This article can be cited as A. Rahman, Statistics-Based Data Preprocessing Methods and Machine Learning Algorithms for Big Data Analysis, International Journal of Artificial Intelligence, vol. 17, no. 2, pp. 44-65, 2019.

Statistics-Based Data Preprocessing Methods and Machine Learning Algorithms for Big Data Analysis

Azizur Rahman[1]

[1]School of Computing and Mathematics, Charles Sturt University,
Wagga Wagga, NSW 2678, Australia
Email: azrahman@csu.edu.au

ABSTRACT

Big data analytics is a very fast growing research domain which embedded the combination of computational (i.e. computer-intensive) and inferential (i.e. statistics-oriented) thinking. Information is increasingly gathered into big data environment such as distinct protein-coding data for identifying various critical diseases and its cure. Data pre-processing techniques are used to make the data clean, noise free and consistent to model in various real life purposes. This paper examines a range of statistics-based data pre-processing methods and machine learning algorithms to assess their performances in the big data analysis setting. Tuberculosis affected protein's amino acid sequences data from the National Center for Biotechnology Information (NCBI) database is utilized for empirical results. Findings reveal that statistics-based pre-processing methods are effective to make the big data useable for significant modelling and analysis with novel machine learning algorithms such as the hidden Markov chain model, Box-Cox and linear transformation, and they also maintain the performance of those algorithms. Although there are significant differences observed between predictive outcomes and performances of the algorithms, results further demonstrate that the hidden Markov chain model produced more accurate, exact and faster analysis with reliable estimates.

Keywords: Big Data, Computational Algorithms, Hidden Markov Chain Model, Normalization, Statistical Thinking, Tuberculosis.

Mathematics Subject Classification: 62P10, 68P05, 93E14

Computing Classification System: I.6

1. INTRODUCTION

In data science, big data and its analysis strategies are at the growing vital focus of 21 century's data centric world. Given the technological developments, almost all modern science, economic, social and business environments are generating and integrating big data from various sources including videos, audios, images, posts, search queries, electronic transactions, emails, health records, social networking activities, science data, sensors and smart-mobile phones and their applications (Eaton et al., 2012). They are stored in databases grow massively and become difficult to capture, form, store, manage, share, analyze and visualize via typical database software tools. In 2015, digital world of data was expanded to 8 zettabytes and predicted to double every two years (Manyika et al., 2011) reaching about 32 zettabytes of data by 2019. In the past, human genome decryption process takes approximately 10 years, but now it takes not more than one week (HPCC, 2018). These create a huge opportunity to advance medical and patient outcomes.

Big data is characterized by its four main components: variety, velocity, volume and veracity (Gerhardt et al., 2012; Rahman, 2018). Variety makes big data really big since it comes from a great variety of sources and generally has in three types: structured, semi structured and unstructured. Structured data can be easily sorted to analyze but unstructured data is random and difficult to analyze. Whereas, semi-structured data does not conform to any fixed fields but contains labels to distinct data elements from wider sources (Shing & Shing, 2011). Volume of data now is larger than terabytes and petabytes which need advanced storage strategies and analysis techniques (Madden, 2012). Velocity is required for big data generation, handling and all processes. For time limited processes, big data should be used as it streams into the organization in order to maximize its value (Shing & Shing, 2011; Madden, 2012). Veracity is a significant issue in big data due to the varying levels of noise and processing errors in raw data. It is difficult to control large data so data security must be provided (Chowdhury et al., 2018a). Besides this, after producing and processing of big data, there should be the potential to analyze it to reveal new insights to optimize decision making for the organization.

To understand typical real world issues, models can be formulated towards making better-informed decisions. However, the data on which the models' are based could be big and complex with the model itself. As a result, the traditional methods show poor performance in handling the highly complicated models optimization. Due to the very fast growing domain in computing such as intelligent algorithms based powerful calculation, the optimal solution of a complex model can be achieved in a short time. For example, an ant colony optimization technique is widely applied in continuous optimization problems, which can improve expert and intelligent systems in terms of data clustering to training neural networks (Chen et al., 2017). This technique seeks for the optimal solution in the pre-specific domains though. Thus, if the initial domains are not estimated correctly, it may not generate the optimal solution. Additionally, fuzzy logic is another approach which is closely related to the probability modelling in statistics. Fuzzy modelling have gained widespread applications in the context of handling complex models and measuring their uncertainty (Narukawa & Torra, 2009; Pozna et al., 2010). A survey on fuzzy systems and control is presented in the study by Precup and

Hellendoorn (2011). Recent applications of fuzzy modelling include process control with focus on adaptive fuzzy control (Blazic et al., 2009; Precup & Hellendoorn, 2011), on the combination between fuzzy control and sliding mode control (Hwang et al., 2009), on kernel-based fuzzy clustering (Graves & Pedrycz, 2010) and biomedical applications (Bustince et al., 2010).

In particular, generic two-degree-of-freedom fuzzy controllers have been proposed in Precup et al. (2009) to deal with servo systems. However, stability is one of the most important problems in the analysis and design of nonlinear control systems (Vrkalovic et al., 2017). A stability analysis method for fuzzy control systems dedicated to nonlinear processes has been formulated in Tomescu et al. (2007). The optimal tuning of fuzzy controllers such as swarm intelligence algorithms (Precup et al., 2015) can guarantee systematic performance specifications in the conditions of model-based tuning (Vrkalovic et al., 2018). A part of the application of these algorithms includes genetic algorithms (Perez et al., 2013), ant colony optimization (Castillo et al., 2015), simulated annealing (Vrkalovic et al., 2017), and the optimal tuning of linear and fuzzy controllers by means of classical algorithms (Precup and Preitl, 2004, 2006; Preitl et al., 2004). The model-free versus model-based tuning remains an open issue though, and the proper adaptation of other algorithms can also be taken into consideration (Kazakov & Lempert, 2015; Vrkalovic et al., 2018).

Formal concept analysis (Ganter & Wille, 1999) is another way to deal with such an open issue for the analysis of complex data. The advances in the theory of fuzzy formal concept analysis and its applications are studied by many researchers (Jiang et al., 2003; Phan-Luong, 2008; Medina & Ojeda-Aciego, 2010; Medina & Ojeda-Aciego, 2013). For instance, a generalisation of the classical dual concept lattices to a multi-adjoint environment in data science is studied by Medina and Ojeda-Aciego (2013) which allows a new perspective to find information from complex databases with incomplete and/or imprecise information. In a recent study by Precup et al. (2015) discusses a number of key methods including support vector machine learning to fuzzy modelling for fault detection and isolation analysis using some intelligent algorithms based techniques. This review study gives special attention to machine learning, data mining, clustering and evolving techniques which are widely applied to industrial problems.

In big data systems fuzzy cognitive maps analysis with migration algorithms for adaptation of model parameters seem to be very useful (Vascak, 2012), particularly when conventional rule-based knowledge discovery methods are insufficient for description of complex dynamic databases. Artificial neural networks (ANNs) approaches can also generate reliable generalized solutions for many complex models designed for pattern classification, function approximation and regression problems. However, there are concerns related to structuring the hidden layers, especially with too many or too small parameters generalization situations. Some researchers use pruning-constructive hybrid algorithms to overcome the common problem of hidden layer architecture in ANNs (Kamruzzaman & Sarker, 2011). A simple and effective pruning algorithm based on the neuroplasticity concepts to find the optimal solution of hidden layer architecture in multilayered ANNs is highlighted by Wagarachchi & Karunananda (2017).

