Open Journal of Geology, 2014, 4, 551-570
Published Online October 2014 in SciRes. http://www.scirp.org/journal/ojg
http://dx.doi.org/10.4236/ojg.2014.410041

Cyclostratigraphy Data as a Proxy for Geoelectrical Data Interpretation: Application to Phosphate Minerals Exploration in Matam (Senegal)

Mapathe Ndiaye, Seydou Coulibaly, Winimi Saratou Zouwinaba, Issa Ba

Laboratoire de Mécanique et Modélisation, University of Thies, Thies, Senegal
Email: mapathe.ndiaye@univ-thies.sn

Received 29 August 2014; revised 25 September 2014; accepted 20 October 2014

Abstract

The vertical electrical soundings on the Matam region phosphates deposits are interpreted by inversion. To retrieve lithology from the obtained resistivities, mechanical drilling was performed to compare directly the lithologies and the geoelectrical resistivities. The obtained relation is not simple and is not sufficient to interpret all the VES. This situation motivated the collection of supplementary information from cyclostratigraphy. Geoelectrical and cyclostratigraphic information is combined using fuzzy logic techniques to build a fuzzy inference system. The obtained results seem to be consistent with the stratigraphy of the investigated region and allow retrieving lithological succession. Further investigations are necessary for more accurate thickness determination.

Keywords

Cyclostratigraphy, Geoelectrical, Vertical Electrical Sounding, Phosphate, Matam, Fuzzy Logic

1. Introduction

The first index of existence of phosphates deposits in Matam (15˚15'22"N; 13˚03'41"N) has been unraveled in the beginning of the nineteen's [1]. Later, it has been substantiated during a drilling survey, in preparation of the Senegal River planning [2]. More recently, the increase of the phosphate economical rate motivates further investigations of the phosphates of Matam [3].

Despite these numerous phases, all the investigations were performed using destructive methods, mainly

How to cite this paper: Ndiaye, M., Coulibaly, S., Zouwinaba, W.S. and Ba, I. (2014) Cyclostratigraphy Data as a Proxy for Geoelectrical Data Interpretation: Application to Phosphate Minerals Exploration in Matam (Senegal). *Open Journal of Geology*, **4**, 551-570. http://dx.doi.org/10.4236/ojg.2014.410041

drilling, until now on. The high operating costs limited the extension of the investigations. To fix this constraint, geophysical investigation mainly geoelectrical methods are more and more used to extend the investigated area.

As for almost all the geophysical methods, there is no simple way to directly retrieve lithological information from measured data [4]. Nevertheless, while it is not possible to definitely eliminate direct investigation methods, their cost can be significantly reduced by combination with geophysical methods. An approach consists in calibrating the geophysical data using available samples from drills.

The aim of this work is to use the available stratigraphic data to define a more accurate interpretation model for geoelectrical data. The calibration is done combining cyclostratigraphy data by fuzzy logic methods.

2. Material and Method

2.1. Geological Settings

The ages in the investigated region vary from the center, occupied by the Polel bute (Quaternary) to the peripheral with respectively, the Formation of Saloum (Miocene), the formation of Matam (Lutetian) and the formation of Gorgol (Ypresian). The phosphates deposits are met in the formation of Matam. It has been proved, in previous exploration stages, that phosphate deposits index are frequently met around the butes resulting from the erosion of sedimentary deposits [5]. We investigated an area around the Polel bute with 15 vertical electrical soundings (VES) named S1 to S15 and 10 mechanical drills named M1 to M10 (**Figure 1**).

2.2. Geoelectrical Model

The VES reached 50 meters depth while mechanical drills were limited to 22.5 meters due to limitation of the drilling machine. Given that these data were collected at different campaigns and that the accessibility of the drilling machine was limited, the coordinates of the soundings and the drills were generally different, expect for a couple of locations. These matching locations were of valuable interest to understand the relationship between the electrical and lithological data.

The VES were interpreted by inversion using IX1D. We obtained 15 geoelectrical resistivity profiles. For 7 geoelectrical profiles (corresponding to VES 1, 2, 4, 9, 10, 11 and 12) the RMS error was acceptable (between 5% and 10%). For the 8 others (corresponding to VES 3, 5, 6, 7, 8, 13, 14 and 15) the RMS error exceeded 50% what make them unreliable for this study. We noticed that the bad VES were generally located on the formation of Saloum. This is probably linked to the advanced weathering of this formation, resulting in a bad soil/electrodes contact.

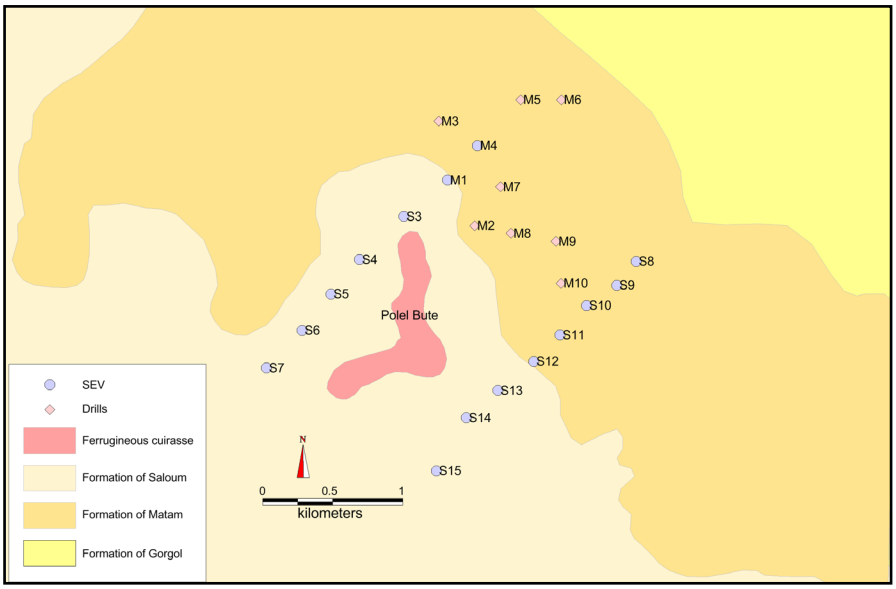

Figure 1. Geological setting around the Polel Bute showing the location of vertical electrical soundings and mechanical drills ([6]; modified).

