www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 14(3), Sep-Dec 2023, 24-30. ISSN: 2007-1558. Doi: 10.61467/2007.1558.2023.v14i3.384

Prediction of the Melting Point of Ionic Liquids with Clustering and Noeuroevolution

Juan Frausto-Solís[1], Juan Javier González-Barbosa[1], Jorge Alberto Cerecedo-Cordoba[1], Juan Paulo Sánchez-Hernández[2], Ocotlán Díaz-Parra[3], Guadalupe Castilla-Valdez[1]*

[1] Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero, Cuidad Madero, Tamaulipas, México
[2] Universidad Politécnica del Estado de Morelos, Jiutepec, Morelos, México.
[3] Universidad Politécnica de Pachuca, Pachuca, México.
*Correspondence: jjgonzalezbarbosa@hotmail.com

Abstract. Ionic liquids (ILs) are salts with a wide liquid temperature range and low melting points and can be fine-tuned to have specific physicochemical properties by the selection of their anion and cation. However, having a physical synthesis of multiple ILs for testing purposes can be expensive. For this reason, an in-silico estimation of physicochemical properties is desired. The selection of these components is limited by the low precision offered by state-of-the-art predictive models. In this paper, we explore the prediction of melting points with clustering algorithms and a novel Neuroevolution approach. We focused our design on simplicity. We concluded that performing clustering analysis in a previous phase of the model generation improves the estimation accuracy of the melting point which is validated in experimentation made in-silico.
Keywords: Ionic Liquids, Clustering analysis, Neuroevolution, Neural Networks, Machine Learning.

Article Info
Received Aug 17, 2023
Accepted Sep 23, 2023

1 Introduction

1.1. Ionic Liquids

Ionic liquids (ILs) are ion mixtures formed by cations and anions in a homogeneous substance. ILs are salts, but these compounds have the particularity that they are liquids at room temperature. In contrast, Sodium Chloride has a melting point of 801°C, which makes it a solid at room temperature. The temperature ranges between ionic liquids and molten salts are sufficiently broad to identify them with their classification.

These salts have been used successfully in a wide range of applications because they have a low melting point and maintain their liquid state in a wide range of temperatures (Donato, Matějka, Mauler, & Donato, 2017) (Santos-López et al, 2017) (Plechkova & Seddon, 2008). The ILs excel when they are used as solvents. The physicochemical properties of these liquids are dependent on the selection of the elements in their composition (i.e., the combination of anion and cation). There is the possibility of designing ILs with specific purposes by making a careful selection of cations and anions. However, this is a complex task because:

- The synthesis of multiple ILs is expensive, especially if they are designed by trial and error.
- Synthetizing multiple ILs becomes a tedious and timely task; if this process is scaled to thousands of combinations of ILs it becomes impractical and sometimes intractable.
- From 10^{12} binary combinations of possible ILs, only a small amount of them can form ILs (Donato, Matějka, Mauler, & Donato, 2017), (Santos-López et al, 2017).

The last issues make it impossible or at least very hard to perform an extensive study of ILs in the traditional manner of synthesis and characterization.

1.2 QSAR and Predictive Models

Since the "traditional" analysis of ILs can be costly, computational estimations are a promising alternative. In particular, the use of QSAR methods.

The Quantitative Structure-Activity Relationships (QSAR) are mathematical models that represent the correlation between independent and dependent variables, specifically the relationships formed between the characteristics derived from compounds of the ILs concerning their biological activity, physical-chemical properties, or their toxicity (Plechkova & Seddon, 2008), (Stanley & Miikkulainen, 2002). These models require the calculation of molecular descriptors. Molecular descriptors are measurements generated from the topology of molecules and their structure. There are three types of molecular descriptors:

- OD-descriptor, descriptors derived from the molecular formula.
- 2D-descriptor, derivative-two of the topological representation of the molecule.
- 3D-descriptor, descriptors derived from the geometric representation.

There is software available in the literature to calculate various descriptors of the three types described. Some examples of this are CODESSA (Katagiri, Hirasawa, Jinglu, & Murata, 2002) and DRAGON (Storn & Price, 1997).

QSAR methods can be used for both classification and regression problems. Classification problems are those where the task is to classify data in one or more dependent sets (or classes in machine learning). On the other hand, regression problems are those in which the system response (dependent variable) is a numerical value.

The QSAR methods use molecular descriptors and with some regression methods (usually linear regression), generate the estimations of the properties of the molecules. Regression methods are predictive modeling methods that seek the inherent relationships between one or several input entities with different outputs. That is, it searches for an unknown function that transforms the independent variables (input) into dependent variables (output). Regression analysis is commonly used in areas such as medicine, finance, chemistry, mechanics, and others; it is an indispensable tool for analyzing and modeling information with unknown behavior.

