Fluid Phase Equilibria 564 (2023) 113597

Contents lists available at ScienceDirect

Fluid Phase Equilibria

journal homepage: www.journals.elsevier.com/fluid-phase-equilibria

Thermodynamically consistent vapor-liquid equilibrium modelling with artificial neural networks

Andres Carranza-Abaid [*], Hallvard F. Svendsen, Jana P. Jakobsen

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), NO-7491, Trondheim, Norway

ARTICLE INFO

Keywords:
Physics-informed machine learning
Hybrid modelling
Activity coefficients
Excess enthalpy
NRTL

ABSTRACT

An integration of Artificial Neural Networks (ANNs) and thermodynamics through the application of Neural Network Programming (NNP) is proposed. Thermodynamic consistency is achieved because the thermodynamic relationships and constraints are transcribed into a specially crafted ANN. Moreover, the developed models allow predicting and extrapolating the model outside the experimental data boundaries.

The Wilson and NRTL models are used as case studies. Modifications to these models based on sigmoid functions are rigorously assessed in order to perform the simultaneous modelling of VLE and excess enthalpy. The automatic differentiation together with the ANN optimization algorithms can find sets of parameters that are better than the ones obtained with traditional gradient-based optimizers.

The frequently disregarded concepts of thermodynamic modelling with ANNs are discussed in-depth. A mathematical analysis of the impossibility of typical fully connected ANNs to formulate thermodynamically consistent equilibrium models is discussed and their use is discouraged (e.g., VLE, LLE, or adsorption.).

1. Introduction

Research on machine learning (ML) and artificial neural networks (ANNs) has been a trend for some years now mainly because of their outstanding predictive features or perhaps because of the excessive media coverage. Although the development of self-driving artificial intelligence (AI) and the creation of complex decision-making AIs [1,2] are quite impressive feats, it is not necessarily an indication that utilizing data-driven-only models is always appropriate. One example of this is in phase equilibria modelling, where it has been common to substitute established thermodynamic models with ANNs. Although ANN models may have better accuracy, it is not necessarily an indication that they are better thermodynamic models than those given by semi-empirical models.

It is a fact that both traditional VLE models and ANNs are approximations of the behavior of matter and energy in equilibrium conditions. However, it should not be forgotten that concepts like partial pressures are measured because of the thermodynamic framework. Therefore, reducing the thermodynamics models to "correlations that fit the thermodynamic data best" understates the essence of equilibrium thermodynamics. The quintessential conclusion that ML models are better than semi-empirical models is "the proposed ML modelling approach is superior to the semi-empirical thermodynamic models because it represents the experimental data more accurately". It is unfair to compare ML and semi-empirical models in this fashion since the former are unconstrained and, therefore, have more degrees of freedom to fit the data. Constraints and thermodynamic consistency are rarely discussed in the ML literature and are often overlooked. For example, in a recent review of ML-based thermodynamic modelling for ionic liquids [3], the thermodynamic consistency is not even mentioned.

If ANNs and machine learning in general are to be recognized as an actual alternative for thermodynamics modelling, the models should be developed within the established thermodynamics laws. The thermodynamic framework includes mathematical relationships between thermodynamic properties that otherwise may seem like unrelated concepts (e.g., the relationship between vapor pressure and enthalpy of vaporization with the Clausius-Clapeyron equation).

The objective of this work is to address the question: "can ANNs be utilized to formulate thermodynamically consistent VLE models?" To address this question, section 2 presents a discussion, with emphasis on thermodynamic consistency, about ML-based approaches that can potentially be used for VLE modelling. Section 3 shows that VLE equations can be transcribed into Algorithmically Structured Neural Networks (ASNNs) by applying the Neural Network Programming (NNP)

[*] Corresponding author.
E-mail address: andres.c.abaid@ntnu.no (A. Carranza-Abaid).

https://doi.org/10.1016/j.fluid.2022.113597
Received 25 May 2022; Received in revised form 26 August 2022; Accepted 2 September 2022
Available online 3 September 2022

method [4]. In order to provide an example that thermodynamics researchers can relate to, the NNP method was applied to transcribe the Wilson and NRTL activity coefficient models as well as some ANN-based modifications. Section 4 presents the application of the ASNNs for modelling VLE and simultaneous VLE and excess enthalpy. We show that machine learning models and thermodynamics consistency are not mutually exclusive concepts.

This work does not present a machine learning method that predicts the binary interaction parameters of unmeasured mixtures. Rather it aims at highlighting the importance of considering thermodynamic relationships instead of only focusing on accuracy. This work will help other researchers towards a more thermodynamics-oriented applications of ML algorithms. Since one of our goals is to make the neural network technology accessible for researchers new in ML, the Matlab codes containing the ASNN-based implementation of the Wilson and NRTL models are presented in the supporting information or can be accessed in github.com/AndresCA91/Non-reactive-VLE-ASNNs.

2. VLE modelling approaches

Three different approaches for VLE modelling have been identified, namely, mechanistic models, data-driven models, and hybrid models. This work restricts the discussion to systems without electrolytes.

Models whose equations are based on first principles are known are mechanistic. These models are of semi-empirical nature because they utilize fitting parameters that allow a better description of VLE. Some examples of mechanistic VLE models are the Equations of State (EoS) based on the van der Waals EoS [5] (e.g., Soave-Redlich-Kwong (SRK) [6] and Peng-Robinson (PR) [7]) or the EoS based on associating theories (e.g., Cubic Plus Association (CPA) [8], and the perturbed chain SAFT (PC-SAFT) [9]). Alternatively, there are models based on excess functions that utilize the concept of ideal solution as a reference state. The first instances of models based on the excess molar Gibbs energy (G^E) were done by Margules in 1895 [10,11], and by van Laar in 1910 [12]. However, the local composition models such as the Wilson model [13], non-random two-liquid model (NRTL) [14], and the universal quasi-chemical theory (UNIQUAC) [15] dramatically improved the accuracy and applicability to more complex problems like liquid-liquid equilibria (LLE). We recommend the available literature [16,17] for a more in-depth review of the theory and applicability of mechanistic models.

2.1. Data-driven models

Machine Learning (ML) is a set of algorithms and tools that find accurate correlations between the input data and can perform prediction and classification tasks. These data-driven algorithms are classified in supervised learning, unsupervised learning, and reinforcement learning. A supervised learning algorithm is more convenient for thermodynamics because the experimental datasets are labeled, and the input / output relationships are known through the thermodynamic framework. From all the supervised learning methods, we consider that Artificial Neural Networks (ANNs) are the most robust ML-based modelling method for VLE. This is due to their universal approximator feature, autodifferentiability and, with the proper selection of transfer functions, continuous. Because of this, this work focuses on ANNs.

ANNs have been utilized in the literature as black boxes to estimate the equilibrium behavior of several thermodynamic systems. The blackbox denomination is because the model is not transparent and implies that it is only based on the data and not on *a priori* knowledge [24] (on the contrary, mechanistic models are known as white-boxes). The first formal attempt in the literature at modelling VLE with ANNs was done by Petersen et al. in 1993 [25]. Back then, they concluded that the UNIFAC model estimations were better than the ANN because the mechanistic model had more knowledge embedded into it. This conclusion is in contradiction with several research contributions that argue that the ANNs provided better accuracy than semi-empirical or empirical models (e.g., [26–30]). It must be remarked that Petersen et al., 1993 modelled the activity coefficients while the other studies modelled the component fugacities; which may explain the difference between the conclusions. Other researchers have also discussed the possibility of utilizing ANNs for VLE modelling (e.g., [26,27,36–40, 28–35]). However, these studies are focused towards analyzing and validating the models from a consequential-statistical perspective rather than from a thermodynamic standpoint. Few attempts have considered thermodynamics while using ANNs for VLE modelling. The first of them was done by Guimaraes and McGreavy [41], who highlighted that understanding the information flow in the ANN is paramount in order to properly utilize them for the description of VLE. Sadly, no other research efforts in this direction have been taken since.

In our previous work [45] we developed a surrogate neural-network-based model for mixtures containing CO_2 and a liquid solvent. We emphasized the importance of considering the Gibbs phase rule in order to be able to replicate the underlying thermodynamic relationships (e.g., relationship between the heat of absorption and partial pressure). It was reported that even if the molar fraction of CO_2 is set to 0, the model estimated a positive value the fugacity of CO_2. This suggests an important structural flaw of the fully connected neural networks (shallow or deep) for VLE modelling.

2.1.1. Fundamentals of ANNs

ANNs are mathematical models that were designed to represent the cognitive processes in biological brains [18]. ANNs are constituted by three types of objects known as layers: input layers, hidden layers, and output layers. Input layers transfer the information fed by the user to the neural network, hidden layers transform the input signal to an output signal, and the output neurons gives the predictions back to the user. Hidden layers are constituted by an arbitrary number of hidden neurons. Each hidden neuron performs a linear or product combination of the inputs and then usually applies a non-linear transformation. Considering this, the number of hidden neurons in a hidden layer can be seen as the number of equations that are evaluated in parallel.

ANNs are well known of being universal approximators [19–22], however, in order to be considered as such, they must have at least have two hidden layers. A neural network with 2 hidden layers in series is known as a shallow neural network (SNN). The first layer usually has a non-linear transfer function (e.g., hyperbolic tangent transfer function) and the last layer linearly combines the results of the preceding layer. The linking of two hidden layer nodes in series is known as MultiLayer Perceptron (MLP).

Fig. 1 shows two graphical representations of the same SNN that models the VLE of a binary mixture. Note that there are two degrees of freedom according to the Gibbs phase rule. The fugacity of component 1 in the vapor phase (f_1) is computed as a function of the liquid molar fraction of component 1 (x_1) and temperature (T). The typical representation shown in Fig. 1a shows that the nodes of the input and output layers are connected to the second layer. However, it provides little information about the neural network. Because of this, this work utilizes the graphical representation of Fig. 1b.

This work represents neural network layers as in Fig. 1b: blue circles are the input layers, red hexagons are the output layers, and squares are the hidden layers. Each input and output layer represent an input or output vector, respectively (e.g., the input vector in Fig. 1 b) is $[x_i, T]^T$). The name of the hidden layers is the text in bold and, unless stated otherwise, the hidden layer output variable name is the same as the layer name. For each hidden layer there can be up to two subscripts and two superscripts. The first superscript provides information if the input layers are combined through a linear combination ($+$) or an element-by-element product ($\odot$). The second superscript tells whether the layer has a bias β or not (no second superscript). The first subscript indicates the number of hidden neurons, and the second subscript is the transfer function utilized. In this work we present the application of the

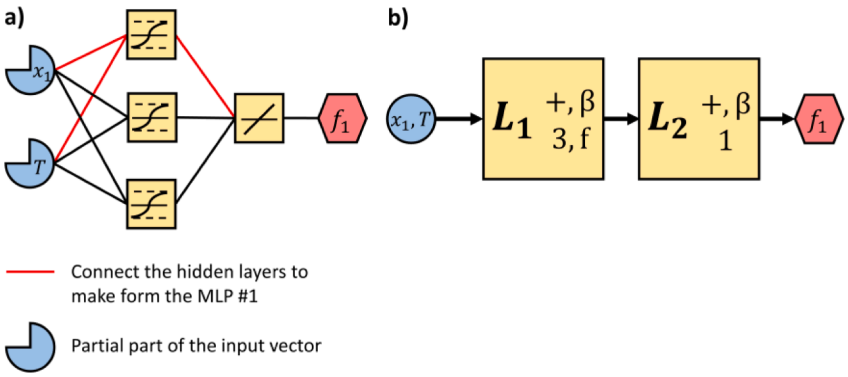
a)
x_1
T
f_1
b)
x_1, T
L_1 +, β 3, f
L_2 +, β 1
f_1
Connect the hidden layers to make form the MLP #1
Partial part of the input vector

Fig. 1. Graphical diagrams of a SNN that models f_1 of a binary mixture as a function of the independent variables x_1, T: a) typical representation (the redlines connect MLP #1), and b) this work.

models proposed by Argatov and Kocherbitov [23]. Therefore, the relevant transfer functions are symbolized with an f for the hyperbolic tangent function (tanh), f$'$ for sech^2 (note that the derivative of tanh is sech^2), e for the exponential function exp, ⊘ for an element-by-element division, and blank for a linear transfer function.

By considering that every hidden layer corresponds to one equation, the SNN shown in Fig. 1 is described with

$$L_1 = \tanh((WI) + (\beta))\ [3]. \tag{1}$$

$$L_2 = (VL_1) + (\delta) = f_1\ [1], \tag{2}$$

where W and V are the weight matrices, β and δ are the biases, I is the input vector, and L is the output of each layer. The numbers inside the square brackets([1] and [3]) indicate the number of hidden neurons in the neural network layer (a vector of equations evaluated simultaneously). The output of every hidden layer should always have as many elements as hidden neurons. Therefore, L_1 is a vector of 3 × 1 dimensions, L_2 is a scalar, W has 3 × 2 dimensions, β has 3 × 1 dimensions, V has 1 × 3 dimensions, δ is a scalar. The role of the weight matrices is not only to be used as fitting parameters but also to reshape the input vectors (e.g., I or L_1) so that the output vector has as many elements as hidden neurons.

