Unlocking the Potential: The Crucial Role of Data Preprocessing in Big Data Analytics

Praveen Kantha
Chitkara University School of Engineering and Technology,
Chitkara University
Himachal Pradesh, India
praveen.kantha@chitkarauniversity.edu.in

Vijay Kumar Sinha
Chitkara University School of Engineering and Technology, Chitkara University,
Himachal Pradesh, India
vk.sinha@chitkarauniversity.edu.in

Durgesh Srivastava
Department of CSE
Chitkara Institute of Engineering & Technology,
Chitkara University, Rajpura, Punjab, India
drdkumar.ptu@gmail.com

Basant Sah
Dept of CSE,
Koneru Lakshmaiah Education Foundation
Guntur, AP
basantbitmtech2008@gmail.com

2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI) | 979-8-3503-3842-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/IDICAIEI58380.2023.10406577

Abstract— **Access to the internet can significantly enhance the capabilities and opportunities in the field of data mining. It provides a vast source of data, tools, and resources that can be leveraged to improve the data mining process. The Internet offers access to a wide range of data sources, including social media, websites, online databases, and more. The effectiveness of the data mining process depends on the ability to extract from a large dataset meaningful patterns and models. The goal of data mining is to uncover previously unknown information inside large databases. However, the information in the current datasets is not always unified and clean. Despite extensive work on the part of developers and fine-tuners, data mining models remain highly dependent on the quality of the data they are fed. The focus of this research is on the steps taken before feeding data into a machine-learning system. Any machine learning algorithm's major success is predicated on the caliber of the input data it uses. Even though many aspects influence how well Machine Learning (ML) performs a job, the representation and quality of the instance data remain key components in the algorithm's overall effectiveness. The process of knowledge discovery becomes increasingly challenging during the training phase when there is an abundance of irrelevant and duplicated information, along with noisy and unreliable data. Data preparation and filtering steps in ML problems are well known to cost a substantial amount of processing time. Data pre-processing produces the final training set. Access to data, secure data handling, a robust network infrastructure, and the support of the IT industry are all critical components of successful data mining endeavors. Hence, this article offers strategies for optimizing data collection performance at every stage of data preprocessing.**

Keywords—Data mining, data cleaning, feature selection, Internet, Security, Network infrastructure

I. INTRODUCTION

'Data' is one of the most significant considerations for every data analyst. In reality, the first and most important issue that any analyst must solve is the representation and quality of the data used for analysis. Three fundamental properties of data sets are the quantity of patterns, dimensions, and classifications. The effectiveness of classifiers is significantly influenced by these characterizations. In this paper, we look into the different ways of preprocessing data and show that if a good preprocessing method is used, a data set should have a better classification performance and a structural variation [1-3]. There are six basic data preparation techniques that are often used: normalized, exponent change, local rate change, global rate change, principal component analysis (PCA), and field deletion. We suggest an essential approach to assess the efficiency of a data preprocessing method[4] since not all data preprocessing methods are effective for any given data set.

II. DATA PREPROCESSING

In each data mining project, the initial stage is referred to as "data preprocessing," involving the conversion of raw data into a practical format. Data collected in the real world is typically unreliable, inconsistent, and inaccurate. Engaging in data preprocessing has proven to be an effective strategy for mitigating these issues. In order to get data ready for analysis, preprocessing is the first step. Some characteristics may be redundant, while others may be irrelevant and noisy, therefore feature selection (FS) is a procedure to choose the most informative features. Preprocessing of the dataset is necessary when it contains useless data that is sparse (missing), noisy (outliers), and inconsistent[5].

III. REASONS FOR APPLYING DATA PREPROCESSING

Data preprocessing is vital because it enables the quality of raw experimental data to be improved. Preprocessing aims to minimize or remove systematic and random sources of error in experimental data, such as baseline drifts and noise contributions linked to instrumental measurements, respectively[6].

Real-world datasets of today are frequently quite huge, which makes them extremely liable to noise, missing, and inconsistent data owing to human and mechanical errors. Affected data is referred to as "dirty" data. Several methods have been developed over the past few decades to preprocess data obtained from real-world applications before the data is processed further for other reasons.

Real-world data is "dirty" for these reasons[8,9]:

- **Incomplete**: missing values for attributes, lacking certain desirable features
  - e.g., Resting BP=“”

- **Noisy:** containing outliers or errors
  - e.g., Resting BP =“-40”
- **Inconsistent:** including inconsistencies in names or codes, such as
  - e.g., was Person_Age = “42”, now Birthday= “03/07/1997”
  - e.g., was rating “1,2,3”, now rating “A, B, C”
  - e.g., discrepancy between duplicate records
- If there is no quality data, there will be no quality mining results.
- Quality judgments must be founded on quality data; for example, identical or inaccurate data might result in erroneous or inaccurate results.

