The Impact of Data Pre-Processing Techniques and Dimensionality Reduction on the Accuracy of Machine Learning

1st Hadeel S. Obaid
College of Engineering
University of Information Technology and Communications
Baghdad, Iraq
hadeel.obaid@uoitc.edu.iq

2nd Saad Ahmed Dheyab
College of Engineering
University of Information Technology and Communications
Baghdad, Iraq
saad.theyab@uoitc.edu.iq

3rd Sana Sabah Sabry
College of Engineering
University of Information Technology and Communications
Baghdad, Iraq
sana.sabah@uoitc.edu.iq

***Abstract*— Data pre-processing is considered as the core stage in machine learning and data mining. Normalization, discretization, and dimensionality reduction are well-known techniques in data pre-processing. This research paper seeks to examine the effects of Min-max, Z-score, Decimal Scaling, and Logarithm to the base 2 on the accuracy of J48 classifier using the NSL-KDD dataset. Experiments were conducted using the above-listed methods and their individual results were compared to each other. Principal component analysis (PCA) and Linear Discriminant Analysis (LDA) were tested for dimensionality reduction; furthermore, a hybrid combination of PCA and LDA was attempted and the performance showed an improved classification accuracy compared to the individual methods.**

Keywords*— *Data Pre-Processing; Min-Max; Z score; Decimal Scaling; Log2; PCA; LDA; J48.

I. Introduction

Machine learning (ML) is a known area of computer science that mainly deals with the discovery of data patterns and data-related irregularities [1]. Data mining deals with the discovery of unrecognized data associations in a given database [2]. It involves the extraction of valid, previously unrecognized but comprehensible information from a large database. The persistent increase in the size of the existing databases has made its human analysis difficult and has created both an opportunity and a need to extract vital information from databases. Currently, careful data integration is permissible, but such data must first be transformed into suitable forms for mining.

Data preprocessing is a major aspect of new knowledge discovery processes which despite being less considered compared to the other steps such as data mining, has often accounted for more than 50 % of the total effort during data analysis [3]. Raw data is usually characterized by several irregularities such as missing values, noise, inconsistencies, and redundancies whose presence influences the performance of subsequent learning steps. Therefore, a proper preprocessing step is often performed to limit the influence of data irregularities (if present) on the performance (quality and reliability) of subsequent processing steps.

Data transformation is another data processing step which mainly involves data generalization, smoothing, normalization, and attribute construction [2]. The accuracy and efficiency of data mining algorithms may be improved by data normalization (a form of data transformation). Better results are achieved with such methods if data has already been normalized, that is, scaled to a specified range such as [0.0, 1.0]. Data transformation requires the transformation of a given dataset to a data mining format [4]. Data transformation can be achieved through either data normalization, aggregation, smoothing, or generalization. The reduction of a huge amount of data requires a long time and, in such condition, data reduction can be performed using dimension reduction, data cube aggregation, data compression, discretization, numerosity reduction, and concept hierarchy generation.

The paper is organized in the following way: Section 2 provided a review of the related work while section 3 discussed dimensional reduction. Section 4 discussed data normalization while section 5 provided a basic introduction of data discretization. Section 6 introduced J48 classifier while section 7 presented the experiments and the results. Section 8 presented the conclusions drawn from the study and the recommendation for future work.

II. Related Work

Several studies have explored data preprocessing; for example, Haddad et al [5] proposed novel machine learning-based two-tier classification models Naïve Bayes and KNN. The results of the study showed a promising gain in false alarm detection rate compared to the existing models. Vasan et al [6] depended on PCA to perform experiments on various classifiers using two benchmark datasets (KDD CUP and UNB ISCX). From the results of the experiments, the first 10 PCs showed effectiveness for classification tasks as they achieved a classification accuracy of about 99.7% and 98.8% on KDD CUP and UNB ISCX respectively. This was almost the same accuracy achieved when using the original 41 features for KDD and 28 features for ISCX. Wimmer and Powell [7] examined the effects of using PCA as a feature reduction method on decision trees (DT). The experiments showed the application of PCA before decision tree induction to enhance the classifiers' classification accuracy. A notable observation was the simplification of the resulting DT after applying PCA.

Different types of normalization methods have been studied by Al Shalabi [2] with each method tested against the ID3 methodology on the HSV dataset. The three factors considered were the number of leaf nodes, the tree growing time, and the accuracy of the classifier. The different learning

978-1-5386-9325-4/19/$31.00 ©2019 IEEE

methods were compared after their application to each normalization method.