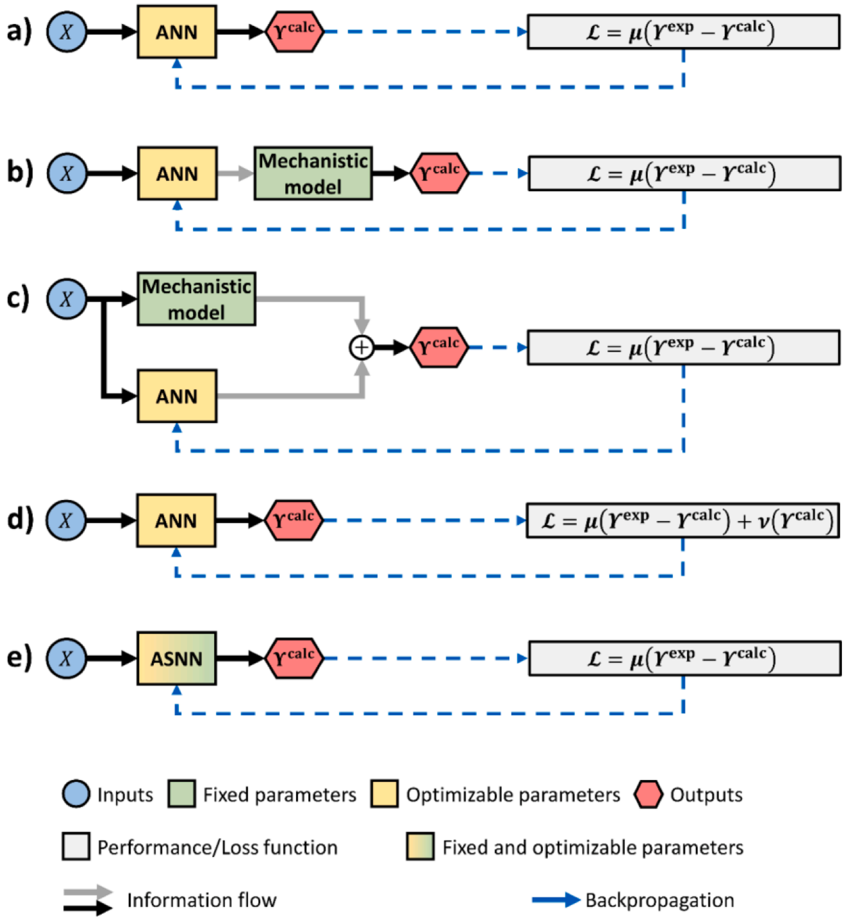
a)
X
ANN
Y^{calc}
$\mathcal{L} = \mu(Y^{exp} - Y^{calc})$
b)
X
ANN
Mechanistic model
Y^{calc}
$\mathcal{L} = \mu(Y^{exp} - Y^{calc})$
c)
X
Mechanistic model
ANN
Y^{calc}
$\mathcal{L} = \mu(Y^{exp} - Y^{calc})$
d)
X
ANN
Y^{calc}
$\mathcal{L} = \mu(Y^{exp} - Y^{calc}) + \nu(Y^{calc})$
e)
X
ASNN
Y^{calc}
$\mathcal{L} = \mu(Y^{exp} - Y^{calc})$
Inputs
Fixed parameters
Optimizable parameters
Outputs
Performance/Loss function
Fixed and optimizable parameters
Information flow
Backpropagation

Fig. 2. Model structure and optimization backpropagation algorithm of different ANN-based VLE models: a) data-driven-only, b) serial, c) parallel, d) gradient-based, and e) NNP. The input layers (X) and output layers (Y^{calc}) contain an arbitrary number of thermodynamic properties (e.g., molar fractions or fugacities).

2.2. (In)applicability of ANNs to VLE

The black-box VLE modelling approach utilizes the model structure and the optimization algorithm shown in Fig. 2 a). The optimization is done by minimizing a loss function $\mathscr{L}$ using a traditional minimization function μ (e.g., Mean-Squared Error (MSE)), and backpropagation method (more details about this method can be found elsewhere [42–44]). The goal of the optimization is to reduce the difference between the calculated VLE ($\mathrm{Y}^{\mathrm{calc}}$) and the experimental VLE data ($\mathrm{Y}^{\mathrm{exp}}$) by tuning the parameters of the ANN.

To visualize this, let us consider the SNN shown in Fig. 1 where x_1 is the CO_2 molar fraction and the CO_2 fugacity is f_1 (the SNN is described by Eqs. (1) and (2)). Considering this, the objective of this exercise is to verify whether if model can be consistent with the limiting conditions (i.e., $x_1 = 0 \, \forall \, f_1 = 0$) or not.

Assuming that $\delta \neq 0$, implies that the operation $WL_1 = -\delta$ for a finite number of combinations. However, the number of combinations in which $f_1 = 0$ is infinite (by varying T). This means that there is an infinite set of simulations in which the model is not consistent with the limits. Moreover, the existence of δ in the model does not make sense from a thermodynamic perspective. It is known that the fugacity of a component in a mixture is a function of the molar compositions, temperature, and pressure. Therefore, δ must be 0 and should not be an optimizable parameter.

Considering $\delta = 0$, the limiting condition $x_1 = 0$, merging Eqs. (1), (2). The following equation is obtained by expressing the result in scalar terms

$$f_1 = V_1 \tanh\left(W_{1,T} T + \beta_1\right) + V_2 \tanh\left(W_{2,T} T + \beta_2\right) + V_3 \tanh\left(W_{3,T} T + \beta_3\right) = 0, \tag{3}$$

where the first subscript in W, V, and β refers to the MLP number, and the second subscript (W) the input variable they are multiplying. The objective is to verify whether Eq. (3) can be equal to 0 or not. The weight matrices must be $W \neq 0$ and $V \neq 0$ if the trivial solution is to be avoided. Therefore, the argument of the hyperbolic functions should be 0. Considering that $\tanh(u) = 0 \Leftrightarrow u = 0$, there is only one temperature in each hyperbolic tangent term where they can be 0 (e.g., $T = -\beta_1 / W_{1,T}$). This suggests that the perceptron will be 0 only if β_1 and $W_{1,\mathrm{T}}$ have opposite signs (assuming that T is absolute). The first scenario in which all the hyperbolic tangent terms can cancel out is if $-\beta_1 / W_{1,\mathrm{T}} = -\beta_1 / W_{2,\mathrm{T}} = -\beta_3 / W_{3,\mathrm{T}}$. The second possible scenario is if the overall calculations done by the hyperbolic tangent cancel out (i.e., $VL_1 = 0$). The number of conditions at which the second condition occurs is infinitely small (tanh function is monotonic) considering that there are infinite scenarios where f_1 should be 0. Therefore, a fully connected shallow neural network or deep neural network cannot be thermodynamically consistent if the composition-dependent variables and temperature are linearly combined.

2.3. Hybrid models

This subsection discusses different hybrid modelling approaches [24] that might be relevant for VLE modelling namely, serial, parallel, gradient-based, and Neural Network Programming (NNP) [4]. The model structure and their optimization are shown in Fig. 2 b) to e).

Serial hybrid models (Fig. 2 b)) are metamodels composed of an ANN that feeds a mechanistic model. In simple terms, serial hybrid models generally substitute low-dimensional parametrizations of semi-empirical models with ANNs (e.g., an ANN can be used instead of Antoine's equation for computing the pure component saturation pressure). Although not labeled as hybrid models, this approach has been utilized for VLE modelling in the literature by reparametrizing already established semi-empirical models such as NRTL [46], PR EoS [47] or PRSV EoS [48].

Parallel hybrid models (Fig. 2 c)) are a metamodel structure constituted of an ANN and a mechanistic model. In these models the input is fed independently to the mechanistic and the ANN. This arrangement is discouraged for VLE modelling because parallel models "add" a correction to the mechanistic model instead of fitting the parameters within the thermodynamic framework. Therefore, the ANN part of the model is thermodynamically inconsistent and the overall metastructure cannot be consistent.

Despite having different structures, serial and parallel hybrid models share a common feature: both utilize a similar performance function $\mathscr{L}$ based on an error function μ. In both cases, the optimization algorithm (or training) seeks to minimize the value of $\mathscr{L}$ by modifying the ANN parameters. This is not the case for gradient-based hybrid models which, utilize an error function μ and additional ν functions that account for the departure between physics laws and the calculations done by the neural network (Fig. 2 d)). In this way, the gradient-based algorithm tries to guide the neural network by reducing the difference between the calculated values and the experimental data while still retaining some physics coherence. These type of hybrid models have become quite popular since the advent of Physics Guided Neural Networks (PGNNs) [49,50] and Physics Informed Neural Networks (PINNs) [51]. More recently, Masi et al. proposed [52] the Thermodynamic-Based Neural Networks, which are based on a similar concept as PINNs. However, they were developed for constitutive modelling and the approach approximates thermodynamic consistency but not completely achieve it (they utilize an error function associated to the departure between the consistency and the calculations done by their neural network model).

Despite their interesting nature and high accuracy, gradient-based hybrid models do not achieve exact coherence with physics laws (i.e., there is a difference between the computations done by the ANN and the first principles equations). Exact coherence is crucial for some research fields like equilibrium thermodynamics, in which their foundation is the exact representation of first principles. Therefore, we consider that in fields like in thermodynamics the coherence with physics laws is equally important as accuracy.

The last hybrid modelling approach is NNP (Fig. 2 e)), which consists in decomposing the equations of a mechanistic models in order to transcript them to the carefully crafted architecture of an Algorithmically Structured Neural Network (ASNNs). Transcribing an equation implies that it is the exact representation of said equation and not a close approximation. This means that the ASNNs are extrapolable, coherent with physics laws (thermodynamically consistent) and automatically differentiable. This last feature implies that the error derived from the computation of the derivatives during the optimization process is only induced by the limitations given by computer precision. Conversely, since the mechanistic part of the serial hybrid models is generally not autodifferentiable, the overall structure is not autodifferentiable either. This implies that serial hybrid models use numerical differentiation which can lead to truncation errors and extended optimization times due to the numerical differentiation algorithms [56]. Another practical advantage of ASNNs over serial models is their parallelization feature. This feature allows computing multiple VLE systems in a single step instead of utilizing "for" or "while" cycles.

Although not labeled as hybrid models, the works done by Focke [53] utilized a model based on neural network averaging, which was crafted to resemble excess functions. Later, Argatov and Kocherbitov [23] took this concept and developed neural networks that resembled the excess functions for the NRTL and Wilson models. They also presented the rigorous derivation of the activity coefficients, however, due to the complexity of the equations, they fitted the data using the excess Gibbs function. Later, Toikka et al. [54] performed an analysis of VLE data in multicomponent systems and they noted that at the limits of the concentration range, the magnitude of the deviations is large. We consider that this error is because the values located in the edges of an excess function are expected to be close to 0, hence, if the value is too small, the uncertainty provided by the experiment will be large. One of the main challenges of these approaches is their lack of robustness for considering non-isothermal data. Nonetheless, in our opinion, these approaches are a step in the right direction towards the actual integration of thermodynamics and machine learning.

3. Methodology

Sections 3.1 and 3.2 describe the ASNNs that transcribe different versions of the Wilson and NRTL models. Section 3.3 describes the transcription of Raoult's and Dalton's law to an ASNN. Section 3.4 presents the solution algorithm for each one of the models.

3.1. Wilson model

Activity coefficients (γ) are defined as the logarithmic transformation of the excess partial molar Gibbs free energy (G^{E}) [55]. The equation that defines the activity coefficient of component i is

$$\ln\gamma_i = \left(\frac{\partial\eta_t\cdot\left[G^{\mathrm{E}}/RT\right]}{\partial\eta_i}\right)_{T,\ P,\eta_{j\neq i}}, \tag{4}$$

where R is the ideal gas constant, T is temperature, η_t is the total number of moles in the mixture, η_i is the number of moles of component i in the mixture and the superscript E stands for excess property. The activity coefficient quantifies the difference between the actual behavior of a liquid mixture with respect to the ideal mixture behavior. The excess partial molar Gibbs free energy and activity coefficient models analyzed in this work are of symmetric nature and use the binary interaction concept.

The first activity coefficient model that is analyzed is the Wilson excess function [13]. However, in this work we utilize the equivalent form of the Wilson model [23]. Therefore, the excess molar Gibbs free energy is given by

$$\frac{G^{\mathrm{E}}}{RT} = \alpha\sum_{i=1}^{n} x_i f\left(\frac{1}{\sum_{j=1}^{n} W_{ij}x_j} - 1\right), \tag{5}$$

where n is the number of components in the mixture, W_{ij} are binary interactions, f is a transfer function (tanh or linear), and α is a constant scaling factor (proposed in this work). The corresponding activity coefficient model was obtained by applying Eq. (4) to Eq. (5) and yielded

$$\ln\gamma_k = \alpha f\left(\frac{1}{\sum_{j=1}^{n} W_{kj}x_j} - 1\right) + \alpha\sum_{i=1}^{n} x_i f'\left(\frac{1}{\sum_{j=1}^{n} W_{ij}x_j} - 1\right)\left(\frac{\sum_{j=1}^{n} W_{ij}x_j - W_{ik}}{\left(\sum_{j=1}^{n} W_{ij}x_j\right)^2}\right). \tag{6}$$

This work proposes the utilization of the scaling factor α as an analogous factor for the "non-randomness" factor of the NRTL model. We observed that by scaling the Wilson G^{E}/RT function, the model accuracy could be improved as long as the same scaling factor is used for every activity coefficient. It must be remarked that this does not affect the thermodynamic consistency of the model because the G^{E}/RT and $\ln\gamma$ functions are equally scaled. For example, let us consider the Gibbs-Duhem equation for a binary mixture that is scaled with the α constant

$$x_1\left(\frac{\partial[\alpha\ln\gamma_1]}{\partial x_1}\right)_{T,P} + x_2\left(\frac{\partial[\alpha\ln\gamma_2]}{\partial x_1}\right)_{T,P} = 0. \tag{7}$$

Considering that the activity coefficient equations $\ln\gamma_1$ proposed by Argatov and Kocherbitov are thermodynamically consistent, it can be readily seen that the scaling constant α is not affected by the partial derivative. Consequently, the thermodynamic consistency of Eq. (7) is independent of α.

