International Journal of Engineering & Technology, 7 (3.24) (2018) 483-490

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Rank of Normalizers Through TOPSIS with the Help of Supervised Classifiers

Ranjit Panigrahi[1]*, Samarjeet Borah[2]

[1,2]Department of Computer Applications
Sikkim Manipal Institute of Technology, Sikkim Manipal University
Majhitar, East Sikkim-737136
**Corresponding author E-mail:ranjit.panigrahi@gmail.com*

Abstract

Classification is a tedious task for gathering and categorizing collected knowledge from the noisy high-dimensional dataset. The classifier suffers a lot when the dimension of the dataset is high and the underlying dataset is in different size and units. To make the classification cost effective, the dataset must be subject to pre-processing. Normalization, as a pre-processor transforms the data into a unit less mode across all the dimensions of the dataset. Practically there are many normalization techniques which are best suitable for different implementation scenarios. Though, it is believed that normalization improves classifiers performance but it is a tedious task to ascertain an optimum normalizer for specific scenarios. In this paper, seven widely used normalization techniques are evaluated through five popular supervised learning classifiers using intrusion detection dataset. Rank to these normalization techniques are allocated using a popular ranking algorithm called as Techniques for Order Preference by Similarity to the Ideal Solution (TOPSIS), thus revealing the best optimum normalizer for intrusion detection environment.

Keywords:*Normalization, Support Vector Machine, SVM, TOPSIS, KNN, Neural Network, Naive Bayes, k-Nearest Neighbors, Linear Regression.*

1. Introduction

In recent information and communication era, the manifestation of data mining and knowledge discovery cannot be escalated. It is one of today's fast growing technical arena that demonstrates its presence between computer science and statistics, and mostly relies on artificial intelligence and data science[1]. Various data mining techniques such as clustering and classification motivates many emerging research domains viz., computer networks, intrusion detection and gene expression. It is indeed to mention that; the knowledge extraction process faces an inherent challenge[2] in underlying dataset. The challenge is due to the curse of many units, shape and size of attributes of the dataset. Therefore, before the actual knowledge exploration starts, the underlying data must pass through a quality check process called data pre-processing. Normalization [3], as a pre-processor transforms the dataset stored in various forms to a unit less format, so that the performance of the classifier believes to be improved. The main objective of normalization techniques is to transform the underlying data to a standard scale. In general practice data normalization uses linear scaling to compress a large dataset. The purpose of data normalization entirely depends on the application type, scope and dataset under consideration.

In this paper, performances of popular classifiers, such as, Support Vector Machine (SVM), Linear Regression (LR), Back-propagation Neural Network (BNN), k-Nearest Neighbours (KNN) and Naive Bayes (NB) are evaluated by normalizing dataset using seven widely used normalization techniques, such as, Decimal Scaling, Statistical, Vector normalization,Ordinal normalization, Linear Normalization – I (LN-I), Linear Normalization – II (LN-II) and Linear Normalization – III (LN-III). The result of each classifier after normalization is used in TOPSIS[4] for allocating a suitable rank[8] to the normalizers.

2. Literature Review

As normalization standardizes a dataset at prepossessing stage, therefore these techniques have a significant impact while classifying knowledge in areas such as skin lesion analysis[5], land cover classification[6] and multimodal biometric system[7]. Similarly, a study on improving Otsu method[8] was conducted using normalization techniques and their ensembles. A novel normalization mechanism was also presented for hiding speaker identity while transmitting voice signal [9]. Normalization techniquescan also be appliedin signature verificationthereby improving the process by 19% and 53% for skilled and random forgeries [10]. Few additional applications are network selection access in heterogeneous wireless networks[11, 12], power forecasting algorithms [13], metasearch[14] etc. In all the experiments, it is found that application of normalization techniques improved the performance of the same.

2.1. Normalization Techniques

The current paper focuses on empirical performance review of various normalization techniques specifically for intrusion detection dataset. The normalizers under evaluation are presented below.

Decimal Scaling:

In a feature vector a data point is normalized by changing the decimal point of values of each feature upto aconvinced place. The number of decimal points to move depends on the maximum absolute value of the feature vector.

A normalized value v_i corresponding to a feature value x_i is acquired using

$$v_i = \frac{X_i}{10^j} \quad (1)$$

where j is the smallest integer in the feature vector such that $\max(|v_i|) < 1$. Normalization by decimal scaling transform original values in the range of 0 and 1.

Statistical Normalization:

In this pragmatic approach, the data resulting from normal distribution are transformed into standard normal distribution. The statistical normalization is represented as

$$v_i = \frac{x_i - \mu}{\sigma} \quad (2)$$

where µ is mean of the given feature vector x having n number of instances. µ can be derived as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i. \quad (3)$$

σ is the standard deviation and normally expressed as

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)} \quad (4)$$

Ordinal Normalization:

In this type of normalization technique, the values of a feature vector are not scaled down, instead its rank is transformed to a range of 0 and 1. Initially, continuous value of a feature vector is ranked and subsequently normalized the rankwithin the range of [0,1]. It is represented as

$$v_i = \frac{r-1}{max(r)-1} \quad (5)$$

where r is the rank of a given value x_i in a feature vector *x*.

