Datamining Techniques for Clustering Seismic Activities

Bala TYODEN, Hamza EROL

Abstract— **Seismic waves are the vibrations from earthquakes that travel through the Earth. They are recorded on instruments called seismographs which record a varying amplitude of ground oscillations including time, locations, and magnitude of an earthquake, and can detect strong earthquakes from sources anywhere in the world. Many methods have been developed for predicting time, magnitude and place of earthquake occurrence but are yet to be precise.**

In this paper, we explore the application of statistical techniques via model-based unsupervised, semi-supervised and supervised learning algorithms for earthquake clustering which can lead to higher classification accuracy and prediction of seismological activities such as earthquake occurrences. The emphasis is on datamining methods which attempt to quantify the probability of an earthquake occurring within specified time, magnitude, and space. The main advantage of this approach is its flexibility in addressing complex problems for time-varying conditions for every point in the region under study which makes it methodologically appealing and useful in practice.

Index Terms— **Earthquake clustering, learning algorithms, prediction, datamining.**

I. INTRODUCTION

Data mining is a process of extracting implicit, previously unknown, but potentially useful information and knowledge from a large quantity of noisy, ambiguous, seldom incomplete and random data in the practical application [1]. Two types of models are used to analyze the relationships in data patterns; descriptive model, which describe patterns and create meaningful subgroups or clusters and predictive model, using the patterns of known results to forecast explicit values. With the advancement of datamining however, prediction of earthquakes is a very difficult and challenging task [2], and cannot operate only at one level of resolution for all occurrences.

Understanding of earthquake dynamics and the development of forecasting algorithms require robust methods in both measurement and analysis of geological data. The Gutenberg-Richter power-law distribution of earthquake sizes postulates that in space and time, largest events are surrounded by a large number of small events for example [3], while [4] infers that the multi-dimensional and multi-resolutional structure of global seismic clusters depend strongly on geological and geophysical conditions which can be investigated through mixture models.

A study of mixture model (mixmod) library [5] was devoted to three kinds of different classification tasks. Its main task is *unsupervised* classification but *supervised* and *semi-supervised* classifications can benefit from its meaningful models and efficient algorithms. The ultimate goal of their proposed mixmod library is to optimize several parsimonious and meaningful models, depending on the type of variables to be considered so that such models can provide simple interpretation of groups with high-dimensionality.

In this study, we proposed the approach of combining unsupervised, semi-supervised and supervised algorithms for model-based clustering to recognize patterns between the clusters and earthquake occurrences, and to advance understanding of effective model-based clustering methods for complex observations such as seismological hazards.

In general, clustering and classification are concerned with assigning labels to observations so that they are partitioned into meaningful groups, or classes. In a model-based situation, classifiers, i.e., functions that map a given observation x to a class label y, are constructed based on probability models. Moreover [6], further explained that when both y and x are known, then the observation is concluded as labeled. We define the data matrix of labelled observations by $\mathbb{X}_1 = ({x^T}_{11}, {x^T}_{12}, \ldots, {x^T}_{1n_1})^T$ and store the observed class labels in indicator matrix $\mathbb{Z}_1 = ({z^T}_{11}, {z^T}_{12}, \ldots, {z^T}_{1n_1})^T$. where D_L refers to labelled data, comprised of the set $\{\mathbb{X}_1, \mathbb{Z}_1\}$ and, D_U the matrix of unlabelled observations which is denoted by $\mathbb{X}_2 = ({x^T}_{21}, {x^T}_{22}, \ldots, {x^T}_{2n_2})^T$, and since D_U does not include the unknown class labels, it is denoted by $\mathbb{Z}_2 = ({z^T}_{21}, {z^T}_{22}, \ldots, {z^T}_{2n_2})^T$.

