Computers and Chemical Engineering 150 (2021) 107344

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Optimal fragrances formulation using a deep learning neural network architecture: A novel systematic approach

Vinícius V. Santana[a,b], Márcio A.F. Martins[b,*], José M. Loureiro[a], Ana M. Ribeiro[a], Alírio E. Rodrigues[a,*], Idelfonso B.R. Nogueira[a,*]

[a] Laboratory of Separation and Reaction Engineering, Associate Laboratory LSRE-LCM, Department of Chemical Engineering, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

[b] Industrial Engineering Program, Polytechnic School, Federal University of Bahia, Rua Prof. Aristides Novis, 2 – Federação, 40210-630, Salvador/Bahia, Brazil

ARTICLE INFO

Article history:
Received 14 October 2020
Revised 20 March 2021
Accepted 19 April 2021
Available online 22 April 2021

Keywords:
Meta-heuristic optimization
Flavor & fragrance
Deep learning
Perfume formulation

ABSTRACT

Human civilization has been economically exploring the enjoyable smell of substances for centuries, giving rise to multi-billion-dollar business. Few works have addressed the formulation of perfumes using a systematic approach based on computational techniques. Thus, the objective of the present work is to develop a novel systematic strategy for optimal perfume design. The strategy comprises a deep learning model trained from high-fidelity simulations, an objective function that reflects the desirable spectrum of the perfume, and a meta-heuristic optimization method. It was applied to define the perfume composition that produces an odor spectrum of pine forest and floral while minimizing non-desirable odors. Hence, we propose an objective function to encode the peculiarities of a fragrance design comprising the question: Which perfume composition attains the desirable odor spectrum across time and space? The results demonstrated the methodology value in fragranced product design by offering a framework to handle the formulation problem.

1. Introduction

The human olfaction is a complex sensorial system that has been recently discovered to discriminate over 1 trillion different smells (Bushdid et al., 2014). The reasons why it detects and responds pleasantly to certain substances are not fully understood. Nonetheless, humans have been economically exploring the enjoyable smell of substances for centuries, giving rise to the Flavor & Fragrance (F&F) multi-billion dollar business (Leffingwell & Associates, 2018).

The fragrance industry is usually organized in two groups of products: Fine Fragrances & Cosmetics (FFC), Consumer Fragrances (CF). The latter usually incorporate pleasant odors to improve consumer acceptance in functional products such as soaps, toothpaste, among others. Nonetheless, CF may include products that go beyond functionality such as rim blocks and all-purpose cleaners (conditioners, shampoos). On the other hand, due to specific customer demands and consequently a higher value associated, the FFC fragrances are typically more refined and richer. Therefore, the production of FFC products is associated with higher budgets with the purpose of being perceived and distinguished from the regular scent.

The F&F industry sales have experienced sustained growth in the last decades (Leffingwell & Associates, 2018) leading manufactures to seek new tools for driving innovation while meeting the growing demand. Even so, the scientific literature is surprisingly scarce on works that develop tools to foster innovation in the field. This is particularly pronounced for fine fragrances (perfumes). This might be a consequence of a collective trend to believe that well-formulated perfumes result from an artistic alchemy.

However, it is indeed a complex task to formulate a perfume. For fine fragrances, a careful selection of ingredients is made first. These ingredients are usually classified into three types of “notes”: top, middle and base (Carles, 1962). Each note is used to cause distinct olfactory impressions or also improve perfume evaporation features: top notes are intended to be the protagonist smell in the first few minutes but should not last for too long – components with high volatility are used for such purpose. The middle notes are considered the “body” and the “heart” of the perfume – they are usually clearly distinguished few minutes/hours after application and after the top note. The base notes must last for hours and even days and can act as a fixative by reducing the evaporation rate of top and middle note chemicals. Nevertheless, middle and even base notes can also sometimes be clearly noticeable in

* Corresponding author.
E-mail addresses: marciomartins@ufba.br (M.A.F. Martins), arodrig@fe.up.pt (A.E. Rodrigues), idelfonso@fe.up.pt (I.B.R. Nogueira).

https://doi.org/10.1016/j.compchemeng.2021.107344
0098-1354/

Nomenclature

A	Area (cm^2)
a	Neural network hidden state
b	Neural network bias vector
C	Molar density (mol/cm^3)
c	Neural network memory state
D	Diffusion coefficient (cm^2/s)
Eth	Ethanol
F	Function
f	Function
$Lina$	Linalool
m	Number of training examples
N	Molar flux ($mol/cm^2/s$)
n	Number of moles/ Number of time lags/ Number of output features/ Exponent in power law
P	Pressure (atm)
Pin	Pinene
T	Time steps
$Tona$	Tonalide
t	Time (s)
u	Vector with input signals
W	Matrix with neural network trainable parameters
w	Weight matrix for perfume optimization
X	Input layer in neural network
x	Mole fraction in liquid phase (dimensionless)
y	Mole fraction in gas phase (dimensionless)
$\hat{y}$	Vector with output values
z	Axial position (cm)