Vector Normalization:

It is the ratio of each data point with the square toot of sum of square of all data points in the series. It also transforms the value in the range of 0 and 1.

$$v_i = \frac{x_i}{\sqrt{\sum_{i=1}^{n} x_i^2}} \quad (6)$$

Linear Normalization – I:

Under this normalization mechanism, the ratio of all the data point with the maximum value of the data point in the series are replaced by the original data points.

$$v_i = \frac{x_i}{\max(x)} \quad (7)$$

Linear Normalization – II (Min-Max Normalization):

This approach provides linear transformation on original range of data using minimum and maximum values of the feature vector [3]. Numerically, it is represented as

$$v_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (8)$$

where $x_i$is the original value of the feature vector, and the highest and lowest values are consideredamong all values of the concerned feature vectors. Ideallyzero value is set for $x_i$whenboth minimum and maximum value within the feature vector are equal.

Linear Normalization – III (Frequency Normalization):

This transformation mechanism is to normalize feature vector by the ratio of anoriginal value with the summed value of the concerned fracture vector. It also scales an attribute into [0,1] and is defined as

$$v_i = \frac{x_i}{\sum_i^n x_i} \quad (9)$$

2.2. Classifiers for Evaluating Normalizers

Five most popular classification methods are taken into consideration for recording of their performance after normalization of underlying data. These are,

2.2.1. Support Vector Machine (SVM)

A Support Vector Machine [15,26,27,28] is a classification and regression prediction tool that models the state by creating a feature space, which is a finite-dimensional vector space [16], each dimension of which represents a feature of the object concerned, called attribute. The main goal is to establish a process in which a set of threshold parameters is trained to classify an unseen object into a particular class [17].

2.2.2. Linear Regression (LR)

Prediction using linear functions are considered to be the basis for many learning algorithms. Linear Regression predicts a real-valued function from a training samples [18]. In other words, it estimates the values of the coefficients used in the test data with the help of training samples.

2.2.3. Back propagation Neural Network

This classification mechanism implements a supervised learning algorithm called Multi-Layer Perceptron (MLP) that usually based on a function$\boldsymbol{f}(.) : \boldsymbol{R^m} \rightarrow \boldsymbol{R^o}$ by training on the concern dataset, where $\boldsymbol{m}$ is the number of features for input and $\boldsymbol{o}$ is the number of features for output[19]. It builds a non-linear function for either classification or regression considering a set of features $\boldsymbol{X} = \boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_m}$ and a target $\boldsymbol{y}$

2.2.4. K-Nearest Neighbors (KNN)

This intelligent classification systemundertake all available instances to classifya new incoming instance grounded on a similarity measure[20].Nearest Neighbour classification is an instance-based classification[21,22] technique responsible for storing instances of the training data. As a lazy learning algorithm, it supports classification from majority of the nearest neighbors of each instance [20].

2.2.5. Naive Bayes(NB) Classifier

The Naive Bayes is a predictive classifier responsible for evaluating a set of prediction of each data point using mixtures of values in a data set. A Naïve Bayes classifier considers the absence or presence of a particular feature and unconnected to the absence or presence of any other feature of class variable [23]. In a supervised learning environment, the probabilities of each point are calculated as

$$P\left(\frac{c}{x}\right) = \frac{P\left(\frac{x}{c}\right) \times P(c)}{P(x)} \quad (10)$$

2.3. TOPSIS – A Ranking Approach

It is a famous classical MultiCriteria Decision Making (MCDM) mechanism [24] used to adopt the the best, from a set of options where each alternative is evaluated against more than oneconditions. Allocation of weight is taken place to each optionfor ultimately selecting the best option having the highest weight. The weight allotment procedure of TOPSIS has been represented in the algorithm algoTOPSIS presented below.

Algorithm:algoTOPSIS

Input: *mxn* matrix

Step 1. Construction of normalized decision matrix.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\left(\sum x_{ij}^2\right)}} \quad (11)$$

Where,

i	=	1,…,m
j	=	1,…,n
x_{ij}	=	Original value of decision matrix.
r_{ij}	=	Normalized value of decision matrix.

Step 2. Construct aweighted normalized matrix.

$$V_{ij} = W_j r_{ij} \quad (12)$$

Where,

W_j = Weight for *j* criterion.

Step 3. Calculate positive and negative ideal solutions.

$$A^* = \{V_1^*, \dots, V_n^*\}$$
$$A' = \{V_1', \dots, V_n'\}$$

Where,

A^*	=	Positive perfect solutions.
A'	=	Negative perfect solutions
V_j^*	=	$\{max(V_{ij}) if j \epsilon J;\ min(V_{ij}) if j \epsilon J'\}$
V_j'	=	$\{min(V_{ij}) if j \epsilon J;\ max(V_{ij}) if j \epsilon J'\}$

Step 4. Calculate the separation measure for each option.

4.1. The separation for positive perfect alternative is –

$$S_i^* = \sqrt{\sum (V_j^* - V_{ij})^2} \text{ for } i=1,\dots,m \quad (13)$$

4.1. The separation for negative perfect alternative is –

$$S_i' = \sqrt{\sum (V_j' - V_{ij})^2} \text{ for } i=1,\dots,m \quad (14)$$

Step 5. Closeness and selection of alternatives

5.1. Calculate the closeness of each alternative to the ideal solution

$$C_i^* = \frac{S_i'}{S_i^* + S_i'} \{0 < C_i^* < 1\} \quad (15)$$

5.2. Select the alternative where C_i^*closest to 1

3. Experiment

3.1. Tools, Simulators and Datasets used

The experiment is conducted with MATLAB 2015 for implementing all the normalization techniques and Weka 3.8 as the simulator for all the classification algorithms.

