The North African Journal of Scientific Publishing (NAJSP)
مجلة شمال إفريقيا للنشر العلمي (NAJSP)
E-ISSN: 2959-4820
Volume 3, Issue 3, 2025
Page No: 161-168
Website: https://najsp.com/index.php/home/index
SJIFactor 2024: 5.49 معامل التأثير العربي (AIF) 2024: 0.71 ISI 2024: 0.696

Data Reduction Via Decision Trees

Khirallah Elferjani[1*], Eman Alamin[2], Faraj A. El-Mouadib[3], Salma almajbri[4]
[1,2,3,4] Computer Science Department, Faculty of Information Technology, Benghazi University, Benghazi, Libya

تقليص البيانات من خلال أشجار القرار

خيرالله الصادق خيرالله الفرجاني[1*]، ايمان الأمين[2]، فرج المؤدب[3]، سالمة المجبري[3]
[4،3،2،1] قسم علوم الحاسوب، كلية تقنية المعلومات، جامعة بنغازي، بنغازي، ليبيا

*Corresponding author: Khirallah.elfarjani@uob.edu.ly

Received: July 08, 2025	Accepted: August 2, 2025	Published: August 11, 2025

Abstract
Due to the wide availability of huge and enormous data that are stored in different repository systems (i.e. databases and data warehouses, etc…) have risen the needs to develop intelligent tools and techniques to infer or extract knowledge from such data. The answer to such need appeared in the field of Data Mining (DM), which is concerned with the discovery of knowledge from huge amounts of data. Eventually, these huge quantities of data slow the discovery process. Therefore, data preprocessing (i.e. data reduction) is an essential step to reduce the data size and in turn speeds up the discovery process. The purpose of this paper is to employ decision trees in the context of data reduction and the ID3 (Iterative Dichotomiser3) method is the decision tree algorithm. The application of this idea was used in the test bed software system's implementation. The system is put to the test using well-known data sets to assess the viability of this method.

Keywords: Decision Tree, Data Reduction, ID3, Data Mining, KDD.

المخص:
نظرًاً لتوفر كميات هائلة وضخمة من البيانات المخزنة في أنظمة مستودعات بيانات مختلفة (مثل قواعد البيانات ومستودعات البيانات وغيرها)، نشأت الحاجة إلى تطوير أدوات وتقنيات ذكية لاستنتاج أو استخراج المعرفة من هذه البيانات. جاء الحل لهذه الحاجة في مجال التنقيب عن البيانات(*Data Mining*) ، الذي يهتم باكتشاف المعرفة من كميات كبيرة من البيانات. ومع ذلك، تؤدي هذه الكميات الضخمة من البيانات إلى إبطاء عملية الاكتشاف. لذلك، يُعد تجهيز البيانات المسبق (مثل تقليل البيانات) خطوة أساسية لتقليل حجم البيانات وبالتالي تسريع عملية الاكتشاف. يهدف هذا البحث إلى توظيف أشجار القرار في سياق تقليل البيانات، حيث تُعتبر خوارزمية *ID3 (Iterative Dichotomiser 3)* هي خوارزمية أشجار القرار المستخدمة. تم تطبيق هذه الفكرة في نظام برمجي تجريبي. وقد تم اختبار النظام باستخدام مجموعات بيانات معروفة بهدف تقييم جدوى هذه الطريقة.

الكلمات المفتاحية: شجرة القرارات، تقليص البيانات، ID3، التنقيب في البيانات، اكتشاف المعرفة من البيانات.

Introduction

Recent advancements in computer technology, particularly the availability of cost-effective, high-capacity storage media, have enabled the accumulation of unprecedented volumes of data. These extensive datasets often encapsulate latent and valuable knowledge that cannot be effectively extracted using traditional data analysis techniques [1,2]. This pressing need for more sophisticated analytical approaches has catalyzed the emergence of the field of Knowledge Discovery in Databases (KDD). KDD broadly encompasses the systematic process of uncovering meaningful patterns, correlations,

and insights from large-scale data repositories. Within this framework, Data Mining (DM) constitutes a pivotal stage, as it involves the application of computational algorithms to extract actionable knowledge from raw data [4]. Although the terms KDD and Data Mining are sometimes used interchangeably in certain scholarly contexts, they represent distinct yet interrelated concepts. The significance of DM has garnered increasing scholarly attention over the past three decades [3,4].