To unravel the correlation between the obtained resistivity and the lithology, we built a "lithology versus geoelectrical" correlation model (GeoE) basing on matching locations between geoelectrical investigations and drilling (**Figure 1**). We used M1 and M4 drills with S2 and S1 VES respectively (**Figure 2**).

For the VES with acceptable RMS error, geoelectrical resistivity varies from 1 to 90 Ω·m. Basing on the matching locations, the correlation between lithology and geoelectrical resistivity was done comparing depth to depth as shown in the **Figure 2**.

To classify a layer, characterized by its geoelectrical resistivity, we plot its resistivity in the model, to obtain a score that can vary from 0% to 100% for a given lithology (**Figure 3**). The score is maximal if the resistivity is in the center of the interval. When the resistivity moves away from the center, the scores decrease linearly to 0% for the limits of the interval.

The GeoE correlation model allows mainly good identification clays, limestone and even phosphates. Nevertheless, the existence of thin alternating laminae, mixed in the cuttings gives various resistivities that overlap with the domain of clays (mix of clay and phosphates) or the domain of phosphates (mix of clay, limestone, limestone grains or coprolites). We observed a gap in the classification interval, between 30 and 43 Ω·m. This gap is probably the place of the facies not found in the geoelectrical model as sands, sandy clays or laterits. To solve this ambiguity, we used additional information from cyclostratigraphic data.

2.3. Cyclostratigraphy Model

Using 10 stratigraphic logs from the drills, we built a transition matrix (TrsM) corresponding the number of transition between two given facies [7]. The transition matrix method allowed retrieving transitions that are more likely characterized by a higher transition value (**Table 1**).

Given that the classification is done from the top to the bottom of the geoelectrical profile (the inverse of the stratigraphic sense), we know a priori the following facies of the resistivity under classification. Therefore, the scores from the TrsM matrix are obtained comparing the transitions to the next lithology for each one of the possible cases, previously obtained from GeoE classification. If the total number of transitions for all lithologies is N, the score for a specific lithology with n transitions to the next facies is n/N × 100.

If classification is still ambiguous after the TrsM step, additional information is collected using the total

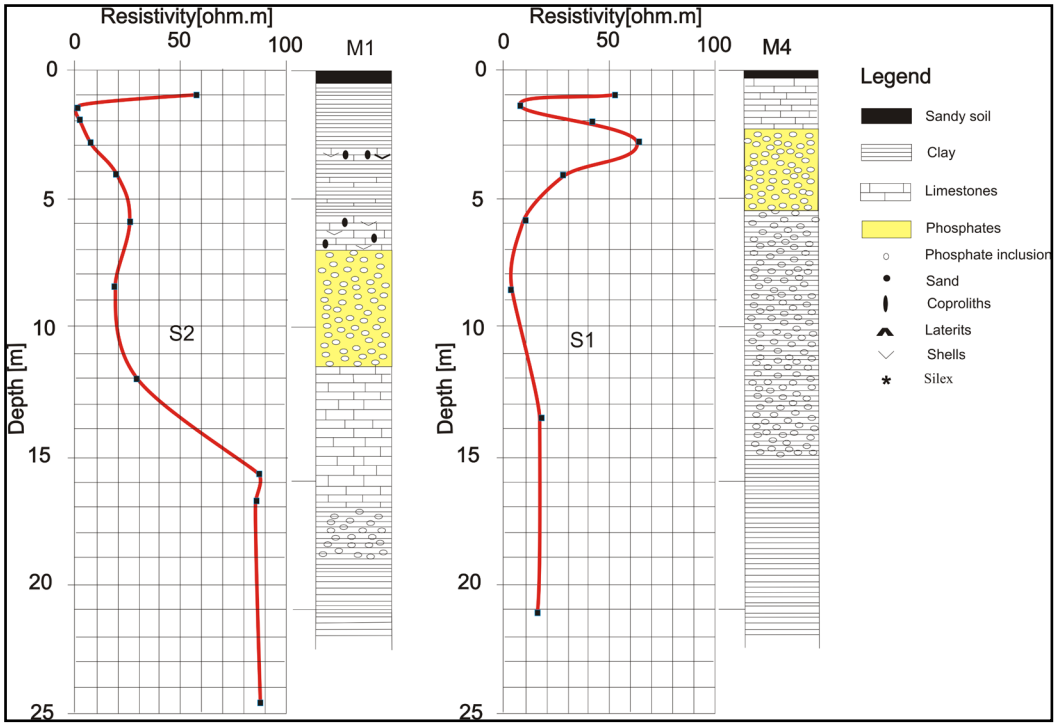

Figure 2. Correlation of the lithology with geoelectrical data for matching locations drills and VES.