IV. TECHNIQUES FOR DATA PREPROCESSING

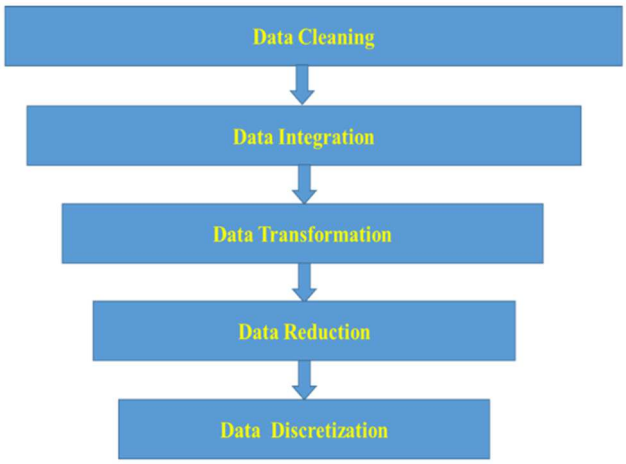

Fig 1: Steps of Data Preprocessing [18]

A. Data Cleaning

Data cleansing offers strategies for dealing with soiled data. Data cleaning methods include value imputation, smoothing noisy data, finding and eliminating outliers, and resolving discrepancies. Dirty datasets may hinder data exploration and analysis. Discretization-based smoothing approaches minimize data by reducing attribute values. Detecting outliers using clustering helps reduce noise[11].

1) Missing Values Handling

There are certain methods for handling missing data that would be best suited for the machine learning algorithm. A method for avoiding instances with uncertain feature values, Since it excludes scenarios with uncertain feature values, this method is the simplest.

- Use a special value to represent the absence of a value; sometimes the absence of a value has significance. The lack of uric acid in a patient's medical record suggests that the patient did not have a kidney function test. Consequently, it makes sense to use a specific value, such as -1, because it can be operated on like a normal value but has a special meaning in the dataset [12].
- To compensate for blanks, replace all the blanks, and enter a new constant value (such as "unknown," "N/A," or "minus infinity"). Sometimes it doesn't make sense to try to anticipate the missing value, thus this method is employed instead.
- Replace missing attribute values in the database with the mean (or median, if the property is discrete).In a database of Indian family incomes, for instance, if the average income of an Indian family is X, that value can be substituted for absent income values[7].

2) Outlier Detection

Outlier analysis is a technique for spotting anomalies using a clustering process. Cluster analysis combines together numbers that are statistically very similar, or 'clusters' them, and labels outlying values as "noise." Identifying anomalies in data is crucial. Outlier detection is used to detect anomalies in a variety of disciplines, like computer detection of network intrusions, analysis of gene expression, illness onset identification including cancer diagnosis, detection of financial fraud, and analysis of human behavior. An outlier in an MRI scan, for example, a malignant tumor may be present. Unusual transactions with credit card patterns might be a sign of fraud, and an unusual network traffic pattern can be a sign that a computer has been compromised and is sending sensitive information to an unapproved site.

Types of Outliers

According to the data and detection methods, outliers may be grouped into the following categories[10,13]:

a) Point Outliers

A point outlier is a particular data instance that differs from the rest of the data. Consider automobile fuel expense as a genuine example. You have a point outlier if your typical daily fuel use is 15 liters but on one particular day, it rises to 60 liters.

b) Contextual Outliers

A contextual outlier, also known as a conditional outlier, is an instance of data that exhibits unusual behavior only in a certain context. For example, credit card spending during the holiday season differs from other times of the year. If monthly spending averages 10,000, then spending 50,000 outside of a holiday season might be seen as unusual. It is contextual outliers

B. Data Integration

The concept of "data integration" involves the process of unifying diverse datasets into a cohesive entity. In various databases, certain attributes representing the same concept might have different names, resulting in duplication and inconsistency. For example, the identifier for soil samples could be denoted as "soil id" in one database and "soil_sample_id" in another. Inconsistencies in attribute naming and values can also exist within soil datasets. The presence of a substantial amount of duplicated data can potentially impede or complicate the data mining process. Further data cleaning activities could be required to detect and remove duplicates that may have formed during the data integration process in order to remedy this issue.

