Investigation the impact of Data Pre-Processing and Dimensionality Reduction to Improve the Accuracy of Machine Learning

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 14, Issue No. 02, 61–70

https://journals.mirlabs.in/index.php/ijcsnt
DOI-10.18486/ijcsnt/14.2.006
IJCSNT

Viranchee V. Dave[1], Shankarshan Panda[2]

Abstract

Data pre-processing stands as a fundamental stage in machine learning and data mining. Within this domain, normalization, discretization, and dimensionality reduction are widely recognized techniques. This research paper aims to investigate the impact of Min-max, Z-score, Decimal Scaling, and Logarithm to the base 2 on the accuracy of the J48 classifier, utilizing the NSL-KDD dataset. Experiments were conducted employing these methods, and their respective outcomes were systematically compared. Additionally, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were assessed for dimensionality reduction. Notably, a hybrid combination of PCA and LDA was explored, demonstrating an enhanced classification accuracy compared to the individual methods.

Keywords

Data Pre-Processing, Min-Max, Z Score, Decimal Scaling, Log2, PCA, LDA, J48

Received: 17 April 2025; **Revised:** 29 July 2025; **Accepted:** 26 August 2025;
Published: 31 August 2025;

[1, 2]Department of Computer Science, Sabarmati University, Ahmedabad, India.

Corresponding author:
Email: pandasankarsanasind@gmail.com

Introduction

Machine learning (ML) constitutes a well-established field within computer science, primarily focused on uncovering patterns and anomalies in data[1]. In parallel, data mining involves identifying previously unnoticed data associations in a given database, extracting meaningful yet previously undiscovered information from large datasets. The continuous expansion of existing databases has made human analysis challenging, creating both an opportunity and necessity to extract crucial information. While careful data integration is now feasible, the data must undergo suitable transformations before mining can commence.

Data pre-processing, often overshadowed by other stages like data mining, emerges as a critical component in new knowledge discovery processes. Remarkably, it frequently consumes over 50% of the total effort in data analysis[3]. Raw data typically exhibits irregularities such as missing values, noise, inconsistencies, and redundancies, all of which can significantly impact subsequent learning steps. Consequently, a thorough pre-processing step is essential to mitigate the influence of data irregularities, if present, on the quality and reliability of subsequent processing steps.

Another crucial aspect is data transformation, encompassing activities like data generalization, smoothing, normalization, and attribute construction[2]. The accuracy and efficiency of data mining algorithms stand to benefit from effective data transformation, thereby enhancing the overall performance of the process.

Data normalization, a type of data transformation, yields improved results when applied to previously normalized data, i.e., scaled to a predefined range like [0.0, 1.0]. The process of data transformation involves converting a given dataset into a format suitable for data mining[4]. This transformation can encompass activities such as data normalization, aggregation, smoothing, or generalization. Handling vast amounts of data can be time-consuming, prompting the use of data reduction techniques like dimension reduction, data cube aggregation, data compression, discretization, numerosity reduction, and concept hierarchy generation.

The paper's structure unfolds as follows: Section 2 provides a review of related work, while Section 3 delves into dimensional reduction. Section 4 explores data normalization, and Section 5 offers a basic introduction to data discretization. Section 6 introduces the J48 classifier, followed by Section 7, which outlines the experiments and presents the results. Section 8 concludes the study with findings and recommendations for future work.

Related Work

Various research studies have delved into the realm of data preprocessing. Haddad et al[5] introduced innovative two-tier classification models based on machine learning, specifically Naïve Bayes and KNN. The outcomes indicated a promising improvement in false alarm detection rates compared to existing models. Vasan et al[6] employed PCA in experiments with various classifiers on benchmark datasets (KDD CUP and UNB ISCX). The first 10 principal components demonstrated efficacy for classification tasks, achieving accuracy rates of approximately 99.7% and 98.8% on KDD CUP and UNB ISCX, respectively. This accuracy was comparable to using the original 41 features for KDD and 28 features for ISCX. Wimmer and Powell[7] explored the impact of using PCA as a feature reduction method on decision trees (DT), observing an enhancement in the classifiers' accuracy and a simplification of resulting DT structures.