Statisticians have developed increasingly advanced methods of regression over the years. Linear regression is still a good choice when a very simple model for a basic prediction task is required. Linear regression also tends to work well in sparse, non-complex data sets. Nevertheless, it has difficulties predicting nonlinear behaviors. Machine learning methods have been used as alternatives to predict nonlinear relationships. Machine Learning is a branch of computer science that studies the development of methods that allow computer programs to include pseudo-learning. Learning is the acquisition of knowledge through study or experience. In the specific case of Machine Learning, the experience usually refers to past information or records available.

The Machine Learning methods have different learning schemes; they determine how the algorithm should learn from a series of input data and how to manipulate them. The selection of the learning method is based on the nature of the data and the goal to achieve with the learning task. The nature of the input data plays an essential role in the algorithm training. The dataset may contain irrelevant or noisy information. Therefore, identifying the omitted and irrelevant variables is a critical issue.

Currently, Artificial Neural Networks (ANNs) are present in many systems thanks to their learning ability, self-organization, and handling of diffuse, noisy, and inconsistent data. Moreover, ANNs, perform efficiently in a wide variety of applications. ANNs excel in the discovery of patterns between inputs and outputs.

1.3 Neuroevolution

Even when ANNs are good at learning patterns from a dataset the topology of the neural network must be fine-tuned by the researcher to obtain quality results. The tuning process is usually done by a trial and error experiment, which is performed based on the experience of the researcher. Nevertheless, there is no good indicator when a good topology is achieved. A good alternative for ANNs is Neuroevolution, which consists of training ANNs through evolutionary approaches. In other words, an evolutionary algorithm modifies the weights and topology of an ANN. These types of networks are also known as TWEANNs (Topology & Weight Evolving Artificial Neural Network algorithms). Historically, TWEANNs have improved their training techniques through the incorporation of new components; examples of such components are minimal topologies and Speciation (Stanley & Miikkulainen, 2002). TWEANNs are commonly used for training with reinforcement learning to accomplish several tasks, particularly for evolution neural networks. In this paper, we incorporate a straightforward implementation of neuroevolutionary neural networks, which consist of two main parts described in the paper: the solutions and the optimizer.

1.3.1. Free-Form Neural Networks

In our implementation, a graph G of neurons is created. Initially, G contains a set of inputs I and an outputs set O. The I set represents all the independent variables of the model. The algorithm will try to find an optimal configuration of G, where I pass information to O in a direct or indirect connection. A direct connection is whenever an input is connected with no intermediaries with O. On the contrary, an indirect connection is whenever I connect to O through internal neurons. As our goal is to let the algorithm create an architecture that yields better results, the connections can be freely made between neurons. As a consequence, ANNs can rapidly deviate to inconsistencies where loops are created, and without a proper way to manage them can lead to unfeasible neural networks. To avoid this issue, we assign an identification number to each neuron. The neurons in the I set, obtain the first available IDs, and the output neurons O obtain the maximum ID values. Additionally, we added a simple rule; each neuron can only connect its output to a neuron with a bigger id. In that way, we guarantee that no internal loops are generated. An example of the free-form neural network is shown in Figure 1.

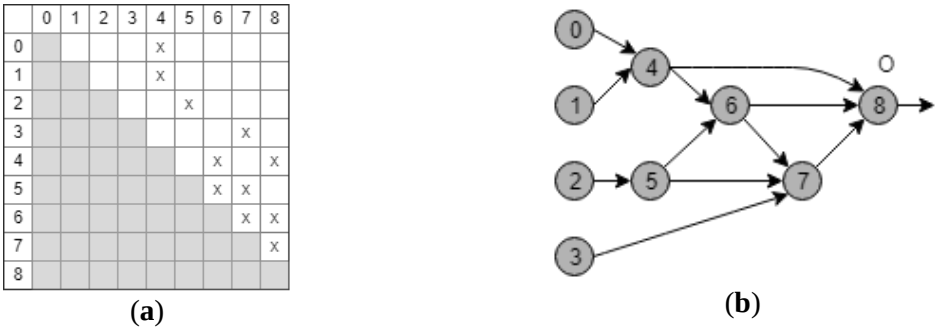

Figure 1. Proposed Architecture: (**a**) The Genotype of the Neural Network. (**b**) The Phenotype.