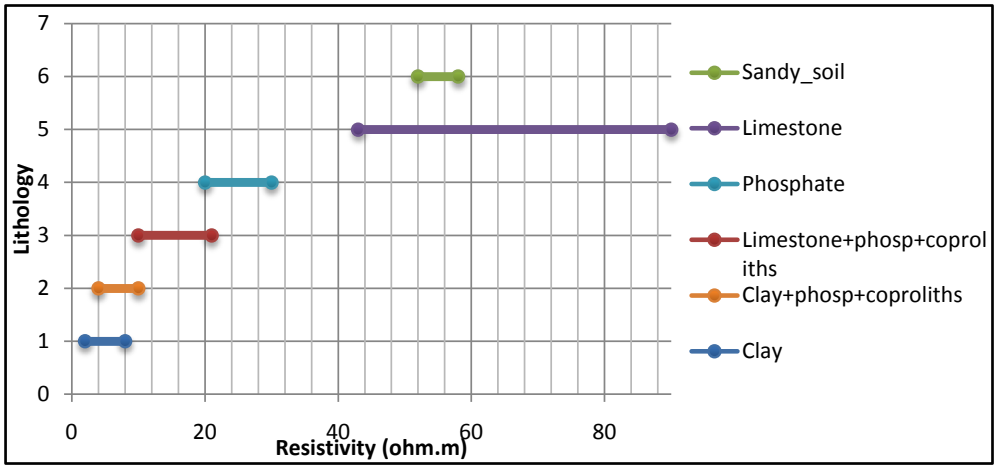

Figure 3. Schematic representation of GeoE model.

Table 1. Transition matrix of the facies met in the 10 drills (Ss: sandy soil, C: clay, L + c + p: Limestone with coproliths and phosphates, L: limestone, P: phosphate, C + c + p: clay with coproliths and phosphate, Sc: sandy clay, La: laterits, S: sand).

	Ss	C	L + c + p	L	P	C + c + p	Sc	La	S
Ss	0	0	0	0	0	0	0	0	0
C	2	0	5	14	8	6	0	2	1
L + c + p	0	4	0	2	0	1	0	1	0
L	2	13	0	0	2	2	0	0	0
P	0	6	2	2	0	1	0	0	1
C + c + p	0	6	1	1	2	0	0	0	0
Sc	1	0	0	0	0	0	0	0	0
La	0	0	0	0	0	0	2	0	0
S	0	1	0	0	0	0	0	1	0

number of occurrences of the lithology (MOcc). For instance, if two lithologies have the same score from the geoelectrical model, and same score from the transition matrix, the lithology with maximum number of occurrences will be considered. In this case, the score of a facies will be m/N × 100 where m is the total number of occurrences of the facies.

The scores obtained respectively from GeoE, TrsM and MOcc are combined and form a fuzzy inference system [8]. In fact, fuzzy logic (FL) allows defining membership function, to allocate each geoelectrical resistivity to a fuzzy set with a degree of membership. The final lithology is obtained considering the higher degree of membership defined by the sum of the scores.

2.4. Classification Process

The steps of the interpretation of a geoelectrical layer to a given lithology are summarized in the flow diagram (**Figure 4**).

Performing all these classification steps can be tedious, especially when the number of VES and/or the number of facies is high, as in our case. To avoid this difficulty we developed an application to automate the process (**Figure 5**). The java source code of the developed application is annexed.

Three steps are necessary to run the classification process (**Figure 4**). First, the operator must input the geoelectrical classes by specifying for a given lithology, the maximum and minimum resistivities (ρ_{min} and ρ_{max}). For classes that are not defined in the GeoE model, maximum and minimum should be 0 and 0. In a next step, the user has to input the transition matrix between all the existing lithologies, in the same order as listed in the

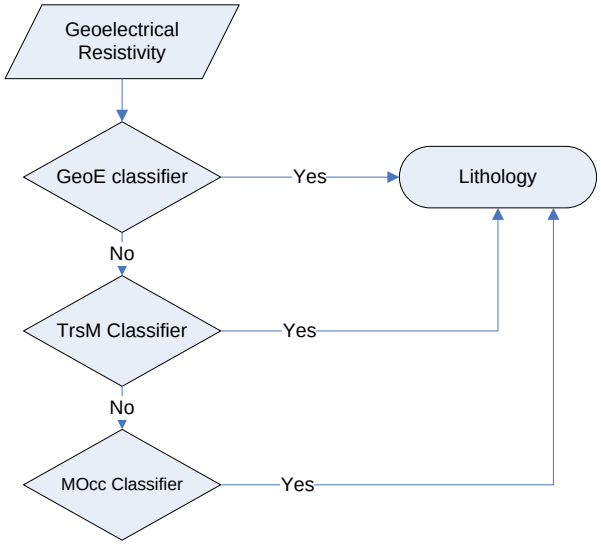

Figure 4. Flow diagram of the classification process. (GeoE: geoelectrical-lithology correlation model; TrsM: transition matrix; MOcc: maximum occurrence).

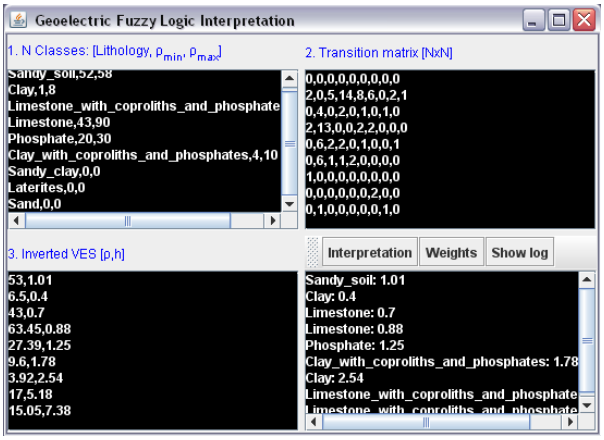

Figure 5. Screenshot of the classification application. We have the input area for classes (1), transition matrix (2) and inverted VES (3). The interpretation shows the thickness for each interpreted lithology.

classes. Strati-signal software [9] allows retrieving automatically the transition matrix in few steps.

