chbr.2020.100033

[5] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Trans. Multimed.*, vol. 19, no. 7, pp. 1609–1624, 2017. https://doi.org/10.1109/TMM.2017.2656064

[6] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," *Intell. Syst. Appl.*, vol. 16, p. 200153, 2022. https://doi.org/10.1016/j.iswa.2022.200153

[7] N. El Rhezzali, I. Hilal, and M. Hnida, "Cheating detection in online exams," in *Digital Technologies and Applications. ICDTA 2023*, in Lecture Notes in Networks and Systems, vol. 668, S. Motahhir and B. Bossoufi, Eds., Cham, Switzerland: Springer, 2023, pp. 431–440. https://doi.org/10.1007/978-3-031-29857-8_44

[8] J. Kerkvliet and C. L. Sigmund, "Can we control cheating in the classroom?" *J. Econ. Educ.*, vol. 30, no. 4, pp. 331–343, 1999. https://doi.org/10.1080/00220489909596090

[9] B. Keresztury and L. Cser, "New cheating methods in the electronic teaching era," *Procedia-Soc. Behav. Sci.*, vol. 93, pp. 1516–1520, 2013. https://doi.org/10.1016/j.sbspro.2013.10.074

[10] D. L. McCabe, K. D. Butterfield, and L. K. Treviño, "Cheating in college: Why students do it and what educators can do about it," *Academy of Management Learning & Education*, vol. 13, no. 1, pp. 143–145, 2014. https://www.jstor.org/stable/43696604

[11] D. A. Rettinger and Y. Kramer, "Situational and personal causes of student cheating," *Res. High. Educ.*, vol. 50, pp. 293–313, 2009. https://doi.org/10.1007/s11162-008-9116-5

[12] B. E. Whitley, "Factors associated with cheating among college students: A review," *Res. High. Educ.*, vol. 39, pp. 235–274, 1998. https://doi.org/10.1023/A:1018724900565

[13] K. Yee and P. MacKown, "Detecting and preventing cheating during exams," Center for Academic Integrity, Rutland Institute for Ethics, Clemson University, Clemson, SC, USA, 2009. Available: http://www.academicintegrity.org

[14] D. Faucher and S. Caves, "Academic dishonesty: Innovative cheating techniques and the detection and prevention of them," *Teach. Learn. Nurs.*, vol. 4, no. 2, pp. 37–41, 2009. https://doi.org/10.1016/j.teln.2008.09.003

[15] A. Javed and Z. Aslam, "An intelligent alarm based visual eye tracking algorithm for cheating free examination system," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 10, pp. 86–92, 2013. https://doi.org/10.5815/ijisa.2013.10.11

[16] R. Bawarith, D. A. Basuhail, D. A. Fattouh, and P. D. S. Gamalel-Din, "E-exam cheating detection system," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 4, p. 5329, 2017. https://doi.org/10.14569/IJACSA.2017.080425

[17] M. Ghizlane, B. Hicham, and F. H. Reda, "A new model of automatic and continuous online exam monitoring," in *2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, 2019, pp. 1–5. https://doi.org/10.1109/SysCoBIoTS48768.2019.9028027

[18] A. C. Ozgen, M. U. Öztürk, O. Torun, J. Yang, and M. Z. Alparslan, "Cheating detection pipeline for online interviews," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 2021, pp. 1–4. https://doi.org/10.1109/SIU53274.2021.9477950

[19] L. C. O. Tiong and H. J. Lee, "E-cheating prevention measures: Detection of cheating at online examinations using deep learning approach – A case study," *arXiv preprint arXiv.2101.09841*, 2021. https://doi.org/10.48550/arXiv.2101.09841

[20] A. Jadi, "New detection cheating method of online-exams during COVID-19 pandemic," *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 4, pp. 123–130, 2021. https://doi.org/10.22937/IJCSNS.2021.21.4.17

[21] N. Dilini, A. Senaratne, T. Yasarathna, N. Warnajith, and L. Seneviratne, "Cheating detection in browser-based online exams through eye gaze tracking," in *2021 6th International Conference on Information Technology Research (ICITR)*, 2021, pp. 1–8. https://doi.org/10.1109/ICITR54349.2021.9657277

[22] A. Barrientos, M. Cuadros, J. Alba, and Á. S. Cruz, "Implementation of a remote system for the supervision of online exams through the use of cameras with artificial intelligence," in *2021 IEEE Engineering International Research Conference (EIRCON)*, 2021, pp. 1–4. https://doi.org/10.1109/EIRCON52903.2021.9613352

[23] M. Soltane and M. R. Laouar, "A smart system to detect cheating in the online exam," in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, 2021, pp. 1–5. https://doi.org/10.1109/ICISAT54145.2021.9678418