Organizations in any industry have big data can benefit from its careful analysis to gain insights and depths to solve real problems (Bakshi, 2012). Big data requires a revolutionary mix of methodologies from different domains including statistics and computing that step forward from traditional data analysis (Figure 1). The Venn diagram depicts that statistics play a vital role on big data analysis, particularly in machine learning and pattern recognition. Algorithms are the code part of big data analysis that contributes all four domains of in data science. Therefore, an assessment of various algorithms is crucial to know which one could be used in big data analysis, especially for genome or proteins sequence data.

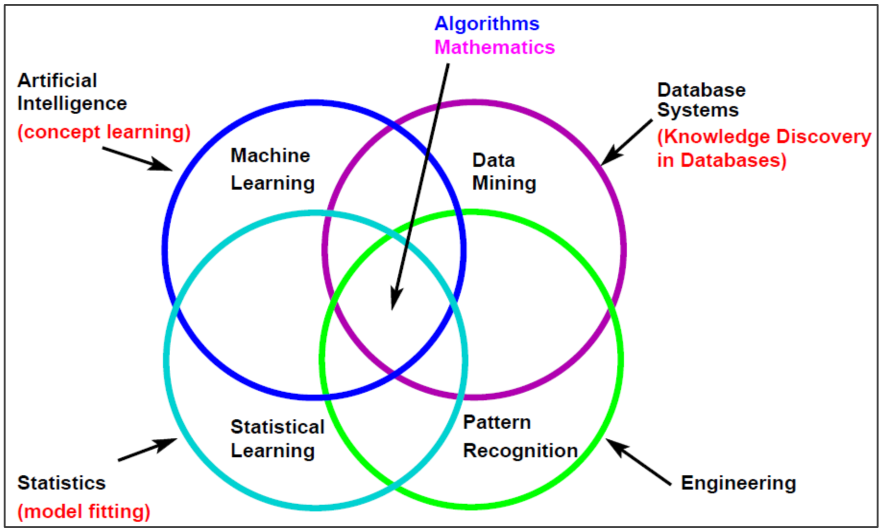

Figure 1. A multi-domain view of big data analysis strategies in data science.

Common analytic problems in data science are depicted in Figure 2. Broadly speaking by whether the output is *continuous* or *discrete* (classes) and whether it is *supervised* (includes desired outputs) or *unsupervised* (doesn't include desired outputs) as optimization, most of the analytic problems are categorised into four groups which are classification, regression, clustering and dimension reduction (Rahman, 2017a). For instances, the main aim of the classification methods is to accurately allocate objects to a discrete set of known classes or groups based on a set of input variables. An example is the development of a diagnostic test, which declares a person to be of class 'diseased' or 'healthy', based on a set of clinical variables. The overall aim of the clustering methods is to combine objects into groups or classes based on a set of discrete input variables. An example is combining the customers into groups based on their responses to a satisfaction survey. One can then inspect these groupings and the customer traits that describe the groups and differentiate between them. This can be used to manage the existing customers or predict satisfaction of new customers.

		Optimization Challenge	
		Supervised Learning	**Unsupervised Learning**
Classes	**Discrete**	classification	clustering
	Continuous	regression	dimension reduction

Figure 2. Common analytic problems in data science by variable class and optimization challenge.

In contrasts, the aim of regression methods is to accurately and precisely estimate or predict the response, given a set of input variables. Typically, the regression model is 'trained' using a set of objects for which the response is known. The analyst might be interested in the estimated values for the objects in the training set, predicting responses for new objects, identifying which input variables are most important in making good predictions, or inspecting the relationships between these variables. Whereas, the aim of dimension reduction is to construct an output variable (or set of variables) based on a set of input variables, where this output variable is unknown. The output variable(s) should be continuous and the new output variables should maximise the information in the data. In big data analytics, dimension reduction tools are commonly available since it can be used to create a small set of output variables that can effectively carry most of the information in a very large set of input variables without compromising any privacy of the original data. The analyst can then inspect these new variables to see which of the original variables are most important in explaining the variation in the data. The new variables can also be used as inputs to regression, clustering and classification problems. So, it is significant to develop new strategies in big data analytics to protect confidentiality of the original data first and then perform robust analysis to achieve insightful information to support business decisions.

In the public health domain, for example, Tuberculosis (TB) disease destroys human tissue and it is considered as a number one disastrous illness for people (Deng et al., 2016). Scientists are trying to discover the fruitful vaccines for the TB disease. It is possible to develop right vaccines by finding the accurate suspected proteins family working behind TB disease. Simulative and automated machine learning systems can help a lot to detect suspected proteins from the genome data. However, the large datasets along with high frequency may hamper the overall process of identification (Rahman et al., 2018).

The main objectives of this paper are to examine a range of statistical data pre-processing methods and machine learning algorithms to determine the best design for model optimisation, and then to assess the performances of the design in the big data analysis setting. It uses the Tuberculosis affected protein's amino acid sequences data from the National Center for Biotechnology Information (NCBI) database for experiential analysis.

The remainder of the paper is as follows. Section 2 presents a range of statistics-based big data preprocessing methods. Section 3 discusses various machine learning algorithms which are based on statistical thinking. Section 4 provides the empirical results with its relevant discussion. The final Section 5 offers the concluding remarks.

2. BIG DATA PREPROCESSING METHODS

Preprocessing is basically implemented on any raw big data before using any kinds of classification or identification. In many real world cases the raw data are not useable due to noises and must be preprocessed. A secured usable data can be achieved via the data preprocessing methods (Figure 3). This section presents three statistical techniques (i.e. decimal scaling, min-max and Z-score normalization) for preprocessing the raw big data which are also known as data standardization tools.

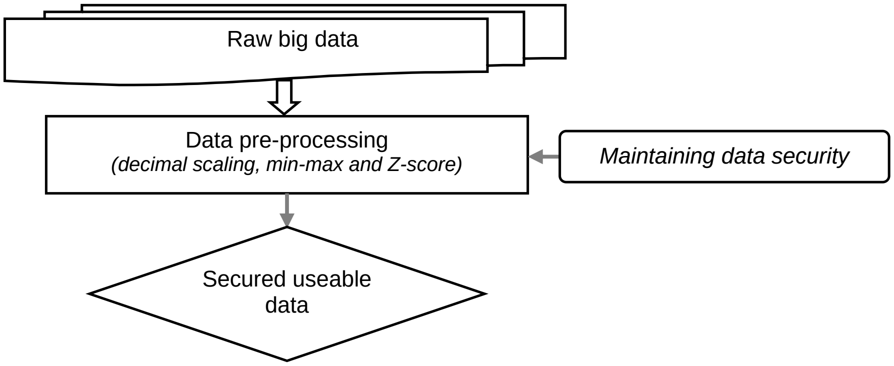
Raw big data
Data pre-processing
(decimal scaling, min-max and Z-score)
Maintaining data security
Secured useable data

Figure 3. A flowchart to obtain secured and noise free dataset from the big data systems.

2.1. Decimal scaling normalization

Decimal scaling is one kind of preprocessing method that aims to equalize the input data to acceptable output. It normalizes the data depending on decimal point of values. In this technique the overall computation is performed in terms of decimal values of data. So, the result is multiplying and dividing it by $POW(10,k)$ with k^{th} exponent. The decimal scale normalization equation can be defined as

$$v_i = \frac{v_i}{10_j} \tag{1}$$

where v_i is the scaled values of given input values.