Finally, the user has to key in the inverted VES (ρ, h) to be interpreted. When he hits the "interpretation" button, the classification is performed. The contribution of GoeE, TrsM and Mocc can be weighted from 0% to 100% using the "Weights" button. When the "show log" button is pressed, the main steps that have led to classify a given layer, the preselected facies and the obtained scores are shown.

3. Results and Discussion

The classification process allocates a lithology to each electrical resistivity basing on the three steps defined in the GeoE model, the TrsM matrix and the MOcc. It is worth to notice that only VES with satisfactory RMS error (RMS < 10%) are interpreted. The results of the classification of the VES are shown in **Table 2**.

To test the developed interpretation method, we used it to classify the VES S1 and S2. The aim is to directly compare the obtained results with the reality represented by drills M4 and M1 respectively. The results are shown in **Figure 6**.

Table 2. Results of the interpretation of VES with acceptable RMS error. For each sounding we have the resistivity ρ (Ω·m) and the interpreted lithology (Litho). For lithology symbols significance, refer to **Table 1**.

S1		S2		S4		S9		S10		S11		S12	
ρ	Litho	ρ	Litho	ρ	Litho	ρ	Litho	ρ	Litho	ρ	Litho	ρ	Litho
53	Ss	57.97	Ss	1371	C	54.75	L	38.09	C	26.6	P	131.22	C
6.5	C	1.8	C	88.47	L	22.9	P	4.68	C + c + p	22.9	P	11.6	L + c + p
43	L	2.9	C	158.93	C	57.1	Ss	16.47	L + c + p	48.4	L	29.76	P
63.45	L	7.8	C	138.8	L	63.45	L	24.3	P	39.23	C	48.8	L
27.39	P	20.53	L + c + p	217.77	C	61.49	L	9.6	C+ c + p	22.9	P	31.19	C
9.6	C + c + p	25.82	P	558.73	L	19.73	L + c + p	3.17	C	28.26	P	42	L
3.92	C	19.73	L + c + p	158.93	C	4.15	C + c + p	8.79	C + c + p	48.4	L	20.78	L + c + p
17	L + c + p	29.98	P	49.39	L	1.69	C	16.97	L + c + p	48.4	L	6.77	C
15.05	L + c + p	88.06	L	32.99	C	2.9	C	9.32	C + c + p	25.07	P		
		85.79	L			7.8	C			6.13	C		
		88.06	L			7.8	C			4.15	C + c + p		

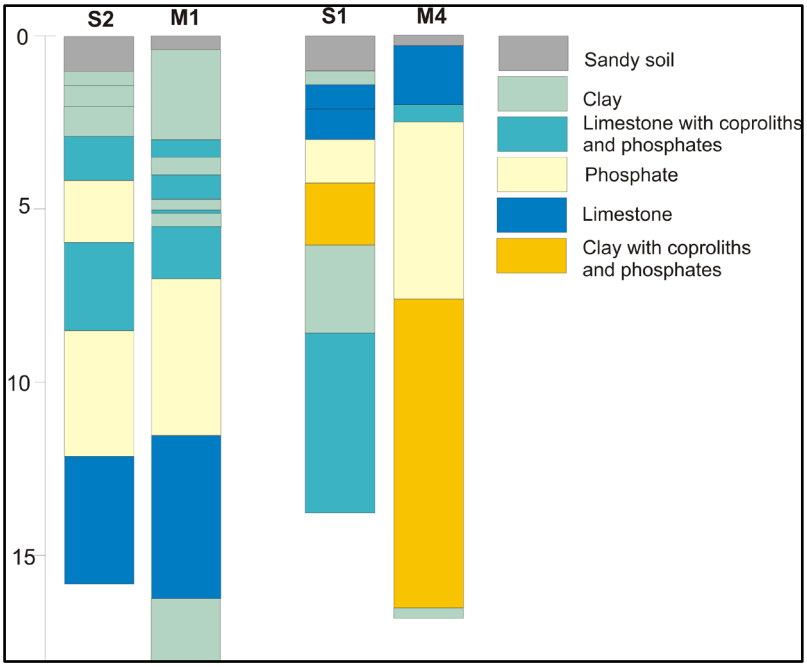

Figure 6. Interpretation of VES S1 and S2 using the developed model and comparison with drills M4 and M1 respectively.

The interpretation of S2 shows a sandy soil on surface followed by clays and Limestone with coprolites and phosphates alternations. The succession of lithologies shown by the interpreted VES is similar with that observed in M1. Nevertheless, we noticed a wrong presence of a phosphate layer, probably due to the thin alternation of clays and limestone layers. In fact, the high resistivities of the limestone and the low resistivities of clays are combined to produce a resistivity in the range of the phosphates.

In the case of S1 interpretation, the lithology succession shows a sandy soil followed by a limestone. We noticed a thin clay layer, whose presence is linked to the decrease of the resistivity of the sandy soil with depth in relation with the water content. As in M2, we noticed the presence of a phosphate layer followed by a mix of clays, phosphates and coprolites. The interpreted lithology succession ends with a Limestone with coprolites and

phosphates layer. The misclassification of this last layer seems to be linked to its content in phosphates and coprolites, which is almost the same as for the clays with phosphates and coprolites.

Therefore, the obtained lithology succession in S1 is, as for S2, consistent with what we observed in the M4 drill.