Data mining systems inherently operate on large-scale datasets, where the magnitude of the data often becomes a bottleneck in the knowledge discovery process due to the computational time required. Moreover, real-world data are frequently characterized by noise, incompleteness, inconsistency, and other imperfections, necessitating a data preprocessing phase to refine and optimize the dataset for the intended mining task. As noted in [5,6], data reduction refers to the process of selecting a subset of the original dataset with reduced volume while maintaining its essential informational integrity. This study introduces the application of decision tree induction as a means of achieving data reduction without compromising the quality or accuracy of the mining results.

The primary objective of this work is to evaluate the effectiveness and practical benefits of employing decision tree induction techniques for data reduction. The validity of this approach is assessed through a series of experimental investigations conducted using a dedicated software testbed, referred to as the Data Reduction System (DRS).

Materials and Methods

Study area:

Data mining functionalities

The data mining functionality to be used is determined by the kind of patterns or regularities sought and the type of available data. In general, the data mining functionalities are classified as being predictive or descriptive. According to [7,8]. As a subset of supervised learning methods, predictive data mining tasks involve drawing conclusions from the available data set to create a model or function that may be used to forecast missing or unavailable class labels or data values [9,10]. Descriptive data mining tasks, on the other hand, are a subset of unsupervised learning techniques that describe the overall characteristics or attributes of the full dataset. Tasks related to predictive mining include classification and prediction. Cluster analysis, association analysis, outlier analysis, characterization analysis, and discrimination analysis are examples of jobs in descriptive mining [11-15]. Many systems and algorithms are available to carry out the various mining functions.

Data reduction

There is variety of repository systems that houses large volumes of data for the purpose of mining. On one hand DM systems are designed to handle massive amounts of data. On the other hand, such cheer volumes of data make the mining process somewhat slow. So, the DM process can benefit from some of the tools and techniques that can be applied to speed it up the process, including data reduction. Data reduction techniques minimize the size of the data by aggregating attribute values, eliminating unused or redundant features [16,17]. Data reduction techniques are used to obtain reduced version of the data set which is the least in volume, yet closely maintains the integrity of the original data. Mining on reduced data set should be more efficient as long as it produces the same (or almost the same) analytical results. There are number of strategies by which data reduction can be accomplished and some of these are [18,19]:

- Data aggregations technique use aggregation operations to produce data cubes with fewer entries cube.
- Dimensionality reduction techniques remove irrelevant (unused), weakly redundant or relevant attributes (dimensions) from the data.
- Data compression techniques uses of some encoding techniques scheme to reduce the data size.
- Discretization and the usage of concept hierarchies can be beneficial in reducing the data by replacing the actual data values (primitive) with higher concepts in the concept hierarchy.
- Numerosity reduction strategies substitute an estimation model for the real data values. Both parametric and nonparametric approaches can be used.

When used before the mining process, these methods can significantly improve both the time needed for the actual mining process and/or the overall quality of the patterns that are mined.

Decision trees

A decision tree is a type of classification procedure with a tree like graphical structure. Given a set of pre-classified instances (objects or examples) in attribute-value representation with their classes, the primary idea is to produce knowledge in the decision tree format [20-22]. A decision tree consists of following:

- Leaf-nodes (class node) that reference a class, or

- Non-leaf nodes (decision node) which indicate an attribute whose values branch to another decision trees.

A decision tree is a tree in which each Non-leaf node branches on the values of the attribute in the given node and each Leaf-node represents a classification (class). Based on the classifications that such trees produce, it is relatively simple to gather important information and make decisions. Decision tree induction algorithms have been used for classification in a wide range of applications. Top-Down Induction Decision Trees (TDIDT) is a method for constructing a decision tree in a top-down fashion. Creating classification rules that may be used to infer an object's class from its attribute values is the induction task. The TDIDT is a popular concept learning approach and the learned concepts can be expressed in the form of "IF – THEN" rules which are more flexible representation than tracing the decision tree.