If a vector V is the range of the input values and also j is the lowest integer value so that max $(|V_i|) < 1$ for all i. Therefore, the decimal scaling normalization depends on movements of decimal points on a specific region.

2.2. Min-max normalization

The min-max normalization method is a strategy which transfers raw data X to Y linearly. It is also a way of feature scaling where numeric feature values are converted within a specific range mostly within 0 and 1. The processed data can be represented in such a way that that the minimum value of X is mapped to 0 whereas the maximum value of Y is mapped to 1. For this study, consider $\min_p$ and $\max_p$ as the minimum and maximum values for detection of annotation for the proteins dataset. The interval within which the desired result needs to be detected is $[\min_p, \max_p]$. Now if any new interval converted to $[new\_\min_p, new\_\max_p]$ then the equation for detecting new protein breaks can be presented as

$$New(p) = \frac{p - \min_p}{\max_p - \min_p} \times (new\_\max_p - new\_\min_p) \tag{2}$$

Usually, min-max normalization preserves the basic properties of original data values. If any kind of data crosses the limit of the interval of the process $[\min_p, \max_p]$ then problem may occur. Therefore, the main focus of min-max normalization is gathering all the data within a certain range mostly the range is within 0 and 1.

2.3. Z-score normalization

The z-score normalization process is one kind of statistical process to standardize vast amount of data depending on the mean value. It is actually a non-dimensional quantity by subtracting mean value from a raw data. Consider X as a random variable then X is normalized by subtracting its desired mean value from the original value and then dividing that by the standard deviation. The z-score process can be written as

$$Z = \frac{X - \mu(X)}{\sigma(X)} \tag{3}$$

where $\mu(X)$ and $\sigma(X)$ is the mean and standard deviation of X , respectively.

This standardization technique is widely used for data normalization because of its ability to calculate the probability of an estimated z-score. This process also helps to compare normal distributions of various variables in the dataset.

2.4. Maintaining data security in preprocessing

Any big database contain large amount of private and sensitive data including healthcare, business, financial or criminal record. These private and sensitive data cannot be share to everyone (Chowdhury et al., 2018b), so privacy protection of data is required in analytics system before employing machine learning. Data preprocessing could be one of the useful methods for avoiding privacy leakage of data in addition to make the data clean, noise free and consistent. A proposed data perturbation and normalization technique is illustrated here for privacy protection in big data (Rahman, 2017b). This method involves with three systematics steps as follows.

Let D be a data matrix of order $r \times k$, representing the original dataset. The rows of the matrix represent objects and the columns of the matrix represent variables.

1. D must be first transformed by a pre-processing normalization technique (e.g., min-max normalization) to get transformed matrix $\underline{D}$ with the same order $r \times k$. The min-max normalization transformed each element of D into the specific interval (0.0,1.0).

2. After step 2, we have obtained a perturbed/scaled new data matrix $\underline{D}$, which is very similar to D , but not identical. Importantly, $\underline{D}$ preserve the properties of D . Thus, $\underline{D}$ can work as a distorted version of D .

3. Now $\underline{D}$ is further shifted by multiplying it with a shifting factor " s ", (i.e. s = a negative number), to increase the security of data. Hence, after applying the shifting factor (-ve number) on $\underline{D}$*,* the order and the value of each element of $\underline{D}$ was changed, i.e. the bigger number become smaller and vice-versa.

3. STATISTICAL THINKING IN MACHINE LEARNING ALGORITHMS

Machine learning is a subfield of data science including artificial intelligence that allows the use of statistical and computing algorithms to parse data, learn from that data, and make informed decisions based on what it has learned to be more accurate in predicting results. Predicted model outcomes in machine learning can be obtained using various statistics-based optimization algorithms though (Figure 4). These algorithms are illustrated in this section.

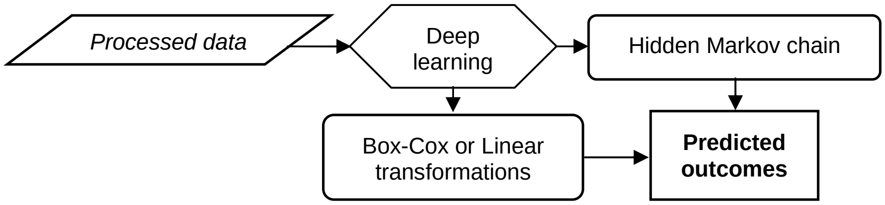

Figure 4. A framework for achieving predicted model outcomes in machine learning.

A straightforward example of a machine learning algorithm is an on-demand movie streaming service. For the service to make a decision about which new movie to recommend to a viewer, machine learning algorithms associate the viewer's preferences with other viewers who have similar drama taste. Whereas, deep learning is a subfield of machine learning that structures algorithms in layers to create an “artificial neural network (ANN)” that can learn and make intelligent decisions on its own. A deep learning model is designed to continually analyze data with a logic structure similar to how a human would draw conclusions (Saez et al., 2016).

3.1. Deep learning

The design of an ANN is inspired by the biological neural network of the human brain. ANN follows non-linear processing and transformation of input data towards standard output and each consecutive layers process previous layer’s output. The fundamental concept an ANN such as a deep neural network (DNN) follows processing the input data through a lot of connected layers which extract data from low level to high level components (Tchurikov et al., 2016; Liao et al., 2016). For example, a multilayer DNN structure is presented in Figure 5.

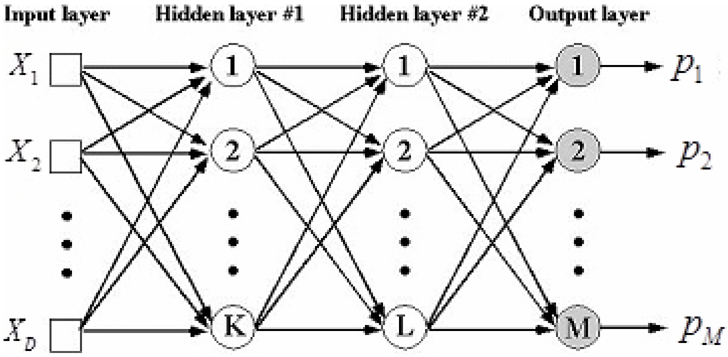

Figure 5. A depiction of multilayer DNN architecture (Eluyode & Akomolafe, 2013).

Consider feature of a particular node as the inputs vector $X_i = (x_{i1}, x_{i2}, ..., x_{im})', i = 1, 2, ..., D$ of the corresponding next layer whose parameters are illustrated by the inputs X_i and weights vector $W_i = (w_{i1}, w_{i2}, ..., w_{im})$, then the non-linear function for the desired output for a specific node in the j^{th} hidden layer is given by

$$X_j = \left(\tanh\left(\sum_{t=1}^{m} w_i \times x_i + b_j \right) \right), \quad (4)$$

where b_j is the bias measure for each node of the j^{th} hidden layer and $\tanh$ is the *hyperbolic tangent* activation function. It is to be noted that the type of activation function used for a model depends on the desired output of the model. Some other activation functions commonly used in DNNs are: threshold, linear, piece-wise, and logistic-Sigmoid, etc.

The vector X_j will be used as input vector of the next hidden layer. For the k^{th} layer with the bias measure b_k for each node, the equation can be written as

$$X_k = \left(\tanh\left(\sum_{t=1}^{m} w_j \times x_j + b_k \right) \right) \tag{5}$$

where X_k be the input vector of the subsequent hidden layer, and the process carries till the final output layer.