The dataset under evaluation is CICNTTor2017 [25] dataset provided by Canadian Institute for Cybersecurity. The dataset consisting of labelled network traffic, including full packet in pcap format and csv. The dataset consists of the fields presented in Table 1. The feature "label" is the class for classification specifying whether a tuple is of type "Tor" or "nonTor". A random subset of 5000 rows are considered under evaluation having 4153 records are labelled as "nonTor" and 847 records are labelled as "Tor". The characteristics of dataset is numeric in nature which contains both integer and double data in verities of units. Therefore, this dataset can be considered as an ideal dataset for evaluating normalizers.

Table 1: CICNTTor2017 dataset with attribute description.

Sl No	Attribute Name	Data Type
1	Flow Duration	Integer
2	Flow Bytes/s	Double
3	Flow Packets/s	Double
4	Flow IAT Mean	Double
5	Flow IAT Std	Double
6	Flow IAT Max	Integer
7	Flow IAT Min	Integer
8	Fwd IAT Mean	Double
9	Fwd IAT Std	Double
10	Fwd IAT Max	Integer
11	Fwd IAT Min	Integer
12	Bwd IAT Mean	Double
13	Bwd IAT Std	Double
14	Bwd IAT Max	Integer
15	Bwd IAT Min	Integer
16	Active Mean	Integer
17	Active Std	Integer
18	Active Max	Integer
19	Active Min	Integer
20	Idle Mean	Integer
21	Idle Std	Integer
22	Idle Max	Integer
23	Idle Min	Integer
24	label	Text

3.2. Methodology

The experiment begins classifying the dataset using each classifier without applying any normalization techniques. The performance outcome of each classifier is tabulated for future reference. Further, the dataset is normalized through each normalizer and the normalized data is passed to each classifier for further classification and subsequent tabulation of performance outcomes. The tabulated results of classifier before and after normalization are compared to ensure the capabilities of normalizers. The qualified normalizers are passed to the TOPSIS ranking algorithm for weight and subsequent rank allocation. The normalizer having the highest rank is considered to be the best optimum approach for the concern classifier. Eventually, a ranking table is prepared for all the classifiers and normalizers to ascertain the best generic normalization approach across all the classifier. The entire implementation procedure can be understood through the algorithm rankNormalizers presented below.

Algorithm: rankNormalizers
Input

Dataset	=	D
Normalizers	=	$N_1, N_2, \dots\dots\dots, N_7$
Classifiers	=	$C_1, C_2, \dots\dots\dots, C_5$

Rankers (TOPSIS) = T

Output
Normalizers with Rank

Step 1 Import the dataset D
Step 2 Import a classifier Cx
Step 3 Classify the dataset D with the classifier Cxand record the performance outputs.
Step 4 Import a normalizer Nx and normalize the dataset D and generate the normalized dataset Dx
Step 5 Classify the normalized Dx dataset with the classifier Cx and record the performance outputs.
Repeat step 4 and 5 for all the normalizers.
Repeat Step 2 to Step 5 for all the classifiers.
Step 6 Call the ranker T and pass the set of normalizers and its correspondence performance outcomes generated by the classifier Cx to get the TOPSIS weight of each normalizer.
Repeat step 6 for all the classifiers.
Step 7 Generate a TOPSIS weight table of all the normalizers along with the corresponding classifiers.
Step 8 Calculate the average weight for each normalizer.
Step 9 Calculate rank of each normalizer.

3.3. Parameters to Evaluate Classifier's Performance

Classification accuracy has been considered as a prominent performance parameter for the classifiers in various literatures. But it is the best practice to evaluate true performance of a classifier using additional measures. Various other performance measures that can be considered to evaluate performance of a classifiers has been outlined in table 2.

Table 2:. Parameters for evaluating classifiers performance

Sl No	Parameters	Descriptions
1	Build Time	It is the model building time using the training instances.
2	Accuracy	It indicates the percentage of correctly classified instances of the supplied dataset.
3	Misclassification rate	It indicates the percentage of incorrectly classified instances of the classifier while classifying the instances.
4	Mean Absolute Error (MAE)	The mean absolute error is the ratio of mean of sum of all the instances along with their absolute error of each instancewith the number of instances in the test set with an original class label of the concern instance.
5	Root Mean Squared (RMS)error	It usually provides the extent up to which the model is from revealing the correct label.
6	Relative Absolute (RA) error	It is also relative to a simple predictor, which is just the average of the actual values where the error is just the total absolute error on contraryto the total squared error.
7	Root Relative Squared (RRS) error	RRS error is the percentage of ratio of RMS error with the RMS error obtained by just predicting the mean of target values.
8	True Positive (TP) Rate	True positive rate is also known as recall or sensitivity, which is defined as the ratio of sum of positively detected instances to sum total of positively detected instances and negatively detected instances.
9	False Positive (FP) Rate	False positive rate is the ratio of sum of negatively detected instances and sum of positively detected instances and negatively detected instances
10	Precision	Precision is also known as Positive Predictive Value (PPV), which is calculated as the ratio of true positives with the sum total of true positives and false negatives.
11	F-Measure	The F-measure or F-score is the harmonic mean between precision and recall

4. Results and Discussion

The output of the rank allocation procedure is outlined into two broad steps. The former step deals with classifiers performance output recording before and after normalization and the later step involves in determining rank of classifiers.

4.1. Classifiers Performance Pre and Postnormalization

In this step, outcome of each classifier using each normalization technique has been recorded separately.