At each stage of the process of constructing a decision tree, the algorithm must pick an attribute that has not been selected yet in a random fashion. This process is repeated until each branch ends up with a leaf-node that will be labeled with one of the given classes in the training set. Such randomness in the process poses serious problems such as:

- It is possible for the number of leaves to match the number of occurrences.
- Because the classification technique does not specify which attributes should be chosen in what order, the decision tree is just a look-up table and is not very helpful. Different attribute selection order produces different decision trees.

Such problems were solved with the emergence of the ID3 (Iterative Dichotomiser3) algorithm that was introduced by Quinlan 1986 [23-25]. The ID3 algorithm has enhanced the original classification Cluster Learning System (CLS) algorithm by adding two features. These are *windowing* and information theoretic heuristic *entropy*. The windowing is to choose a random subset of the training set to use for the initial tree. The tree is then used to categorize the remaining input cases. The process is completed if the tree correctly classifies these input cases and is accepted for the training set. If not, the misclassified cases are added to the window and the procedure is repeated until all the objects in the tree have the correct categorization assigned to them by the tree. The information theoretic heuristic is used to decide on the order on which the attributes are selected. Calculating the proportions of positive and negative training states currently available at a node is the first stage in applying theoretical heuristic information. In the case of the root node, these are all instances in the training set. A value known as the information needed for the node is calculated using the following formula:

$$I(p, n) = -p \log_2 p - q \log_2 q$$

Where p is the proportion of positive cases $p/(p + n)$ and q is the proportion of negative cases $n/(p + n)$ at the node. This measure is known as E-score (entropy). All available attributes at the node are considered for calculation. Available attributes are those that have not been used so far in the path from the root to the current node.

$$E(A) = \sum_{i=1}^{k} \frac{(p_i + n_i)}{(p + n)} I(p_i, n_i)$$

$$Gain\ (A) = I\ (p, n) - E(A)$$

A simplified version of the ID3 algorithm is as follows:

- **If** all the states have the same class value, then returns the decision tree (Simply this value not really a tree)
- **Else**
  A. Compute Gain values for all attributes and choose an attribute with the highest value and create a node for that attribute.
  B. Make a branch from this node for all value of the attribute
  C. determine all possible values of the attribute to branches.
  D. Follow each branch by dividing the dataset to be only states whereby the value of the branch is present and then go back to 1.

The benefits of utilizing decision trees, as stated by [3], include reasonable training time, quick application, easy interpretation, simple implementation, and ability to handle big number of features.

Design and implementation

The DRS is designed and implemented into several processors (steps); each of which has a special purpose, requirements and limitations. The data entry process is to provide the system with the data. The provided data must comply with the following conditions:

- The data set must be stored in a database (DBF).
- The data values must be nominal.
- The data have no missing values.
- The class attribute (supervised data) must be the last one.
- The class attribute value must be binary.

The preprocessing step is to remove the irrelevant attributes that do not contribute to the systems' objectives. The attributes that must be removed are such as:

- Single valued attributes.
- Unique values attribute (i.e. record numbers).

The main process is performed twice for the DRS to meet its objectives. The first one uses the full set of data, while the second one uses the reduced set. The output process shows the system's ultimate outcome, which includes the decision trees both before and after data reduction, along with the processing times for both cases.

Cace Study

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Systemica 1(1), pp. 145-157, 1990. This study evaluates cars according to the following concept structure: "car acceptability":

- price means overall price
- buying means buying price
- maint means price of the maintenance
- tech means technical characteristic
- comfort means comfort
- doors mean number of doors
- persons mean capacity in terms of persons
- lug-boot means the size of luggage boot
- safety means estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept [5], the model includes three intermediate concepts: price, tech, comfort. The Car Evaluation Database includes examples where the structural details have been deleted., i.e. directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety. The database size and run time criteria are used to compare the two subsystems. The comparison criterion is the run time and data size. The two processing times, one using the original data and the other using the reduced data for creating the same decision tree we will be compared. The data set used in this study is the Car Evolution data set [5]. The original data set consisted of 1728 objects, 6 attributes and one class attribute as describe in Table (1).