Subscripts and superscripts

a	Output signal
b	Input signal
c	Memory state
ca	Hidden state to memory state neural network mapping
cx	Input layer to memory state neural network mapping
D	Desorbent
eq	Equilibrium
f	Forget gate
fa	Hidden state to forget gate neural network mapping
fx	Input layer to forget gate neural network mapping
g	Gas phase
gl	Gas/liquid interface
i	Component/Row index of weight matrix /Training example
j	Component/Column index of weight matrix /Feature
k	Time instant/Solvent
l	Lower limit/ Liquid phase
M	Molar
max	Maximum
min	Minimum
mix	Mixture
o	Output gate/ Number of fragrances
oa	Hidden state to output gate neural network mapping
ox	Input layer to output gate neural network mapping
sat	Saturation
T	Total
u	Update gate/ Upper limit
ua	Hidden state to update gate neural network mapping
ux	Input layer to update gate neural network mapping
z	Axial direction
0	Initial condition

Greek symbols

Γ	Multiplicative gate in neural network
γ	Activity coefficient (dimensionless)
Δ	Range
θ	Vector with parameters
σ	Sigmoid function
Ψ	Odor intensity (dimensionless)

the first seconds of perception of the fragrance, which makes the perfume formulation an even more complex problem. Therefore, the present work proposes the formulation of an objective function that takes into consideration the perception through space and time, adding this degree of complexity to the mathematical problem.

The notes are the building blocks of the perfumes but are not stand-alone chemicals in a real perfume formulation. They are incorporated in a solvent matrix – generally a blend of solvents comprised mostly by ethanol. Unfortunately, the human olfactory system is highly sensitive to ethanol and responds unpleasantly to it. It can be challenging for non-experienced perfumers to quickly choose how much of each fragrance to use while avoiding excessive ethanol smell.

Looking from that perspective, perfumes are liquid solutions of fragrances in a proper solvent. For the fragrances to be perceived, their vapors have to diffuse through the surrounding space and be detected by a human nose. From this point of view, a perfume allows the incorporation of several consolidated methods from chemical engineering disciplines; the process of transferring chemicals from liquid to vapor phase is very well described by models for vapor–liquid phase equilibrium and gas molecular diffusion. The human detection can be analyzed by using models from psychophysics. Several works have addressed the experience of scenting perfume from this perspective (Almeida et al., 2019; Mata et al., 2005a; M. Teixeira et al., 2013; Teixeira et al., 2010a, 2010b, 2009; Wakayama et al., 2019). These works have addressed the perfume release and propagation by considering the evaporation of odorants from human skin and modeling it with a skin absorption rate and air evaporation, proposing a phenomenological model to represent the phenomena, opening space to the application of engineering tools to solve F&F industry problems.

The use of high-fidelity simulation to optimize a perfume, though a non-conventional technique, was introduced in (Mata et al., 2005b, 2005a) through the perfume ternary diagram (PTD®) method. It offers a quick guide to the choice of a ternary perfume and a solvent composition to optimize its odor features. According to those works, the composition must be chosen such that: (1) The top note should be noticed strongly near the perfume or shortly after the application, (2) The base note should be noticed long after the application and (3) ethanol should not be noticed since it is unpleasant and not appreciated. In spite of its simplicity and capacity to provide an efficient guide for simple perfume mixtures (three components), the method can easily lose its effectiveness in more complex scenarios. In fact, real perfume mixtures can contain up to 100 components (M. A. Teixeira et al., 2013). On the other hand, high-fidelity simulation relies on mathematical models that are traditionally grounded on the natural sciences where phenomenology is important and may demand significant computational power.

From that perspective, some works in the literature have proposed the use of Artificial Intelligence (AI) tools to help the design of perfumes. Zhang et al., 2018 trained an artificial neural network model that learned how to classify a fragrance molecule among 20

possible descriptors with great accuracy. It encoded the molecular structure into numerical predictors using a fragment (group) based representation. Researchers from Google introduced a graph neural network for quantitative structure-odor relationship (QSOR) modeling (Sanchez-Lengeling et al., 2019). The work proposes to encode molecular structures of odorants as graphs in which nodes are atoms and edges are chemical bonds. The application of AI from academia in F&F related problems is still limited to the classification point of view. In the industry, there are already models being used for perfume design, such as International Business Machines (IBM) AI tool from Symrise ("IBM Research and Symrise are working together - Symrise," 2018) company and Carto tool from Givaudan ("The new AI tool that represents the future of fragrance formulations | Givaudan," 2019), though the details are kept in secrecy. However, no reports were found applying AI in the field in order to use its computationally cheaper models for expensive simulations, as suggested in order fields (von Rueden et al., 2020). Furthermore, up to the present data, it was not found a systematical approach to perform an optimal perfume formulation considering its whole spectrum. Addressing a similar issue, Zhang et al., 2020 proposed a systematic methodology to maximize the sensorial quality of cosmetics.