The main limitation of the interpretation model resides in the determination of the thickness of the layers. We have at least identified two possible sources of errors:

-The first source of error is linked to the nature of the drill data. We used destructive drilling instead of coring. And due to possible contamination of the cuttings, the observed drilling succession and thickness is not accurate. To improve the model, the use of cores is advised. Moreover, observing the cores will allow a better appreciation of the structure of the investigated rocks, given that the measured resistivity does not rely directly on the lithology but on the structure of the rock, as advocated by the Archie law [10].

-The second source of error is linked to the nature of geoelectrical prospection. The measured resistivity values are controlled by the Dar Zarouk parameters [11]. Therefore, according the existing lithology and the observed thicknesses, the principles of suppression and equivalence cannot always be ruled out. Additionally, as for all geophysical methods, the resolution of geoelectrical investigation decreases with depth. The contribution of the cyclostratigraphic data as a priori knowledge is therefore of a valuable input.

4. Conclusions

The interpretation method developed in this study can contribute to phosphates exploration costs reduction. The classification seems to be satisfactory, as obtained lithological successions are in conformity with those observed in the investigated region.

The lithology of the first layer should be considered with caution in certain conditions. In fact, the dryness of the soils surface under the climatic conditions causes the resistivities to be higher than the normal in superficial layers. Moreover, the first layers suffer of the lack of TM data as the number of transitions increases with depth.

We also notice that undefined resistivity classes are not found in the classification results. It is therefore necessary to have a more complete GeoE model, with resistivity ranges for all possible lithologies.

The classification should be improved using spatial information. In fact, the proximity of mechanical drills should be weighted in such a manner that the nearest information will contribute more in the classification.

It has been advocated that the region is affected by a rapid lateral variation of facies [5] probably due to the synsedimentary tectonics. All these reasons motivate the necessity to widen the understanding of the depositional environment of the phosphates that is a key step to retrieve more accurate information from geoelectrical data.

References

[1] Chudeau, R. (1916) Le Golfe éocène du Sénégal, Bulleting SGF. Paris (4) t. 16, 303-308.

[2] Baud, L. (1938) Rapport du 19 mai 1938 sur les calcaires phosphatés de la région de Matam et de Kanel. Rap. MAS, Arch. Dir Mines, Dakar.

[3] Flicoteaux, R., Lappartient, J., Parron, C. and Popoff, M. (1979) Les formations tertiaires du fleuve Sénégal. Influence des phénomènes diagénétiques et d'altération sur les caractères lithostratigraphiques. Comm. X' coll. géol. afr., Montpellier.

[4] Ollier, G. and Mahtis, B. (1991) Lithologic Interpretation from Geophysical Logs in Holes 737B, 738C and 742A. *Proceedings of the Ocean Drilling Program, Scientific Results*, **1**, 263-289.

[5] Pascal, M. (1986) Les minéralisations phosphatées de la rive droite du fleuve Sénégal. Itinéraires géologiques dans le Sud-Ouest mauritanien. Lab. Géol. Univ. Nice et Ecole norm. sup. Nouakchott, 25-43.

[6] Noel, J.B., Barusseau, J.P., Roger, J., Dabo, B., Diagne, E., Duvail, C., Sagna, R., Sarr, R. and Serrano, O. (2009) Carte géologique du Sénégal à 1/200 000, feuilles de Matam et Semme, Première édition mars 2009.

[7] Harper, C.W.J. (1984) Improved Methods of Facies Sequence Analysis, Facies Models: Response to Sea Level Change. Geological Association of Canada, Reprint Series 1, 11-13.

[8] Jang, J.S.R. (1997) Fuzzy Inference Systems. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River, 73-91.

[9] Ndiaye, M., Davaud, E. and Jorry, S. (2014) A Computer-Assisted Method for Depositional Model Determination. *International Journal of Geosciences*, **5**, 178-183. http://dx.doi.org/10.4236/ijg.2014.52019

[10] Roberts, J.N. and Schwartz, L.M. (1985) Grain Consolidation and Electrical Conductivity in Porous Media. *Physical Review B*, **31**, 5990. http://dx.doi.org/10.1103/PhysRevB.31.5990

[11] Niwas, S. and Singhal, D.C. (1981) Estimation of Aquifer Transmissivity from Dar-Zarrouk Parameters in Porous Media. *Journal of Hydrology*, **50**, 393-399. http://dx.doi.org/10.1016/0022-1694(81)90082-2