Table (1): The data set details.

Attribute No.	Attribute Name	Attributes Values
1	Buying	Vhigh, high, med, low
2	Maint	Vhigh, high, med, low
3	Doors	2, 3, 4, 5more
4	Persons	2, 4, more
5	Lug_boot	Small, med, high
6	Safety	Low, mid, high
7	Class	unacc, acc, good, vgood

The program is designed to accept all values in nominal form. So, it will change the values to a string which number (2) is written as (two) and number (3) is written as (three). Since the decision will be either to accept or not to accept the car, the final attribute will be as follow: (unacc) written as (unacc), and (acc, good and vgood) written as (acc). After applying the program, the result rules were described in figure (1). Which all attributes are used (original data). And the run time will be described in Figure (2), which the run time is consuming in (6.385s).

In this study, the processing times for creating the same decision tree using the original and reduced data sets will be compared. Considering that the first time involves using the algorithm and creating the decision tree process, the second time involves creating the decision tree process and reducing the processing time. We get the same rules after the data reducing (remove the unused attributes), which described in Figure (3). Removed attribute is "doors". And the run time will be described in Figure (4), which the run time is consuming in (5.132s). The decision tree for car evaluation data set is describe in Figure (5).

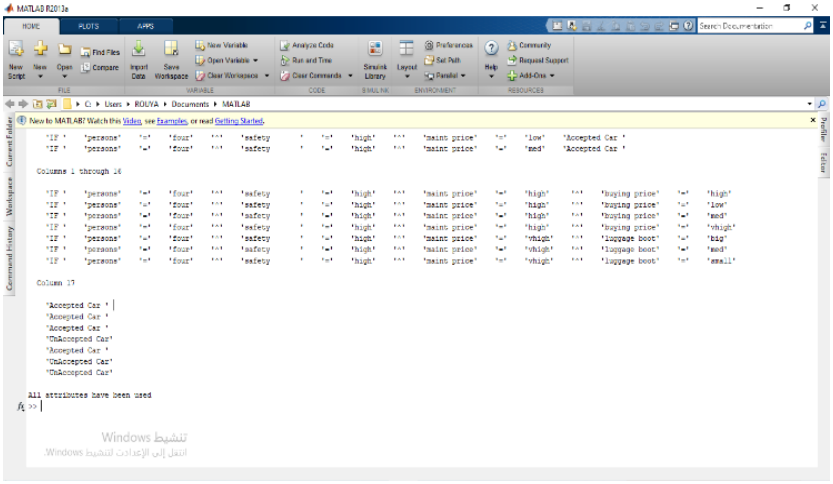

Figure (1): describe the result rules with original data

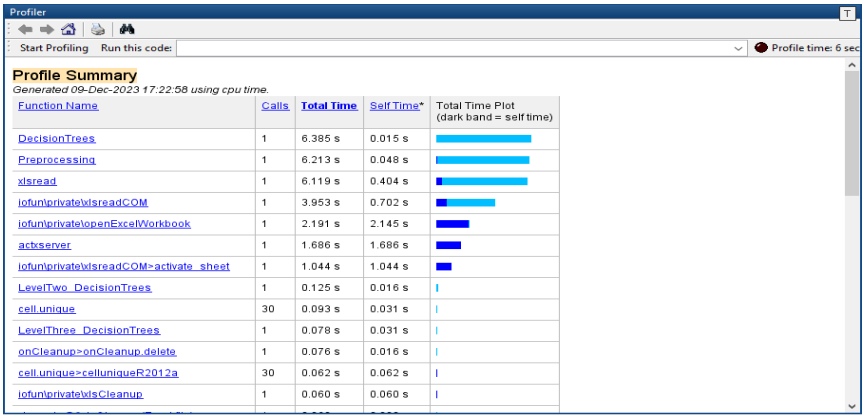

Figure (2): describe the run time with original data.