Ramírez-Gallego et al [3] analyzed, summarized, and categorized the contributions on data preprocessing that cope with streaming data. The study was conducted by using the most relevant contributions while the analysis was performed in terms of their reduction rates, predictive performance, memory usage, and computational time. The existing preprocessing techniques have been analyzed by Ramasamy [8] in terms of their prediction accuracy after preprocessing. Evidently, the prediction accuracy was reportedly increased to up to 90 % after raw data preprocessing using the existing techniques. The highest prediction accuracy was achieved by the Naïve Bayes classifier. It also achieved a low error rate compared to Logistic. The extraction of the most relevant features and a proper training of the network will pave the way to achieving highly promising outcomes.

Saranya & Manikandan [9] analyzed the feasibility of achieving privacy using the normalization techniques. They also compared the results of these techniques. The experimental results showed the Min-Max normalization method to achieve the minimum misclassification error. They executed all the three normalization methods on a dataset consisting of 10 elements. Using the values of (10, 15, 20, 24,30, 37, 40, 45, 50, 60), the results clearly showed Min-Max normalization to achieve the least rate of misclassification error compared to Decimal Scaling and Z-Score.

III. Dimensionality Reduction

Dimension reduction is mainly aimed at representing a high dimensional (HD) dataset in a low-dimensional (LD) space while keeping the HD structures (outliers and clusters) of the dataset [10]. An advantage of dimension reduction over other methods is its scalability in HD data visualization; however, its drawback is a loss of information during data transformation into the LD projection. The next section discussed two common dimension reduction algorithms – PCA and LDA.

A. Principal Components Analysis (PCA)

The PCA is a method for feature extraction which generates new linear combinatorial features of the initial features [6]. It maps each example of a given dataset present in a d dimensional space to a k dimensional subspace such that $k < d$ and the new set of generated k dimensions referred to as the Principal Components (PC). Each PC is directed towards a maximum variance with the exception of the variance which has already been accounted for in all its preceding components. Subsequently, the first component covers the maximum variance while each subsequent component covers lesser value of variance. The PC can be represented thus:

$$PC_i = a_1X_1 + a_2X_2 + \ldots + a_dX_d \quad (1)$$

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

B. Linear Discriminant Analysis (LDA)

The LDA is mainly used as a reduction method when there is a need to reduce computation time complexity and achieve a better classification [5]. As a dimension reduction technique, the LDA is mainly used in image processing, signal processing, bankruptcy, and market analysis problems. Despite the effectiveness of PCA in extracting the most efficient features from a given dataset, it is not efficient in discrimination tasks. The LDA transforms a highly dimensional feature to a lower dimensional space through the selection of an optimal projection matrix while preserving the vital information for data classification. The LDA process requires the definition of two scatter matrices; the first matrix is SB which is defined as inter-class scatter matrix, while the second matrix is SW which is defined as the intra-class scatter matrix.

IV. Normalization

The normalization of an attribute is done by scaling the values in such a manner that they fall into a specified range. Normalization is an important step for classification frameworks that involves distance measurements or neural networks [2]. When performing classification using neural network backpropagation algorithm, the speed of the learning phase will be increased by normalizing the values of the input for every measured attribute in the training set. Regarding the distanced-based techniques, normalization ensures that attributes with initially low ranges are not outweighed by those with initially larger ranges. Some of the current data normalization techniques include decimal scaling, Z-score normalization, and min-max normalization.

A. Min-Max

The aim of min-max normalization is to linearly transform the original data [11]. Assume minA and maxA as the minimum and maximum values of an attribute A. Min-max normalization maps a value, vi, of A to $\ddot{V}_i$ in the range [new-minA, new-maxA] by computing:

$$\ddot{V}_i = (v_i - \text{minA}) / (\text{maxA} - \text{minA}) * (\text{new}_{\text{maxA}} - \text{new}_{\text{minA}}) + \text{new}_{\text{minA}} \quad (2)$$

The relationship between the original values of a dataset is preserved after min-max normalization. An “out-of-bounds” error is likely to occur if a future normalization input case falls outside the original data range for A.