1.3.2. Evolutionary Training

In this work, a genetic algorithm was used to perform the training and topology search of the neural networks. Like the traditional manner, our algorithm has a selection, crossover, and mutation phase. At first, the algorithm starts with a set of simple free-form neural networks (only a few connections each) which is known as population. In the genetic algorithm, the population is recombined multiple times with an optimization goal of minimizing a prediction error. The algorithm stops iterating when a certain number of evaluations is reached. Since our phenotype is a special case of a neural network, we needed to define new crossovers and mutations. In our method, we used the conventional GNP Crossover (Katagiri, Hirasawa, Jinglu, & Murata, 2002) and a crossover based on the differential crossover (Storn & Price, 1997). Moreover, a set of mutation algorithms was designed:

- AddNeuron: adds a new neuron to the topology and connects it to existing neurons.
- RemoveNeuron: selects a random neuron and connects the incoming connections to the outputs, then removes the neuron.
- AddWeight: adds a random connection between neurons with a random weight.
- RemoveWeigth: selects a random weight from the topology, if the elimination of the connection converts the network to unfeasible, that means that the removal operation fails.
- WeigthMutate: change the weight of a random connection.

These mutations are designed to alter existing neural networks so that no unfeasible networks are created. In Figure 2, an example of four of the five mutations is shown.

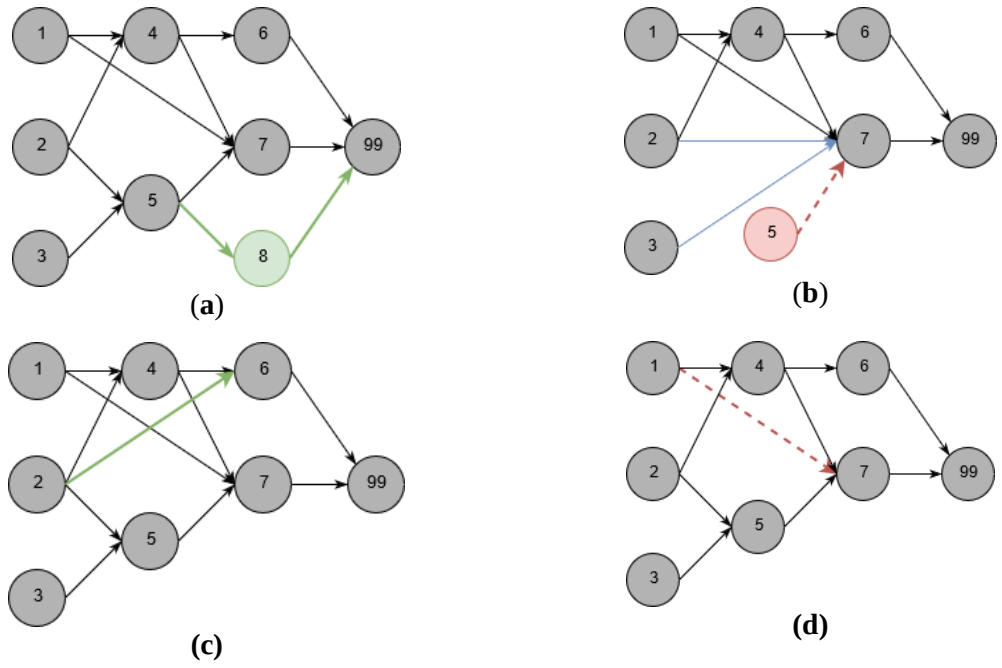

Figure 2. Proposed mutations
(**a**) AddNeuron: adds a new neuron to the network;
(**b**) RemoveNeuron: deletes an existing neuron by redirecting the connections to its output.
(**c**) AddWeight: creates a random link between a pair of neurons previously unconnected.
(**d**) RemoveWeigth deletes an existing connection.

1.3.3. Clustering of Ionic Liquids

The analyzed literature shows that further work needs to be carried out to predict the melting point of ILs. For that reason, we analyzed and separated the ILs into subsets. The predictive models need to learn just from a section of the dataset at a time. A divide-and-conquer strategy was proposed in 2007 that generates subsets of ILs (Varnek, Kireeva, Tetko, Baskin , & Solov'ev, 2007). This idea raises the problem of determining the best subsets. A possible solution is the generation of subgroups with the opinion of an expert. Another idea is the creation of subsets by using clustering algorithms.