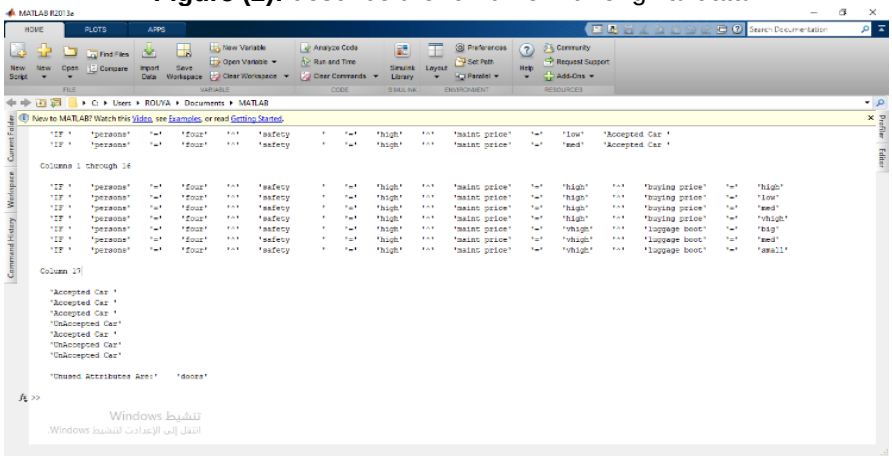

Figure (3): The rules with reduced data.

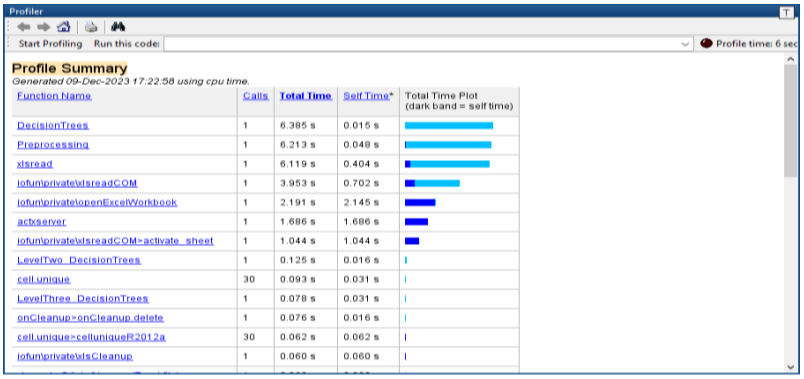

Figure (4): describe the run time with reduced data.