Initially, the performance of SVM classifier has been judged using build time, accuracy (%) and misclassification (%) rate before and after normalization of datasets. Figure 1 represents the same. It has been seen that–

- All the classifiers consume less amount of model build time for raw dataset.
- The ordinal normalization leads to highest accuracy with lowest amount of misclassification rate.

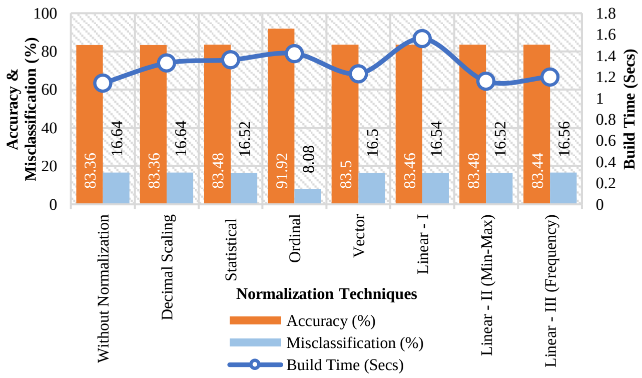

Fig. 1: Accuracy, Misclassification and Build Time of SVM classifier

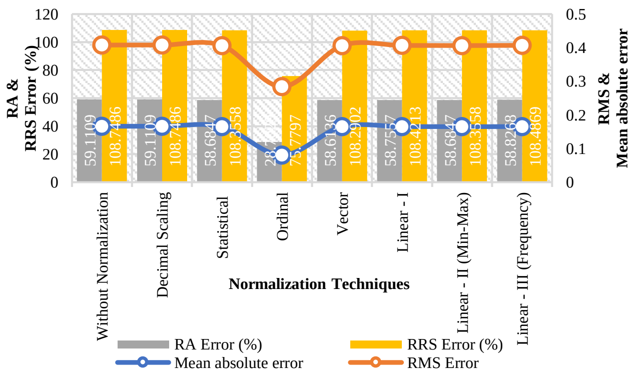

Fig.2: Error rate of SVM classifier

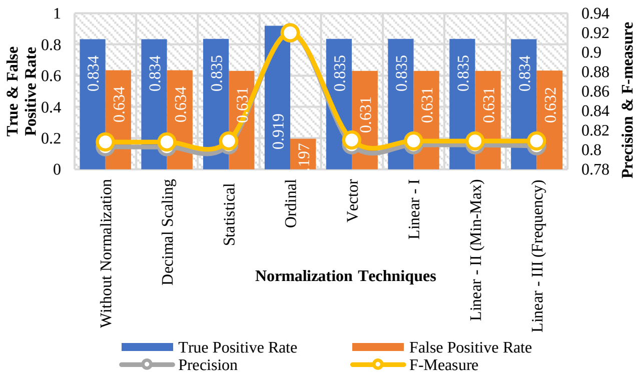

Fig. 3: True & False Positive rate, Precision and F-Measure of SVM classifier

Similarly, when the classification errors have been considered, it has been seen that the

- Ordinal normalization of data is responsible for claiming least amount of Mean absolute error, RMS Error, RA Error and RRS error (Figure 2).
- On the other hand, Decimal scaling normalization has no effect on boosting the performance of the classifier, hence generates highest amount of classification errors (Figure 3).

Further, Figure 3 also reflects

- Ordinal normalization outperforms its peer techniques in terms of TP Rate, FP Rate, Precision and F-Measure.

When build time, accuracy(%) and misclassification(%) has been taken into consideration to evaluate all the normalization techniques using Linear Regression classifier (Figure 4 to Figure 6) it has been seen that the Vector normalization plays a crucial role in boosting the performance of the classifier. Therefore, the general outcomes when the Linear Regression is deployed along with normalizers would be –

- Linear Regression consumes less amount of model build time when the dataset is normalized using Vector normalization.
- The Vector normalization also leads to highest accuracy with lowest amount of misclassification rate.

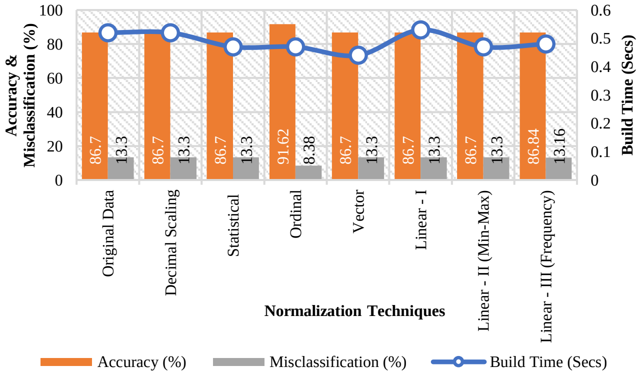

Fig. 4: Accuracy, Misclassification and Build Time of Linear Regression classifier

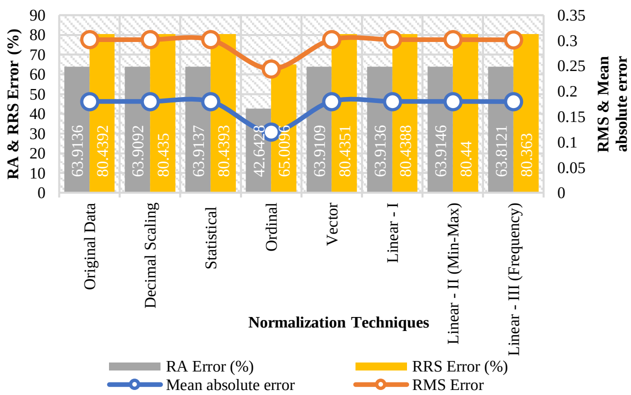

Fig .5: Error rate of Linear Regression classifier

Again, for classification errors, it has been seen that the

- Ordinal normalization of data is responsible for claiming least amount of Mean absolute error, RMS Error, RA Error and RRS error (Figure 5).