[24] E. R. Cavalcanti, C. E. Pires, E. P. Cavalcanti, and V. F. Pires, "Detection and evaluation of cheating on college exams using supervised classification," *Informatics in Education*, vol. 11, no. 2, pp. 169–190, 2012. https://doi.org/10.15388/infedu.2012.09

[25] J. A. Ruiperez-Valiente, P. J. Munoz-Merino, G. Alexandron, and D. E. Pritchard, "Using machine learning to detect 'multiple-account' cheating and analyze the influence of student and problem features," *IEEE Trans. Learn. Technol.*, vol. 12, no. 1, pp. 112–122, 2019. https://doi.org/10.1109/TLT.2017.2784420

[26] O. Harmon and J. Lambrinos, "Are online exams an invitation to cheat?" *J. Econ. Educ.*, vol. 39, no. 2, pp. 116–125, 2008. https://doi.org/10.3200/JECE.39.2.116-125

[27] S. Papadakis *et al.*, "Revolutionizing education: Using computer simulation and cloud-based smart technology to facilitate successful open learning," in *Joint Proceedings of the 10th Illia O. Teplytskyi Workshop on Computer Simulation in Education, and Workshop on Cloud-based Smart Technologies for Open Education (CoSinEi and CSTOE 2022) co-located with ACNS Conference on Cloud and Immersive Technologies in Education (CITEd 2022)*, Kyiv, Ukraine, 2022. https://doi.org/10.31812/123456789/7375

[28] S. Papadakis *et al.*, "Unlocking the power of synergy: The joint force of cloud technologies and augmented reality in education," in *Joint Proceedings of the 10th Illia O. Teplytskyi Workshop on Computer Simulation in Education, and Workshop on Cloud-based Smart Technologies for Open Education (CoSinEi and CSTOE 2022) co-located with ACNS Conference on Cloud and Immersive Technologies in Education (CITEd 2022)*, Kyiv, Ukraine, 2022. https://doi.org/10.31812/123456789/7399

[29] K. Lavidas *et al.*, "Determinants of humanities and social sciences students' intentions to use artificial intelligence applications for academic purposes," *Information*, vol. 15, no. 6, p. 314, 2024. https://doi.org/10.3390/info15060314

[30] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, nos. 2–3, pp. 146–162, 1954. https://doi.org/10.1080/00437956.1954.11659520

[31] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975. https://doi.org/10.1145/361219.361220

[32] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," in *Document Retrieval Systems*, London, U.K.: Taylor Graham Publishing, 1988, pp. 132–142.

[33] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill, 1986.

[34] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. New York, NY: Wiley, 2000. https://doi.org/10.1002/0471722146

[35] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

[36] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018

[37] E. R. Nabila, H. Imane, and H. Meriem, "Identifying near-duplicate Pairs in exams: Similarity detection using Kshingling Minhashing and LSH," in *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2023, pp. 1–6. https://doi.org/10.1109/SITA60746.2023.10373698

[38] A. Z. Broder, "Identifying and filtering near-duplicate documents," in *Combinatorial Pattern Matching. CPM 2000,* in Lecture Notes in Computer Science, vol. 1848, R. Giancarlo and D. Sankoff, Eds., Berlin, Germany: Springer, 2000, pp. 1–10. https://doi.org/10.1007/3-540-45123-4_1

[39] W. Gomaa and A. Fahmy, "A survey of text similarity approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013. https://doi.org/10.5120/11638-7118

[40] A. Z. Broder, "On the resemblance and containment of documents," in *Proceedings.* Compression *and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, 1997, pp. 21–29. https://doi.org/10.1109/SEQUEN.1997.666900

[41] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *J. Comput. Syst. Sci.*, vol. 60, no. 3, pp. 630–659, 2000. https://doi.org/10.1006/jcss.1999.1690

[42] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, and R. Motwani, "Finding interesting associations without support pruning," *IEEE Trans. Knowl. Data Eng.,* vol. 13, no. 1, pp. 64–78, 2001. https://doi.org/10.1109/69.908981

[43] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 2006, pp. 459–468. https://doi.org/10.1109/FOCS.2006.49

[44] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[45] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge, UK: Cambridge University Press, 2011. https://doi.org/10.1017/CBO9781139058452

8 AUTHORS

Nabila El Rhezzali. ITQAN Team, LyRica Lab, School of Information Sciences, Rabat, Morocco (E-mail: nabila.el-rhezzali@esi.ac.ma).

Imane Hilal is an Associate Professor and a researcher at School of Information Sciences in Rabat (Morocco). ITQAN Team, LyRica Lab, Information Sciences School, Rabat, Morocco.

Meriem Hnida. ITQAN Team, LyRica Lab, School of Information Sciences, Rabat, Morocco.