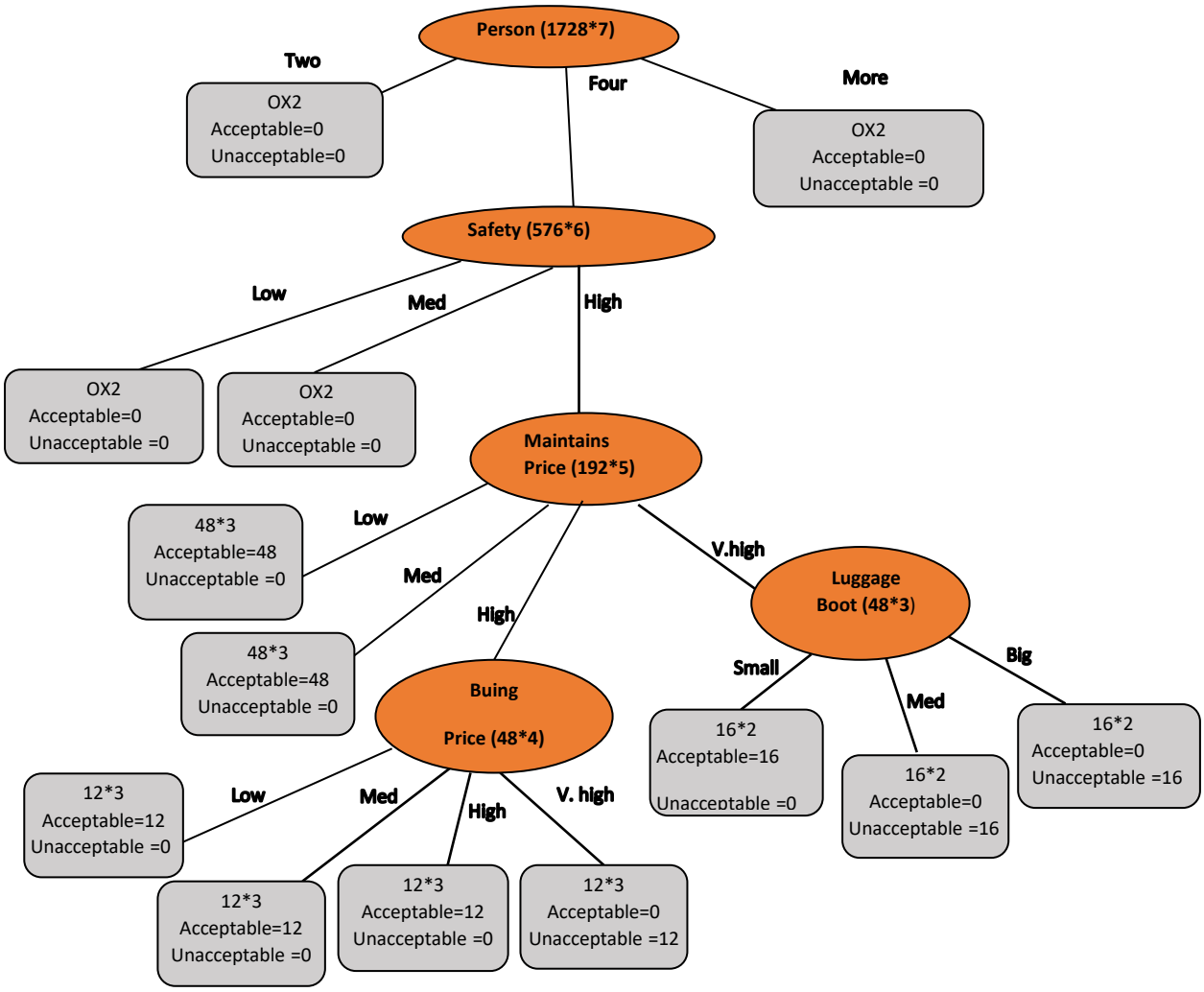

Figure (5): The decision tree for car evaluation data set.

This study worked on a device with the following specifications: Toshiba satellite, processor intel(R) core (TM) i7, cpu@2.20GHZ and Ram is 6GB. windows 10 pro.

Conclusion

From the findings of this study, it can be deduced that the time required to derive rules from a reduced dataset, obtained by eliminating non-contributory attributes, is shorter than the time required to obtain the same rules from the original dataset containing all attributes. Specifically, the post-reduction rule extraction time comprises two components: the preprocessing time required for data reduction and the subsequent processing time for rule generation. In contrast, the pre-reduction extraction time consists of the data loading time and the processing time required to generate rules from the complete dataset. Thus, after reduction, the total computational time is the sum of data loading,

preprocessing (data reduction), and rule extraction times, whereas before reduction, it is limited to the sum of data loading and rule extraction times.

Refrenceses

[1] M. Khaleel, Y. Nassar, and H. J. El-Khozondar, “Towards utilizing Artificial Intelligence in scientific writing,” *Int. J. Electr. Eng. and Sustain.*, pp. 45–50, 2024.

[2] B. Ahmaderaghi, E. Barlaskar, O. Pishchukhina, D. Cutting, and D. Stewart, “Enhancing students’ performance in computer science through tailored instruction based on their programming background,” in *2024 IEEE Global Engineering Education Conference (EDUCON)*, 2024.

[3] Z. Wang, Q. Shen, S. Bi, and C. Fu, “AI empowers data mining models for financial fraud detection and prevention systems,” *Procedia Comput. Sci.*, vol. 243, pp. 891–899, 2024.

[4] M. M. Sabry Aly *et al.*, “The N3XT approach to energy-efficient abundant-data computing,” *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 107, no. 1, pp. 19–48, 2019.

[5] M. Khaleel, A. Jebrel, and D. M. Shwehdy, “Artificial intelligence in computer science: Https://doi.org/10.5281/zenodo.10937515,” *Int. J. Electr. Eng. and Sustain.*, pp. 01–21, 2024.