As previously mentioned, the robustness of NNP allows the customization of neural network architectures. Therefore, the W_{ij} can be set as a function of temperature with the proper architecture. A parametrization of the Wilson models is

$$\tau_{ij} = A_{ij} + \frac{B_{ij}}{T} + C_{ij}T \tag{8}$$

$$W_{ij} = \exp\left(\tau_{ij}\right). \tag{9}$$

There are $p = 2$ temperature-dependent parameters in Eq. (8). At the pure component limit every W_{ii} must be 0 so that $G^{\mathrm{E}}/RT = 0$. Therefore, A_{ii}, B_{ii} and C_{ii} must be 0 as well. For every VLE system, there are $s_1 = n(n-1)$ non-zero temperature independent binary interactions, $s_2 = n(n-1)p$ non-zero temperature-dependent binary interactions, and $s_3 = n^2p$ temperature-dependent binary interactions (including the ones where $i = j$).

NNP was applied to the activity coefficient equation (Eq. (6)) in order to transcript the Wilson-based model to the architecture of an ASNN. The corresponding ASNN is presented in Fig. 3 and the set of equations is

$$\alpha = \beta_\alpha\ [1] \tag{10}$$

$$A = \beta_A\ [s_1] \tag{11}$$

$$B = \beta_B\ [s_3] \tag{12}$$

$$B_T = (\kappa_1 t)\odot(\kappa_3 B)\ \left[n^2\right] \tag{13}$$

$$W = \exp(\kappa_2 A + \kappa_4 B_T)\ \left[n^2\right] \tag{14}$$

$$L_1 = (\kappa_6 x)\odot(\kappa_5 W)\ \left[n^2\right] \tag{15}$$

$$L_2 = \kappa_7 L_1\ [n] \tag{16}$$

$$L_3 = 1\oslash(\kappa_8 L_2)\ [n] \tag{17}$$

$$L_4 = \tanh(\kappa_8 L_3 - \kappa_9)\ [n] \tag{18}$$

$$L_5 = \kappa_8 L_3\ [n] \tag{19}$$

$$L_6 = -\kappa_{10} W + \kappa_6 L_2\ \left[n^2\right] \tag{20}$$

$$L_7 = \mathrm{sech}^2(\kappa_8 L_3 - \kappa_9)\ [n] \tag{21}$$

$$L_8 = (\kappa_6 x)\odot(\kappa_6 L_3)\odot(\kappa_6 L_5)\odot(\kappa_5 L_6)\odot(\kappa_6 L_7)\ \left[n^2\right] \tag{22}$$

$$L_9 = \kappa_8 L_4 + \kappa_7 L_8\ [n] \tag{23}$$

$$L_{10} = \exp(\kappa_8 L_9 \odot \kappa_9 \alpha)\ [n]. \tag{24}$$

The input vector x contains the molar fractions, t is a temperature-dependent input vector (it may have as many elements as the user specifies), function f is tanh, and function f' is sech^2. It can be seen that Eq. (10) corresponds to the scaling factor, Eq. (11) to the temperature independent parameters, and Eq. (12) to the temperature-dependent parameters. The weight matrices κ have fixed values in order to transcript the Wilson model. In general, the κ weight matrices are used to transform the optimizable parameters (Eqs. (10)–(12)) to a form that ensures that every W_{ii} is equal to 0 or to perform the sums shown in Eq. (6). The description of the weight matrices is summarized in Table 1. The implementation of the κ matrices (in Matlab code) can be consulted in the supporting information.

As seen in Eqs. (10), (11), and (12), the number of elements in the vector is equal to the number of non-zero parameters. The ASNN is formulated so that it is not necessary to modify the optimization gradient in order to maintain the parameters zero parameters equal to 0 (e.g., $A_{ii} = 0$).

Note that the Wilson model with a linear transfer function can be

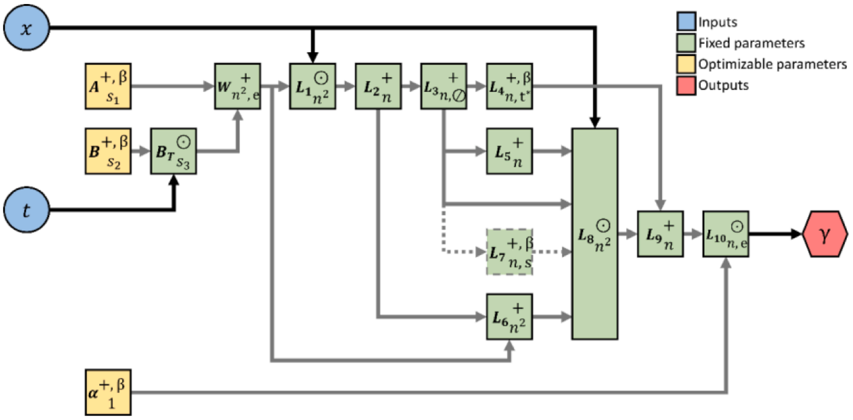

Fig. 3. Architecture of the ASNN that transcripts the Wilson-based activity coefficient models. Some connections were colored differently to make the diagram easier to follow.

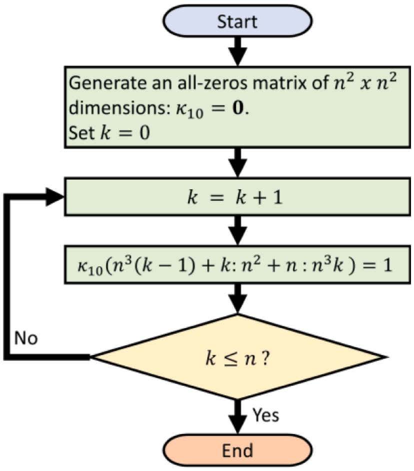
Start
Generate an all-zeros matrix of n^2 x n^2 dimensions: $\kappa_{10} = \mathbf{0}$.
Set $k = 0$
$k = k + 1$
$\kappa_{10}(n^3(k-1)+k : n^2+n : n^3k) = 1$
$k \leq n$?
No
Yes
End

Fig. 4. Algorithm to generate the M_C matrix. Note that the algorithm indexes the matrix κ_{10} as if it were a vector.

Table 1
Definition of the κ weight matrices of the Wilson and NRTL models.
The last column shows an example of the matrix when $n = 2, p = 1, s_1 = n(n-1) = 2$, $s_2 = n(n-1)p = 2$, and $s_3 = n^2p = 4$.

Var.	Description	Dimensions	Example
κ_1	Every row of an identity matrix of p x p dimensions is repeated n^2 number of times	s_3 x p	$\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$
κ_2	An identity of matrix of s_1 x s_1 dimensions is modified by placing an n number of all-zeros rows vectors with s_1 elements every $n+1$ rows starting above the first row of the identity matrix.	n^2 x s_1	$\begin{bmatrix}0&0\\1&0\\0&1\\0&0\end{bmatrix}$
κ_3	A block diagonal matrix formed of p number of κ_2 matrices	s_3 x s_1p	$\begin{bmatrix}0&0\\1&0\\0&1\\0&0\end{bmatrix}$
κ_4	κ_5 is horizontally stacked p times	n^2 x s_3	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
κ_5	Identity matrix of n^2 x n^2 dimensions	n^2 x n^2	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
κ_6	Every row of κ_8 is repeated n times	n^2 x n	$\begin{bmatrix}1&0\\1&0\\0&1\\0&1\end{bmatrix}$
κ_7	Horizontally stacking κ_8 times	n x n^2	$\begin{bmatrix}1&0&1&0\\0&1&0&1\end{bmatrix}$
κ_8	Identity matrix of n x n elements	n x n	$\begin{bmatrix}1&0\\0&1\end{bmatrix}$
κ_9	Column vector of n elements	n x 1	$\begin{bmatrix}1\\1\end{bmatrix}$
κ_{10}	See the algorithm in Fig. 4	n^2 x n^2	$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$

easily obtained by applying the linear transfer function to L_4, removing layer L_7, and using

$$L_8 = (M_A x) \odot (M_A L_3) \odot (M_A L_5) \odot (I_2 L_6) \; \left[n^2\right] \tag{25}$$

instead of Eq. (22). The generalized form of the Wilson ASNN allows the quick modification of the transfer function by only changing the transfer function in layer L_4, the transfer function in L_7, and the associated biases in L_4 and L_7.

3.2. NRTL model

3.2.1. Polynomial Parametrization

3.2.1.1. Activity coefficient. The general form of the excess molar Gibbs free energy of a degree-2 homogeneous neural network is [23]

$$\frac{G^{\mathrm{E}}}{RT} = \sum_{i=1}^{n} x_i f\left(\frac{\sum_{j=1}^{n} V_{ij} x_j}{\sum_{j=1}^{n} W_{ij} x_j}\right), \tag{26}$$

and its associated activity coefficient model is

$$\ln\gamma_k = f\left(\frac{\sum_{j=1}^{n} V_{kj} x_j}{\sum_{j=1}^{n} W_{kj} x_j}\right) + \sum_{i=1}^{n} x_i f'\left(\frac{\sum_{j=1}^{n} V_{ij} x_j}{\sum_{j=1}^{n} W_{ij} x_j}\right)\left(\frac{V_{ik}\sum_{j=1}^{n} W_{ij} x_j - W_{ik}\sum_{j=1}^{n} V_{ij} x_j}{\left(\sum_{j=1}^{n} W_{ij} x_j\right)^2}\right). \tag{27}$$

In this case, there are two sets of parameters W_{ij} and V_{ij} ($W_{ii} = 1$ and $V_{ii} = 0$). However, the original NRTL model developed by Renon and Prausnitz utilized the following parameter definitions [56]

$$W_{ij} = \exp\left(-\alpha\tau_{ij}\right) \tag{28}$$

$$V_{ij} = \tau_{ij} W_{ij}. \tag{29}$$

This parametrization is convenient since it reduces the number of fitting parameters and scales the function with the non-randomness parameter (α). In a similar fashion as with the Wilson model, τ_{ij} is usually a temperature-dependent polynomial (e.g. [57,58]).

The NNP method was applied to decompose the NRTL (considering that the function f in Eqs. (26) and (27) is tanh). The NRTL ASNN is shown in Fig. 5 and the set of equations is

$$\alpha = \beta_\alpha \ [s_1] \tag{30}$$

$$A = \beta_A \ [s_1] \tag{31}$$

$$B = \beta_B \ [s_3] \tag{32}$$

$$B_T = (\kappa_1 t) \odot (\kappa_3 B) \ \left[n^2\right] \tag{33}$$

$$\tau = \kappa_2 A + \kappa_4 B_T \ \left[n^2\right] \tag{34}$$

$$W = \exp((-\kappa_2 \alpha) \odot (\kappa_5 \tau)) \ \left[n^2\right] \tag{35}$$

$$V = (\kappa_5 \tau) \odot (\kappa_5 W) \ \left[n^2\right] \tag{36}$$

$$L_1 = (\kappa_6 x) \odot (\kappa_5 W) \ \left[n^2\right] \tag{37}$$

$$L_2 = \kappa_7 L_1 \ [n] \tag{38}$$

$$L_3 = (\kappa_6 x) \odot (\kappa_5 V) \ \left[n^2\right] \tag{39}$$

$$L_4 = \kappa_7 L_3 \ [n] \tag{40}$$

$$L_5 = 1 \oslash (\kappa_8 L_2) \ [n] \tag{41}$$

$$L_6 = f((\kappa_8 L_4) \odot (\kappa_8 L_5)) \ [n] \tag{42}$$

$$L_7 = \kappa_8 L_5 \ [n] \tag{43}$$

$$L_8 = (\kappa_{10} W) \odot (\kappa_6 L_4) \ \left[n^2\right] \tag{44}$$

$$L_9 = (\kappa_{10} V) \odot (\kappa_6 L_2) \ \left[n^2\right] \tag{45}$$

$$L_{10} = -\kappa_5 L_8 + \kappa_5 L_9 \ \left[n^2\right] \tag{46}$$

$$L_{11} = f'((\kappa_8 L_4) \odot (\kappa_8 L_5)) \ [n] \tag{47}$$

$$L_{12} = (\kappa_6 x) \odot (\kappa_6 L_5) \odot (\kappa_6 L_7) \odot (\kappa_5 L_{10}) \ \odot (\kappa_6 L_{11}) \ \left[n^2\right] \tag{48}$$

$$L_{13} = \exp(\kappa_8 L_6 + \kappa_7 L_{12}) \ [n]. \tag{49}$$

The κ matrices have the same definition as for the Wilson model (see Table 1). The classic NRTL model can be obtained by removing Eq. (47), applying the linear transfer function in Eq. (42) and utilizing

$$L_{12} = (M_A x) \odot (M_A L_5) \odot (M_A L_7) \ \odot (I_2 L_{10}) \ \left[n^2\right] \tag{50}$$

instead of Eq. (48).

For the sake of generality, the ASNN was developed so that the non-randomness factor α can be utilized as a binary interaction. Therefore, as seen in Eq. (30), the number of hidden neurons is s_1. However, the user can set α to one of the "recommended" non-randomness values by assigning this parameter value to the layer and not optimizing the layer. Alternatively, if a symmetric α (i.e., $\alpha_{12} = \alpha_{21}$) wants to be optimized, the number of hidden neurons in Eq. (30) should be set to 1 and the κ_2 weight matrix should be modified. The new weight matrix is calculated by summing every row of k_2 to produce a column vector of n^2 elements.