Al Shalabi[2] investigated different normalization methods, testing each against the ID3 methodology on the HSV dataset. The study considered factors such as the number of leaf nodes, tree growing time, and classifier accuracy. Ramírez-Gallego et al[3] conducted an analysis, summarization, and categorization of contributions on data preprocessing for streaming data. The study evaluated contributions based on reduction rates, predictive performance, memory usage, and computational time. Ramasamy[8] analyzed existing preprocessing techniques in terms of their prediction accuracy. The study reported a notable increase in prediction accuracy, reaching up to 90%, after raw data preprocessing. The Naïve Bayes classifier achieved the highest prediction accuracy and a lower error rate compared to Logistic.

Saranya & Manikandan[9] explored the feasibility of achieving privacy through normalization techniques and compared their results. Experimental findings revealed that the Min-Max normalization method achieved the minimum misclassification error. Applying all three normalization methods to a dataset with values of (10, 15, 20, 24, 30, 37, 40, 45, 50, 60) showed that Min-Max normalization consistently yielded the lowest misclassification error rate compared to Decimal Scaling and Z-Score.

Dimensionality Reduction

Dimension reduction primarily aims to represent a high-dimensional (HD) dataset in a low-dimensional (LD) space while retaining the HD structures, such as outliers and clusters, of the dataset[10]. One advantage of dimension reduction, especially in visualizing HD data, is its scalability. However, a drawback is the inevitable loss of information during the transformation of data into the LD projection. The following section discusses two common dimension reduction algorithms, namely PCA and LDA.

A. Principal Components Analysis (PCA)

Principal Components Analysis (PCA) serves as a feature extraction method that generates new linear combinations of the initial features[6]. It maps each example in a given dataset from a d-dimensional space to a k-dimensional subspace, where k $<$ d, and the new set of generated k dimensions is referred to as Principal Components (PC). Each PC is oriented towards maximum variance, excluding the variance already accounted for in preceding components. The first component covers the maximum variance, with subsequent components covering decreasing amounts of variance. The PC can be mathematically represented as follows:

$$PC_i = a_1X_1 + a_2X_2 + \ldots + a_dX_d \tag{1}$$

where PCi is Principal Component ‘i’; Xj is the original feature ‘j’; aj is the numerical coefficient for Xj.

B. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is primarily employed as a reduction method when aiming to reduce computation time complexity and enhance classification[5]. Used in image processing, signal processing, bankruptcy, and market analysis problems, LDA complements PCA’s effectiveness in feature extraction but focuses on discrimination tasks. LDA transforms a highly dimensional feature into a lower-dimensional space by selecting an optimal projection matrix while preserving vital information for data classification. The LDA process involves defining two scatter matrices: SB, the inter-class scatter matrix, and SW, the intra-class scatter matrix.

Normalization

Attribute normalization involves scaling values to fit within a specified range, a crucial step for classification frameworks utilizing distance measurements or neural networks[2]. Normalization significantly enhances the learning speed in neural network back propagation algorithms, particularly when applied to input values for each measured attribute in the training set. In distance-based techniques, normalization ensures that attributes with initially smaller ranges do not get overshadowed by those with larger ranges. Common data normalization techniques include decimal scaling, Z-score normalization, and min-max normalization.

A. Min-Max

Min-max normalization aims to linearly transform the original data[11]. Assuming minA and maxA as the minimum and maximum values of an attribute A, the transformation maps a value, vi, of A to Yi in the range [new-minA, new-maxA] using the formula:

$$Y_i = \frac{v_i - min_A}{max_A - min_A} * (newmax_A - newmin_A) + newmin_A \tag{2}$$

The relationship between the original dataset values is preserved after min-max normalization. However, an "out-of-bounds" error may occur if a future normalization input case falls outside the original data range for A.

B. Z-Score

Z-score normalization relies on the mean and standard deviation (SD) of the attribute for normalizing values[11]. The transformation of a value, vi, of A to Yi is calculated as:

$$Y_i = \frac{v_i - \bar{A}}{\sigma_A} \tag{3}$$

Where $\bar{A}$ and σ_A represent the mean and SD respectively, of A,

$$A = \frac{1}{n} * (v_1 + v_2 + \ldots + v_n) \tag{4}$$

Where σ_A is the computed square root of the variance of A.

Z-score normalization is particularly useful when outliers dominate the min-max normalization process or when the actual minimum and maximum of attribute A are unknown.