Annexes

The source code of the developed application

package model;

```
import java.text.NumberFormat;
public class Classification {
	private Class[]classes = null;
	private double[][]transitions = null;
	private Lithology[]lithologies = null;
	private double[][]geoelecProfile = null;
	private String otherFacies = "None";
	private int goodClasses = 0;
	public Classification(int nbLayers){
		lithologies = new Lithology[nbLayers];
		for (int i = 0; i < nbLayers; i++) {
			lithologies[i] = new Lithology("",0);
		}
	}
	public void classifyAll(double weightLG,double weightTM,double weightMO){
		int m = lithologies.length;
		for (int i = 0; i < m; i++) {
			lithologies[i] = classify(i,weightLG,weightTM,weightMO);
		}
	}
	private Lithology classify(int index, double weightLG,double weightTM,double weightMO){
		//reset scores
		int classificationStep = 0;
		int n = classes.length;
		for (int i = 0; i < n; i++) {
			classes[i].reset();
		}
		//new classification
		double resistivity = geoelecProfile[index][0];
		double thickness = geoelecProfile[index][1];
		int indexOfClass = -1;
		computeLGScore(resistivity,weightLG);
		indexOfClass = maxScoreIndex();
		classificationStep++;
		if(!stopClassification()){
			computeTMScore(transitions,weightTM,index);
			indexOfClass = maxScoreIndex();
			classificationStep++;
		}
		if(!stopClassification()){
			computeMOScore(weightMO);
			indexOfClass = maxScoreIndex();
			classificationStep++;
		}
		String name = "Unknown";
		Class cl = null;
		if(indexOfClass>=0){
			name = classes[indexOfClass].getName();
			cl = classes[indexOfClass];
		}
```

```
		Lithology litho = new Lithology(name,thickness);
		String log ="\u03C1 = "+resistivity;
		log+="\nh = "+thickness;
		log+="\nClassified as : "+name;
		if(classificationStep>1){
			log+="\nClassified in "+classificationStep+" steps.";
		}else{
			if(indexOfClass>-1){
				log+="\nClassified in "+classificationStep+" step.";
			}else{
				log+="\nUnclassified.\n";
			}
		}
		if(cl != null){
			NumberFormat format = NumberFormat.getInstance();
			format.setMaximumFractionDigits(2);
			log+="\nPreselected Facies : "+otherFacies;
			log+="\nScores :";
			log+="\n\u25CFGeoE="+format.format(cl.getScoreLG());
			log+="\n\u25CFTrsM="+format.format(cl.getScoreTM());
			log+="\n\u25CFMOcc="+format.format(cl.getScoreMO())+"\n";
		}
		litho.setLog(log);
		return litho;
	}
	private void computeLGScore(double resistivity, double weightLG){
		otherFacies = "None";
		int n = classes.length;
		goodClasses = 0;
		boolean[]inside = new boolean[n];
		//find the number of possible classes
		for (int i = 0; i < n; i++) {
			inside[i] = classes[i].contains(resistivity);
			if(inside[i]){
				goodClasses++;
			}
		}
		//if we have one class the score is 100%
		//otherwise the score is 100/n with n the number of classes
		if(goodClasses == 0){return;}else{otherFacies = "";}
		//Now set the score for each class
		for (int i = 0; i < n; i++) {
			if(inside[i]){
				Class cl = classes[i];
				double min = cl.getMin();
				double max = cl.getMax();
				double mid = (max-min)/2.0;
				double v = resistivity-min;
				double score = Math.abs(v-mid)/mid*weightLG;
				//double score = (max-resistivity)/(max-min)*100;
				//System.out.println("LG score : "+score);
				classes[i].setScoreLG(score);
				otherFacies+="\n\u25CF"+classes[i].getName();
			}
		}
```

```
}
private void computeTMScore(double[][]transitions,double weightTM, int index){
	String[]facies = getFacies();
	int n = facies.length;
	double totalTM = 0;
	double maxScore = maxScore();
	for (int i = 0; i < n; i++) {
		//only classes with a positive score
		double lg = classes[i].getScoreLG();
		double tm = classes[i].getScoreTM();
		double mo = classes[i].getScoreMO();
		double total = lg+tm+mo;
		if(total == maxScore){
			if(index>0){
				String currentFacies = classes[i].getName();
				String nextFacies = lithologies[index-1].getFacies();
				int nextIndex = getIndex(nextFacies, facies);
				int currentIndex = getIndex(currentFacies, facies);
				if(currentIndex>=0 && nextIndex >=0){
					double scoreTM = transitions[currentIndex][nextIndex];
					totalTM+=scoreTM;
					classes[i].setScoreTM(scoreTM);
				}
			}
		}
	}
	//compute percent
	for (int i = 0; i < n; i++) {
		double scoreTM = classes[i].getScoreTM();
		if(totalTM==0){
			scoreTM = 0;
		}else{
			scoreTM = scoreTM/totalTM*weightTM;
		}
		classes[i].setScoreTM(scoreTM);
	}
}
private void computeMOScore(double weightMO){
	int n = classes.length;
	double totalMO = 0;
	double maxScore = maxScore();
	for (int i = 0; i < n; i++) {
		//only classes with a positive score
		double lg = classes[i].getScoreLG();
		double tm = classes[i].getScoreTM();
		double mo = classes[i].getScoreMO();
		double total = lg+tm+mo;
		if(total == maxScore){
			double score = columnSum(i);
			classes[i].setScoreMO(score);
			totalMO+=score;
		}
	}
	//compute percent
	for (int i = 0; i < n; i++) {
```

```
                double scoreMO = classes[i].getScoreMO();
                if(totalMO==0){
                        scoreMO = 0;
                }else{
                        scoreMO = scoreMO/totalMO*weightMO;
                }
                classes[i].setScoreMO(scoreMO);
        }
}
private int maxScoreIndex(){
        double maxScore = maxScore();
        int n = classes.length;
        for (int i = 0; i < n; i++) {
                double lg = classes[i].getScoreLG();
                double tm = classes[i].getScoreTM();
                double mo = classes[i].getScoreMO();
                double total = lg+tm+mo;
                if(total==maxScore && total>0){
                        return i;
                }
        }
        return -1;
}
private boolean stopClassification(){
        int n = classes.length;
        int classified = 0;
        double maxScore = maxScore();
        int nbMaxScore = 0;
        for (int i = 0; i < n; i++) {
                double lg = classes[i].getScoreLG();
                double tm = classes[i].getScoreTM();
                double mo = classes[i].getScoreMO();
                double total = lg+tm+mo;
                if(total > 0){
                        classified++;
                }
                if(total == maxScore){
                        nbMaxScore++;
                }
        }
        //Classification stops when we are out of LG limits
        /*if(classified == 0){
                return true;
        }*/
        //Classification stops when there is one possible class
        if(goodClasses == 1){
                return true;
        }
        return false;
}