[6] C. Llatas, B. Soust-Verdaguer, L. C. Torres, and D. Cagigas, “Application of Knowledge Discovery in Databases (KDD) to environmental, economic, and social indicators used in BIM workflow to support sustainable design,” *J. Build. Eng.*, vol. 91, no. 109546, p. 109546, 2024.

[7] M. Resell *et al.*, “Knowledge discovery in databases of proteomics by systems modeling in translational research on pancreatic cancer,” *Proteomes*, vol. 13, no. 2, 2025.

[8] B. C. Pijanowski *et al.*, “Soundscape analytics: A new frontier of knowledge discovery in soundscape data,” *Curr. Landsc. Ecol. Rep.*, vol. 9, no. 4, pp. 88–107, 2024.

[9] M. J. Belizón, D. Majarín, and D. Aguado, “Human resources analytics in practice: A knowledge discovery process,” *Eur. Manag. Rev.*, vol. 21, no. 3, pp. 659–677, 2024.

[10] C. Chen, L. Hao, B. Bai, and G. Zhang, “Knowledge discovery from database: MRI radiomic features to assess recurrence risk in high-grade meningiomas,” *BMC Med. Imaging*, vol. 25, no. 1, p. 14, 2025.

[11] Z. Hamid, F. Khalique, S. Mahmood, A. Daud, A. Bukhari, and B. Alshemaimri, “Healthcare insurance fraud detection using data mining,” *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, p. 112, 2024.

[12] V. S. R. K. Kumar and S. Brooks, “Visual analysis of oceanic data for marine ecosystems,” *Ecol. Inform.*, vol. 82, no. 102762, p. 102762, 2024.

[13] A. Razaque *et al.*, “Review of malicious code detection in data mining applications: challenges, algorithms, and future direction,” *Cluster Comput.*, vol. 28, no. 3, 2025.

[14] C. Zhang and J. Han, “Data mining and knowledge discovery,” in *Urban Informatics*, Singapore: Springer Singapore, 2021, pp. 797–814.

[15] A. Kousis and C. Tjortjis, “Data Mining algorithms for smart cities: A bibliometric analysis,” *Algorithms*, vol. 14, no. 8, p. 242, 2021.

[16] P. K. Vishwakarma and N. Jain, “A review of data analysis methodology,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022.

[17] Q. Zheng, Y. Xu, H. Liu, B. Shi, J. Wang, and B. Dong, “A survey of tax risk detection using data mining techniques,” *Engineering (Beijing)*, vol. 34, pp. 43–59, 2024.

[18] A. Sayal, A. Gupta, C. Vasundhara, Yashas, V. Gupta, and H. Maheshwari, “Crime detection using data mining techniques,” in *2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, 2024.

[19] A. Alam and A. Mohanty, “Predicting students’ performance employing educational data mining techniques, machine learning, and learning analytics,” in *Communications in Computer and Information Science*, Cham: Springer Nature Switzerland, 2023, pp. 166–177.

[20] M. Hajihosseinlou, A. Maghsoudi, and R. Ghezelbash, “A novel scheme for mapping of MVT-type Pb–Zn prospectivity: LightGBM, a highly efficient gradient boosting decision tree machine learning algorithm,” *Nat. Resour. Res.*, vol. 32, no. 6, pp. 2417–2438, 2023.

[21] V. G. Costa and C. E. Pedreira, “Recent advances in decision trees: an updated survey,” *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4765–4800, 2023.

[22] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, “Decision trees: from efficient prediction to responsible AI,” *Front. Artif. Intell.*, vol. 6, p. 1124553, 2023.

[23] K. Wang, L. Wang, and J. Sun, “The data analysis of sports training by ID3 decision tree algorithm and deep learning,” *Sci. Rep.*, vol. 15, no. 1, p. 15060, 2025.

[24] S. Gupta *et al.*, “The iterative dichotomiser 3 method improves soft skills training decisions using survey data,” in *2024 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, 2024, pp. 13–19.
[25] M. A. A. Abdelaziz, M. A. Ghonim, J. Wu, and A. M. A. Almandooh, “ID3-driven insights: the proactive prosumer’s role in technological innovation,” *TQM J.*, 2024.