The performance outcome in figure 6 reflects

- Ordinal normalization outperforms its counterpart in terms of TP Rate whereas Linear – III plays a significant role in reducing dales positive rate.
- Moreover, so far as Precision and F-Measure is concerned, Ordinal normalization kept a position far ahead than its peer.

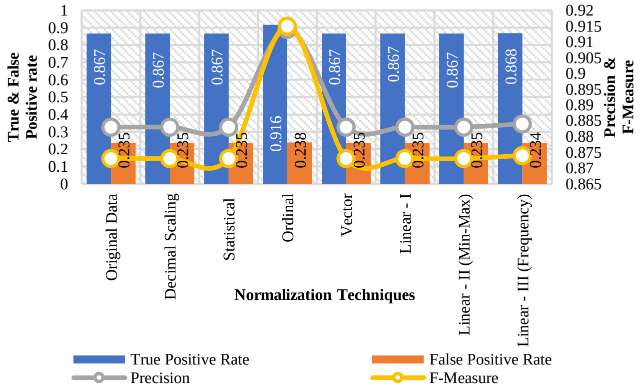

Fig. 6: True & False Positive rate, Precision and F-Measure of Linear Regression classifier

Performance outcome of Back Propagation Neural Network classifier has been represented in figure 7 to figure 9. It clearly shows that –

- The classifier consumes huge amount of time to build the model when any of the normalization techniques used. Rather it claims very less amount of time when the raw data is deployed without any normalization.
- The accuracy of the classifier improved with the significant decrease in misclassification (%) rate for Ordinal normalization.

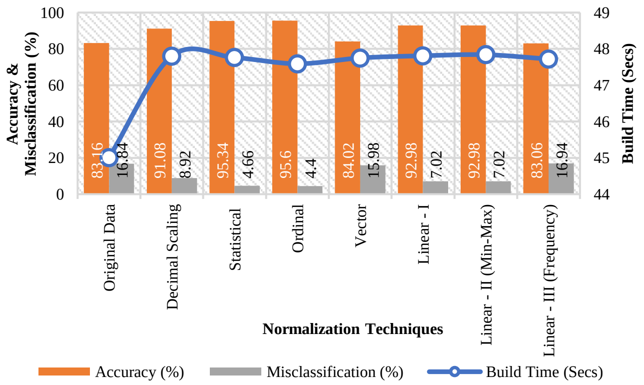

Fig. 7: Accuracy, Misclassification and Build Time of Back Propagation Neural Network classifier

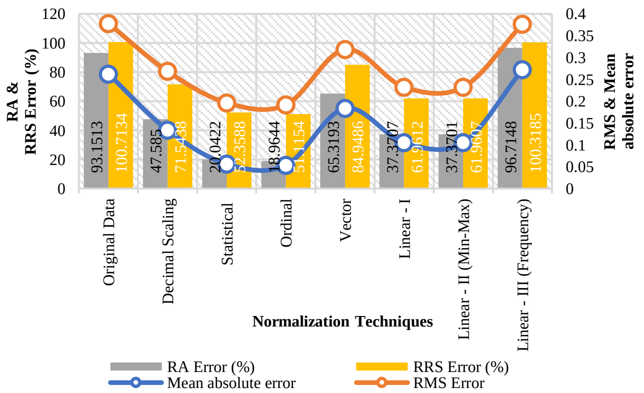

Fig. 8: Error Rate of Back Propagation Neural Network classifier

Again, for classification errors, it has been seen that the

- Ordinal normalization of data is responsible for claiming least amount of Mean absolute error, RMS Error, RA Error and RRS error (Figure 8).

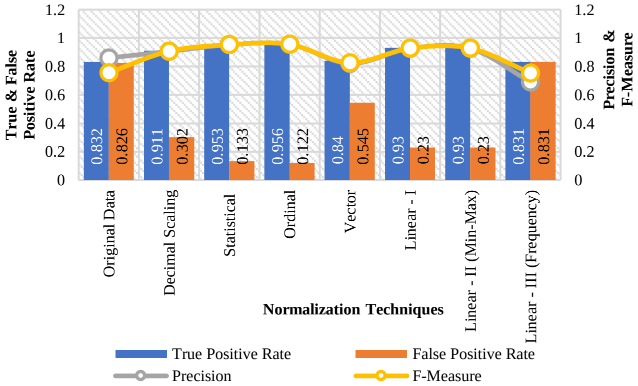

Fig. 9: True & False Positive rate, Precision and F-Measure of Neural Network classifier

The performance outcome recorded in figure 9 also reflects

- Ordinal normalization outperforms its counterpart in terms of TP Rate, FP Rate, Precision and F-Measure.