private String[]getFacies(){
        int n = classes.length;
        String[] facies = new String[n];
        for (int i = 0; i < n; i++) {
                facies[i] = classes[i].getName();
```

```
		}
		return facies;
	}
	private int getIndex(String name, String[]facies){
		int n =	facies.length;
		for (int i = 0; i < n; i++) {
			if(name.equalsIgnoreCase(facies[i])){
				return i;
			}
		}
		return -1;
	}
	private double maxScore(){
		int n = classes.length;
		double maxScore = 0;
		for (int i = 0; i < n; i++) {
			double lg = classes[i].getScoreLG();
			double tm = classes[i].getScoreTM();
			double mo = classes[i].getScoreMO();
			double total = lg+tm+mo;
			if(total>maxScore){
				maxScore = total;
			}
		}
		return maxScore;
	}
	public String toString(){
		String out = lithologies[0].getFacies()+": "+lithologies[0].getThickness();
		int m = lithologies.length;
		for (int i = 1; i < m; i++) {
			out+= "\n"+lithologies[i].getFacies()+": "+lithologies[i].getThickness();
		}
		return out;
	}
	public String log(){
		String out = lithologies[0].getLog();
		int m = lithologies.length;
		for (int i = 1; i < m; i++) {
			out+= "\n"+lithologies[i].getLog();
		}
		return out;
	}
	public Class[] getClasses() {
		return classes;
	}
	public void setClasses(Class[] classes) {
		this.classes = classes;
	}
	public double[][] getTransitions() {
		return transitions;
	}
	public void setTransitions(double[][] transitions) {
		this.transitions = transitions;
	}
	public double[][] getGeoelecProfile() {
```

```
		return geoelecProfile;
	}
	public void setGeoelecProfile(double[][] geoelecProfile) {
		this.geoelecProfile = geoelecProfile;
	}
	private double columnSum(int column){
		int n = transitions.length;
		double sum = 0;
		for (int i = 0; i < n; i++) {
			sum+=transitions[i][column];
		}
		return sum;
	}
}
```

```
package model;
public class Class {
    private double min = 0;
    private double max = 0;
    private String name = "";
    private double scoreLG = 0;
    private double scoreTM = 0;
    private double scoreMO = 0;

    public double getScoreTM() {
        return scoreTM;
    }
    public void setScoreTM(double scoreTM) {
        this.scoreTM = scoreTM;
    }
    public Class(double min, double max, String name){
        this.min = min;
        this.max = max;
        this.name = name;
    }
    public boolean contains(double resistivity){
        return resistivity>=min && resistivity <=max;
    }
    public double getMin() {
        return min;
    }
    public void setMin(double min) {
        this.min = min;
    }
    public double getMax() {
        return max;
    }
    public void setMax(double max) {
        this.max = max;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
```

```
    }
    public double getScoreLG() {
        return scoreLG;
    }
    public void setScoreLG(double scoreLG) {
        this.scoreLG = scoreLG;
    }
    public void reset(){
        scoreLG = 0;
        scoreTM = 0;
        scoreMO = 0;
    }
    public void ToString(){
        System.out.println(name+":  min="+min+",  max="+max+",  LG="+scoreLG+",  TM="+scoreTM+",
MO="+scoreMO);
    }
    public void setScoreMO(double scoreMO) {
        this.scoreMO = scoreMO;
    }
    public double getScoreMO() {
        return scoreMO;
    }
}
package model;
public class Lithology {
    private double thickness = 0;
    private String facies = "";
    private String log = "";
    public Lithology(String facies, double thickness){
        this.facies = facies;
        this.thickness = thickness;
    }
    public double getThickness() {
        return thickness;
    }
    public void setThickness(double thickness) {
        this.thickness = thickness;
    }
    public String getFacies() {
        return facies;
    }
    public void setFacies(String facies) {
        this.facies = facies;
    }
    public void ToString(){
        System.out.println(facies+": "+thickness+" "+log);
    }
    public String getLog() {
        return log;
    }
    public void setLog(String log) {
        this.log = log;
    }

}
```

```
package view;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
public class Input extends JPanel {
    private JLabel label;
    private JTextArea text;
    public Input(String indication){
            super(new BorderLayout());
            label = new JLabel(indication);
            label.setForeground(Color.BLUE);
            label.setPreferredSize(new Dimension(200,32));
            label.setFont(new Font("Arial", Font.PLAIN, 12));
    text = new JTextArea();
            text.setBackground(Color.BLACK);
            text.setForeground(Color.WHITE);
            text.setFont(new Font("Arial", Font.BOLD, 12));
            text.setCaretColor(Color.ORANGE);
            text.setSelectionColor(Color.RED);
            JScrollPane scroll = new JScrollPane (text);
    scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
    scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
            add(label, "North");
            add(scroll,"Center");
    }
    public Input(){
            super(new BorderLayout());
    text = new JTextArea();
            text.setBackground(Color.BLACK);
            text.setForeground(Color.WHITE);
            text.setFont(new Font("Arial", Font.BOLD, 12));
            text.setCaretColor(Color.ORANGE);
            text.setSelectionColor(Color.RED);
            text.setEditable(false);
            JScrollPane scroll = new JScrollPane (text);
    scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
    scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
            add(scroll,"Center");
    }

    public void setText(String input){
            text.setText(input);
    }

    public String getText(){
            return text.getText();
    }
}
```

```
package view;
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JToolBar;
import javax.swing.WindowConstants;
import model.Classification;
import model.Data;
import model.Class;
public class ViewClassification extends JFrame implements ActionListener{
    private Input classeView;
    private Input matrixView;
    private Input sevView;
    private Input interpretation;
    private JButton weight;
    private JButton ok;
    private JButton log;
    private String line = "\n";//System.getProperty("line.separator");
    private Classification classification;
    private double weightLG = 100;
    private double weightTM = 100;
    private double weightMO = 100;
    public ViewClassification(){
        super(" Geoelectric Fuzzy Logic Interpretation ");
        setSize(550,400);