C. Decimal Scaling

Decimal scaling involves shifting the decimal point of attribute values, a normalization process dependent on the absolute maximum value of the attribute[11]. The normalization of a value v of A to v' is performed using the formula:

$$V_i' = \frac{v_i}{10^j} \tag{5}$$

Where j represents the least integer that satisfies Max ($|V_i'|$) <1.

Discretization

Discretization refers to the conversion of continuous variables into discrete ones, achieved by dividing the range of values of continuous variables into a finite number of subranges known as intervals, buckets, or bins[12]. Discretization algorithms partition continuous variables into a defined number of intervals, treating them as distinct categories. To prevent biases, logarithm to the base 2 is applied to discretize continuous-valued features before projecting the result to an integer[5]. The transformation for each continuous-valued variable z is carried out using the following equation:

$$\text{if } (z \geq 2)\, z = \int (z + 1) \tag{6}$$

J48 Classifier

This represents a straightforward C4.5 decision tree primarily employed for classification tasks[13]. Although it generates a binary tree, the decision tree method is widely utilized for solving classification problems. The classifier constructs a tree to model the classification process, and this tree is then applied to each tuple in the database for classification. Notably, the J48 classifier does not consider missing values during tree construction, meaning the value of a missing item can be predicted based on the information gleaned from the values of other attributes. The primary objective is to partition the data into ranges based on the attribute values for items in the training set. The J48 classifier allows the classification of attributes using either decision trees or the rules generated from them.

Experiments and Results

Our experiments aimed to assess the efficacy of various preprocessing methods, including Log2, Min-max, Z-score, decimal scaling, LDA, and PCA, in enhancing classification accuracy. These methods were evaluated using the NSL-KDD benchmark dataset, an improved version of the KDD'99 dataset with 42 attributes. The enhancements involved removing redundant instances from the training dataset to prevent biased classification results[14]. The NSL-KDD dataset includes 20% of attributes for training (25192 instances) and 2254 instances for testing. The class attribute labeled 42 indicates whether each instance represents a normal connection or an attack.

Several improvements were made to the NSL-KDD dataset for intrusion detection systems[15]:

1. The training dataset lacks redundant records to keep the classifier unbiased for more frequent records.

2. The testing dataset has no duplicate records, enhancing learner performance and preventing bias from methods favoring frequent records.

3. An inverse relationship between the number of records selected at each difficulty level group and the percentage of records in the KDD dataset improves classification rates for different machine learning methods.

In our experiments, Log2 was employed as a data discretization method, and three normalization techniques were studied. Dimensional reduction was performed using LDA and PCA, while the J48 classifier, an extension of ID3, was used for classification[16]. The WEKA data mining tool, offering tree pruning options to address potential overfitting, was utilized for the experiments.

The first experiment involved applying Logarithm to base 2 to the dataset, implemented in C sharp as it was not provided in WEKA. Subsequently, the J48 classifier was applied to assess classification accuracy, with 70% of the dataset used for training and 30% for testing. Normalization techniques, including Min-max, Z-score, and decimal scaling, were implemented in WEKA, followed by the application of the J48 classifier to determine classification accuracy. The results obtained were presented in Table I, and Figure 1 provided a visual comparison of the results.

Table 1. The accuracy of the evaluated preprocessing methods

Pre-Processing Method	Classification Accuracy (%)	Time for Model Building (Sec)	Time for Model Testing (Sec)
Log2	99.66	16.4	0.12
Min-max	99.66	17.7	0.06
Z-score	99.66	21.05	0.04
Decimal Scaling	99.66	18.43	0.13

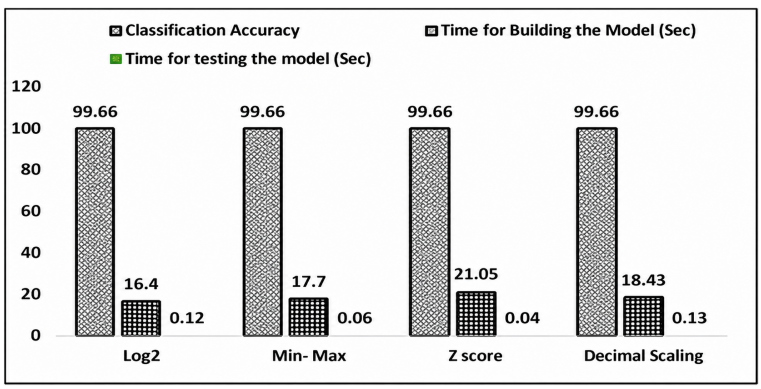

Figure 1. A comparison of the preprocessing methods

To evaluate the performance of the above methods, the results in Table I were compared. As seen in Table I, the value of the classification accuracy for all the pre-processing techniques was the same (99.66 %). However, the values of the time to build the model and the time of testing the model varied. Log2 was observed as a faster technique in terms of the time to build the model (16.4 sec). However, Min-max was faster (17.7 sec) among the normalization methods in terms of building the model while Z-score was the slowest (21.05 sec). Meanwhile, Z-score was faster than the others (0.04 sec) in testing the model.