3.2.1.2. *Excess enthalpy.*

Utilizing the NRTL model for the estimation of both VLE and excess enthalpy (H^{E}) has been suggested and done previously [57,59–62]. Therefore, it is paramount that the ASNN can utilize H^{E} data to fit binary interaction parameters. The excess molar enthalpy can be estimated with the partial derivative of the excess molar Gibbs energy with respect to temperature

$$\left(\frac{\partial G^{\mathrm{E}}/RT}{\partial T}\right)_{x,\,P} = -\frac{H^{\mathrm{E}}}{RT^2} \ . \tag{51}$$

Differentiating Eq. (26) with respect to T, using the NRTL parametrization Eqs. (28) and ((29)), using a polynomial for the temperature dependency, and integrating the result into an ASNN yields

$$B_{dT} = (\kappa_1 t_{dT}) \odot (\kappa_3 B) \ \left[n^2\right] \tag{52}$$

$$\tau_{dT} = \kappa_4 B_{dT} \ \left[n^2\right] \tag{53}$$

$$H_1 = (\kappa_5 L_1) \odot \left(\kappa_7^{\mathrm{T}} L_5\right) \odot (\kappa_5 \tau_{dT}) \ \left[n^2\right] \tag{54}$$

$$H_2 = (\kappa_5 L_3) \odot \left(\kappa_7^{\mathrm{T}} L_5\right) \odot (\kappa_2 \alpha) \ \left[n^2\right] \tag{55}$$

$$H_3 = (\kappa_5 H_2) \odot (\kappa_5 \tau_{dT}) \ \left[n^2\right] \tag{56}$$

$$H_4 = (\kappa_7 H_1) \odot (\kappa_7 H_2) \ [n] \tag{57}$$

$$H_5 = \kappa_7 H_1 - \kappa_7 H_3 + \kappa_8 H_4 \ [n] \tag{58}$$

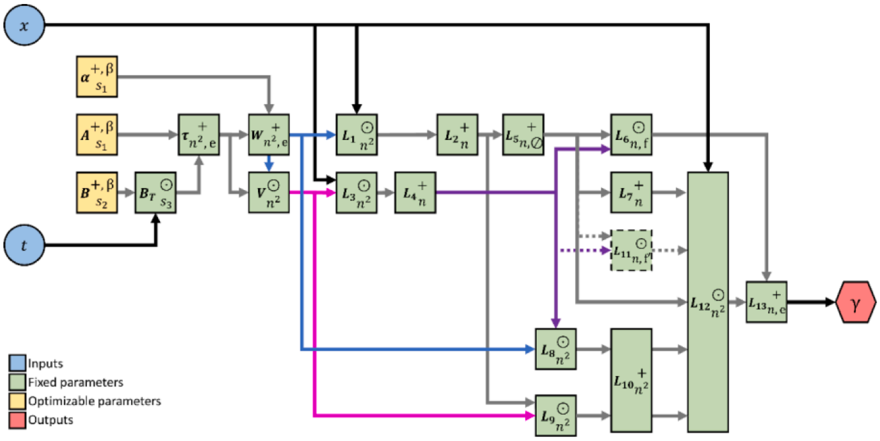

Fig. 5. Architecture of the ASNN that transcripts the NRTL-based activity coefficient models with the polynomial parametrization. Some connections were colored differently to make the diagram easier to follow.

$$H_6 = (\kappa_8 x) \odot (\kappa_8 H_5)\ [n] \tag{59}$$

$$H_7 = \kappa_9^{\mathrm{T}} H_6 = -H^{\mathrm{E}}/RT^2\ [1], \tag{60}$$

where t_{dT} is an input vector that contains the derivative of the input t (e.g., if the input in $t = 1000/T$, $t_{dT} = -1000/T^2$). Eqs. (54)–(60) are transferable to other parametrizations of the temperature dependency. Since the equations presented in this section are exact representations of the thermodynamic functions, the input or the output should **NOT** be normalized without modifying the ASNN correspondingly.

3.2.2. Sigmoid function parametrization

This parametrization stacks temperature-dependent sigmoid functions with the general form

$$\tau_{ij} = C_{ij}\tanh\left(A_{ij} + B_{ij}(T)\right) + D_{ij}, \tag{61}$$

where $B(T)$ is an arbitrary function dependent on temperature. In essence, stacking more than one equation of this form implies that a SNN is being used to find τ_{ij} instead of a polynomial. Note that the number of D_{ij} parameters is independent of the number of hyperbolic functions. This parametrization aims at capturing highly nonlinear temperature dependencies; hence, it is more convenient for applications where there is available data for H^{E} rather than only VLE.

Generalizing Eq. (61) to an arbitrary number of sigmoid functions and an arbitrarily large $B_{ij}(T)$ function provides the following equations

$$\alpha = \beta_\alpha\ [s_1] \tag{62}$$

$$A = \beta_A\ [s_4] \tag{63}$$

$$B = \beta_B\ [s_5] \tag{64}$$

$$C = \beta_C\ [s_4] \tag{65}$$

$$D = \beta_D\ [s_1] \tag{66}$$

$$B_T = (\kappa_{11} t) \odot (\kappa_{12} B)\ [s_6] \tag{67}$$

$$B_{dT} = (\kappa_{11} t_{dT}) \odot (\kappa_{12} B)\ [s_6] \tag{68}$$

$$Q_1 = \kappa_{13} A + \kappa_{14} B_T\ [s_7] \tag{69}$$

$$Q_2 = \tanh(\kappa_{15} Q_1)\ [s_7] \tag{70}$$

$$Q_3 = (\kappa_{13} C) \odot (\kappa_{15} Q_{12})\ [s_7] \tag{71}$$

$$Q_4 = \mathrm{sech}^2(\kappa_{15} Q_1)\ [s_7] \tag{72}$$

$$Q_5 = (\kappa_{13} C) \odot (\kappa_{14} B_{dT}) \odot (\kappa_{15} Q_4),\ \ [s_7] \tag{73}$$

$$\tau = \kappa_{16} Q_3 + \kappa_2 D\ \left[n^2\right] \tag{74}$$

$$\tau_{dT} = \kappa_{16} Q_5\ \left[n^2\right] \tag{75}$$

where q is the number of stacked hyperbolic functions, $s_4 = n(n-1)q$, $s_5 = n(n-1)pq$, $s_6 = n^2pq$, and $s_7 = n^2q$. Matrices κ_{11} - κ_{16} are used to transform the optimizable parameters of Eqs. (63)–(66) so that $\tau_{ii} = 0$ and to set up the hyperbolic tangent functions in parallel. The form of Eqs. (54)–(60) are independent of the temperature-dependent parametrizations.

3.3. VLE model

The VLE data usually report liquid molar fractions (x_i), vapor molar fractions (y_i), temperature (T), partial pressures (p_i), and/or total pressure (P). Therefore, thermodynamic laws must be transcribed to the ASNN in order to relate all these variables. Specifically, Raoult's law

$$y_i P = x_i \gamma_i p_i^{sat} \tag{76}$$

and Dalton's law

$$P = \sum_1^n y_i P. \tag{77}$$

The transcription of these equations to an ASNN yields the following set of equations for an n number of components (Fig. 6)

$$L_R = (\kappa_8 x) \odot (\kappa_8 \gamma) \odot (\kappa_8 p^{\mathrm{sat}}) = P_i\ [n] \tag{78}$$

$$L_D = 1 \oslash \left(\kappa_9^{\mathrm{T}} L_R\right) = 1/P\ [n] \tag{79}$$

$$L_y = (\kappa_8 L_R) \odot (\kappa_8 L_D) = y\ [n]. \tag{80}$$

This ASNN x, γ, and the pure component saturation pressure (p^{sat}) as inputs while the outputs are p_i, P^{-1}, and y.

3.4. Assembling the VLE and excess enthalpy ASNNs

The VLE models analyzed in this work are summarized and presented in Table 3. The first column contains the identifier. The second column describes the scaling factor for the Wilson model and the non-randomness factor for the NRTL model. All optimizable α are unconstrained in the VLE models. The third column refers to the transfer function used in the excess function. Lastly, the "equations" column shows the hidden layer equations that correspond to the VLE ASNN. The VLE + H^{E} models analyzed in this work are the classic formulation of the NRTL model with polynomial parametrization and the NNRTL model with a SNN parametrization. In this case, the non-randomness factor is constrained to $\alpha \geq 0$.

Table 4

4. Results and discussion

4.1. VLE modelling

This subsection compares the different VLE models summarized in Table 3. The thermodynamic systems and the number of datapoints used are shown in Table 5. Since the operating pressures shown in Table 5 are low, it is reasonable to assume ideal gas law for modelling the vapor phase behavior, hence, the component partial pressure p_i is numerically equivalent to its fugacity. The pure liquid component fugacities were calculated with the extended Antoine equation

$$p_i^{sat} = \exp\left(K_1 + \frac{K_2}{T + K_3} + K_4 T + K_5 \ln(T) + K_6 T^{K_7}\right), \tag{81}$$

where the empirical parameters K were taken from Aspen Plus v8.6 (the parameters are reported in the supporting information).

Before training the ASNN, the temperature-dependent input vector should be defined. This section defines t as

$$t = [1000\,/\,T]. \tag{82}$$

In order to train ASNNs, the data must be divided into three subsets: training, validation and testing. From these 3 datasets, only the training dataset is used for fitting the parameters, the rest are used for verifying that the model is being overfitted. In this work, 80 % of the datapoints were used for training, 10 % for validation and 10 % for testing. The performance function $\mathscr{L}$ used to train the ASNNs is

$$\mathscr{L} = \sum_1^N \left(\frac{\mathrm{Y}^{\exp} - \mathrm{Y}^{\mathrm{calc}}}{\mathrm{Y}^{\exp}}\right)^2, \tag{83}$$

where Y is any output from the ASNN model (p, P, and/or y) and N is the total number of datapoints. This performance function is indirectly set

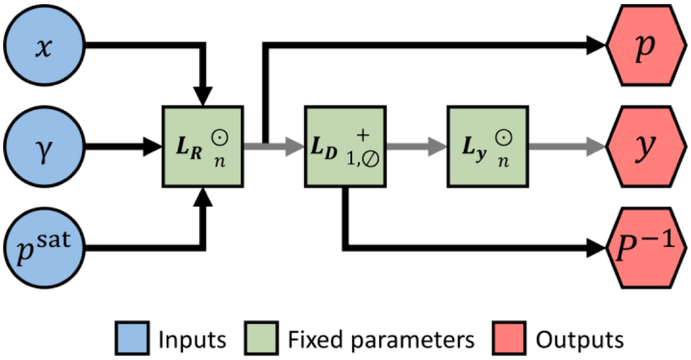

Fig. 6. ASNN that transcripts the modified Raoult's and Dalton's laws.

up by dividing each datapoint weight by $(\mathrm{Y}^{\exp})^2$ and calculating the MSE. We want to remark that all thermodynamic properties (p, P and y) were utilized in order to show that the ASNN can utilize every type of data.

After the data has been collected, divided and the performance function has been selected, the ASNNs for the 7 VLE systems were trained. Since the initial values of the fitting parameters and the training data are chosen randomly, the optimization yields different parameters in each run. Therefore, the procedure was done 20 times and the model with the lowest AARD was selected. The Levenberg-Marquardt training algorithm [67–69] provided the set of parameters with the best performance for these VLE models.

The model AARDs are shown in Table 6. The models that have 4 fitting parameters (A_{ij}, B_{ij}) either do not have α (Wilson and Wilson (H)) or α was fixed (like the NRTL model). The models with 5 fitting parameters on the other hand, optimize the binary interaction parameters together with α. The non-randomness factor α in the traditional NRTL-based models is fixed to 0.45 in system 3 while for the remaining systems is 0.30. The ASNN models were programmed and optimized by using the machine learning framework available in Matlab 2020b (nntrain). The models in the last two columns were optimized with the numerical gradient optimization function "fmincon" also found in Matlab. The average absolute relative difference ($AARD$) presented in Table 6 is is computed with

$$AARD \Big/ 100\ \% = \frac{1}{3N_p}\sum_{1}^{N_p}\left|\frac{p_i^{\exp}-p_i^{\mathrm{calc}}}{p_i^{\exp}}\right| + \frac{1}{3N_P}\sum_{1}^{N_P}\left|\frac{P^{\exp}-P^{\mathrm{calc}}}{P^{\exp}}\right| + \frac{1}{3N_y}\sum_{1}^{N_y}\left|\frac{y_i^{\exp}-y_i^{\mathrm{calc}}}{y_i^{\exp}}\right|. \tag{84}$$

The results in Table 2 suggest that better sets of fitting parameters might be found by using the Levenberg-Marquardt training algorithm instead of the numerical gradient method. In every system using the ASNN-based NRTL or Wilson model provided either equal or better results than using optimizers with numerical gradients. For the case of the Wilson model, the ASNNs provided equal or better sets of parameters than the ones using a numerical optimizer in most cases.

According to Table 2, modifying the Wilson and NRTL models with the sigmoid functions do to not improve the model performance significantly. The NRTL (H) has slightly better fittings because there is an there is an extra degree of freedom because α is optimizable. These results might suggest that the non-linearities caused by the mixture non-ideality are more related to the binary interaction modelling functions rather than the mathematical form of the excess function. In general, all models have reasonable performance for every studied VLE system, however, larger deviations are observed in system 3. P −xy diagrams of the different models are presented in Fig. 7 and their corresponding parameters are shown in Table 7.

Fig. 7 a) – b) show that the Wilson and hyperbolic Wilson models have poor modelling performance. This can be attributed to the highly

Table 2
Definition of the κ weight matrices of NRTL model with a SNN parametrization. The last column shows an example of the matrix when $n = 2$, $p = 2$, $q = 2$. Therefore, $s_4 = 4$, $s_5 = 8$, $s_6 = 16$, and $s_7 = 8$.