The task of classification technique can be performed using a variety of techniques. The first approach is supervised learning, where labelled observations are used to build a classification rule from which to group the remaining unlabelled observations. On the contrary, the classifier used in unsupervised learning or clustering relies solely on unlabelled observations. Similar to supervised learning, a semi-supervised learning approach includes missing labels $\mathbb{Z}_2$. In contrast to supervised learning, semi-supervised learning makes use of both labelled and unlabelled data, which we denote by $D_0 = D_U \cup D_L$.

Although the inclusion of unlabelled data has proven to be beneficial in many classification applications, it is possible that including unlabelled observations may lead to a larger classification error, for example., when the postulated model is incorrect [7]. In addition, they show that labelled samples are exponentially more valuable than unlabelled samples in reducing classification error when a two component Gaussian mixture with unknown mixing proportions is considered. In

Bala TYODEN, Department of Statistics, Faculty of Arts and Sciences, Çukurova University, Adana. TURKEY.

Prof. Dr. Hamza EROL, Department of Statistics, Faculty of Arts and Sciences, Çukurova University, Adana. TURKEY.

such cases, it may seem reasonable to assign more weight to labelled observations in the estimation procedure.

II. METHOD

For our proposed approach, the kmeans and PCA algorithms were used in the unsupervised setting to extract the number of components that adequately represent the data. In the semi-supervised setting, the performance of Locally Weighted Learning (LWL) algorithm was compared with the Naïve Bayes and Instance-Based Learning (IBk) algorithms, while in the supervised setting, the Random Forest was compared with J48 and Simple Classification and Regression Trees (SimpleCART) algorithms.

A. *Unsupervised Learning*

Kmeans

The kmeans algorithm is an algorithm for putting N data points in an I-dimensional space into K clusters. Each cluster is parameterized by a vector $m^{(k)}$ called its mean. The data points will be denoted by $\{x^{n}\}$ where the n runs from 1 to the number of data points N. Each vector X has I components x_i. Assuming X resides in is a real space and defines distances between points, for example,

$$d(x,y) = \frac{1}{2}\sum_{i}(x_i - y_i)^2 \qquad (1)$$

the kmeans $\{m^{(k)}\}$ are initialized to random values K-means is then an iterative two-step algorithm. In the assignment step, each data point n is assigned to the nearest mean. In the update step, the means are adjusted to match the sample means of the data points that they are responsible for until convergence.

Kmeans is undoubtedly the most widely used partitional clustering algorithm. Unfortunately, due to the non-convexity of the model formulations, expectation-maximization (EM) type algorithms converge to different local optima with different initializations. Recent studies [8] have identified that the global solution of Kmeans cluster centroids lies in the principal component analysis (PCA) subspace, because the PCA subspace is much smaller than the original space, searching in the PCA subspace is both more effective and efficient. Based on this insight, we shall proceed with the principal component analysis.

Principal Component Analysis (PCA)

The PCA aims to reduce the dimensionality of a high-dimensional data set consisting of a large number of interrelated variables and at the same time to retain as much as possible of the variation present in the data set [9]. The principal components (PCs) are new variables that are uncorrelated and ordered such that the first few retain most of the variation present in all of the original variables. Assume $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$ as m vectors in $\mathbb{R}^{d}$, the vector dimension can be reduced using a linear transformation. A matrix $W \in \mathbb{R}^{n,d}$, where $(n < d)$, induces a mapping $\mathbf{x} \mapsto W\mathbf{x}$ where $W\mathbf{x} \in \mathbb{R}^{d,n}$ is the lower dimensionality representation of $\mathbf{x}$. Then, a second matrix $U \in \mathbb{R}^{n,d}$, can be used to (approximately) recover each original vector $\mathbf{x}$ from its compressed version. That is, for a compressed vector $y = W\mathbf{x}$, where y is in the low dimensional space $\mathbb{R}^{n}$, we can construct $\tilde{\mathbf{x}} = U\mathbf{y}$, so that $\tilde{\mathbf{x}}$ is the recovered version of $\mathbf{x}$ and resides in the original high dimensional space $\mathbb{R}^{d}$.