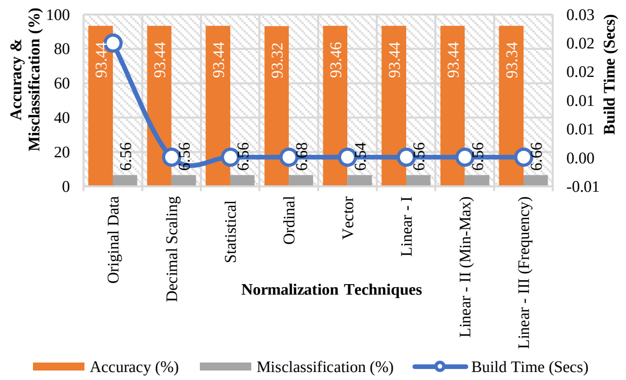

Fig. 10: Accuracy, Misclassification and Build Time of KNN classifier

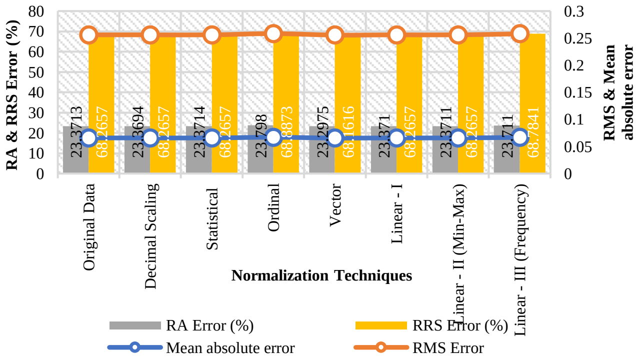

Fig. 11: Error rate of KNN classifier

Similarly, performance outcome of KNN classifier for all the normalizers has been depicted in Figure 10 to Figure 12. It clearly shows that –

- The model build time for KNN classifier while using any of the normalization process is lesser than that of the model build time required for raw data.
- Model build time of the classifier for all the normalizers remain same.
- The classifier claims highest amount of accuracy with lowest amount of misclassification (%) rate, when deployed with Vector normalizer.

Again, for classification errors, it has been seen that the

- Vector Normalization of data claims least amount of Mean absolute error, RMS Error, RA Error and RRS error (Figure 11).

Secondly, the performance outcome of Figure 12 reflects

- Ordinal normalization outperforms its counterpart in terms of TP Rate, FP Rate, Precision and F-Measure.

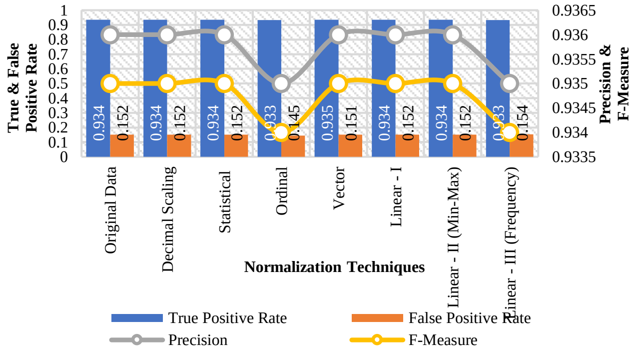

Fig. 12: True & False Positive rate, Precision and F-Measure of KNN classifier

Various performance outcome of Naïve Bayes classifier has been outlined in Figure 13 to Figure 15. The outcomes regarding various normalizers is that –

- The classifier consumes very less amount of time for both Ordinal and Linear Normalization – II to build the model.
- The Ordinal Normalization boosts the accuracy of the classifier to the highest with the decrease in misclassification (%) rate.

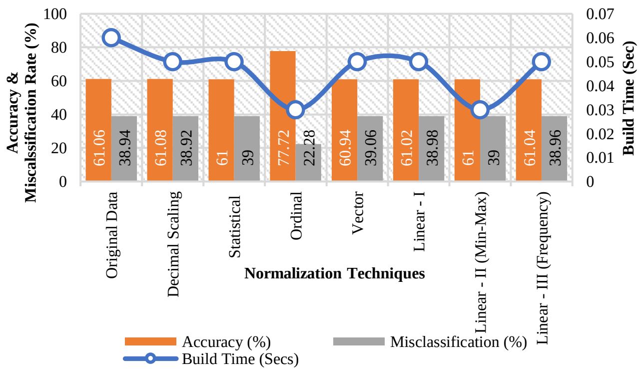

Fig. 13: Accuracy, Misclassification and Build Time of Naïve Bayes classifier

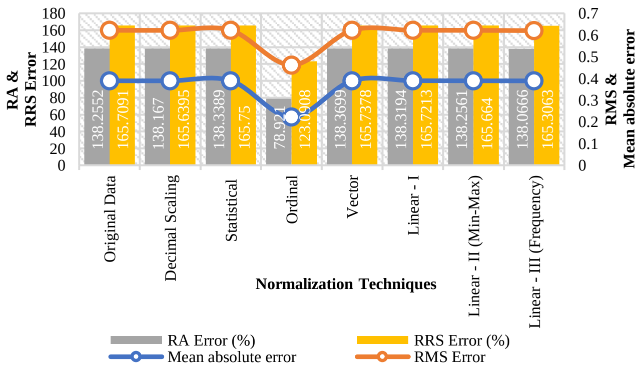

Fig. 14: Error rate of Naïve Bayes classifier

Again, for classification errors, it has been seen that the

- Ordinal normalization of data is responsible for claiming least amount of Mean absolute error, RMS Error, RA Error and RRS error (Figure 14).

Now, coming to the Figure 15 about TP Rate, Precision and F-Measure, it is evident that

- Ordinal normalization outperforms its counterpart in terms of TP Rate, Precision and F-Measure. So far as FP rate is concern Decimal Scaling, Linear – I and Linear – II works very well for the Naïve Bayes classifier.