For higher performance for each hidden layer multiple nodes are adjusted. All the weights are treated as a tensor consists of various combinations of features of a specific layer node. The weight function can be defined as

$$W_{ij}(t+1) = W_{ij}(t) + \eta \frac{\partial C}{\partial W_{ij}} + \varepsilon(t) \tag{6}$$

where η is the learning rate, C is the cost function, and $\varepsilon(t)$ is an stochastic error term.

The cost function is fully depends on the type of the process like supervised or unsupervised learning (Figure 2). For example, from the perspective of supervised learning the cost function can be implemented as $C = -\sum p_j \log(I_j)$ where p_j actually indicates the target probability at immediate iteration I_j. Hence, the deep learning is a different way of processing multidimensional and ordered data though the processed input data.

The DNN actually uses an optimisation process of achieving the optimal set of weights for the links between nodes by minimising the cost function which will make the model to yield the correct expected output corresponding to the given input. A range of techniques including the ant colony optimization, backpropagation, genetic algorithm, particle swarm optimization algorithm can be utilised for the optimisation (see, e.g. Deng, 2010; Kamruzzaman & Sarker, 2011; Perez et al., 2013; Deng & Dong, 2014; Castillo et al., 2015; Chen et al., 2017; Wagarachchi & Karunananda, 2017). This study uses the backpropagation algorithm which has been described in details in the literature (e.g., Rojas, 1996; Deng, 2010; Eluyode & Akomolafe, 2013; Deng & Dong, 2014). A quick description of this algorithm is provided here.

An iterative approach is used in backpropagation to find the correct set of weights by minimising the cost function i.e. error function. The inputs $X_1, X_2, ..., X_D$ are applied to the DNN and the expected (target) output is compared with the actual (computed) output $p_1, p_2, ..., p_M$ to get the error (Figure 4). A typical cost function for backpropagation technique is the mean squared error (MSE) function: $MSE = \sum_{m=1}^{M} (e_m - p_m)^2$, where e is the expected output, p is the actual output, m indexes output nodes and M is the number of output node. The error estimate is then used to adjust the weights such that it is minimized. This process is repeated until the error is within a benchmark value. However, it is to be noted that the suitable number of nodes in hidden layers is determined heuristically, while the number of nodes in output layer depends on the task. Generally speaking, for the regression problems, the number of nodes in the output layer could be the number of dependent variables in the model.

Now, if w_{ml} and w_{mb} denote the output-hidden layer weights and its corresponding bias weights respectively, then they can be updated by $w_{ml}(t+1) = w_{ml}(t) + \eta\Delta_m p_l + \alpha[\delta w_{ml}(t)]$ and $w_{mb}(t+1) = w_{mb}(t) + \eta\Delta_m p_b + \alpha[\delta w_{mb}(t)]$; where, η, Δ_m, p_l, p_b, α, δw_{ml}, δw_{mb}, and t represent the learning rate, the hidden-output layer error measure for node m, the output of hidden node l, the hidden-output bias, the momentum rate, the previous weight change, the previous weight change for the hidden-output bias, and the iteration index respectively.

The hidden-output error for node m can be obtained as $\Delta_m = p_m \times (1 - p_m)(e_m - p_m)$. Also, the hidden-hidden layers weights, w_{lk}, to be updated by $w_{lk}(t+1) = w_{lk}(t) + \eta\Delta_{lk} + \alpha[\delta w_{lk}(t)]$, where Δ_l, p_k and δw_{lk} denote the error measure of hidden layer node l, the output of preceding hidden layer node k and the previous weight change for the hidden-hidden bias node. Then, Δ_l can be easily estimated as $\Delta_l = p_l(1 - p_l)\sum_{l=1}^{L} w_{ml}\Delta_m$.

If w_{kd} and w_{kb} denote the input-hidden layer weights and its corresponding input-hidden bias weights respectively, then they can be updated by $w_{kd}(t+1) = w_{kd}(t) + \eta\Delta_k x_d + \alpha[\delta w_{kd}(t)]$ and $w_{kb}(t+1) = w_{kb}(t) + \eta\Delta_k p_b + \alpha[\delta w_{kb}(t)]$, where, x, d, δw_{kd}, δw_{kb} and Δ_k indicate the DNN input, the indexes of the input characteristics, the previous weight change, the previous weight change for the input-hidden bias node and the error measure of node k in the hidden layer following the input layer. Hence, the error measure Δ_k for input-hidden node k can be computed by using $\Delta_k = p_k(1 - p_k)\sum_{k=1}^{K} w_{lk}\Delta_l$.

Furthermore, the gradient descent can be very slow if the learning rate η is too small, and can oscillate widely if η is too large. The momentum rate α is chosen between 0 and 1, 0.9 is a good value though. After computing all partial derivatives the DNN weights are updated in the negative gradient direction with corrections for the weights where the learning constant defines the step length of the correction at each iteration. It is very important to make the corrections to the weights only after the backpropagated error has been computed for all units in the network (Rojas, 1996; Deng & Dong, 2014).

3.2. Hidden Markov chain model

The Markov chain model is a liability based approach strictly following the Markov property (Li et al., 2016; Doerks et al., 2012; Anandakumar & Shanmughavel, 2008), which provides a probable solution depending on specific current situation of some dynamic variables. An improved version of Markov chain is the Hidden Markov chain model (HMCM). It is a process of probability ordination over consequent identification of tasks and one of the most powerful approaches of machine learning for statistical prediction to extract information from training and robust data. HMCM is well developed statistical process that can also handle vast amount of data robustly along with being computationally proficient. By focusing on two consecutive datasets the HMCM can predicts what should be the result for that datasets after specific duration. For example, the prediction of annotation of hypothetical proteins can be performed by HMCM.

Consider the proteins sequence as $P = p_1, p_2, ..., p_n$ and the overall model is $\theta = \left(X, Y, \pi\right)$ with the variables X and Y and the tasks probabilities π . Then for every fixed state sequence $I = i_1 i_2 ... i_T$, the probability of the proteins sequence given the model $f\left(P \mid \theta\right)$ can be defined as

$$f\left(P \mid I, \theta\right) = b_{i_1}\left(p_1\right) \times b_{i_2}\left(p_2\right) \times ... \times b_{i_k}\left(p_k\right) \times ... \times b_{i_T}\left(p_T\right) \tag{7}$$

where T is the state sequence length for I and $b_{i_k}\left(p_k\right)$ is the probability of k^{th} state value in P .

The probability of same state sequence I can be expressed as

$$f\left(I \mid \theta\right) = \left(\pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} ... a_{i_{(k-1)} i_k} ... a_{i_{(T-1)} i_T}\right) \tag{8}$$

where π_{i_1} is the probability of task at the initial state i_1 and $a_{i_{(k-1)} i_k}$ is the probability of the state moving from $i_{(k-1)}$ to i_k .

Now probability of P and I is the product of (7) and (8) which can be presented as

$$f\left(P, I \mid \theta\right) = f\left(P \mid I, \theta\right) \times f\left(I \mid \theta\right). \tag{9}$$

Hence, the overall probability for information extraction can be written as,

$$f\left(P, I \mid \theta\right) = \sum_{\forall i} \pi_{i_1} b_{i_1}\left(p_1\right) a_{i_1 i_2} b_{i_2}\left(p_2\right) .. a_{i_{(k-1)} i_k} b_{i_k}\left(p_k\right) ... a_{i_{(T-1)} i_T} b_{i_T}\left(p_T\right) . \tag{10}$$

At the initial task for $T = 1$ the probability is approximately π_1 which is defined by $b_{i_1}\left(p_1\right)$. Similar way when the state is shifting from $i_{(T-1)}$ to i_T then the state transition probability occurs $a_{i_{(T-1)} i_T}$ and generates p_T with probability $b_{i_T}\left(p_T\right)$. Therefore, an apparent location of a specified task (i.e. probability of getting annotation of hypothetical proteins for tuberculosis in this research) can be identified successfully by HMCM.