B. *Semi-supervised Learning*

Semi-supervised learning is a situation in which in the training data, some of the samples are not labelled. Semi-supervised estimators in are able to make use of this additional unlabelled data to better capture the shape of the underlying data distribution and generalize better to new samples. These algorithms can perform well when we have a very small amount of labelled points and a large amount of unlabelled points. In semi-supervised classification we start from a training data with s labelled instances and u unlabelled samples, often $u \gg s$. It is of importance to manage a better classifier or clustering result than from the unlabelled observations alone for instance [10].

Locally Weighted Learning (LWL)

LWL methods are non-parametric and performs prediction by local functions which only uses a subset of the data. The basic idea behind LWL is that instead of building a global model for the whole function space, for each point of interest a local model is created based on neighbouring data of the query point. The data point becomes a weighting factor which expresses the influence of the data point for the prediction. In general, data points which are in the close neighbourhood to the current query point are receiving a higher weight than data points which are far away. In LWL, the processing of the training data is shifted until a query point is addressed. This approach makes LWL a very accurate function approximation method where it is easy to add new training points. LWL has the advantages to solve function approximations problem accurately.

Following a standard regression model,

$$(y) = f(x) + \epsilon \qquad (2)$$

is assumed with a continuous function $f(x)$ and noise ϵ. The basic cost function of LWL is defined by [11] is,

$$J = \frac{1}{2}\sum_{i=1}^{n} w_i(x_q)(y_i - x_i\beta_q)^2 \qquad (3)$$

with the components, $D = \{(x_i, y_i) | i = 1,2,\dots,n\}$ where each data point x_i belongs to a corresponding output value y_i, the query point X_q, which is the position where we want a prediction $\hat{y}_q$,weights w_i describe the relevance of the corresponding training set (x_i, y_i) for the current prediction. They are dependent on the query point and are computed by a weighting function and the regression coefficient β_q of our linear model, which we want to obtain for doing the prediction. The goal is to find a β_q that minimizes equation (3) for the current query point X_q. An important difference to global least square methods is that β_q is dependent of the current query point. One advantage of LWL is the possibility to switch very easily between different weighting functions.

C. *Supervised Learning*

Supervised learning attempts to discover an optimal representation of a data set with known class memberships.

The goal of supervised learning is typically the building of a classifier for classifying unlabelled items. Given a set of training examples of the form $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, a learning algorithm seeks a function $g: X \rightarrow Y$, where *X* is the input space and *Y* is the output space. The function g is an element of some space of possible functions *G*, usually called the *hypothesis space*. It is sometimes convenient to represent *g* using a scoring function f: $X \text{ x } Y \rightarrow \mathbb{R}$ such that *g* is defined as returning the *y* value that gives the highest score, $g(x) = arg \max_y f(x, y)$.

Many learning algorithms are probabilistic models where *g* takes the form of a conditional probability model $g(x) = P(y|x)$, or *f* takes the form of a joint probability model $f(x, y) = P(x, y)$. For example, naive Bayes and linear discriminant analysis are joint probability models, whereas logistic regression is a conditional probability model.

There are two basic approaches to choosing *f* or *g*: empirical risk minimization and structural risk minimization. Empirical risk minimization seeks the function that best fits the training data. Structural risk minimize includes a penalty function that controls the bias/variance tradeoff. In both cases, it is assumed that the training set consists of a sample of independent and identically distributed pairs (x_i, y_i). In order to measure how well a function fits the training data, a loss function $L: Y \text{ x } Y \rightarrow \mathbb{R} \geq 0$ is defined [12]. For training example (x_i, y_i), the loss of predicting the value $\hat{y}$ is $(y_i, \hat{y})$. The risk *R(g)* of function *g* is defined as the expected loss of *g*. This can be estimated from the training data as,