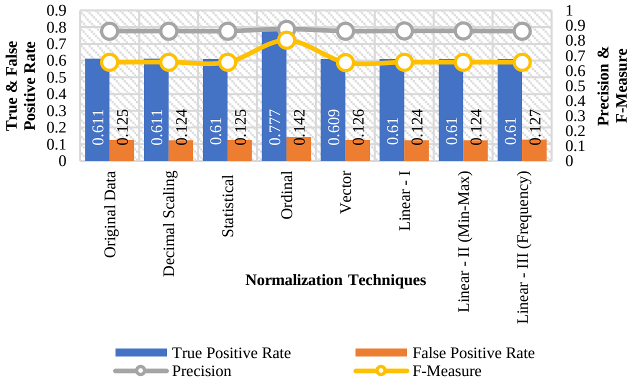

Fig. 15: True & False Positive rate, Precision and F-Measure of Naïve Bayes classifier

4.2. Weight and Rank Allocation to Normalizers

Considering the performance measures, performance outcome of individual classifiers for all the normalization techniques are subjected to a multiclass decision-making approach called TOPSIS to ascertain weight and corresponding rank of each normalizers. Table 3 presents weight and rank of each normalizers.

Table 3: Weight and Rank of normalizers through various classifiers using TOPSIS

Normalization Techniques	Rank after Classification										Average Weight	Final Rank
	Support Vector Machine		Linear Regression		Neural Network		k Nearest Neighbor		Naïve Bayes			
	TOPSIS Weight	Rank	TOPSIS Weight	Rank	TOPSIS Weight	Rank	TOPSIS Weight	Rank	TOPSIS Weight	Rank		
Decimal Scaling	0.9973	2	0.0065	6	0.6925	5	0.2399	3	0.0109	3	0.38942	5
Statistical	0.9985	1	0.0377	3	0.9845	2	0.2398	6	0.0096	5	0.45402	3
Ordinal	*0.9861*	*4*	*0.9808*	*1*	*1*	*1*	*0.8914*	*1*	*0.9901*	*1*	*0.96968*	*1*
Vector	0.0018	7	0.0555	2	0.3781	6	0.3474	2	0.0088	7	0.15832	7
Linear - I	0.9837	6	0	7	0.8092	3	0.2399	3	0.0102	4	0.4086	4
Linear - II (Min-Max)	0.9885	3	0.0377	3	0.8092	3	0.2399	3	0.4605	2	0.50716	2
Linear - III (Frequency)	0.9859	5	0.0376	5	0	7	0.021	7	0.0093	6	0.21076	6

It is clearly evident that Ordinal normalization achieves highest amount of weight and rank for all the classifiers except Support Vector Machine. But for SVM statistical normalizer, it not only achieves the highest weight but also score the highest rank.

At this point of time it was really tough to identify the winners among the normalizers. Therefore, the average weight and the corresponding rank there upon is calculated to get the final rank list (Table 3, Figure 16). From Figure 16, it can be seen that the Ordinal normalization attracts the highest score with the top most rank, and hence declared as winner among its peers.

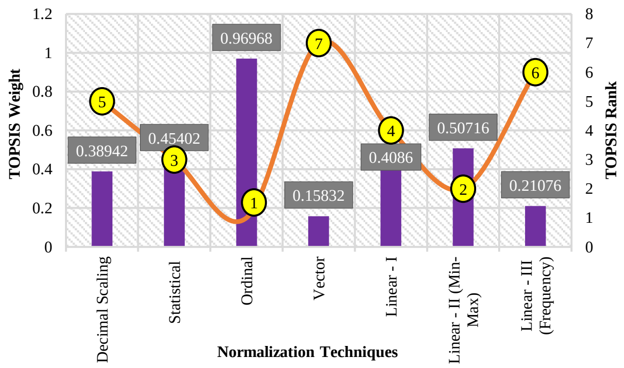

Fig. 16: Overall Rank of Normalization techniques with TOPSIS weight.

4. Conclusion

Data normalization is an important pre-processing task prior to classification. Poorly normalized data may lead to improper classification results. In this research work, an attempt has been made to rank selected normalization techniques. The study is quite specific, since, it is tested in the domain of cyber security only (univariate/multivariate). The normalizers selected for this study are - decimal scaling, statistical, ordinal, vector, linear, min-max and frequency normalization. Further, impact of normalized data can be measured based on how it is influential in classification accuracy. Therefore, a set of widely used classifiers i.e. Support Vector Machine, Linear Regression, Back propagation Neural Network, K-Nearest Neighbors and Naive Bayes are used to examine the same. Performance is measured in terms of build time, accuracy, misclassification rate, RMS error,mean absolute error, RRS error, RA error, true positive rate, false positive rate, precision and f-measure. The experiments are conducted on Weka 3.8 environment. Finally, rank allocation is performed using TOPSIS. Dataset used for this experiment is CICNTTor2017. From the experiment, it is found that ordinal normalization achieves highest amount of weight and rank for all the classifiers except Support Vector Machine where statistical normalizer is found to be the best one. The study is conducted on a large dataset specific to computer intrusion data with well recognized classification techniques. Therefore, the research work carried out here will be more applicable to development of fast and efficient intrusion detection systems using appropriate normalizers at the pre-processing stage. Further studies can be made in other application areas with a larger set of classifiers or datasets.