Var.	Description	Dimensions	Example
κ_{11}	Vertically stacking q times κ_1	s_6 x p	$\begin{bmatrix}\kappa_1\\ \kappa_1\end{bmatrix}$
κ_{12}	Block diagonal matrix formed of q number of κ_3 matrices (Kronecker product of κ_3 and and an identity matrix of q x q dimensions).	s_6 x s_5	$\begin{bmatrix}\kappa_3 & 0\\ 0 & \kappa_3\end{bmatrix}$
κ_{13}	Block diagonal matrix formed of q number of κ_2 matrices (Kronecker product of κ_2 and and an identity matrix of q x q dimensions).	s_7 x s_4	$\begin{bmatrix}\kappa_2 & 0\\ 0 & \kappa_2\end{bmatrix}$
κ_{14}	Block diagonal matrix formed of q number of κ_4 matrices (Kronecker product of κ_4 and and an identity matrix of q x q dimensions).	s_7 x s_6	$\begin{bmatrix}\kappa_4 & 0\\ 0 & \kappa_4\end{bmatrix}$
κ_{15}	Identity matrix	s_7 x s_7	$\begin{bmatrix}1 & 0 & 0 & \cdots & 0\\ 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$
κ_{16}	Horizontally stacking q times κ_5	n^2 x s_7	$\begin{bmatrix}\kappa_5 & \kappa_5\end{bmatrix}$

0 is a null matrix with the appropriate dimensions<

Table 3
Characteristics of the VLE models analyzed in this work.

Model	α	Transfer function f	Equations
Wilson	1	Linear	(10)-(20),(25),(23),(24),(78)-(80)
Wilson (H)	1	tanh	(10) - (24), (78) - (80)
Wilson (S)	Optimizable	Linear	(10) - (20), (25), (23), (24), (78) - (80)
Wilson (S – H)	Optimizable	tanh	(10) - (24), (78) - (80)
NRTL	Fixed	Linear	(30) - (46), (50), (49), (78) - (80)
NRTL (H)	Optimizable (symmetric)	tanh	(30) - (49), (78) - (80)

Table 4
Characteristics of the VLE + H^{E} models analyzed in this work.

Model ID	Parametrization	Equations
NRTL	Polynomial	(30) - (46), (50), (49), (78) - (80), (52) - (60)
NNRTL	Hyperbolic tangent functions	(62) - (75), (35) - (46), (50), (49), (78) - (80), (54) - (60)

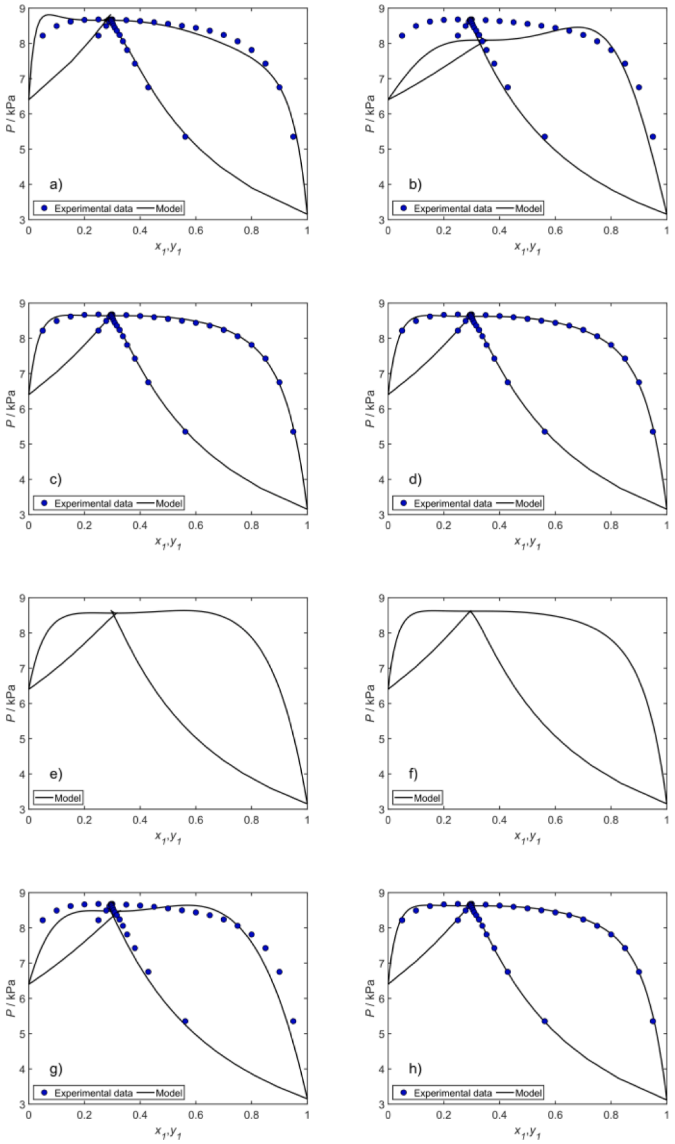

Fig. 7. $P - xy$ diagrams of the Ethanol (1) – Cyclohexane (2) system at T = 283 K with: a) Wilson, b) Wilson (H), c) Wilson (S), d) Wilson (S – H), e) NRTL 1 (α = 0.3), f) NRTL 2 (α = 0.45), g) NRTL (H), h) NRTL (inconsistent, α = 0.45). Experimental data: [65].

non-ideal interactions between the two components that cause the azeotropic singularity. Furthermore, it seems that the non-idealities also “push” the ethanol vapor fraction towards the positive azeotrope. It can be seen that both models struggle to model the VLE either by under or overestimating the experimental pressure over the entire experimental range. On the other hand, Fig. 7 c) and Fig. 7 d) show that adding a scaling factor to the Wilson model greatly enhance the performance of the model. The scaling factor adjusts the numerator of the mixing function in order to modify, as needed, the values of the excess molar Gibbs energy and the activity coefficients. This helps to distribute the effect of the molecule interactions between the binary interactions and the adjustable empirical scaling factor.

The usefulness of scaling the numerator and denominator perhaps was the main inspiration for the addition of the non-randomness factor in the NRTL model. In fact, if we analyze the scaled Wilson and NRTL equations, it can be observed that the scaling and the non-randomness factors have an analogous purpose. In the NRTL model, α reduces the effect of the exponential term while increasing the effect of τ_{ij} (see Eqs. (28), (29)). Conversely, the scaling factor in the modified Wilson model alters the effect of the exponential term. To illustrate this, Fig. 7 e), f) highlight the effect of the non-randomness factors in the NRTL model. Although both models seem to provide good modelling performance, the model in Fig. 7 f) has a smooth behavior in the azeotropic region while the model in Fig. 7 e) does not. In view of this, the non-randomness factor can be considered as an adjustable scaling empirical factor that scales the excess molar Gibbs energy model to better represent the azeotropic points and liquid phase splitting.

The scaled Wilson model seems to have a better performance than the original Wilson model. On the contrary, the NRTL model properly estimates the ternary data because α multiplies the binary interaction parameters (Eq. (28)) instead of multiplying the entire excess molar Gibbs energy function. Therefore, the robustness of the scaled Wilson model is highly reduced.

The $P - xy$ diagrams shown in Fig. 7 g) and Fig. 7 h) correspond to the hyperbolic NRTL models where the main difference is that the model represented in Fig. 7 g) is thermodynamically consistent while the one shown in Fig. 7 h) is not (system 3* in Table 5). The inconsistency was set so that none of the binary interaction parameters were fixed (i.e., $A_{ii} \neq 0$ and $B_{ii} \neq 0$, therefore, there are 9 fitting parameters). Table 6 shows that the inconsistent hyperbolic NRTL model outperforms its consistent counterpart. However, the $P - xy$ diagrams do not show evident signs of thermodynamic inconsistency.

In order to show a clearer picture about the effects of utilizing inconsistent thermodynamic models, Fig. 8 is presented. Fig. 8 b) shows that the inconsistent model does not comply with the definition of a symmetric excess Gibbs energy model (G^E/RT when x_i = 1). This is despite being statistically closer to the experimental data than the model used in Fig. 8 a). This points out that in the case of models based on neural networks, utilizing statistical figures only, is not enough to prove the validity of a thermodynamic model. Thus, the thermodynamic consistency must be checked to conclude that ANN-based models are better than traditional thermodynamic models.

Another main criticism towards machine learning models can be their apparent lack of extrapolability. Although in strict terms, every semi-empirical model is only valid within the measured experimental data, the models developed with machine learning provide physically unfeasible results outside the experimental data boundaries. This is not a challenge for NNP-based VLE models since the basic thermodynamic relationships are integrated in the ASNN. This implies that the ASNNs may be capable of providing reasonable predictions even if it used outside the training operating conditions. For instance, Fig. 9 (dashed line) shows that the scaled Wilson model can provide thermodynamically physically feasible VLE predictions outside the data training limits.

Table 5
VLE systems modelled

System	System	Number of datapoints	T / K	P / kPa	Refs.
1	Benzene (1) - Isooctane (2)	31	308 - 348	12 - 86	[63]
2	Carbon Tetrachloride (1)				
- Cyclohexane (2)	19	349 - 353	101.3	[64]	
3	Ethanol (1) - Cyclohexane (2)	89	283 - 347	8 - 101.3	[64, 65]
4	Methanol (1) - Ethanol (2)	11	338 - 350	101.3	[66]
5	Ethanol (1) - Water (2)	18	351 - 362	101.3	[66]
6	Methanol (1) - Water (2)	20	338 - 370	101.3	[66]
7	Methanol (1) - Ethanol (2) - Water (3)	49/86*	338 - 370	101.3	[66]

* the model was trained with 49 binary VLE datapoints and tested against 86 datapoints consisting of binary and ternary VLE

Table 6
AARD of the VLE models for the studied systems

System	ASNN						fmincon	
	Wilson	Wilson (H)	Wilson (S)	Wilson (S) (H)	NRTL	NRTL (H)	Wilson	NRTL
1	0.40	0.83	0.40	0.40	0.42	0.40	0.40	0.43
2	1.16	1.31	1.10	0.96	1.20	1.04	1.26	1.44
3	3.68	8.07	1.92	1.95	1.82	2.00	3.90	1.96
3*	2.94	6.44	1.35	1.40	1.27	1.46	N/A	N/A
4	0.09	0.12	0.08	0.12	0.16	0.08	0.13	0.16
5	0.67	2.23	0.42	0.41	0.43	0.35	0.79	0.45
6	0.60	0.47	0.55	0.53	0.56	0.40	0.61	0.57
7	1.46	2.67	1.46	1.88	1.30	1.40	2.14	2.13
Average	**1.15**	**2.24**	**0.85**	**0.89**	**0.84**	**0.81**	**1.32**	**1.02**
No. of fitting parameters	4	4	5	5	4	5	4	4

Table 7
Activity coefficient parameters for the ethanol (1) - cyclohexane (2) system (the parameters are presented in the form of Eq. (8))

Model	A_{21}	A_{12}	B_{21}	B_{12}	α
Wilson	-0.3015	-0.4121	-0.2185	-0.2845	1
Wilson (H)	-0.6428	-0.8404	-0.1927	-0.1679	1
Wilson (S)	0.7656	0.0159	0.0840	0.2052	-1.3352
Wilson (S - H)	0.7191	-0.0080	0.0722	0.2011	-1.4924
NRTL 1	-0.6496	1.1809	0.6107	0.3698	0.45
NRTL 2	-0.5330	2.1541	0.4219	-0.0361	0.30
NRTL (H)	0.6509	0.1943	0.1176	0.1769	-0.7723

The accuracy-related advantages of the ASNN-based models are mostly likely to be caused by the automatic differentiation algorithm. Moreover, the ASNN training algorithm utilizes an arbitrary selected portion of the data to optimize the model while the other portion is used to avoid model overfitting. Both of these features are essential in machine learning modelling, and it seems that they can provide an advantage for VLE modelling because allows finding better sets of fitting parameters. This suggests that using all the experimental data for parameter fitting is not necessarily the best option. One of the interesting features that can be exploited for VLE modelling while using ASNN-based VLE models, is the division of data. Although the data division ratios are fixed by the user, the algorithm chooses the datapoints for fitting the parameters randomly. Considering this, the data division algorithm can be used to automatically find them. This might be helpful to discriminate the datapoints without previous knowledge of which datapoint is "good" for the parameter fitting. Unfortunately, the stochastic nature of the process also has the drawback of not guaranteeing that the model will find the same set of optimized parameters (although the values can be close).

4.2. Simultaneous VLE and excess enthalpy modelling

This subsection discusses the advantages of utilizing ANN optimization frameworks and sigmoid functions for the simultaneous fitting of VLE data and H^E. Table 8 presents the modelled thermodynamic systems and the number of individual datapoints for each thermodynamic variable. The same performance function (Eq. (83)) used in section 4.2 is used in this section. However, the temperature-dependent input vectors (t and t_{dT}) are different for the models presented in this work. The temperature-dependent vectors for the NRTL model are

$$t = \begin{bmatrix} 1000/T \\ 0.01T \end{bmatrix} \tag{85}$$

$$t_{dT} = \begin{bmatrix} -1000/T^2 \\ 0.01 \end{bmatrix}, \tag{86}$$

while for the NNRTL model t and t_{dT} only contain the first element of

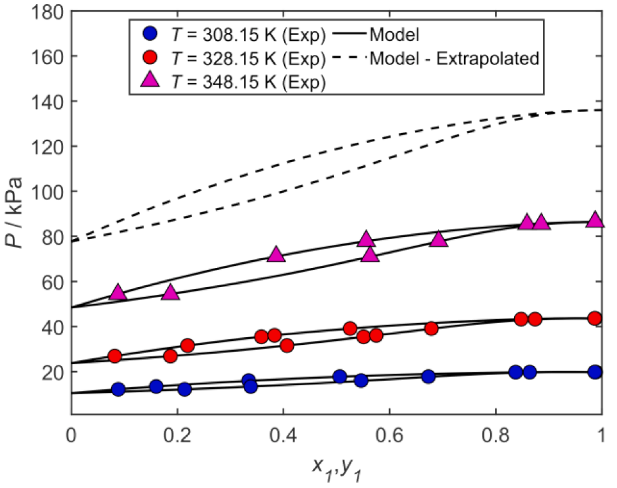

Fig. 9. $P - xy$ diagrams of the Benzene (1) – Isooctane (2) system. The dashed line values are calculated with the Wilson model at 363.15 K (extrapolated). Parameters: $A_{12} = -1.0486$, $A_{21} = 0.3153$, $B_{12} = 0.3183$, $B_{21} = 0.0797$, $\beta_\alpha^s = -1.080$. Experimental data: [63].