$$R_{emp}(g) = \frac{1}{N} \sum_i L(y_i, g(x_i)) \qquad (4)$$

Random Forest

Through repeated sampling of sets of predictor variables at the tree splitting stage, random forests offer a natural approach to handling collinearity among attributes. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges as to a limit as the number of trees in the forest becomes large [13] thus, the generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. The Random forests algorithm addresses these problems by measuring the importance of a particular variable and comparing the out-of-bag errors for the trees in the ensemble with the out-of-bag errors when the values for that variable are permuted randomly. Differences are averaged over all trees, and divided by the standard error. The random forest is a predictor consisting of a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$, where $\Theta_1, \Theta_2 \ldots$ are i.i.d. outputs of a randomizing variable Θ. These random trees are combined to form the aggregated regression estimate.

III. Results

Data was obtained from selected occurrences of seismic activities from the European–Mediterranean Seismological Centre (EMSC) which comprises of measurements of earthquake activities from ten different periods. Six thousand earthquakes observations were measured within a decade. The measurements recorded are; the independent variable, *Year* – year of earthquake occurrences (2004 to 2014), and the dependent variables; *Time* -time of earthquake occurrence (UTC); *Depth* – distance from the top surface to the actual point event (km); *Magnitude*– size of the earthquake (Richter's scale); *Latitude* – angular distance of point north to south of equator (degrees); and *Longitude* – angular distance of point east to west of equator (degrees).