The second experiment was dimensional reduction using LDA and PCA algorithms. For each algorithm, five different number of features (5, 10, 20, 30, 40) were used to test the classification accuracy. After applying these algorithms, the J48 classifier was applied for the classification purpose. Tables II and

III showed the results of the LDA and PCA, respectively. Figures 2 and 3 compared the results of feature reduction using LDA and PCA, respectively.

Table 2. Dimensional reduction results using LDA

No. of features	Classification accuracy (%)	Time for model building (sec)	Time for model testing (sec)
5	53.36	0.02	0.12
10	99.08	0.04	4.6
20	99.11	0.04	6.72
30	98.99	0.04	10.57
40	99.02	0.04	15.21

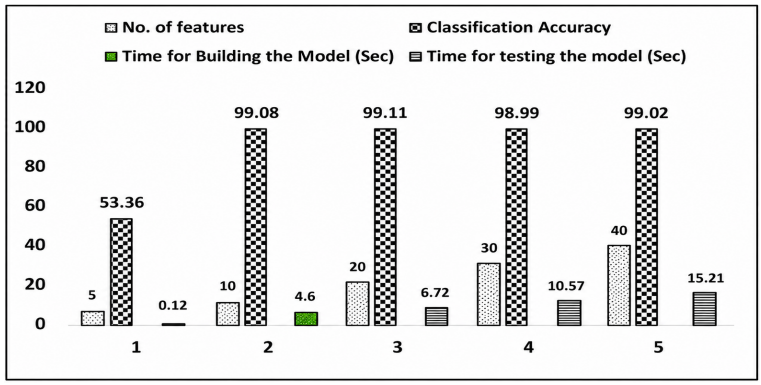

Figure 2. Results of feature reduction using LDA

Table 3. Dimensional reduction results using PCA

No. of features	Classification accuracy (%)	Time for model building (sec)	Time for model testing (sec)
5	99.28	0.03	2.78
10	99.50	0.04	5.12
20	99.52	0.04	12.69
30	98.53	0.04	17.46
40	99.53	0.04	22.47

Examining Table II reveals that the accuracy with only five features was relatively low (53.36%), while the optimal accuracy of 99.11% was achieved with 20 features. Notably, the model-building time

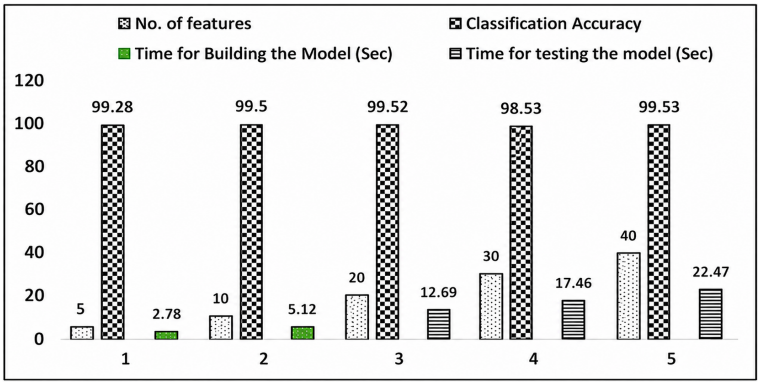

Figure 3. Results of feature reduction using PCA

was 0.04 sec, and the model-testing time was 6.72 sec. In Table III, high accuracy was also attained with 20 features (99.52%). The corresponding model-building time was 0.04 sec, and the model-testing time was 12.69 sec.

In the second experiment, it was observed that the model exhibited increased speed and higher accuracy when reducing the number of features to 50% for both LDA and PCA. Considering our dataset with only 42 attributes and 22544 instances, the data preprocessing time remained manageable. Given the critical importance of processing time for extensive datasets, reducing attributes by 50% proved advantageous, yielding high accuracy with minimal time.