While as discussed in the introduction section a number of optimization processes can be doable (e.g. Phan-Luong, 2008; Medina & Ojeda-Aciego, 2013; Precup et al., 2015; Chen et al., 2017), to optimize the model parameters this study relays on the maximum likelihood statistical process to maximize $f\left(P,I\,|\,\theta\right)$. It is considered as non-convex, non-linear optimization problem with constraints on π, X and Y. This optimization can also be design as maximizing the log likelihood, i.e. log $f\left(P,I\,|\,\theta\right)$. The EM (i.e. Expectation-Maximization) algorithm is the best technique to resolve this optimisation problem. Although a detailed demonstration of the method is provided in an early study by Dempster et al. (1977), a brief outline is as follows.

For the hidden Markov chain model, the EM method can be designed with the state sequences function $I\left(\theta,\theta^{(k)}\right)$ which is to maximize over the parameter θ at each iteration. The function $I\left(\theta,\theta^{(k)}\right)$ can be written as the following sum of three separable elements:

$$I\left(\theta,\theta^{(k)}\right)=I_1\left(\theta,\theta^{(k)}\right)+I_2\left(\theta,\theta^{(k)}\right)+I_3\left(\theta,\theta^{(k)}\right)$$

$$=\sum_{r=1}^{N}\varphi_{r_1}^{(k)}\log\pi_r+\sum_{r=1}^{N}\;\sum_{s=1}^{N}\;\sum_{t=1}^{T-1}\varphi_{rst}^{(k)}\log a_{rs}+\sum_{r=1}^{N}\;\sum_{t=1}^{T}\varphi_{rt}^{(k)}\log b_r(p_t)\,,$$

where $\varphi_{r_1}^{(k)}$, $\varphi_{rst}^{(k)}$ and $\varphi_{rt}^{(k)}$ are some unknown probabilistic measures which can be estimated by using the forward-backward iterative calculation process (Rahman, 2008a; Rahman et al. 2010) with the lattice structure of the HMCM (Rabiner, 1989; Phan-Luong, 2008; Medina & Ojeda-Aciego, 2010).

Now each element can be maximised individually. The maximising solution for the element $I_1\left(\theta,\theta^{(k)}\right)$ is $\pi_r=\dfrac{\pi_r^{(k)}b_r^{(k)}(p_1)}{\sum_{s=1}^{N}\pi_s^{(k)}b_s^{(b)}(p_1)}$ and for $I_2\left(\theta,\theta^{(k)}\right)$ is $a_{rs}=\dfrac{\sum_{t=1}^{T-1}\varphi_{rst}^{(k)}}{\sum_{t=1}^{T-1}\varphi_{rt}^{(k)}}$. However, the maximising solution for the element $I_3\left(\theta,\theta^{(k)}\right)$ depends on the outputs of the model. When the outputs are discrete, the solution is fairly straightforward as $b_r(z)=\dfrac{\sum_{t=1}^{T}\varphi_{rt}^{(k)}\delta(p_t-z)}{\sum_{t=1}^{T}\varphi_{rt}^{(k)}}$, where z represents a possible output and δ represents a difference measure indicator. However, when the outputs of the model are continuous, the solution needs to be estimated by using analytic process which to be subject to any special forms of the output distribution. For instance, the maximising solution for the parameters of multivariate Normal distribution output is obtatined for

$b_r(y) = \frac{Q}{|\Sigma_r|^{1/2}} e^{-\frac{1}{2}\left((y-\mu_r)'\Sigma_r^{-1}(y-\mu_r)\right)}$, where Q is the normalising component (see, e.g. Rahman, 2008b; Rahman & Upadhyay, 2015; Rahman & Harding, 2016)), $\mu_r(z) = \frac{\sum_{t=1}^{T}\varphi_{rt}^{(k)}P}{\sum_{t=1}^{T}\varphi_{rt}^{(k)}}$ and $\Sigma_r = \frac{\sum_{t=1}^{T}\varphi_{rt}^{(k)}\left(p_t-\mu_r^{(k+1)}\right)\left(p_t-\mu_r^{(k+1)}\right)'}{\sum_{t=1}^{T}\varphi_{rt}^{(k)}}$.

Although the EM algorithm is widely used to achieve convergence to a local maximum, the HMCM objective function may have multimodality issue in many situations. In such a case, the optimization problem would be much more challenging though (Granat, 2003).

3.3. Box-Cox transformation

In many cases the normality assumptions do not exists, and an appropriate transformation of these types of data can make it useable for estimation. Box Cox transformation (Box & Cox, 1964) algorithms can be used for such a sophisticated task. The overall transformation process of Box Cox is illustrated as below.

Consider an input vector as $X = \left(x_1, x_2, ..., x_n\right)$ on which the algorithm will be embedded. Transformation of the input vector by Box-Cox method (Bakshi, 2012) can be defined as

$$x_i^{(\delta)} = \begin{cases} \delta^{-1}\left(x_i^{(\delta)}-1\right), if\ \delta \neq 0 \\ \log\left(x_i\right) \qquad ,if\ \delta = 0 \end{cases} \tag{11}$$

for an unknown optimal parameter δ, $x^{\delta} = D\beta + \varepsilon$, where, x^{δ} is the δ-transformed data, D is the design data matrix, β is the parameters vector and ε is the random error term that needs to be adjusted under the basic normality assumptions $x^{\delta} \square N\left(D\beta, \sigma^2 I_n\right)$. Equation (11) performs acceptably when the input vectors are $x^i > 0$ for $i = 1, 2, ..., n$. For further accuracy the overall process requires adjustment in the estimation process.

The basic aim in the Box-Cox transformation model is to find an optimal estimation on the transformation parameter δ, and there are a range of approaches employed by researchers which go from standard statistical methods (Box & Cox, 1982; Chen et al., 2002; Zeng & Lin, 2007) to fuzzy logics (Liu et al., 2005; Rajasekaran & Pai, 2011). This paper uses an iteration based optimisation process which includes two main steps. First obtain the Profile log-likelihood function

$PLL\left(\delta\right) = -\frac{n}{2}\log\left(\frac{RSS\left(\delta\right)}{n}\right) + \left(\delta - 1\right)\sum_{i=1}^{n}\log\left(x_i^{(\delta)}\right)$ for each possible δ values, where of $RSS(\delta)$ is the residual sum of squares when using $x_i^{(\delta)}$ in the model. Then calculate the maximum likelihood estimate (MLE), $\hat{\delta}$, of δ that is the maximizer of $PLL\left(\delta\right)$.

Also one can easily find an approximate value of δ by using alternative technique as follows. Consider $\delta \in [-a, +a]$ for the ordered values $-a = \delta_0 < \delta_1 < \delta_2 < \cdots < \delta_{2n-1} < \delta_{2n} = +a$, where $\delta_i = \delta_0 + \left(\frac{i}{n}\right)a$, a is some constant, and n is a very large number. For instance, as $n = 100$ and $a = 1$, then $\delta_0 = -1, \delta_1 = -0.99, ..., \delta_{199} = 0.99, \delta_{200} = 1$. Now estimate all values of the residual sum of squares when using $x_i^{(\delta)}$ in the model i.e. $RSS(\delta_0), RSS(\delta_1), ..., RSS(\delta_{199}), RSS(\delta_{200})$. The value δ_i corresponding to the smallest $RSS(\delta_i)$ is the suitable approximate estimate $\hat{\delta}$ to be used. Hence, $RSS(\delta_i) \leq RSS(\delta_j); \forall ij$ and $j \neq i$.