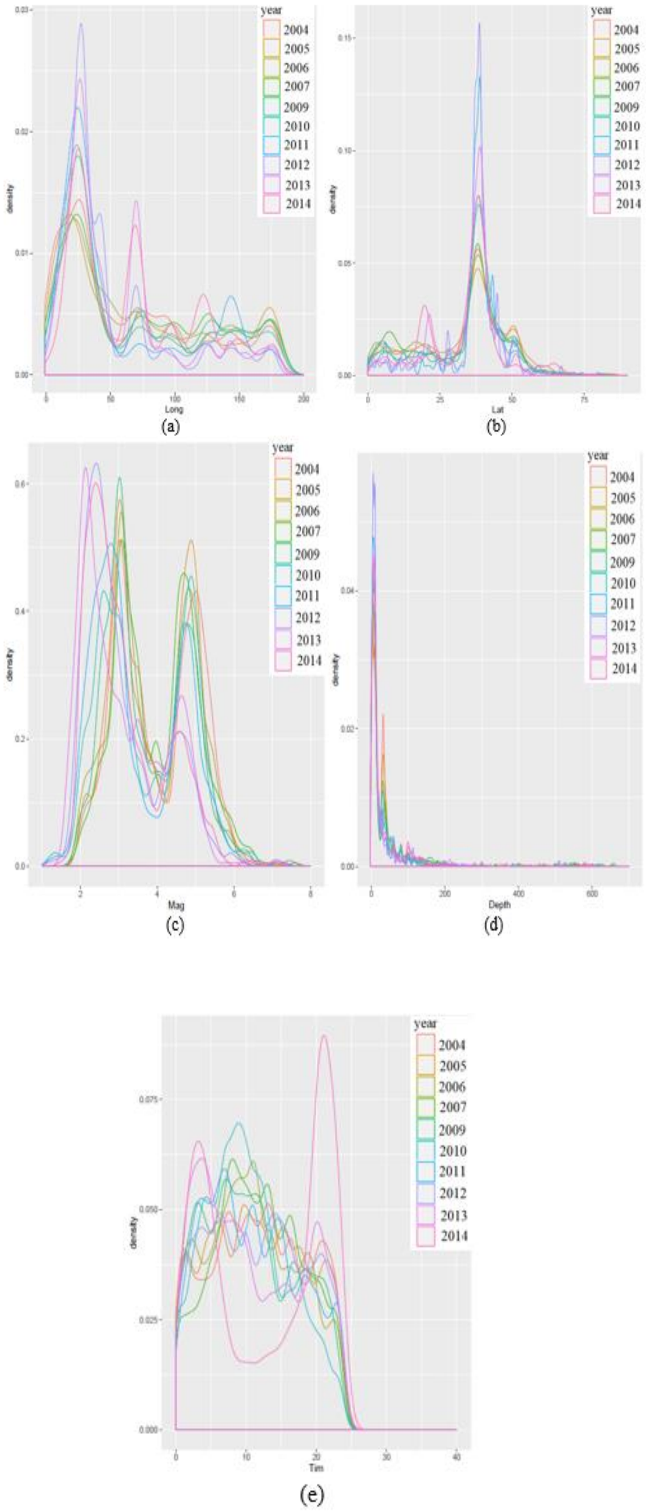

Figure 1. Geometric Densities of earthquakes at (a) Longitude and (b) Latitude, (c) Magnitude of occurrences, (d) Depth from epicenters, and (e) Time of occurrences.

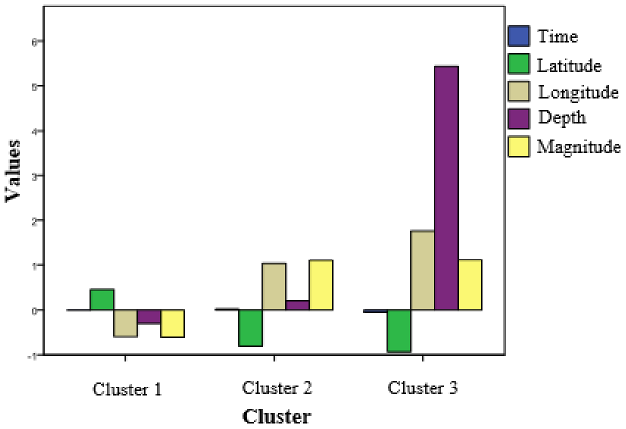

Figure 2. Kmeans plots for density Clusters (k=3)

From the kmeans computation, a typical earthquake occurrence in cluster 1 has a standardized score of -0.00970 time, .45431 latitude, -0.60081 longitude, -.30248 depth and -0.61285 magnitude. Cluster 1 has the lowest magnitude average values of the three clusters suggesting lighter earthquake magnitudes in cluster1, moderate magnitudes in cluster2 and very high magnitudes in cluster3. It is worth noting that the mean value of 0 was used as the standardized score.

Table 1. Total variance of Principal Components Analysis.

Comp	*Eigenvalues* Total	 % of Variance	 Cum. %	 Total	*Rotation Sums of Squared Loadings* Total
1	2.380	47.590	47.590	2.380	2.256
2	1.000	20.009	67.599	1.000	1.001
3	.748	14.965	82.563	.748	1.372
4	.610	12.205	94.769		
5	.262	5.231	0.9872		

From Table 1, the principal components are consistent with the three cluster solution as suggested by the kmeans, accounting for 82.60% of the total variability in the model. The loadings for the third factor are *Magnitude* and *Longitude*, for the second factor are *Depth* and *Latitude* while *Time* appears invariant of both components as shown in Figure 3.

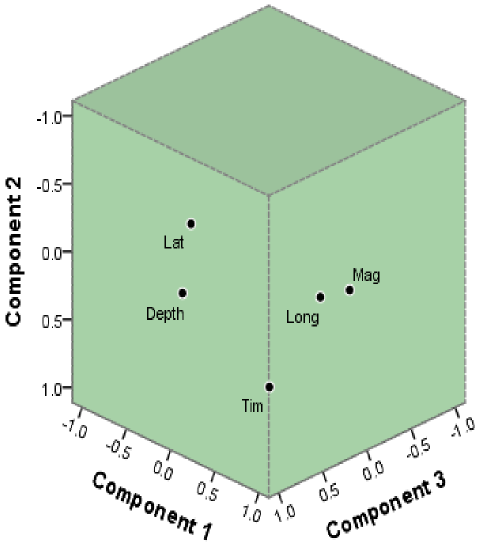

Figure 3. Rotated space component plot for earthquake data

Performance Comparison of Algorithms

A succinct performance comparison between semi-supervised and supervised learning functions as well as the sensitivity and specificity of the classifiers in clustering the earthquake data, are presented.