Thus, the objective of the present work is to develop a systematic strategy for optimal perfume design taking into consideration all the four pillars that compose the fragrance the solvent, and top, middle, and base notes. The strategy considers the use of a phenomenological model to generate an extensive database to train an AI model, an objective function that precisely reflects how desirable is the construction of the smell of the perfume and a robust optimization method to maximize it. Among the AI techniques, the Deep Neural Networks (DNN) architecture was chosen because its notorious capacity to model dynamic systems, already demonstrated in other fields (Miguel et al., 2020). A Deep Learning (DL) based model can efficiently replace the high-fidelity simulation and save considerable CPU time.

Several algorithms can be used to minimize the objective function: either deterministic (gradient-based) or meta-heuristic. One must be careful when using a deterministic approach since the global optimum is not guaranteed for non-convex problems. Meta-heuristic methods (such as Non-Sorted Genetic Algorithm (NSGA), Differential Evolution (DE), Particle Swarm Optimization (PSO)) are flexible and relatively easy to implement. Also, they can deal with complex objective functions. Additionally, one kind of derivative-free algorithm is very popular within the machine learning community: Bayesian Optimization (BO). BO has two main components: a Bayesian statistical model for modeling the objective function, and an acquisition function for deciding in which part of the search domain to sample in the next iteration. It is extensively used for optimizing hyperparameters of deep neural networks and several areas due to its ability to optimize expensive black-box derivative-free functions (Frazier, 2018; Shahriari et al., 2016).

In this work, a meta-heuristic technique was used to solve the optimization problem, the Particle Swarm Optimization (PSO), which have been highlighted as an effective method to find minima in complex optimization tasks (Cuadros Bohorquez et al., 2020; Nogueira et al., 2019; Schwaab et al., 2008a). Moreover, its computational performance can be enhanced with parallelization and its results can be directly applied in the evaluation of the optimal point uncertainty (Nogueira et al., 2020; Schwaab et al., 2008b). In this way, coupling a surrogate model with a powerful optimization method seems more appropriate for designing and optimizing real fragrance structures following specific requirements, as it will be further demonstrated.

The proposed methodology was applied in a case study where the goal is to find the optimal composition of a perfume that will provide an equivalent high-level odor characteristic of pine forest and floral through space and time, while minimizing as much as possible the ethanol odor.

2. The proposed methodology

This section is divided into two parts. The first part details the stages to identify the deep learning models that will represent the fragrance release and propagation, through time and space. The second part deals with the underlying optimization problem formulation for fragrance performance.

2.1. The perfume diffusion process and model

Understanding the release of chemicals from a liquid mixture and their propagation through the air is an important step to design perfume products with the desired performance. A real perfume experience would require the odorants to evaporate from the liquid phase, travel through the surrounding air space and reach a human nose to be perceived. A way to mimic this process chain is by framing the release and propagation of perfumes as the well-known Stefan tube device (Heinzelmann et al., 1965; Lee and Wilke, 1954). Fig. 1 illustrates it. The core idea is that odorant molecules in liquid *i* (perfume) are able to escape to the gas phase (air) and diffuse through it in the z direction. On top of the tube, a passing gas stream quickly removes any chemical on that point preventing its accumulation in detectable amounts.

In a real perfume experience, transport mechanisms other than molecular diffusion may play an important role in the traveling of odorants through the surrounding air, such as forced or natural convection driven by temperature gradients, for instance. The mathematical model based on molecular diffusion measured with Stefan tube tests was used because it offers a simple and effective tool for evaluating perfume behavior. It has been already validated experimentally in the literature (M. A. Teixeira et al., 2013) serving, therefore, as a solid base for developing the methodology here pro-