References

[1] Jordan MI, Mitchell TM, Machine learning: Trends, perspectives, and prospects, Science Journal, 2015; 349(6245):255-260

[2] Lobur M, Stekh Y, Artsibasov V, Challenges in knowledge discovery and data mining in datasets, Perspective Technologies and Methods in MEMS Design, Polyana, 2011; 232-233

[3] Eftekhary M, Gholami P, Safari S, et al., Ranking Normalization Methods for Improving the Accuracy of SVM Algorithm by DEA Method, Modern Applied Science,2012; 6(10):26-36

[4] Sianaki OA, Intelligent Decision Support System for Energy Management in Demand Response Programs and Residential and Industrial Sectors of the Smart Grid. (PhD), Curtin University, 2015, Retrieved from http://espace. library. curtin. edu.au/R?func=dbin-jump-full&local_base=gen01-era02&object_id=240088 (240088)

[5] Jailani R, Hashim H, Nasir Taib M, Normalization techniques for psoriasis skin lesion analysis, Asian Conference on Sensors and the International Conference on new Techniques in Pharmaceutical and Biomedical Research, 2006; 151-153

[6] Ahmed B, Abdullah Md., Noman Al, Land cover classification for satellite images based on normalization technique and Artificial Neural Network, International Conference on Computer and Information Engineering (ICCIE), Rajshahi, 2015; 138-141

[7] Kabir W, Ahmad MO, Swamy MNS, A novel normalization technique for multimodal biometric systems, International Midwest Symposium on Circuits and Systems (MWSCAS), 2015; 1-4

[8] Kasmin F, Abdullah A, Prabuwono AS, The effect of normalization techniques and their ensembles towards Otsu method, International Conference on Intelligent Systems Design and Applications (ISDA), 2012; 931–936

[9] Patel R, Shrawankar U, Thakare VM, Normalization techniques for hiding speakers identity, International Conference on Data Science & Engineering (ICDSE), 2012; 75-80

[10] Aguilar JF, Garcia JO, Rodriguez JG, Target dependent score normalization techniques and their application to signature verification, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2005; 35(3): 418–425

[11] Lahby M, Cherkaoui L, Adib A, Performance analysis of normalization techniques for network selection access in heterogeneous wireless networks, International Conference on Intelligent Systems: Theories and Applications (SITA-14), Rabat, 2014; 1-5

[12] Baghla S, Bansal S, Effect of normalization techniques in VIKOR method for network selection in heterogeneous networks, IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2014; 1–6

[13] Gensler A, Sick B, Vogt S, A review of deterministic error scores and normalization techniques for power forecasting algorithms, IEEE Symposium Series on Computational Intelligence (SSCI), 2016; 1–9

[14] Sever H, Tolun MR, Comparison of Normalization Techniques for Metasearch. In: Yakhno T. (eds) Advances in Information Systems, ADVIS, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2002;2457

[15] Cortes C, Vapnik V, Support-vector network, Machine Learning, 1995; 20:1–25.

[16] Burges CJC, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998; 2(2): 1-43.

[17] Pengwenlong Gu, Khatoun R, Begriche Y, et al. Support Vector Machine (SVM) Based Sybil Attack Detection in Vehicular Networks, IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, March 2017

[18] Poole D, Mackworth A, Linear Regression and Classification, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010

[19] Pedregosa F, Varoquaux G, Gramfort A, et al., Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011;12-2825-2830.

[20] d.umn.edu [Internet]. Duluth(MN): University of Minnesota; [cited 2017 July 20]. Available from: www.d.umn.edu/ ~deoka001 / downloads/ K_Nearest_ Neighbor _ Algorithm.pdf

[21] Zhu X, Goldberg A, Introduction to Semi-Supervised Learning, Morgan and Claypool Publishers (CA, USA), 2009

[22] Guleria P, Sood M, Educational Data Classification and Clustering Using K-Means and K-Nearest Neighbours, International Journal of Control Theory and Applications, 2016; 10(40):181-189.

[23] Muthulakshmi S, Porkodi R, A Survey on Classification Algorithms in Data Mining of Bioinformatics, International Journal for Scientific Research & Development, 2016 ; 4(3): 612-617.

[24] Hwang CL, Yoon K, Multiple Attributes Decision Making Methods and Applications. Springer, Berlin Heidelberg, 1981.

[25] Lashkari AH, Gil GD, Mamun M ,Ghorbani A, Characterization of Tor Traffic using Time based Features, 3rd International Conference on Information Systems Security and Privacy, 2017; 1:253-262

[26] Pradeep Kumar Mallick, Debahuti Mishra, Srikanta Patnaik and Kailash Shaw, "A Hybrid Approach for Simultaneous Gene Clustering and Gene Selection for Pattern Classification", Indian Journal of Science and Technology, Volume : 9(21), DOI: 10.17485/ijst/2016/v9i21/94175, June 2016.

[27] Pradeep Kumar Mallick, Debahuti Mishra, Srikanta Patnaik and Kailash Shaw, "A Semi-Supervised Rough Set and Random Forest Approach for Pattern Classification of Gene Expression Data", Int. J. of Reasoning-based Intelligent Systems (IJRIS), Vol.8, No.3/4, pp.155 – 167, 2016.

[28] Pradeep Kumar Mallick, Debahuti Mishra, Srikanta Patnaik and Kailash Shaw, "A Novel Supervised Gene Clustering Approach By Mining Interdependent Gene Patterns" International Journal of Pharma and Bio Sciences, Volume: 7(4): (B), pp: 20 - 32 ,2016.