Table 2. Classifier Performance for Semi-Supervised and Supervised Learning

	n = 6000					
	Classifier					
	Naïve Bayes	**LWL**	**IBK**	**J48**	**Simple CART**	**Random Forest**
Statistic	Semi-Supervised			Supervised		
CCI	0.6737	0.6890	0.8073	0.9130	0.9148	0.9318
ICI	0.3263	0.3110	0.1927	0.0870	0.0852	0.0682
Kappa	0.3641	0.3161	0.6353	0.8362	0.8378	0.8706
MAE	0.1861	0.2063	0.0965	0.0548	0.0576	0.0642
CC	0.9177	0.9872	0.8073	0.9500	0.9558	0.9948

Correctly Classified Instances = CCI, Incorrectly Classified Instances = ICI, Mean absolute error = MAE, Coverage of cases (α=0.95)

Table 3. Classifiers Accuracy Rates

	n = 6000					
	Classifier					
	Naïve Bayes	**LWL**	**IBK**	**J48**	**Simple CART**	**Random Forest**
Statistic	Semi-Supervised			Supervised		
TP Rate	0.674	0.6890	0.8070	0.9130	0.915	0.932
FP Rate	0.296	0.386	0.1560	0.0680	0.08	0.062
Precision	0.654	0.612	0.8060	0.9130	0.914	0.931

True Positive = TP
False Positive = FP

IV. Conclusion

In the unsupervised learning approach, we obtained the estimates of hidden variables as well as defining all plausible clusters or latent coordinates in the PCA model. The findings suggests that three component solution is adequate for a parsimonious solution of the earthquake dataset. This was supported by the kmeans algorithm and echoed by the PCA which was consistent with three cluster solution accounting for 82.60% of the total variability in the model. The loadings for the third factor are *Magnitude* and *Longitude*, for the second factor are *Depth* and *Latitude* while *Time* appears invariant of both components.

The supervised learning algorithms (J48, Simple CART and Random Forest), had higher classification accuracies than the semi-supervised learning algorithms (Naïve Bayes, Locally Weighted Learning and Instance-Based Learning). The Random Forest algorithm consistently outperformed the other classifiers with correctly classified instances and showing a better chance classification rate (Kappa = .876). The sensitivity and specificity result also indicated that the Random Forest had higher accuracy (TP = .932) and precision rates than the other learning algorithms.

Further research is needed to explore further possibilities of fitting a mixture distribution of non-Gaussian distribution(s) within the exponential family with mixing parameters to find a distribution that best fits the variable with the highest importance (magnitude) in the variables under investigation. This can increase the classification accuracy thus leading to higher prediction chances of future earthquake occurrences.

REFERENCES

[1] Baoa, F., Heb X., Zhaoc F. (2012). Applying Data Mining to the Geosciences Data. International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012), Volume 33, P685–689

[2] Geller, R. S., Jackson, D.D., Kagan, Y.Y., and Mulargja, F. (1997), Earthquakes Cannot be Predicted, Science 275, 1616-1617.

[3] Wiemer, S., and Wyss, M. (2002). Mapping spatial variability of the frequency-magnitude distribution of earthquakes, in Advances in Geophysics, 45:259-302

[4] Erol, H. (2013). A Model Selection Algorithm for Mixture Model Clustering of Heterogeneous Multivariate Data. [978-1-4-799-0611-1]. IEEE

[5] Lebret R., Iovleff S., Langrognet F., Biernacki C., Celeux G., Govaertg. (2015). Rmixmod: The R Package of the Model-Based Unsupervised, Supervised and Semi-Supervised Classification Mixmod Library. Journal of Statistical Software, pp.241-270.