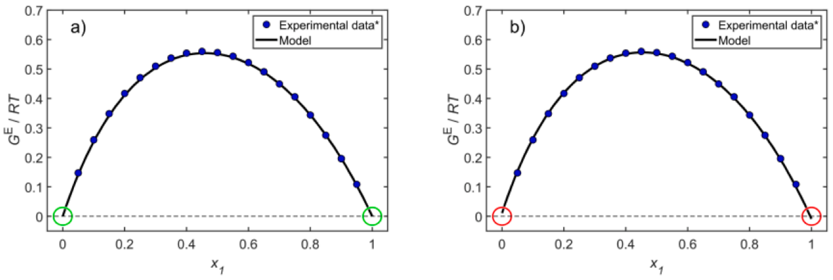

Fig. 8. Excess molar Gibbs energy diagrams of the Ethanol (1) – Cyclohexane (2) system at 293.15 K. The solid lines were computed using the same models as: a) Fig. 7 g), and b) Fig. 7 h) respectively. *Calculated from the experimental data: [65].

Table 8
Modelled systems using VLE and excess enthalpy data.

ID	System	Number of datapoints			T / K	P / kPa	Refs.
		y	P	H^E			
8	Acetone (1) - Hexane (2)	124	50	49	253 - 329	10 - 101.325	[70,71]
9	Dichloromethane (1) - Acetone (2)	142	134	137	273 - 348	9 - 309	[70,71]
10	Acetone (1) - Water (2)	86	116	141	288 - 401	1 - 369	[70,71]
11	Ethanol (1) - Water (2)	676	354	154	273 - 423	12 - 988	[61,66, 71–75]

both vectors. This means that $p = 2$ for the NRTL model and $p = 1$ for the NNRTL model. The NNRTL models developed in this section have two stacked hyperbolic tangent functions ($q = 2$).

Considering the above, the ASNNs were trained using the Bayesian regularization algorithm [69,76,77] ("trainbr" in Matlab 2020b). The AARD of the optimized models is presented in Table 9 together with the optimization of the NRTL model optimized with fmincon. The optimized parameters of the models with the lowest AARD are presented in Table 10. The AARD was computed using

$$AARD \Big/ 100\ \% = \frac{1}{4N_p}\sum_{1}^{N_p}\left|\frac{p_i^{\text{exp}} - p_i^{\text{calc}}}{p_i^{\text{exp}}}\right| + \frac{1}{4N_P}\sum_{1}^{N_P}\left|\frac{P^{\text{exp}} - P^{\text{calc}}}{P^{\text{exp}}}\right|$$

$$+\frac{1}{4N_y}\sum_{1}^{N_y}\left|\frac{y_i^{\text{exp}} - y_i^{\text{calc}}}{y_i^{\text{exp}}}\right| + \frac{1}{4N_P}\sum_{1}^{N_H}\left|\frac{H^{\text{E,exp}} - H^{\text{E,calc}}}{H^{\text{E.exp}}}\right|. \quad (87)$$

The ID column in Table 8 consists of a number that refers to the system number and a letter that indicates whether the model uses a non-optimizable α (the models whose ID have a suffix "a" fix the scaling factor to 0.3), a symmetric optimizable α (suffix "b") or an unsymmetric α (suffix "c"). The addition of an optimizable symmetric α adds 1 degree of freedom while the unsymmetric α adds 2 (e.g., the NNRTL 8c model has 14 fitting parameters from the SNN functions and 2 from α).

The Bayesian regularization algorithm (trainbr) showed better accuracy (lower AARD) for training VLE + H^E models than the trainlm. In fact, for VLE modelling the Bayesian Regularization is a competitive optimizer compared to the used Levenberg-Marquardt algorithm (however, this algorithm was slightly superior).

The results show that models with unsymmetric α have 35 % lower AARD than those with fixed symmetric α and 10 % with respect to the ones with an optimizable symmetric α. In general, utilizing a SNN to parametrize the temperature (NNRTL) is better than utilizing polynomial-based parametrizations (NRTL). However, the advantages of the NNRTL model are more evident at systems where H^E has a highly unsymmetrical and non-monotonic behavior of H^E (e.g., Fig. 10 c) – d)). For simpler H^E behaviors the advantages are diminished (Fig. 10 a)) or simply the polynomial parametrization is better (Fig. 10 b)).

Table 10
Vectors containing the binary interaction parameters.

ID	α	A	B	C	D
8b	0.4169	-5.4097 -1.6853 6.3013 0.2180	2.2558 0.4013 -1.5695 -0.2621	1.9995 1.8285 -0.4896 -1.1646	-0.3379 0.9710
9c	0.0266 0.5235	0.5022 0.5880	-0.5274 0.1332 -0.0370 -0.0267		
10c	-0.1971 -0.6958	-2.4321 -1.8265 7.7666 -2.1862	1.2992 1.0256 -2.0550 0.5955	0.9532 1.2459 -0.3967 -2.5575	0.5897 -0.6404
11c	1.6830 0.1951	0.8750 5.4679 -0.0039 6.9744	1.5013 -2.1492 -0.4357 -2.1980	1.1558 -0.8690 3.8923 -0.5611	1.5563 2.5896

The results in Table 9 show that utilizing the ASNNs together with the neural network optimization framework is enough to improve the available thermodynamic models even if the same parametrization for the temperature-dependence is being utilized. This suggests that the reparameterization of established thermodynamic models with neural networks might be caused more by the optimization algorithms rather than from the ANN. In fact, we tried to increase the number of stacked sigmoid functions (q) to check whether the accuracy of H^E could be improved. The results suggested that the maximum effective parameters used by the NNRTL is between 15 – 16 (out of 16). The number of effective parameters in the Bayesian regularization algorithm reports how many parameters are actually having an impact on the model output.

As previously discussed, it is recommendable to check the thermodynamic models in order to ensure that the model is thermodynamically correct. Fig. 11 shows that the ASNN model can estimate both the VLE and H^E of the ethanol (1) – water (2) system over a wide range of temperatures, pressures, and molar compositions. In order to further test the method, we compared the model predictions against VLE data [78] that, in addition being above the maximum VLE temperature data, were not used for the parameter fitting of the ASNN. This demonstrates that the VLE model is capable of accurately predicting extrapolated "unseen" data (the AARD of P is 2.9 % and y_i is 4.0 %) despite being a neural network.

Table 9
AARD / % of the VLE data (P, p, and y) and H^E of the different NRTL model variants developed. The AARD values in bold correspond to the models used in the H^E plots shown in Fig. 13.

ID	NRTL (fmincon)			NRTL (ASNN)			NNRTL		
	VLE	H^E	Overall	VLE	H^E	Overall	VLE	H^E	Overall
8a	4.4	5.5	4.7	4.2	5.4	4.5	4.2	5.1	4.4
8b	3.9	5.7	4.4	3.9	5.4	4.3	**4.3**	**2.9**	**3.9**
8c	3.9	5.4	4.3	4.3	4.1	4.2	4.3	2.7	3.9
9a	3.4	2.6	3.2	3.4	1.4	2.9	3.5	2.1	3.1
9b	3.4	1.4	2.9	3.3	1.5	2.9	3.4	1.3	2.9
9c	3.4	1.3	2.9	**3.3**	**1.4**	**2.8**	3.5	1.4	3.0
10a	8.8	27.5	13.5	7.3	26.3	12.1	6.1	16.7	8.7
10b	7.6	22.9	11.4	6.6	21.9	10.4	6.7	16.7	9.2
10c	6.7	17.0	9.3	6.2	16.5	8.8	**6.4**	**15.2**	**8.6**
11a	10.6	60.0	22.9	3.4	62.4	18.1	3.8	44.4	13.9
11b	6.2	32.6	12.8	5.2	34.9	12.6	5.0	23.2	9.5
11c	5.6	32.2	12.3	4.7	25.5	9.9	**5.2**	**20.7**	**9.1**

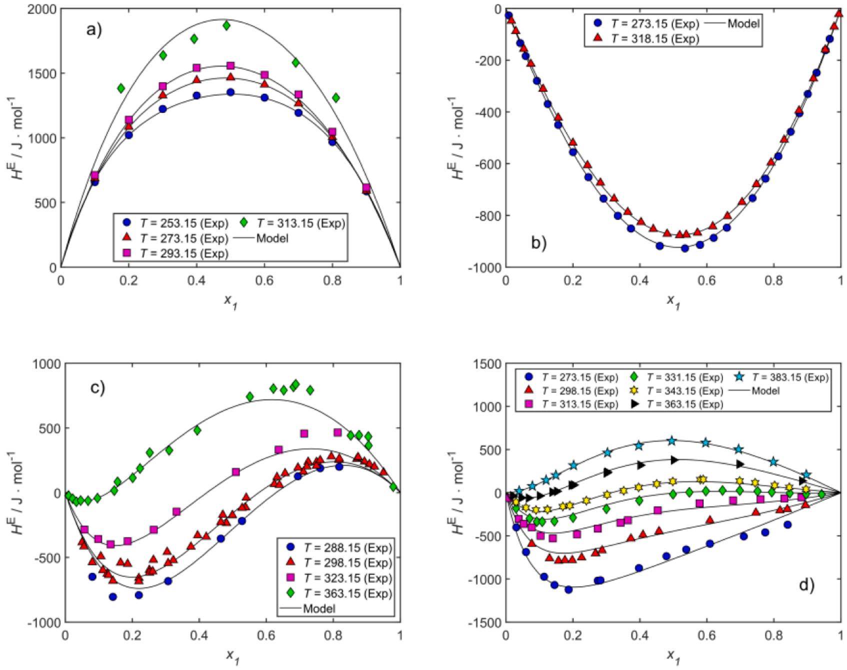

Fig. 10. Excess enthalpy plots of a) Acetone (1) – Hexane (2) [Model 8b], b) Dichloromethane (1) - Acetone (2) [Model 9c], c) Acetone (1) – Water (2) [Model 10c] and d) Ethanol (1) – Water (2) [Model 11c]. The fitting parameters are presented in Table 10. Experimental data: [71].

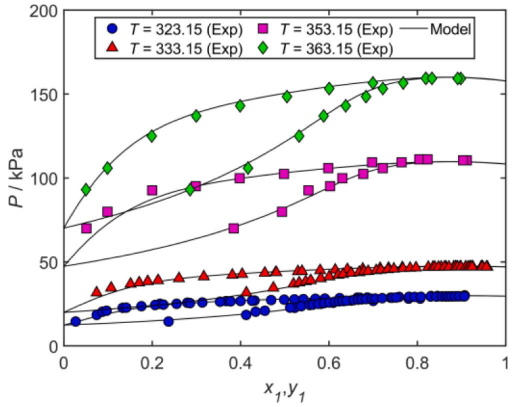

Fig. 11. VLE plot of the ethanol (1) – water (2) system. Experimental data at 323.15 – 333.15 K [73] and at 353.15 – 363.15 K [78]. [Model 11c].

4.3. *Applications to other systems*

4.3.1. *Non-ideal vapor phase*

It is reasonable to assume that the vapor phase behaves as an ideal gas in the systems studied in this work. Nonetheless, this might not be acceptable for other VLE systems. If it is desired to include an equation of state for the vapor phase, the algorithms shown in Fig. 12 can be utilized. The algorithm shown in Fig. 12a can be utilized whenever *T*, *P* and all molar fractions of both phases are known. Conversely Fig. 12b can be utilized for cases in which the total pressure or vapor molar fraction values are missing. In the likely case that an ANN-based equation of state (EoS) is developed in the future, it will be enough to couple the ANN-based EoS with the ASNN shown in Fig. 6. Alternatively, a surrogate model can be utilized to transform the fugacities to molar fractions.

Note that the algorithms shown in Fig. 12 are for parameter fitting only. Therefore, if the user wants to use the ASNN activity coefficient model for other phase-equilibria calculations, the ASNN should include the appropriate computational algorithms for them.

4.3.2. *Liquid-Liquid Equilibrium*

There are systems relevant to industrial applications which require Liquid-Liquid Equilibrium (LLE) calculations. LLE can be described for a system with two liquid phases as

$$x_i^I\gamma_i^I = x_i^{II}\gamma_i^{II}, \tag{88}$$

where *I* and *II* are liquid phases 1 and 2, respectively. Utilizing feedforward ASNNs is not straightforward in LLE. This is because the activity coefficient γ is a function of the component molar fraction and the variables are not separable like in VLE. However, the semi-empirical parameters can be regressed with a feedforward ASNN by using an approximation. This approach would imply to utilize the ASNN shown in Fig. 13 and given in the following set of equations

$$L_a^I = \left(\kappa_8 x^I\right) \odot \left(\kappa_8 \gamma^I\right)\ [n] \tag{89}$$

$$L_a^{II} = \left(\kappa_8 x^{II}\right) \odot \left(\kappa_8 \gamma^{II}\right)\ [n] \tag{90}$$

$$L_{\Delta a} = \kappa_8 L_{R1} - \kappa_8 L_{R2}\ [n]. \tag{91}$$

The goal of the optimization algorithm is to minimize the difference between L_a^I and L_a^{II} by utilizing the experimental data and tuning the fitting parameters so that $L_{\Delta a}$ is minimized ($L_{\Delta a} \cong 0$). Therefore, the output Δ should be set equal to 0 for the experimental datapoints and x^I, x^{II}, and *T* are the inputs. It should be noted that the input nodes that contain the activity coefficients in Fig. 13 must utilize the same activity coefficient model and set of fitting parameters.