For the third experiment, a hybrid approach utilizing both LDA and PCA with a selection of 20 features (10 from each) was employed. The J48 classifier was then applied for classification, resulting in an increased accuracy of 99.56% compared to LDA or PCA individually. However, the model-building and testing times experienced slight increments (15.22 sec and 0.22 sec, respectively). Nevertheless, these outcomes were promising, particularly in the context of handling extensive datasets.

Conclusion and Future Work

Data pre-processing stands as a pivotal phase in data mining. This study delves into the impact of pre-processing methods on the classification outcomes of the J48 classifier, utilizing the NSL-KDD dataset in the experiments. Pre-processing techniques such as Min-max, Z-score, decimal scaling, and Log2 were applied to the original dataset. Despite all evaluated methods yielding a classification accuracy of 99.66%, Log2 emerged as the fastest among them in terms of processing time.

In the realm of dimension reduction, PCA and LDA were employed. The experiments indicated that the optimal ratio for feature reduction was 50% of the original dataset for both PCA and LDA, enabling high accuracy with minimal time consumption. This reduction proves particularly beneficial when dealing with extensive datasets, commonly referred to as big data. The application of the hybrid PCA-LDA method not only enhanced classification accuracy but also incurred a slight increase in processing time.

In future work, our focus will extend to the exploration of additional preprocessing techniques and the investigation of alternative dimension reduction methods, as well as their potential combinations.

References

[1] G. D. S. H. Hamed Haddad Pajouh, "Two-tier network anomaly detection model: a machine learning approach," Journal of Intelligent Information Systems, Springer, vol. 48, p. 61–74, 2015.

[2] B. S. K. Keerthi Vasan, "Dimensionality reduction using PrincipalComponent Analysis for network intrusion detection," Elsevier, vol. 8, p. 510—512, 2016.

[3] L. P. Hayden Wimmer, "Principle Component Analysis for Feature Reduction and Data Preprocessing in Data Science," in Information Systems & Computing Academic Professionals (ISCAP), Las Vegas, Nevada USA, 2016.

[4] M. D. a. N. Ramasamy, "A Comparison of the Perceptive Approaches for Preprocessing the DataSet for Predicting Fertility Success Rate," International Journal of Computer Technology and Applications, pp. 255-260, 2016.

[5] G. C.Saranya, "A Study on Normalization Techniques for Privacy Preserving Data Mining," International Journal of Engineering and Technology (IJET), vol. 5, no. 3, 2013.

[6] I. C. N. R. John Wenskovitch, "Towards a Systematic Combination of Dimension Reduction," IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, vol. 24, 2018.

[7] M. K. a. J. P. Jiawei Han, Data Mining Concepts and Techniques, Elsevier, 2012.

[8] F. Y. Zeynel Cebeci, "Comparison of Chi-square based algorithms for discretization of continuous chicken egg quality traits," Journal of Agricultural Informatics, vol. 8, 2017.

[9] M. S. S. S. Tina R. Patil, "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," International Journal of Computer Science and Applications, vol. 6, 2013.

[10] S. K. S. Preeti Aggarwal, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," Elsevier, 2015.

[11] M. C. Uzair Bashir, "Performance Evaluation of J48 and Bayes Algorithms for Intrusion Detection System," International Journal of Network Security & Its Applications (IJNSA), 2017.

[12] G. S. Tomar, S. Verma and A. Jha, "Web Page Classification using Modified Naïve Bayesian Approach," *TENCON 2006 - 2006 IEEE Region 10 Conference*, Hong Kong, China, 2006, pp. 1-4, doi: 10.1109/TENCON.2006.344193.

[13] K. K. Arora, G. S. Tomar and S. S. Agrawal, "Studying the Role of Data Quality on Statistical and Neural Machine Translation," *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, 2021, pp. 199-204, doi: 10.1109/CSNT51715.2021.9509604.

[14] K. RATHI, V. SHARMA, S. GUPTA, A. BAGWARI and G. S. TOMAR, "Home Appliances using loT and Machine Learning: The Smart Home," *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, Al-Khobar, Saudi Arabia, 2022, pp. 329-332, doi: 10.1109/CICN56167.2022.10008294.

[15] A. C. Gaganjot Kaur, “Improved J48 Classification Algorithm for the Prediction of Diabetes,” International Journal of Computer Applications, 2014.