C. Data Transformation

If an attribute's value is low in comparison to those of other attributes, then the attribute won't have much of an impact on data mining since both its mean and standard deviation will be low. This means that data transformation is the act of transforming data from its original format into a more suitable one for data mining[14,15].

Data transformation refers to changing the way data are represented so that they may be used as inputs for data mining models and to simplify the data mining model's optimization procedure.

1) Normalization

Data normalization is a preprocessing method that organizes the provided data into a finer format. The performance of a machine learning algorithm is significantly influenced by the quality of the datasets it utilizes. Thus, data normalization is an essential transformation method that may increase accuracy and provide higher performance in evaluated datasets. Recognizing the importance of transformation methods in data mining algorithms, the normalization approach is applied here to boost generalization and learning capacity with few mistakes. When normalization is used, all characteristics are transformed into a normalized score or a range (0, 1).

Min-max normalization and z-score normalization are the two most often used approaches for this scope. This is seldom used for ML algorithm input data, particularly for geographical data.

- Min-max normalization: A feature may be transformed from its original range [lb, ub] to a new range [lbnew, ubnew] via min-max normalization. The typical target range is [0, 1] or [-1, 1]. The normalized value, v', for a sample with value v is calculated as in (1).

$$V' = \frac{v - lb}{vb - lb}(Ubnew - lbnew) + lbnew \quad (1)$$

- Z-score normalization: If the inherent range of a feature is uncertain or outliers are present, the use of min-max normalization might not be suitable and can yield unintended outcomes. Another approach to normalization includes recentering the data to achieve a mean of 0 and a standard deviation of 1. Once you have the attribute's mean and standard deviation, you can express the transformation as depicted in Equation (2).

$$V' = \frac{v - \mu}{\sigma} \quad (2)$$

Note that if μ and σ μ are not known, they can be replaced with the sample mean and standard deviation.

D. Data Reduction

It is often observed that when complicated data analysis and mining procedures are applied to enormous datasets, they take so long that the whole process of analysis or mining the data becomes impractical. In such situations, data reduction techniques come to the scene. The integrity of the original data may be preserved while using data reduction methods to portray a dataset in a more compact way. Data reduction involves minimizing the dimensions (the total number of attributes) or the volume of data. The term 'dimension' in Data warehousing provides structured labeling information. However, it's not always required to consider all dimensions (attributes). Certain approaches to reduce dimensions involve applying a dimensional reduction transformation to a dataset, resulting in new data samples with fewer attributes than the original dataset. These transformations come with unique requirements. Data may have their dimension reduced while maintaining their maximum variance using principal component analysis, or PCA. This is accomplished by performing matrix multiplication with $A = (a_1, a_2, \ldots \ldots a_p)$T and the dataset X, then retaining the top k dimensions. (a_1 represents the normalized eigenvector corresponding to the ith largest eigenvalue of the dataset's covariance matrix [16].

1) Feature selection

Feature selection serves as an alternative method for reducing dimensionality by eliminating irrelevant or correlated attributes from a dataset while retaining the remaining attributes that are relatively independent. Finding a sufficiently effective subset of features for variable prediction is the aim of feature selection. The following are the three kinds of feature selection methods:

a) Filter

In most cases, filters are applied before any further processing is done. No machine learning methods are required for feature selection. Instead, features are chosen based on how well they correlate with the result (target) variable in a battery of statistical tests.

Using a filter approach, you may choose a feature based on attribute-level criteria like information gain, correlation, or chi-square test. The filter technique disregards the data mining paradigm. This strategy takes care of both rating individuals and choosing subsets. Evaluation functions including distance, information, dependency, and consistency (but no classifier) are used to rank the individuals[17]. Using just the most crucial aspects of the data, filter methods determine which genes are important. This model works independently of the induction technique, making it quicker than the wrapper approach and producing a superior generalization[18]. However, it favors selecting subsets with a large number of characteristics (or perhaps all of the features), necessitating the use of a threshold to narrow down the features to consider.

b) Wrapper

The wrapper method creates the necessary subset of features by using any GP in conjunction with the classifier as an evaluation function. Using a blind search, the wrapper method attempts to identify a specific subset of characteristics. The optimal subset is found using a random search, and this can't be guaranteed without obtaining every potential subset[19]. In this approach, feature selection is inherently NP-hard, and with each iteration, the search frequently becomes unmanageable for the user. In an effort to speed things up, greedy methods like forward selection and backward elimination are offered[20].