This simplified approach should only be utilized to optimize the fitting parameters but not for performing LLE calculations. This is because the system is over-specified in accordance with the Gibbs phase rule. Over-specifying the LLE system was necessary since feedforward ASNNs cannot perform the iterative calculations needed in LLE. Because of this, more complex ASNN architectures such as Recurrent Neural Networks (RNNs) might be needed.

5. Conclusions

The application of Neural Network Programming (NNP) hybrid

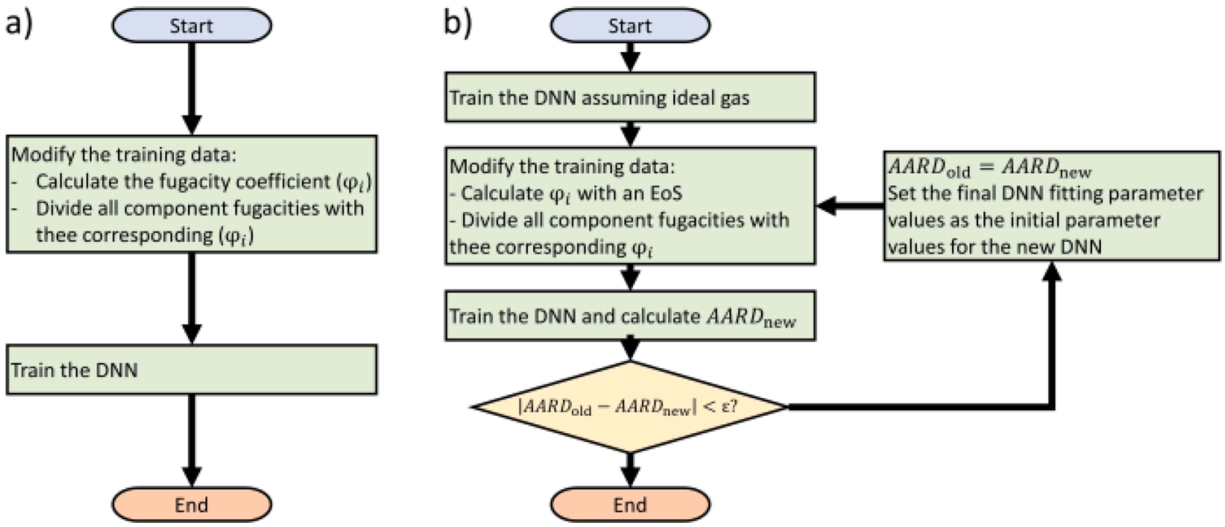
a)
Start
Modify the training data:
- Calculate the fugacity coefficient (φ_i)
- Divide all component fugacities with thee corresponding (φ_i)
Train the DNN
End
b)
Start
Train the DNN assuming ideal gas
Modify the training data:
- Calculate φ_i with an EoS
- Divide all component fugacities with thee corresponding φ_i
Train the DNN and calculate $AARD_{\text{new}}$
$|AARD_{\text{old}} - AARD_{\text{new}}| < \varepsilon$?
$AARD_{\text{old}} = AARD_{\text{new}}$
Set the final DNN fitting parameter values as the initial parameter values for the new DNN
End

Fig. 12. Algorithms for training the activity coefficient ASNN for systems with a non-ideal vapor phase. The tolerance value ε is arbitrary.

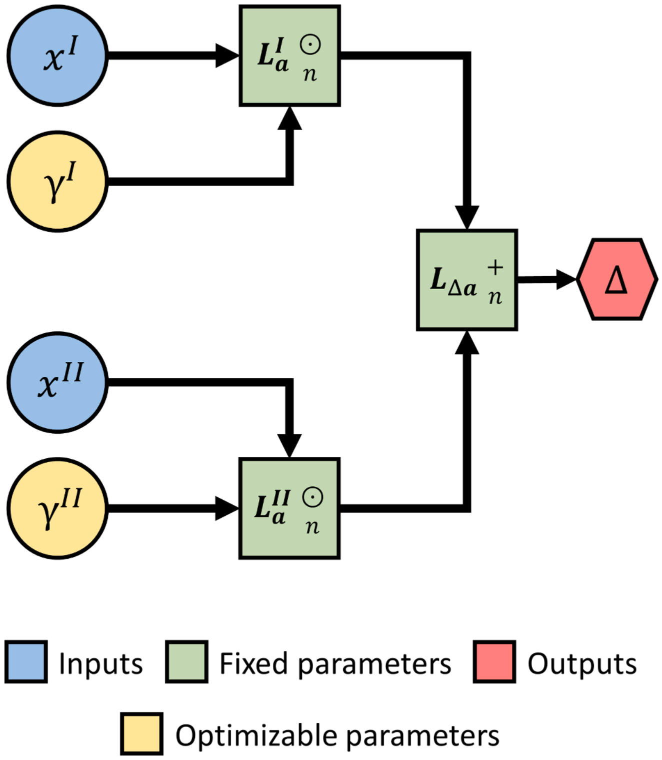

Fig. 13. ASNN that optimizes the fitting parameters by minimizing the difference between the activity of phase I and phase II.

modelling technique was utilized to develop thermodynamically consistent, accurate and extrapolable VLE models based on Algorithmically Structured Artificial Neural Networks (ASNNs). Utilizing NNP for developing thermodynamic models helps to guarantee that the thermodynamic consistency is preserved. Furthermore, it is ensured that the pure component thermodynamic properties (e.g., vapor pressure) and the definition of ideal mixture can be recovered from the model whenever the model is used under limiting conditions (e.g., for symmetric excess functions if $x_i = 1$, $G^E/RT = 0$). These conditions are paramount for any thermodynamic model to be regarded as such.

By using established thermodynamic models as examples, it was demonstrated that it is possible to formulate thermodynamically consistent models based on ASNNs. This works provides the insight of the needed facets and paths for developing new and thermodynamically consistent models entirely based on neural network technology. The thermodynamic consistency can easily be overlooked when the data are separated from thermodynamic principles. Therefore, developing models within the ASNN framework can help to analyze which modifications can be performed on the model without altering the validity of the equations.

It was shown that the utilization of ANN optimization frameworks is a feasible alternative for improving the available thermodynamic models without the need of flooding the model with dozens or hundreds of parameters. Thermodynamic models and parameterizations have a threshold on how many temperature-dependent parameters can be effectively utilized to fit the thermodynamic properties.

Utilizing machine learning should not be considered as a substitution for thermodynamic modelling but rather a tool for its improvement. If a neural network model comply with thermodynamic constraints is because its architecture and parameters are not compatible with thermodynamics (as in typical fully connected ANNs).

Thermodynamic models for mixtures that are based on data-driven techniques should be carefully developed and evaluated because the typical ANN architectures (shallow neural networks or fully connected neural networks) do not ensure thermodynamic consistency or proper behavior in the pure component limits. The validation of ANN-based thermodynamic models should not be created only with statistical figures like AARD or mean squared error since they can be misleading. The relevant limits and constraints should be tested as well since they are the core of the thermodynamic framework.

Using machine learning training algorithms (ASNN training) might provide better sets of fitting parameters than regular gradient methods. Furthermore, several modifications for the Wilson and NRTL models were analyzed. Utilizing a hyperbolic tangent modification for the excess molar Gibbs free energy function (as proposed by Focke) does not significantly improve or reduce the accuracy of the NRTL model. Further evaluations are required for definite conclusions.

Stacking hyperbolic tangent functions for substituting the typical low-dimensional polynomial temperatures is a feasible alternative for the simultaneous modelling of VLE and H^E. For example, the AARD of the H^E in the ethanol (1) – water (2) system was just 20.7 %. As opposed to conventional neural network models, the number of fitting parameters did not increase drastically by using this modification (7 parameters are used in the original NRTL and up to 16 parameters with a NRTL with a neural-network-based parametrization).

The ASNNs presented in this work might seem like complex or extensive models. However, these representations allow the parallel calculation of multiple VLE steps in a single function call. This means that if 1,000 VLE datapoints are to be computed, the model will only be called one time instead of 1,000 times and avoiding the typical "for" cycles used. This proves quite convenient for situations in which large thermodynamic databases are utilized for parameter fitting.

Due to the efficient matricial nature of the calculations performed in the ASNN, the simultaneous optimization of multiple VLE systems can be performed. This implies that large databases can be utilized to test different parametrizations of the VLE system (e.g., perhaps a SNN parametrization as shown in this work). A similar approach that the one presented here can be done to transcribe other activity coefficient models (e.g., UNIQUAC/UNIFAC) and perform a simultaneous optimization of all the fitting parameters. This would allow verifying whether there is a correlation between some component properties and the binary interactions.

CRediT authorship contribution statement

Andres Carranza-Abaid: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Hallvard F. Svendsen:** Formal analysis, Resources, Writing – review & editing, Supervision. **Jana P. Jakobsen:** Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We want to acknowledge Tobias Neumann for the insightful discussions. This research was funded by the Faculty of Natural Sciences of the Norwegian University of Science and Technology (NTNU).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.fluid.2022.113597.

References

[1] O. Vinyals, I. Babuschkin, W.M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J.P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T.L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, D. Silver, Grandmaster level in StarCraft II using multi-agent reinforcement learning, Nature 575 (2019) 350–354, https://doi.org/10.1038/s41586-019-1724-z.

[2] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature 529 (2016) 484–489, https://doi.org/10.1038/nature16961.

[3] F. Yusuf, T. Olayiwola, C. Afagwu, Application of Artificial Intelligence-based predictive methods in Ionic liquid studies: a review, Fluid Phase Equilib. 531 (2021), 112898, https://doi.org/10.1016/j.fluid.2020.112898.

[4] A. Carranza-Abaid, J.P. Jakobsen, Neural network programming: integrating first principles into machine learning models, Comput. Chem. Eng. 163 (2022), 107858, https://doi.org/10.1016/j.compchemeng.2022.107858.

[5] J.D. Van der Waals, On the Continuity of the Gaseous and Liquid States, Leiden, 1873.

[6] G. Soave, Equilibrium constants from a modified Redlich-Kwong equation of state, Chem. Eng. Sci. 27 (1972) 1197–1203, https://doi.org/10.1016/0009-2509(72)80096-4.

[7] D.-Y. Peng, D.B. Robinson, P-R.a New Equation of State, Proc. Natl. Acad. Sci. U. S. A. 15 (1976) 11–18.

[8] G.M. Kontogeorgis, E.C. Voutsas, I.V. Yakoumis, D.P. Tassios, An equation of state for associating fluids, Ind. Eng. Chem. Res. 35 (1996) 4310–4318, https://doi.org/10.1021/ie9600203.

[9] J. Gross, G. Sadowski, Perturbed-Chain SAFT: an equation of state based on a perturbation theory for chain Molecules, Ind. Eng. Chem. Res. 40 (2001) 1244–1260, https://doi.org/10.1021/ie0003887.

[10] M. Margules, "Über die Zusammensetzung der gesättigten Dämpfe von Misschungen". Sitzungsberichte der Kaiserliche Akadamie der Wissenschaften Wien Mathematisch-Naturwissenschaftliche Klasse II., 1895. https://archive.org/details/sitzungsbericht10wiengoog.

[11] N.A. Gokcen, Gibbs-duhem-margules laws, J. Phase Equilib. 17 (1996) 50–51, https://doi.org/10.1007/BF02648369.

[12] J.J. van Laar, Über Dampfspannungen von binären Gemischen, Z. Für Phys. Chem. 72U (1910) 723–751, https://doi.org/10.1515/zpch-1910-7236.

[13] G.M. Wilson, Vapor-liquid equilibrium. XI. A new expression for the excess free energy of mixing, J. Am. Chem. Soc. 86 (1964) 127–130, https://doi.org/10.1021/ja01056a002.

[14] H. Renon, J.M. Prausnitz, Local compositions in thermodynamic excess functions for liquid mixtures, AIChE J. 14 (1968) 135–144, https://doi.org/10.1002/aic.690140124.

[15] D.S. Abrams, J.M. Prausnitz, Statistical thermodynamics of liquid mixtures: a new expression for the excess Gibbs energy of partly or completely miscible systems, AIChE J. 21 (1975) 116–128, https://doi.org/10.1002/aic.690210115.

[16] G.M. Kontogeorgis, G. Folas, Thermodynamic Models for Industrial Applications, Wiley, Chichester, 2010.

[17] J.M. Prausnitz, R.N. Lichtenthaler, E. Gomes de Azevedo, Molecular Thermodynamics of Fluid-Phase Equilibria, 3rd ed., Prentice-Hall, 1999.

[18] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, 1962.

[19] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314, https://doi.org/10.1007/BF02551274.

[20] K.I. Funahashi, On the approximate realization of continuous mappings by neural networks, Neural Networks 2 (1989) 183–192, https://doi.org/10.1016/0893-6080(89)90003-8.

[21] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366, https://doi.org/10.1016/0893-6080(89)90020-8.

[22] K. Hornik, Approximation Capabilities of Multilayer Neural Network, Neural Networks 4 (1991) 251–257.

[23] I. Argatov, V. Kocherbitov, A note on artificial neural network modeling of vapor-liquid equilibrium in multicomponent mixtures, Fluid Phase Equilib. 502 (2019), 112282, https://doi.org/10.1016/j.fluid.2019.112282.