The k-Means algorithm is widely accepted as a reliable clustering algorithm (Pérez-Ortega, Romero, & Almanza-Ortega, 2018); k-Means separates the full dataset into k-groups named clusters (Pérez-Ortega et al, 2022). The algorithm works as follows. At first, the algorithm chooses k random centroids; a centroid is a point in space. The main loop consists of two phases. In the first phase, all the points are assigned to the nearest centroid to form k clusters. Then, each cluster is averaged to obtain updated centroids. If the position of centroids changes from the previous iteration the algorithm continues otherwise the algorithm halts.

In Cerecedo et al. 2019 (Cerecedo-Cordoba, González-Barbosa, Frausto-Solis J., & Gallardo-Rivas, 2019), we explored the usage of clustering algorithms to separate ILs into subgroups of similar characteristics. Here we prove that this is an effective solution for problems with data of a complex nature, at least for the data sets tested.

In our proposal, the predictive models were trained just in the corresponding subsets of ILs as is shown in Figure 3.

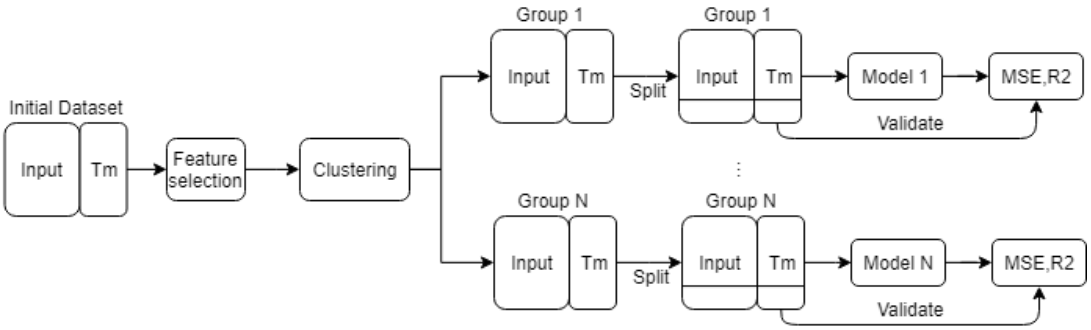

Figure 3. Clustering and prediction of ILs.

2 Materials and Methods

The ILs used in this study are from (Huo, Xia, Zhang, & Ma, 2009), the ILs were prepared in Molfiles with the JCHEM software (ChemAxon, 2018). We only use the liquids based on the Imidazole molecule. The prepared files in this paper contain the 2D structures of the cations and anions used.

Molecular descriptors are calculated values derived from the topology of the ILs; we obtained such data from a set of well-defined algorithms in the Padel software (Yap, 2011), which performs the molecular descriptor calculations of the ions. The calculations were performed for each anion and cation in the dataset, and the resulting molecular descriptors were merged to obtain a whole dataset.

The Padel software generates a large group of variables; in consequence, a feature selection is needed. In a preliminary study, we found that regression trees can filter and select a smaller set of suitable quality variables. We performed a feature selection with a regression tree algorithm selecting as valuable variables the top branches of the tree.

The final dataset is composed of 134 ILs with 30 molecular descriptors selected. To guarantee the robustness of the model we made a random split of the data with 80% of the data for training and 20% for testing. Also, we perform a repeated double cross-validation (Baumann & Baumann, 2014). This process comprises two nested cross-validation loops; this process has been shown to have a good performance in estimating an unbiased prediction of true error (Varma & Simon, 2006). In the outer loop, a split of the data is made, one portion to be used by the inner loop and the other to create an external validation of the model. The inner loop performs a cross-validation with the reserved data. This process is repeated multiple times. In our case is repeated 100 times. All the results are averaged.

3 Results

A k-Means clustering algorithm was used where K is equal to four. Thirty program executions for this experiment were performed. The models used in this experiment are regression trees, SVRs, and neuroevolution.

The results obtained are shown in Table 1 which were measured with Mean Average Error (MAE) show in Equation (1), and Mean Absolute Percentage Error (MAPE) presented in Equation (2).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (1)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (2)$$

where y_i is the actual melting point and $\hat{y}_i$ is the predicted melting point.

Table 1. Experimental results of the proposed method.