        Container pane = getContentPane();
        pane.setLayout(new GridLayout(2,2,5,5));
        classeView = new Input("<html>1. N Classes: [Lithology, \u03C1<sub>min</sub>,
\u03C1<sub>max</sub>]</html>");
        classeView.setText(Data.CLASSES);
        matrixView = new Input("2. Transition matrix [NxN]");
        matrixView.setText(Data.TRANSITIONS);
        sevView = new Input("3. Inverted VES [\u03C1,h]");
        sevView.setText(Data.SEV);
        JPanel panel = new JPanel(new BorderLayout());
        JToolBar toolbar = new JToolBar();
        ok = new JButton("Interpretation");
        ok.addActionListener(this);
        toolbar.add(ok);
        weight = new JButton("Weights");
        weight.addActionListener(this);
        toolbar.add(weight);
        log = new JButton("Show log");
        log.addActionListener(this);
        toolbar.add(log);
        interpretation = new Input();
        panel.add(toolbar,"North");
        panel.add(interpretation,"Center");
        pane.add(classeView);
```

```
        pane.add(matrixView);
        pane.add(sevView);
        pane.add(panel);
        setContentPane(pane);
        setVisible(true);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
public static void main(String[] args) {
        // TODO Auto-generated method stub
        new ViewClassification();
}
@Override
public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();
        if(source == ok){
                Class[]classes = getClasses();
                if(classes == null){
                        String message = "There is an error in the Classes input.";
                        message+="\nClasses must be in the form: Clay, min, max";
                        JOptionPane.showMessageDialog(this, message);
                        return;
                }
                double[][]transitions = getTransitions();
                if(transitions == null){
                        String message = "There is an error in the Transition matrix.";
                        message+="\nMatrix must be in the form N x N (with N classes)";
                        JOptionPane.showMessageDialog(this, message);
                        return;
                }
                int n = transitions.length;
                if(classes.length!=n||transitions[0].length!=n){
                        String message = "There is an error in the Transition matrix.";
                        message+="\nMatrix must be in the form N x N (with N classes)";
                        JOptionPane.showMessageDialog(this, message);
                        return;
                }
                double[][] geoelecProfile = getGeoelecProfile();
                if(geoelecProfile == null){
                        String message = "There is an error in the geoelectric profile.";
                        message+="\nThe inerted VES must be in the form: resistivity,thickness";
                        JOptionPane.showMessageDialog(this, message);
                        return;
                }
                classification = new Classification(geoelecProfile.length);
                classification.setClasses(classes);
                classification.setTransitions(transitions);
                classification.setGeoelecProfile(geoelecProfile);
                classification.classifyAll(weightLG,weightTM,weightMO);
                interpretation.setText(classification.toString());
        }else if(source == log){
                if(classification == null){
                        String message = "There is no log file to show.";
                        JOptionPane.showMessageDialog(this, message);
                        return;
                }
```

```
            interpretation.setText(classification.log());
        }else if(source == weight){
            String message = "Enter the GeoE, TrsM and MOcc weights (in percent).";
            message+="\ne.g. 50,25,25";
            String option = JOptionPane.showInputDialog(this, message);
            try {
                String[]weights = option.split(",");
                weightLG = Double.parseDouble(weights[0]);
                weightTM = Double.parseDouble(weights[1]);
                weightMO = Double.parseDouble(weights[2]);
            } catch (Exception e1) {
                // TODO Auto-generated catch block
                message = "Your entered wrong values.";
                message+="\nWeights must be in the form V1,V2,V3";
                JOptionPane.showMessageDialog(this, message);
            }
        }

    }
    public Class[]getClasses(){
        Class[]classes = null;
        try {
            String text = classeView.getText();
            String[]lines = text.split(line);
            int n = lines.length;
            if(n>1)
            classes = new Class[n];
            for (int i = 0; i < n; i++) {
                String[] e = lines[i].split(",");
                double min = Double.parseDouble(e[1]);
                double max = Double.parseDouble(e[2]);
                classes[i]= new Class(min, max, e[0]);
            }
        } catch (Exception e) {
            // TODO Auto-generated catch block
            classes = null;
        }
        return classes;
    }
    public double[][]getTransitions(){
        double[][]transitions = null;
        try{
            String text = matrixView.getText();
            String[]lines = text.split(line);
            int n = lines.length;
            if(n>1)
            transitions = new double[n][n];
            for (int i = 0; i < n; i++) {
                String[]e = lines[i].split(",");
                for (int j = 0; j < n; j++) {
                    transitions[i][j]=Double.parseDouble(e[j]);
                }
            }
        }catch (Exception e){
            transitions = null;
```

```
            }
            return transitions;
        }
        public double[][]getGeoelecProfile(){
            double[][]geoelecProfile = null;
            try{
                String text = sevView.getText();
                String[]lines = text.split(line);
                int n = lines.length;
                geoelecProfile = new double[n][2];
                for (int i = 0; i < n; i++) {
                    String[]e = lines[i].split(",");
                    geoelecProfile[i][0]= Double.parseDouble(e[0]);
                    geoelecProfile[i][1]= Double.parseDouble(e[1]);
                }
            }catch (Exception e){
                geoelecProfile = null;
            }
            return geoelecProfile;
        }
    }
```

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.