c) Hybrid Method

It involves blending one or more traditional feature selection techniques, encompassing methods like filters and wrappers. The feature subset obtained through one approach

serves as the input for another selection algorithm, resulting in combinations like filter-filter, filter-wrapper, or filter-filter-wrapper [21] [22]. In most cases, filters are employed to choose the first feature subset or to assist in eliminating duplicate features. To initially choose the feature subset, you can apply a combination of multiple filtering techniques vertically[23][24]. The features are then sent to the wrapper function so that it may choose the best ones. Different assessment criteria are used in this procedure. As a result, it improves efficiency and prediction accuracy while having a lower computing cost for high-dimensional data.

E. Data Discretization

Discretization refers to the process of converting numerical attributes into categorical (nominal) attributes or segmenting a numerical range into discrete subgroups. For instance, a range such as 10-65 can be discretized into three subgroups: (10-23), (24-39), and (40-65). Data discretization techniques minimize the number of values associated with a continuous attribute by splitting its range into intervals [25][26]. These intervals can then replace the actual data values, simplifying the presentation of mining results and rendering them more user-friendly at the knowledge level. Data discretization can be applied either before or after the data mining process. Many real-world datasets feature continuous properties, and certain machine learning algorithms that accommodate both continuous and discrete features have demonstrated superior performance compared to those designed exclusively for discrete values [27][28]. Discretization entails:

- Segment continuous attribute ranges into intervals.
- Note that specific classification algorithms require categorical attributes.
- Employ discretization to diminish data size.
- Get the data ready for in-depth analysis.

IV. CONCLUSION

The paper provides a concise overview of pre-processing and post-processing procedures. It begins with pre-processing methods, including a full discussion of several data cleaning methodologies, and then ignores noisy data, unbalanced data handling, and dimensionality reduction. the significance of data preparation in the context of data mining cannot be overstated, given the inherent challenges posed by real-world data, including its incompleteness, noise, and inconsistency. A wide range of crucial procedures, such as data integration, conversion of data, data reduction, and data purification, are involved in data preparation. The most important of these procedures is cleaning up data, which is essential for addressing problems with noisy data, missing values, outliers, and inconsistent data. Its meticulous execution is the linchpin to enhancing the quality and dependability of the data at the heart of any analytical endeavor. Data integration plays a pivotal role in unifying data from various origins into a unified and consistent repository, achieved through processes like analyzing metadata correlations, resolving data conflicts, and addressing semantic discrepancies. Data transformation is essential for molding data into suitable forms for mining, with normalization being a prime example of how attribute data can be adapted to fit within a predefined range. Conversely, data reduction methods provide a way to condense data into a more concise format while minimizing the loss of essential information. Methods such as data cube accumulation, reduction of dimensionality, compression of data, numerosity elimination, and discretization are examples of these approaches. Concept hierarchies play a pivotal role in organizing attribute or dimension values into progressive levels of abstraction, proving particularly advantageous in multilevel mining scenarios. Their automatic generation can be based on attributes' distinct values, enabling efficient categorization of data. Methods such as data segmentation utilizing segmentation rules, distribution analysis, and analysis of clustering are crucial components of the data preparation toolkit in the case of numerical data. Despite the proliferation of various methods and techniques in the field of data preparation, it remains a dynamic and vital realm of ongoing research. The constantly changing landscape of data and the ever-expanding range of data sources continue to drive the demand for inventive approaches and tools in data preparation, underscoring its enduring and indispensable role in the data mining process.

REFERENCES

[1] Guo, Y. Ping, N. Liu, and S. S. Luo, “A two-level hybrid approach for intrusion detection,” Neurocomputing, vol. 214, pp. 391–400, 2016.

[2] A. Dubey, U. Gupta, and S. Jain. "Medical data clustering and classification using TLBO and machine learning algorithms." Computers, Materials and Continua 70.3, 4523-4543, 2021

[3] D. Srivastava, R. Singh, and V. Singh, “Performance Evaluation of Entropy Based Graph Network Intrusion Detection System (E-Ids)”, in Jour of Adv Research in Dynamical & Control Systems, Vol.- 11, 02-Special Issue, 2019

[4] A. Dubey, A. Kumar, et al. "Performance estimation of machine learning algorithms in the factor analysis of COVID-19 dataset." Computers, Materials and Continua, vol.66, 1921-1936, 2020.