3.4. Linear transformation

In statistical perspective the linear transformation is a mapping based system by which the inputted datasets are added or multiplied to reorganize them in a specific manner to manipulate. Linear transformation takes input as a vector and multiply or add the inputted vector with another vector thereafter analyze it. The basic aim is to transfer the high dimensional linear subspaces data onto lower dimensional subspaces matrices. Consider A and B as two matrix spaces where a and b are vectors. Then the following properties can be pursued in the linear transformation method:

$$f(a+b) = f(a) + f(b) \tag{12}$$

Now if φ is any scalar then

$$f(\varphi a) = \varphi f(a) \tag{13}$$

Therefore, the combination of (12) and (13) creates a new property which is also followed by linear transformation for any large number of sequence of anything. For this work the sequence is proteins sequence. If we consider proteins sequences as $P = p_1, p_2, ..., p_n$ with scalars as $\varphi = \varphi_1, \varphi_2, ..., \varphi_n$, then the linear transformation property can be expressed as

$$f(\varphi_1 p_1 + \varphi_2 p_2 + ... + \varphi_n p_n) = \varphi_1 f(p_1) + \varphi_2 f(p_2) + ... + \varphi_n f(p_n) \tag{14}$$

There are wide applications of linear transformation in bioinformatics as well as big data analysis. For our work it is implemented for predicting multiple possibilities regarding annotation of suspected proteins responsible for Tuberculosis.

A linear transformation algorithm in the supervised learning setting is demonstrated in Goldberger et al. (2004). A comparative analysis of kernel-based fuzzy methods is presented in Graves & Pedrycz (2010). The connections between metric learning and kernel learning specially that arise when studying metric learning as a linear transformation learning problem are studied by Jain et al. (2012). While each of these algorithms was shown to yield improved classification performance over the baseline metrics, their constraints do not generalize outside of their particular problem domains, especially when it needs to satisfy arbitrary linear constraints on the Mahalanobis distance matrix. Nevertheless, the iterative technique based simplex algorithm generates a sequence of feasible iterates p^k for the original problem, where each iterate typically has the same number of nonzero (strictly positive) components as there are rows in A. This iterate is then used to generate dual variables λ (i.e. the Lagrange multipliers) and u (i.e. the slack variable) such that the optimality conditions $Ap = b,\ A'\lambda^k + u^k = c$ and $\left(p^k\right)' u^k = 0$ are satisfied (Gallier, 2013).

If the remaining constraint $u^k \geq 0$ is also satisfied, then the optimal solution is to be achieved and the algorithm terminates. Otherwise, one of the negative components of the slack variable u^k needs to be chosen to get a large value of the corresponding component of p. When this occurs, the algorithm stops and implies to the new iterate p^{k+1}. Each iteration of the simplex method is relatively inexpensive. It maintains a factorization of the submatrix of A that corresponds to B, and updates this factorization at each step to account for the fact that one column of B has changed. Typically, the optimisation with the linear transformation model is quick and accurate, and the simplex methodology converges in a number of iterates that is about two to three times the number of columns in A (Methling et al., 2017).

4. EMPIRICAL RESULTS AND DISCUSSION

This section provides the results and discussion of the data pre-processing techniques and statistics based optimisation algorithms in machine learning. The focus is here though to assess the performances of these statistical approaches in a big data analysis setting. The study uses tuberculosis affected protein’s amino acid sequences data from the National Center for Biotechnology Information (NCBI) databases for the empirical analyses. Further details about the data are available in Rahman et al. (2018). Data analytics programming codes and relevant datasets to be available on author’s webpage at https://researchoutput.csu.edu.au/en/persons/azrahmancsueduau.

4.1. Comparison of data preprocessing techniques

Figure 6 presents the preprocessing results of the data using the decimal scaling, min-max and Z-score normalizations techniques. Findings reveal that the Z-score method normalizes datasets better than other two techniques. As the data size increases, the Z-score method performs the best followed by the decimal scaling. In particular when the data size is about 180,000kb, the Z-score method was able to clean the data well that have provided 187 proteins information after normalization. This figure is around 3.5 times and 1.2 times higher than the number of proteins provided by min-max and decimal scaling normalizations.

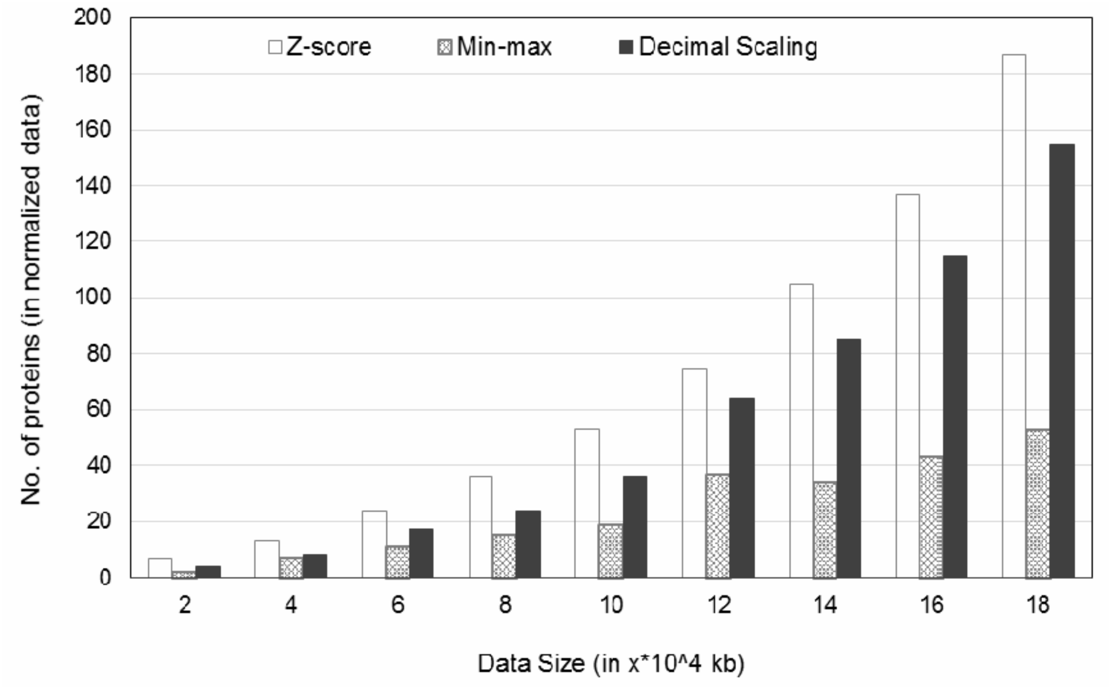

Figure 6. Outcomes of data preprocessing by three methods.

4.2. Performance of machine learning algorithms

Table 1 presents the results from HMCM versus the Box-Cox transformation (BCT) and linear transformation (LT) algorithms for the prediction of number of proteins in different size of datasets. Findings reveal that HMCM algorithm is preferable due to its significantly high level of accuracy and ability to predict exact number of proteins responsible for TB in increasing size of the data from 20 to 180 million sequences. Specifically, according to the first input data (twenty million) the performance of HMCM model for detecting responsible proteins for TB is 55.56% better than the performance of other two models, and while the data size increases to 180 million base pairs, the performance of HMCM model is even better (i.e. 55.83%) than Box-Cox transformation.