B. Z- Score

For Z-score normalization, the normalization of the values for an attribute, A, is based on the mean and standard deviation (SD) of the attribute [11]. The normalization of a value, vi, of A to $\ddot{V}_i$ is done by computing:

$$\ddot{V}_i = (v_i - \bar{A}) / \sigma A \quad (3)$$

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

$$A = 1/n * (v_1 + v_2 + \ldots + v_n) \quad (4)$$

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

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

C. Decimal Scalingcc

This form of normalization is performed by moving around the decimal point of the attribute values [11]. This movement is dependent on the absolute maximum value of the attribute. The normalization of a value v of A to v' is done thus:

$$Vi' = v_i / 10^j \quad (5)$$

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

V. Discretization

Discretization is a process of converting continuous variables into discrete ones by splitting the range of values of the continuous variables into a finite number of subranges called intervals, buckets or bins [12]. Discretization algorithm partition continuous variables into finite numbers of intervals which are treated as categories. The continuous-valued features are discretized using logarithm to the base 2 before projecting the value of the result to an integer to avoid biases [5]. For each continuous-valued z, this step uses the equation below:

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

VI. J48 Classifier

This is a simple C4.5 decision tree mainly used for classification tasks [13]. It generates a binary tree, but the decision tree method is the most applied method in solving classification problems. This classifier constructs a tree to model the classification process. The constructed tree is applied to each tuple in the database to bring about its classification. J48 classifier does not consider missing values while building a tree, i.e., the value of the missing item can be predicted using the information gained from the values of the other attributes. The major aim is to partition the data into ranges with respect to the attribute values for items contained in the training set. J48 classifier allows the classification of attributes using either decision trees or the rules generated from them.

VII. Experiments and Results

Our experiments were aimed at examining the effectiveness of some available preprocessing methods (Log2, Min-max, Z-score, decimal scaling, LDA, and PCA) on the classification accuracy. The performance of these preprocessing methods was evaluated on NSL-KDD benchmark dataset. The NSL-KDD dataset was built with 42 attributes as an improved form of KDD'99 dataset. The improvement on KDD'99 dataset was made by removing duplicate instances to get rid of biased classification results [14]. There are several versions of this dataset; out of the total number of attributes in the dataset, 20 % serve as the training data (total of 25192 instances) while the test dataset contained 2254 instances. There are different versions of this dataset with differences in the number of instances; however, the number of attributes per case is 42. The class attribute is labeled 42 in the data set and indicates the class of each instance (normal connection or an attack). The following improvements were made on the NSL-KDD dataset for intrusion detection systems [15]:

1. The dataset does not contain redundant records in the training dataset; this will keep the classifier unbiased for more frequent records.
2. In the testing dataset, there are no duplicate records; therefore, the performance of the learners will be better and unbiased by the methods that have better detection rates on these frequent records.
3. There is an inverse relationship between the number of records selected at each difficulty level group and the percentage of records in the KDD dataset. This inverse relationship gives rise to different classification rates for different machine learning methods and improves the efficiency and evaluation accuracy of each learning technique.

Logarithm to base 2 was examined as a data discretization method; three normalization techniques were also studied. LDA and PCA were performed for dimensional reduction while J48 classifier was used for classification. J48 is an extension of ID3 [16] which is publicly available in the WEKA data mining tool as an open source Java implementation of the C4.5 algorithm. The WEKA tool provides several tree pruning options. Pruning is used as a tool for précising in case of any potential over-fitting. Classification is performed recursively in the other algorithms until every single leaf is pure; that is, the data classification must be as perfect as possible. This algorithm generates the rules for the generation of the specific data identity with the objective of progressively generalizing a decision tree until a balance between accuracy and flexibility is achieved.

The first experiment was conducted by applying the Logarithm to base 2 on our dataset. The Log2 was implemented in C sharp because it is not provided in WEKA. Then, the J48 classifier was applied to examine the classification accuracy. During the evaluation, 70 % of the dataset was used for training and 30 % for testing. For normalization, Min-max, Z-score, and decimal scaling were implemented in WEKA. J48 classifier was also applied after data normalization to find the classification accuracy. The results achieved were shown in Table I while Figure 1 provided a comparison between the results.

TABLE I. 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

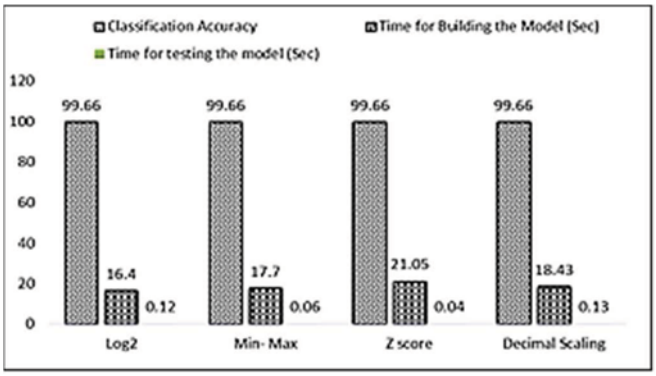

Fig.1. A comparison of the preprocessing methods

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 II. 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