[5] A. A. Aburomman and M. Bin Ibne Reaz, “A novel SVM-kNN-PSO ensemble method for an intrusion detection system,” Applied Soft Computing Journal, vol. 38, pp. 360–372, 2016.

[6] S. O. Al-mamory and F. S. Jassim, “On the Designing of Two Grains Levels Network Intrusion Detection System,” Karbala International Journal of Modern Science, Elsevier, vol. 1, pp. 15–25, 2015.

[7] W. Bul’ajoul, A. James, and M. Pannu, “Improving network intrusion detection system performance through quality of service configuration and parallel technology,” Journal of Computer and System Sciences, vol. 81, pp. 981–999, 2015.

[8] K. Zheng, Z. Cai, X. Zhang, Z. Wang, and B. Yang, “Algorithms to speedup pattern matching for network intrusion detection systems,” Computer Communications, vol. 62, pp. 47–58, 2015.

[9] S. Rastegari, P. Hingston, and C. P. Lam, “Evolving statistical rulesets for network intrusion detection,” Applied Soft Computing Journal, vol. 33, pp. 348–359, 2015.

[10] J. Cervantes, F. García Lamont, A. López-Chau, L. Rodríguez Mazahua, and J. Sergio Ruíz, “Data selection based on decision tree for SVM classification on large data sets,” Applied Soft Computing, vol. 37, pp. 787–798, 2015.

[11] M. S. Gondal, A. J. Malik, and F. A. Khan, “Network Intrusion Detection using Diversity-based Centroid Mechanism,” 12th International Conference on Information Technology - New Generations, pp. 224–228, 2015.

[12] A. Dastanpour and A. Selamat, “Comparison of Genetic Algorithm Optimization on Artificial Neural Network and Support Vector Machine in Intrusion Detection System,” IEEE Conference on Open Systems (ICOS), pp. 72–77, 2014.

[13] Y. Choi, D. H. Kim, K. N. Plataniotis, and Y. M. Ro, “Classifier ensemble generation and selection with multiple feature

representations for classification applications in computer-aided detection and diagnosis on mammography," Expert Systems with Applications, vol. 46, pp. 106– 121, 2016.
[14] C. A. Ronao and S. B. Cho, "Anomalous query access detection in RBAC-administered databases with PART and PCA," Information Sciences, vol. 369, pp. 238–250, 2016.
[15] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," Information Sciences, vol. 378, pp. 484–497, 2017.
[16] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," 2014 International Conference on Computing, Networking and Communication, pp. 797–801, 2014.
[17] A. Karim, R. Salleh, M. Shiraz, S. Shah, I. Awan, and N. Anuar, "Botnet detection techniques: review, future trends, and issues," Computer and Electronics, vol. 15, pp. 943–983, 2014.
[18] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, "A review on feature selection in mobile malware detection," Digital Investigation, vol. 13, pp. 22–37, 2015.
[19] A. Karim, S. Adeel, A. Shah, R. Bin Salleh, M. Arif, and R. Noor, "Mobile Botnet Attacks – an Emerging Threat : Classification , Review and Open Issues," TIIS 9, vol. 9, pp. 1471–1492, 2015.
[20] D. Srivastava, R.Singh and V.Singh, "Analysis of different Hybrid methods for Intrusion Detection Systems," International Journal Of Computer Sciences And Engineering 7(5):757-764, May 2019
[21] S. Garasia, D. Rana, and R. Mehta, "Http Botnet Detection Using Frequent Patternset Mining," Intl. Journal of Engineering Science and Advanced Technology (IJESAT), pp. 619–624, 2012.
[22] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using Machine Learning Techniques to Identify Botnet Traffic," Local Computer Networks, Proceeding. 2006 31st IEEE Conference, pp. 967–974, 2006.
[23] I. Mohammad, R. Pandey and A. Khatoon, "A Review of types of Security Attacks and Malicious Software in Network Security," International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), vol. 4, pp. 413–415, 2014.
[24] D. Srivastava, N. Sainis and R. Singh, "Classification of various Dataset for Intrusion Detection System", in International Journal of Emerging Technology and Advanced Engineering, Volume 8, Issue 1, January 2018.
[25] KDD Cup 1999 Intrusion Detection Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html[Accessed on: Feburary 2017]
[26] M. Xu, N. Ye, "Probabilistic networks with undirected links for anomaly detection", In Proceedings of IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, pp. 175–179, 2000.
[27] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in IEEE Conference on Communications and Network Security (CNS), pp. 247–255, 2014.
[28] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using PARTs," Information Sciences, vol. 278, pp. 488-497, 2014.