[6] Vrbik, I. and Mcnicholas, P.D. (2015). 'Fractionally-supervised classification', Journal of Classification 32(3), 359-381

[7] Vandewalle, V., Biernacki, C., Celeux, G., and Govaert, G. (2008). “Are Unlabeled Data Useful in Semi-Supervised Model-Based Classification? Combining Hypothesis Testing and Model Choice”, in Proceedings of the First Joint Meeting of the Societe Francophone de Classification and the Classification and Data Analysis Group of SIS, pp. 433–436

[8] Xu, Q., Ding, C., Liu, J. (2015). PCA-guided search for K-means. Pattern Recognition Letters Vol. 54, pp 50-55

[9] Ding, C., and He, X. (2004). K-means clustering via principal component analysis. In Proceedings of the Twenty-First International Conference on Machine Learning, ICML ’04, pages 225–232.

[10] Martinez-Uso, A., Pla, F., Sotoca, J.M (2010). A Semi-supervised Gaussian Mixture Model for Image Segmentation. In ICPR, pp.2941-2944.

[11] Englert, P. (2012). Locally Weighted Learning [online]. Darmstadt, Germany.

[12] VAPNIK, V. (1992). Principles of risk minimization for learning theory. Advances in Neural Information Processing Systems 4, pp. 831-838.

[13] Biau, G. (2012). Analysis of a random forests model, Journal of Machine Learning Research, Vol. 13, pp. 1063-1095.

Bala Tyoden holds a degree in Statistics, and M.Sc. in Statistics and Management Sciences, from the University of the West of England, Bristol, United Kingdom. He has worked with government and non-governmental organizations both in Nigeria and in the United Kingdom. He worked as a consultant for various multi-lateral institutions on research and development for developing countries. Due to his commitment to excellence, he was awarded a scholarship to pursue his doctoral programme (PhD) in statistics at Çukurova University, Adana, Turkey. He is interested in the areas of multivariate statistics, datamining and machine learning. He is a member of the Nigerian Statistics Society (NSS) as well as the Royal Statistics Society (RSS) of England, United Kingdom.

Hamza Erol received a BSc degree in 1989 from Mathematics Department, with computer sciences minor, of Faculty of Arts and Sciences, Middle East Technical University in Ankara Turkey. He worked at Mathematics Department of Faculty of Arts and Sciences, Çukurova University in Adana Turkey as a computer consultant between 1989 and 1992. He completed his MSc degree in 1991 from Mathematics Department of Institute of Natural Sciences of Çukurova University. He completed PhD degree in 1995 from Mathematics Department of Institute of Natural Sciences of Çukurova University. He worked as a computer science lecturer at Mathematics Department of Çukurova University for the time period 1992-1995. He served in 1996 his compulsory military service as a computer programmer lieutenant at Turkish Air Forces headquarters in Ankara Turkey. He worked as an Assistant Professor at Mathematics Department of Çukurova University for the time period 1995-1999. He worked as an Associate Professor at Mathematics Department of Çukurova University for the time period 1999-2002. He participated in the establishment of Statistics Department of Çukurova University. He worked as an Associate Professor at Statistics Department of Çukurova University for the time period 2002-2005. He was awarded the title Professor of Statistics at Statistics Department of Çukurova University in 2005 for Statistical Information Systems Section. He commissioned and participated in the establishment of Computer Engineering Department of Faculty of Engineering and Architecture, Çukurova University in 2003. He taught undergraduate and graduate level courses at Computer Engineering Department for the time period 2004-2011. He was department chair of Statistics Department of Çukurova University for the time period 2005-2008. He was chair of Electrical Machines, Statistical Information Systems, Remote Sensing and Geographical Information Systems Departments at Institute of Natural Sciences of Çukurova University. He worked as the dean of Computer Sciences Faculty and head of Software Engineering Department at Abdullah Gül University Kayseri Turkey between the period 2011 and 2014. He is currently department chair of Statistics at Çukurova University