[24] J. Sansana, M.N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L.H. Chiang, M.S. Reis, Recent trends on hybrid modeling for Industry 4.0, Comput. Chem. Eng. 151 (2021), 107365, https://doi.org/10.1016/j.compchemeng.2021.107365.

[25] R. Petersen, A. Fredenslund, P. Rasmussen, Artificial neural networks as a predictive tool for vapor-liquid equilibrium, Comput. Chem. Eng. 18 (1994) S63–S67, https://doi.org/10.1016/0098-1354(94)80011-1.

[26] A. Dey, S.K. Dash, B. Mandal, Equilibrium CO2 solubility and thermophysical properties of aqueous blends of 1-(2-aminoethyl) piperazine and N-methyldiethanolamine, Fluid Phase Equilib. 463 (2018) 91–105, https://doi.org/10.1016/j.fluid.2018.01.030.

[27] B. Vaferi, Y. Rahnama, P. Darvishi, A. Toorani, M. Lashkarbolooki, Phase equilibria modeling of binary systems containing ethanol using optimal feedforward neural network, J. Supercrit. Fluids. 84 (2013) 80–88, https://doi.org/10.1016/j.supflu.2013.09.013.

[28] C. Si-Moussa, S. Hanini, R. Derriche, M. Bouhedda, A. Bouzidi, Prediction of high-pressure vapor liquid equilibrium of six binary systems, carbon dioxide with six esters, using an artificial neural network model, Braz. J. Chem. Eng. 25 (2008) 183–199, https://doi.org/10.1590/S0104-66322008000100019.

[29] S. Garg, A.M. Shariff, M.S. Shaikh, B. Lal, H. Suleman, N. Faiqa, Experimental data, thermodynamic and neural network modeling of CO2 solubility in aqueous sodium salt of l-phenylalanine, J. CO2 Util. 19 (2017) 146–156, https://doi.org/10.1016/j.jcou.2017.03.011.

[30] P. Pakzad, M. Mofarahi, A.A. Izadpanah, M. Afkhamipour, Experimental data, thermodynamic and neural network modeling of CO2 absorption capacity for 2-amino-2-methyl-1-propanol (AMP) + Methanol (MeOH) + H2O system, J. Nat. Gas Sci. Eng. 73 (2020), 103060, https://doi.org/10.1016/j.jngse.2019.103060.

[31] R. Sharma, D. Singhal, R. Ghosh, A. Dwivedi, Potential applications of artificial neural networks to thermodynamics: vapor-liquid equilibrium predictions, Comput. Chem. Eng. 23 (1999) 385–390, https://doi.org/10.1016/S0098-1354(98)00281-6.

[32] M.C. Iliuta, I. Iliuta, F. Larachi, Vapour-liquid equilibrium data analysis for mixed solvent-electrolyte systems using neural network models, Chem. Eng. Sci. 55 (2000) 2813–2825, https://doi.org/10.1016/S0009-2509(99)00529-1.

[33] L. Govindarajan, P.L. Sabarathinam, Prediction of vapor-liquid equilibrium data by using radial basis neural networks, Chem. Biochem. Eng. Q. 20 (2006) 319–323.

[34] H. Karimi, F. Yousefi, Correlation of vapour liquid equilibria of binary mixtures using artificial neural networks, Chin. J. Chem. Eng. 15 (2007) 765–771, https://doi.org/10.1016/S1004-9541(07)60160-8.

[35] C.A. Faúndez, F.A. Quiero, J.O. Valderrama, Correlation and prediction of VLE of water+congener mixtures found in alcoholic beverages using an artificial neural network, Chem. Eng. Commun. 198 (2011) 102–119, https://doi.org/10.1080/00986445.2010.493127.

[36] M. Lashkarbolooki, B. Vaferi, A. Shariati, A. Zeinolabedini Hezave, Investigating vapor-liquid equilibria of binary mixtures containing supercritical or near-critical carbon dioxide and a cyclic compound using cascade neural network, Fluid Phase Equilib. 343 (2013) 24–29, https://doi.org/10.1016/j.fluid.2013.01.012.

[37] F. Gharagheizi, QSPR studies for solubility parameter by means of genetic algorithm-based multivariate linear regression and generalized regression neural network, QSAR Comb. Sci. 27 (2008) 165–170, https://doi.org/10.1002/qsar.200630159.

[38] D. Bastani, M.E. Hamzehie, F. Davardoost, S. Mazinani, A. Poorbashiri, Prediction of CO2 loading capacity of chemical absorbents using a multi-layer perceptron neural network, Fluid Phase Equilib. 354 (2013) 6–11, https://doi.org/10.1016/j.fluid.2013.05.017.

[39] Z. Lyu, H. Ma, H. Zhang, W. Ying, Solubility of carbon dioxide in methanol from 213.15 K to 273.15 K: Measurement and modeling, Fluid Phase Equilib. 471 (2018) 40–54, https://doi.org/10.1016/j.fluid.2018.04.014.

[40] H. Ahmadian Behrooz, R.B. Boozarjomehry, Prediction of limiting activity coefficients for binary vapor-liquid equilibrium using neural networks, Fluid Phase Equilib. 433 (2017) 174–183, https://doi.org/10.1016/j.fluid.2016.10.033.

[41] P.R.B. Guimaraes, C. McGreavy, Flow of information through an artificial neural network, Comput. Chem. Eng. 19 (1995) 741–746, https://doi.org/10.1016/0098-1354(95)87123-3.

[42] P. Werbos, Beyond regression: New tools for prediction and analysis in the behavioral sciences, Harvard University, 1974.

[43] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (1990) 1550–1560, https://doi.org/10.1109/5.58337.

[44] J. Schmidhuber, Deep Learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.

[45] A. Carranza-Abaid, H.F. Svendsen, J.P. Jakobsen, Surrogate modelling of VLE: integrating machine learning with thermodynamic constraints, Chem. Eng. Sci. X. 8 (2020), 100080, https://doi.org/10.1016/j.cesx.2020.100080.

[46] H.E. Reynel-Ávila, A. Bonilla-Petriciolet, J.C. Tapia-Picazo, An artificial neural network-based NRTL model for simulating liquid-liquid equilibria of systems present in biofuels production, Fluid Phase Equilib. 483 (2019) 153–164, https://doi.org/10.1016/j.fluid.2018.11.009.

[47] S. Rajasekaran, G.A. Vijayalakshmi Pai, Neural networks, fuzzy logic and genetic algorithm: synthesis and applications, Asoke K. Ghosh, New Delhi, 2011. https://books.google.no/books?hl=es&lr=&id=bVbj9nhvHd4C&oi=fnd&pg=PP1&ots=tIxxDENzK3&sig=KBJk_sAlB8BN6YOale6SD8GQGjk&redir_esc=y.

[48] T. Hatami, M. Rahimi, H. Daraei, E. Heidaryan, A.A. Alsairafi, PRSV equation of state parameter modeling through artificial neural network and adaptive network-based fuzzy inference system, Korean J. Chem. Eng. 29 (2012) 657–667, https://doi.org/10.1007/s11814-011-0235-x.

[49] A. Daw, A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling, (2017). http://arxiv.org/abs/1710.11431.

[50] A. Karpatne, G. Atluri, J.H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: a new paradigm for scientific discovery from data, IEEE Trans. Knowl. Data Eng. 29 (2017) 2318–2331, https://doi.org/10.1109/TKDE.2017.2720168.

[51] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, https://doi.org/10.1016/j.jcp.2018.10.045.

[52] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based Artificial Neural Networks for constitutive modeling, J. Mech. Phys. Solids. 147 (2021), 104277, https://doi.org/10.1016/j.jmps.2020.104277.

[53] W.W. Focke, Mixture models based on neural network averaging, Neural Comput. 18 (2006) 1–9, https://doi.org/10.1162/089976606774841576.

[54] A.M. Toikka, G.K. Misikov, A.V. Petrov, Analysis of Data on Vapor–Liquid Equilibrium in Multicomponent Systems Using Artificial Neural Networks, Theor. Found. Chem. Eng. 55 (2021) 403–409, https://doi.org/10.1134/S004057952103026X.

[55] A.Z. Panagiotopoulos, Essential Thermodynamics, Drios Press, San Bernardino, CA, 2014.

[56] H. Renon, J.M. Prausnitz, Local compositions in thermodynamic excess functions for liquid mixtures, AICHE J. 14 (1968) 135–144.

[57] K. Tochigi, J. Rarey, J. Gmehling, Recommended NRTL model parameters by simultaneous correlation of VLE, infinite dilution activity coefficients and excess enthalpy data, J. Chem. Eng. Japan. 42 (2009) 376–380, https://doi.org/10.1252/jcej.08we123.

[58] K. Tochigi, H. Inoue, K. Kojima, Determination of azeotropes in binary systems at reduced pressures, Fluid Phase Equilib. 22 (1985) 343–352, https://doi.org/10.1016/0378-3812(85)87030-8.

[59] Y. Demirel, H. Gecegörmez, Simultaneous correlation of excess gibbs energy and enthalpy of mixing by the UNIQUAC Equation, Can. J. Chem. Eng. 67 (1989) 455–461, https://doi.org/10.1002/cjce.5450670316.

[60] K.A.G. Schmidt, Y. Maham, A.E. Mather, Use of the NRTL equation for simultaneous correlation of vapour-liquid equilibria and excess enthalpy: applications to aqueous alkanolamine systems, J. Therm. Anal. Calorim. 89 (2007) 61–72, https://doi.org/10.1007/s10973-006-8307-6.

[61] E.C. Voutsas, C. Pamouktsis, D. Argyris, G.D. Pappa, Measurements and thermodynamic modeling of the ethanol-water system with emphasis to the azeotropic region, Fluid Phase Equilib. 308 (2011) 135–141, https://doi.org/10.1016/j.fluid.2011.06.009.

[62] M.R. Gennero de Chialvo, A.C. Chialvo, Determination of the domain of the function F(HE,GE) = 0 for Wilson, NRTL, LEMF, and UNIQUAC equations, Ind. Eng. Chem. Res. 33 (1994) 1035–1039, https://doi.org/10.1021/ie00028a035.

[63] S. Weissman, S.E. Wood, Vapor-liquid equilibrium of benzene-2,2,4-trimethylpentane mixtures, J. Chem. Phys. 32 (1960) 1153–1160, https://doi.org/10.1063/1.1730865.

[64] K.S. Yuan, J.C.K. Ho, A.K. Koshpande, B.C.-Y. Lu, Vapor-liquid equilibria, J. Chem. Eng. Data 8 (1963) 549–559, https://doi.org/10.1021/je60019a024.

[65] J. Nagai, N. Ishii, II - IV. Parts, Soc. Chem. Ind. Jap. 38 (1935), 86–10, http://www.ddbst.com/en/EED/VLE/VLE/Ethanol%3BCyclohexane.php.

[66] K. Kurihara, M. Nakamichi, K. Kojima, Isobaric vapor-liquid equilibria for methanol + ethanol + water and the three constituent binary systems, J. Chem. Eng. Data 38 (1993) 446–449, https://doi.org/10.1021/je00011a031.

[67] K. Levenberg, F. Arsenal, A method for the solution of certain non-linear problems in least squares, Q. Appl. Math. 1 (1943) 536–538.

[68] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math. 11 (1963) 431–441, https://doi.org/10.1137/0111030.

[69] M.T. Hagan, M.B. Menhaj, Training feedforward networks with the Marquardt algorithm training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Netw. 5 (1994) 989–993, https://doi.org/10.1006/brcg.1996.0066.

[70] J. Gmehling, U. Onken, W. Arlt, Vapor-liquid equilibrium data collection : 3+4 : Aldehydes and ketones, ethers, DECHEMA Chemistry Data Series, Frankfurt, 1979.

[71] C. Christensen, J. Gmehling, R. Peter, U. Weidlich, Heats of mixing data collection : 1 : Binary systems, DECHEMA Chemistry Data Series (1984).

[72] K. Kurihara, T. Minoura, K. Takeda, K. Kojima, Isothermal vapor-liquid equilibria for methanol + ethanol + water, methanol + water, and ethanol + water, J. Chem. Eng. Data 40 (1995) 679–684, https://doi.org/10.1021/je00019a033.

[73] M.K. C, Dutta Choudhury, Excess free energy of binary mixtures of n-butylamine with ethyl alcohol & n-propyl alcohol, Indian J. Chem. Sect. A 14 (1976) 553–556.

[74] A.F. Cristino, S. Rosa, P. Morgado, A. Galindo, E.J.M. Filipe, A.M.F. Palavra, C. A. Nieto de Castro, High-temperature vapour-liquid equilibrium for the water-alcohol systems and modeling with SAFT-VR: 1. Water-ethanol, Fluid Phase Equilib. 341 (2013) 48–53, https://doi.org/10.1016/j.fluid.2012.12.014.

[75] J.A. Larkin, Thermodynamic properties of aqueous non-electrolyte mixtures I. Excess enthalpy for water + ethanol at 298.15 to 383.15 K, J. Chem. Thermodyn. 7 (1975) 137–148, https://doi.org/10.1016/0021-9614(75)90261-X.

[76] D.J.C. MacKay, Bayesian interpolation, Neural Comput. 4 (1992) 415–447, https://doi.org/10.1162/neco.1992.4.3.415.

[77] F.Dan Foresee, M.T. Hagan, Gauss-Newton approximation to bayesian learning, in: IEEE Int. Conf. Neural Networks - Conf. Proc. 3, 1997, pp. 1930–1935, https://doi.org/10.1109/ICNN.1997.614194.

[78] I.C. Arango, A.L. Villa, Isothermal vapor - liquid and vapor-liquid-liquid equilibrium for the ternary system ethanol+water+diethyl carbonate and constituent binary systems at different temperatures, Fluid Phase Equilib. 339 (2013) 31–39, https://doi.org/10.1016/j.fluid.2012.11.026.