Table 1. Prediction results from HMCM, Box-Cox and linear transformations.

Data (in x*10^4 kb)	**HMCM**	**BCT**	**LT**
2	7	2	2
4	13	7	7
6	24	11	12
8	36	15	16
10	53	19	21
12	75	27	31
14	105	34	50
16	137	43	70
18	187	53	95

A further analysis has revealed that the positive predictive power of HMCM algorithm is about 95.45% with a likelihood ration estimate of 9.68. These findings also confirm that HMCM has a very high predictive accuracy and utility for detecting proteins responsible for TB.

5. CONCLUSIONS

Data science including big data and its analysis strategies are significant for this data centric era. Big data is essentially a part of everyday living and business, but how effectively we can process the raw data and then analyze it to reveal new insights to optimize decision making is an important question? Statistical thinking plays a vital role in data preprocessing to make the data clean, noise free and consistent, and then to analysis or model in various real life purposes. This research has examined a variety of statistics-based data preprocessing techniques with machine learning optimisation algorithms and evaluated their performances utilizing the NCBI's big dataset. It has been found that among the three data preprocessing methods discussed, the Z-score normalization method has outperformed than the min-max and decimal scaling methods with the increasing data size.

Findings have also demonstrated that statistical machine learning algorithms such as the hidden Markov chain model (HMCM), Box-Cox transformation and linear transformation are important methodologies in data science. These algorithms are useable for significant modelling and analysis of big data, for example – detection of responsible proteins for TB disease. Comparison of these methodologies has revealed a clear difference in statistical concepts, its predictive outcomes and performances measures. Empirical results have demonstrated that HMCM algorithm is the best one to use for big data analysis in terms of the accuracy, predictive ability and utility, especially when the data size really big. A future research should explore further why this model is the best by comparing and integrating it with many other advanced statistical concepts such as high dimensional data analysis techniques with applications to other areas.

6. ACKNOWLEDGEMENTS

I would like to sincerely thank the two anonymous reviewers and the two editors for their valuable comments and stimulus which were used to improve this final version. I also acknowledge technical support of the Data Science Research Unit (DSRU) at the Charles Sturt University, Australia.

REFERENCES

Anandakumar, S., Shanmughavel, P., 2008, Computational annotation for hypothetical proteins of mycobacterium tuberculosis. *Journal of Computational Science and System Biology* **1**(1), 50–62.

Bakshi, K., 2012, Considerations for big data: Architecture and approach. *Aerospace Conference IEEE*, Big Sky Montana, March.

Blažic, S., Škrjanc, I., Gerkšic, S., Dolanc, G., Strmcnik, S., Hadjiski, M., Stathaki, A., 2009, Online fuzzy identification for an intelligent controller based on a simple platform. *Engineering Applications of Artifficial Intelligence* **22**(5), 628–638.

Box, G.E.P., Cox, D.R., (1982), An analysis of transformations revisited, rebutted. *Journal of American Statististics Association* **77**, 209–210.

Bustince, H., Pagola, M., Barrenechea, E., Fernandez, J., Melo-Pinto, P., Couto, P., Tizhoosh, H.R., Montero, J., 2010, Ignorance functions: an application to the calculation of the threshold in prostate ultrasound images. *Fuzzy Sets and Systems* **161**(1), 20–36.

Castillo, O., Neyoy, H., Soria, J., Melin, P., Valdez, F., 2015, A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot. *Applied Soft Computing* **28**, 150–159.

Chen, G., Lockhart, R., Stephens, M., 2002, Box-Cox transformations in linear models: large sample theory and tests of normality. *The Canadian Journal of Statistics* **30**(2), 177–209.

Chen, Z., Zaou, S., Luo, J., 2017, A robust ant colony optimization for continuous functions. *Expert Systems with Applications*. **81**, 309–320.

Chowdhury, M., Rahman, A., Islam, R., 2018a, Malware analysis and detection using data mining and machine learning classification. In J. Abawajy, K-K. R. Choo, & R. Islam (eds.), *International Conference on Applications and Techniques in Cyber Security and Intelligence – Applications and Techniques in Cyber Security and Intelligence* (Vol. 580, pp. 266–274). (Advances in Intelligent Systems and Computing; Vol. 580). Springer-Verlag Ltd., London.

Chowdhury, M., Rahman, A., Islam, M.R., 2018b, Protecting data from malware threats using machine learning technique. In *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1691-1694). (United States: IEEE, Institute of Electrical and Electronics Engineers). IEEE, New York.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977, Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistics Society* **39**(1), 1–38.

Deng, L., 2010, An overview of deep-structured learning for information processing. *Proceedings of the Asia-Pacific Signal and Information Processing Association*, vol. 1, pp. 2–4.

Deng, L., Dong, Y., 2014, Deep learning: methods and applications. *Foundations and Trends in Signal Processing* **7**(3), 197–387.

Deng, S.P., Zhu, L., Huang, D.S., 2016, Predicting hub genes associated with cervical cancer through gene co-expression networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **13**(1), 27–35.

Doerks, T., van Noort, V., Minguez, P., Bork, P., 2012, Annotation of the M. tuberculosis hypothetical orfeome: Adding functional information to more than half of the uncharacterized proteins. *Plos One* **7**(4), e34302.

Eaton, C., Deroos, D., Deutsch, T., Lapis, G., Zikopoulos, PC., 2012, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill, New York.

Eluyode, O.S., Akomolafe, D.T., 2013. Comparative study of biological and artificial neural networks. *European Journal of Applied Engineering and Scientific Research* **2**(1), 36-46.

Gallier, J., 2013, *Fundamentals of Linear Algebra and Optimization*. University of Pennsylvania, Philadelphia, PA.

Ganter, B., Wille, R., 1999, *Formal Concept Analysis – Mathematical Foundations*. Springer Verlag, NY.

Gerhardt, B., Griffin, K., Klemann, R., 2012, *Unlocking value in the fragmented world of big data analytics*. Cisco Internet Business Solutions Group, June.

Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R., 2005, Neighbourhood components analysis. *Advances in Neural Information Processing Systems* **17**, 513–520.

Granat, R.A., 2004, *Regularized deterministic annealing EM for hidden Markov models*. University of California Press, Los Angeles, CA.

Graves, D., Pedrycz, W., 2010, Kernel-based fuzzy clustering and fuzzy clustering: a comparative experimental study. *Fuzzy Sets and Systems* **161**(4), 522–543.

HPCC Systems Homepage, 2018, http://hpccsystems.com/, last access 2018/08/15.

Hwang, C.-L., Wu, H.-M., Shih, C.-L., 2009, Fuzzy sliding-mode under-actuated control for autonomous dynamic balance of an electrical bicycle. *IEEE Transactions on Control Systems Technology* **17**(3), 658–670.

Jain, P., Kulis, B., Davis, J. V., Dhillon, I. S., 2012, Metric and kernel learning using a linear transformation. *Journal of Machine Learning Research* **13**(1), 519–547.

Jiang, G., Ogasawara, K., Endoh, A., Sakurai, T., 2003, Context-based ontology building support in clinical domains using formal concept analysis. *International Journal of Medical Informatics* **71**(1), 71–81.