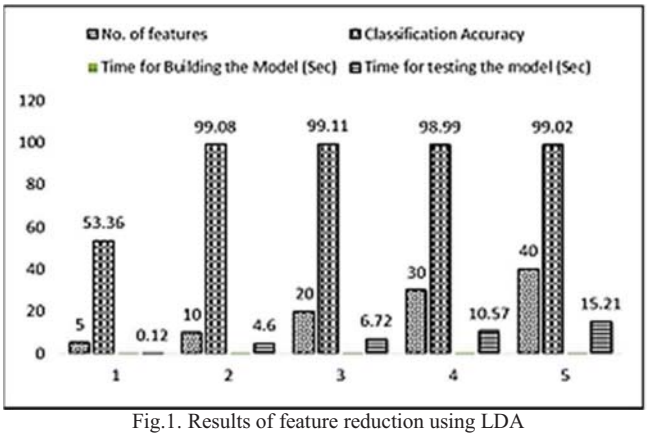

Fig.1. Results of feature reduction using LDA

TABLE III. 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

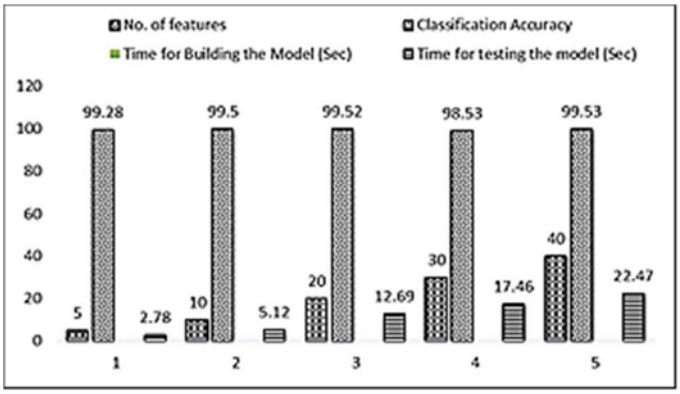

Fig.3. Results of feature reduction using PCA

From Table II, the accuracy for five features was low (53.36 %) while the best accuracy of 99.11 % was achieved when using 20 features. Besides that, the model building time was 0.04 sec while the model testing time was 6.72 sec. In Table III, a high accuracy was also obtained for 20 features (99.52 %). The model building time was 0.04 sec while the model testing time was 12.69 sec.

From the second experiment, the model was observed to be fast and with higher accuracy when the number of features for both LDA and PCA was reduced to 50 %. For our dataset with just 42 attributes and 22544 instances, the time of data preprocessing was not prolonged; since the processing time for very big data is crucial, therefore, reducing the attributes for big data to 50 % with high accuracy and minimum time is beneficial.

For the third experiment, a hybrid of the two algorithms (LDA and PCA) was used by choosing 20 features (10 from PCA and 10 from LDA). The J48 classifier was then used for classification. The accuracy of this experiment was increased to 99.56 % compared to that of LDA or PCA individually. However, the results for model building and testing times were slightly increased (15.22 sec and 0.22 sec, respectively). Still, these were promising results, especially when dealing with massive datasets.

CONCLUSION AND FUTURE WORK

Data pre-processing is a critical phase in data mining. In this paper, we studied the effects of pre-processing methods on the classification results of J48 classifier. NSL-KDD dataset was used in the experiments while Min-max, Z-score, decimal scaling, and Log2 were applied on the original dataset for pre-processing. From the experimental results, the classification accuracy for all the evaluated methods was found to be 99.66 % although Log2 was the fastest among all the evaluated methods. For dimension reduction, PCA and LDA were applied. Furthermore, the ideal ratio of features reduction was found to be 50 % of the original dataset for

PCA and LDA experiments to obtain high accuracy with minimum time. This reduction is very helpful especially when dealing with big data. The application of the hybrid PCA-LDA method improved the classification accuracy and slightly increased the time. In the future work, efforts will be put into examining more preprocessing techniques and investigating other dimension reduction methods, as well as their combinations.

REFERENCES

[1] J. Furnkranz, "Machine Learning and Data Mining," Springer-Verlag Berlin Heidelberg, 2012.

[2] Z. S. a. B. K. Luai Al Shalabi, "Data Mining: A Preprocessing Engine," Journal of Computer Science, pp. 735-739, 2006.

[3] B. K.,. S. G. M. ´. F. H. Sergio Ramírez-Gallego, "A survey on data preprocessing for data stream mining: Current status and future directions," Elsevier, p. 39–57, 2017.

[4] S. Srivastava, "Weka: A Tool for Data preprocessing, Classification, Ensemble, Clustering and Association Rule Mining," International Journal of Computer Applications, vol. 88, 2014.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

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

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

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