Trial	MAE Training	MAPE training	MAE testing	MAPE testing
1	15.35	4.28	21.85	6.26
2	16.59	4.78	25.63	7.94
3	12.48	3.61	35.44	9.50
4	12.53	3.69	30.06	8.63
5	15.75	4.81	14.81	4.64
6	19.89	5.79	25.88	8.37
7	10.04	3.06	20.99	6.57
8	13.36	3.96	20.39	6.60
9	13.05	3.86	32.69	9.84
10	11.58	3.26	20.96	5.80
11	12.22	3.39	20.61	6.18
12	20.00	5.83	20.73	6.38
13	7.67	2.21	25.55	7.70
14	16.92	4.95	13.95	4.05
15	15.54	4.60	27.38	8.40
16	19.49	5.75	23.20	7.34
17	9.97	3.10	17.18	5.04
18	10.02	3.14	22.22	6.64
19	14.87	4.67	30.07	8.07
20	14.65	4.38	18.34	6.15
21	13.71	4.05	19.92	5.18
22	10.76	3.29	12.14	3.82
23	13.03	3.98	18.45	5.30
24	8.45	2.71	39.87	13.02
25	11.27	3.32	34.37	9.89
26	21.05	6.26	30.03	8.48
27	18.03	5.26	24.13	7.13
28	16.06	4.64	21.33	6.77
29	13.02	4.02	14.16	4.32
30	17.54	5.16	23.56	6.65
average	14.16	4.19	23.53	7.02

4 Discussion

In this work, we studied the prediction of melting temperatures of ILs with clustering and a neuroevolutionary algorithm as an effective model for the possible analysis and design of ILs. Part of the study presented is based on the separation of ILs into possible groupings or clusters, which allows us to obtain more robust models compared to an algorithm without grouping. Also, we studied the use of a heterogeneous combination of algorithms to predict the different clusters defined by our research group. In our study, the best algorithms for each cluster were automatically selected by the algorithm, and we found that 78% of the time the neuroevolution obtained better results than the other methods. Additionally, the double cross-validation results indicate that the algorithm performs close to our first approach with only a difference of 1.05%. Even though this is a significant accomplishment, a disadvantage of this method is the complexity of the created neural networks.

The neuroevolutionary network presented in this paper for ILS had a very good performance when SVRs and Regression Trees were used. This strategy would transform the neuroevolution from a simple predictive model to a hybrid between bagging and boosting ensemble methods. This idea is planned to be explored in future work.

References

Baumann, D., & Baumann, K. (2014). Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation. *J. Cheminform, 6*, 1-19.

Cerecedo-Cordoba, J., González-Barbosa, J., Frausto-Solis J., & Gallardo-Rivas, N. (2019). Melting Temperature Estimation of Imidazole Ionic Liquids with Clustering Methods. *J. Chem. Inf. Model, 59*(7), 3144-3153.

ChemAxon. (2018). *MarvinSketch*. Obteined from https://chemaxon.com/marvin

Donato, K., Matějka, L., Mauler, R., & Donato, R. (2017). Recent Applications of Ionic Liquids in the Sol-Gel Process for Polymer-Silica Nanocomposites with Ionic Interfaces. *Colloids and Interfaces 1,5*.

Huo, Y., Xia, S., Zhang, Y., & Ma, P. (2009). Group Contribution Method for Prediting Melting Points of Imidazolium and Benzimidazolium Ionic Liquids. *Ind. Eng. Chem. Res., 48*, 2212–2217.

Katagiri, H., Hirasawa, K., Jinglu, H., & Murata, J. (2002). Comparing some graph crossover in genetic network programming. *IEEE, 2*, 1263-1268.

Pérez-Ortega, J., Roblero-Aguilar, S., Almanza-Ortega, N., Frausto-Solis, J., Zavala-Díaz, J., Hernández, Y., & Landero-Nájera, V. (2022). Hybrid Fuzzy C-Means Clustering Algorithm Oriented to Big Data Realms. *Axioms, 11*(8), 377.

Pérez-Ortega, J., Romero, D., & Almanza-Ortega, N. (2018). Balancing effort and benefit of K-means clustering algorithms in Big Data realms. *PLoS ONE, 13*(9).

Plechkova, N., & Seddon, K. (2008). Applications of ionic liquids in the chemical industry. *Chem. Soc. Rev.*, 37, 123-150.

Santos-López, G., Argüelles-Monal, W., Carvajal-Millan, E., López-Franco, Y., Recillas-Mota, M., & Lizardi-Mendoza, J. (2017). Aerogel from Chitosan solutions in ionic liquids. *Polymers (Basel)*, 9, 1-13.

Stanley, K., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.*,10, 99-127.

Storn, R., & Price, K. (1997). Differential Evolution - A simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.*, 11, 341-359.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*, 1-8.

Varnek, A., Tetko, I., Baskin , I., Solov'ev, V., & Kireeva, N. (2007). Exhaustive QSPR studies of a large diverse set of ionic liquids: how accurately can we predict melting points? *J. Chem. Inf. Model*, 47, 1111-1122.

Yap, C. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *J. Compt. Chem., 32*, 1466-1474.