Kamruzzaman, S.M., Sarker, A.M.J., 2011, A new data mining scheme using artificial neural networks. *Sensors* **11**(5), 4622–4647.

Kazakov, A.L., Lempert, A.A., 2015, On mathematical models for optimization problem of logistics infrastructure. *International Journal of Artificial Intelligence* **13**(1), 200–210.

Li, X., Jin, X., Wang, H., Zhang, X., Lin, Z., 2016, Structure, evolution, and comparative genomics of tetraploid cotton based on a high-density genetic linkage map. *DNA Repair* **23**(2), 127–136.

Liao, S., Tammaro, M., Yan, H., 2016, The structure of ends determines the pathway choice and Mre11 nuclease dependency of DNA double-strand break repair. *Nucleic Acids Research*, **15**(2), 135–142.

Liu, Y., Chen, G., Ying, M., 2005, Fuzzy logic, soft computing and computational intelligence. *Eleventh International Fuzzy Systems Association World Congress*, vol. 3, pp. 1376–1381.

Madden, S., 2012, From databases to big data. *IEEE Internet Computing* **16**(1), 4–6.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, AH., 2011, *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.

Medina, J., Ojeda-Aciego, M., 2010, Multi-adjoint t-concept lattices. *Information Sciences* **180**(5), 712–725.

Medina, J., Ojeda-Aciego, M., 2013, Dual multi-adjoint concept lattices. *Information Sciences* **225**(1), 47–54.

Methling, T., Braun-Unkhoff, M., Riedel, U. 2017, A novel linear transformation model for the analysis and optimisation of chemical kinetics. *Combustion Theory and Modelling* **21**(3), 503–528.

Narukawa, Y., Torra, V., 2009, Multidimensional generalized fuzzy integral. *Fuzzy Sets and Systems* **160**(6), 802–815.

Perez, J., Milanés, V., Godoy, J., Villagrá, J., Onieva, E., 2013, Cooperative controllers for highways based on human experience. *Expert Systems with Applications* **40**, 1024–1033.

Phan-Luong, V., 2008, A framework for integrating information sources under lattice structure. *Information Fusion* **9**, 278–292.

Pozna, C., Precup, R.-E., Tar, J.K., Škrjanc, I., Preitl, S., 2010, New results in modelling derived from Bayesian filtering. *Knowledge-Based Systems* **23**(2), 182–194.

Precup, R.-E., Angelov, P., Costa, B.S.J., Sayed-Mouchaweh, M., 2015, An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry* **74**, 75–94.

Precup, R.-E., Hellendoorn, H., 2011, A survey on industrial applications of fuzzy control. *Computers in Industry* **62**(3), 213–226.

Precup, R.-E., Preitl, S., 2004, Optimisation criteria in development of fuzzy controllers with dynamics. *Engineering Applications of Artificial Intelligence* **17**(6), 661–674.

Precup, R.-E., Preitl, S., 2006, PI and PID controllers tuning for integral-type servo systems to ensure robust stability and controller robustness. *Electrical Engineering* **88**(2), 149–156.

Precup, R.-E., Preitl, S., Petriu, E.M., Tar, J.K., Tomescu, M.L., Pozna, C., 2009, Generic two-degree-of-freedom linear and fuzzy controllers for integral processes. *Journal of the Franklin Institute* **346**(10), 980–1003.

Preitl, S., Precup, R.-E., Fodor, J., Bede, B., 2006, Iterative feedback tuning in fuzzy control systems. Theory and applications. *Acta Polytechnica Hungarica* **3**(3), 81–96.

Rabiner, L.R., 1989, A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**(2), 257–286.

Rahman A., Nimmy S.F., Sarowar, G., 2018, Developing an automated machine learning approach to test discontinuity in DNA for detecting tuberculosis. In: Xu J., Cooke F., Gen M., Ahmed S. (eds.). *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*. Lecture Notes on Multidisciplinary Industrial Engineering, vol. 1 pp. 277–286. Springer, Switzerland.

Rahman, A., 2008a, A review of small area estimation problems and methodological developments. *NATSEM Discussion Papers Series* **DP66**, 1–56.

Rahman, A., 2017b, An analysis of statistics-based data preprocessing mechanisms for privacy protection in big data. *The 4th Cyber Security Symposium*, Wagga Wagga, Australia, June.

Rahman, A., 2018, *Some novel modelling techniques in data science. The Big-Data and Statistical Sciences Workshop*, Charles Sturt University, Australia, Aug. 7 (2018).

Rahman, A., Harding, A., 2017, *Small Area Estimation and Microsimulation Modeling*. CRC Press, Boca Raton.

Rahman, A., Harding, A., Tanton, R., Liu, S., 2010, Methodological issues in spatial microsimulation modelling for small area estimation. *International Journal of Microsimulation* **3**(2), 3–22.

Rahman, A., Upadhyay, S.K., 2015, A Bayesian reweighting technique for small area estimation. In U. Singh, A. Loganathan, S. K. Upadhyay, & D. K. Dey (eds.), *Current Trends in Bayesian Methodology with Applications, 1st Ed.* (vol. 1, pp. 503–519). CRC Press, Florida.

Rahman, A., 2008b, *Bayesian Predictive Inference for Some Linear Models Under Student-t Errors*. VDM Verlag, Saarbrucken.

Rahman, A., 2017a, Microdata versus Big Data: Computational and inferential challenges in data science. 1. *7th International Conference on Cloud Computing, Data Science and Engineering*, Noida, India.

Rajasekaran, S., Pai, G.A.V., 2011, *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*. PHI Learning Private Ltd., New Delhi.

Rojas, R., 1996, *Neural Networks*. Springer-Verlag, Berlin.

Sáez, J.A., Luengo, J., Herrera, F., 2016, Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure. *Neurocomputing* **176**(1), 26–35.

Singh, S., Singh, N., 2011, Big data analytics. *IEEE International Conference on Communication, Information & Computing Technology*, Mumbai India, Oct.

Tchurikov, N.A., Yudkin, D.V., Gorbacheva, M.A., Kulemzina, A.I., Grischenko, I.V., Fedoseeva, D.M., Sosin, D.V., Kravatsky, Y.V., Kretova, O.V., 2016, Hot spots of DNA double-strand breaks in human rDNA units are produced in vivo. *Scientific Reports* **6**, 25866.

Tomescu, M.L., Preitl, S., Precup, R.-E., Tar, J.K., 2007, Stability analysis method for fuzzy control systems dedicated controlling nonlinear processes. *Acta Polytechnica Hungarica* **4**(3), 127–141.

Vascak, J., 2012, Adaptation of fuzzy cognitive maps by migration algorithms. *Kybernetes* **41**(4), 429–443.

Vrkalovic, S., Lunca, E.-C., Borlea, I.-D., 2018, Model-free sliding mode and fuzzy controllers for reverse osmosis desalination plants. *International Journal of Artificial Intelligence* **16**(2), 208–222.

Vrkalovic, S., Teban, T.-A., Borlea, I.-D., 2017, Stable Takagi-Sugeno fuzzy control designed by optimization. *International Journal of Artificial Intelligence* **15**(2), 17–29.

Wagarachchi, M., Karunananda, A., 2017, Optimization of artificial neural network architecture using neuroplasticity. *International Journal of Artificial Intelligence* **15**(1), 112–125.

Zeng, D., Lin, D.Y., 2007, Maximum likelihood estimation in semiparametric regression models with censored data*. Journal of the Royal Statistical Society: Series B* **69**(4), 507**–**564.