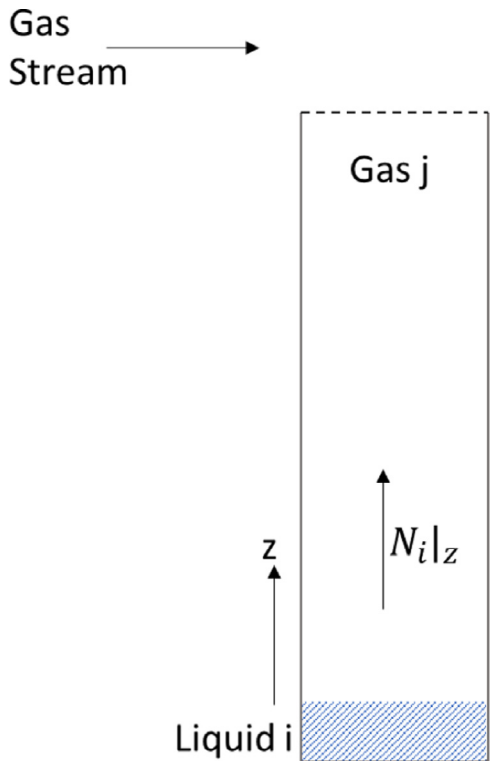

Fig. 1. Illustration of the device for perfume release and propagation study.

posed. In fact, more efforts are necessary from the scientific community to develop and validate models that account for all these transport mechanisms simultaneously.

The model presented below was proposed by Mata et al. (2005a). In the referred work it was identified a first principle model by using mass conservation, Fick's law of diffusion and phase equilibria equations with good agreement between experimental observations and predictions. The model was verified experimentally in M. A. Teixeira et al. (2013). Thus, this model was implemented here to serve as an accurate representation of the perfume release and propagation to generate the database for the empirical model development. It assumes:

- The radial flux in the gas phase is negligible;
- The air is not soluble in the liquid phase nor react with it, besides it can be considered a stagnant gas;
- The gas phase is an ideal gas mixture;
- The mass transfer resistance in the liquid phase is negligible;
- The liquid/gas interface is at chemical equilibrium.

The diffusion for a given chemical component relative to a stationary coordinate z can be expressed as (Bird et al., 1961):

$$N_{i,\,z} = -C_T D_{i,j} \frac{\partial y_i}{\partial z} + y_i \left(N_{i,z} + N_{j,z}\right) \tag{1}$$

Considering species j as the air, which is stagnant, Eq. (1) can be rearranged to:

$$N_{i,\,z} = -\frac{C_T D_{i,j}}{(1-y_i)} \frac{\partial y_i}{\partial z} \tag{2}$$

Thus, the differential material balance of component i can be written as:

$$C_T \frac{\partial y_i}{\partial t} = \frac{-\partial N_{i,\,z}}{\partial z} = -\frac{\partial}{\partial z}\left(-\frac{C_T D_{i,j}}{(1-y_i)} \frac{\partial y_i}{\partial z}\right) \tag{3}$$

By applying the derivative operator on the right-hand side of Eq. (3), it turns out to be:

$$\frac{\partial y_i}{\partial t} = \frac{D_{i,j}\left[\left(\frac{\partial y_i}{\partial z}\right)^2 + (1-y_i)\frac{\partial^2 y_i}{\partial z^2}\right]}{(1-y_i)^2} \tag{4}$$

Eq. (4) describes the relationship of the gas mole fraction with respect to time t and position z. In a similar way, a material balance in the liquid phase can be written considering a sink term as the flux at the liquid/gas interface. It can be written as:

$$\frac{dn_i}{dt} = A_{gl} \frac{C_T D_{i,j}}{(1-y_i)} \frac{\partial y_i}{\partial z}\Big|_{z\,=\,0} \tag{5}$$

To find a unique solution to Eqs. (4) and (5), initial and boundary conditions are necessary. They can be written as:

$$I.C1:\ t = 0,\ 0 < z \leq z_{max} \rightarrow y_i = 0 \tag{6}$$

$$I.C2:\ t = 0,\ z = 0 \rightarrow n_i = n_i^0,\ n_i^0 \in \mathbb{R}_{\geq 0} \tag{7}$$

$$\begin{aligned} &B.C:\ t > 0,\ z = 0 \rightarrow\ y_i = y_i^{eq} = \tfrac{\gamma_i x_i P_i^{sat}}{P} \\ &t > 0,\ z\ = z_{max} \rightarrow y_i = 0 \end{aligned} \tag{8}$$

where z is the axial position, t is the time, $N_{i,z}$ is the molar flux of component i on the z direction. C_T is the total molar concentration, $D_{i,\,j}$ is the diffusion coefficient of component i in components j, y_i is the mole fraction of component i in the gas phase, $N_{j,z}$ is the molar flux of components j in the z direction, n_i is the number of moles of component i, A_{gl} is the liquid/gas interfacial area, n_i^0 is the number of moles in the liquid phase at $t = 0$, γ_i is the activity coefficient of component i, x_i is the equilibrium mole fraction of component i in the liquid phase, y_i^{eq} is the gas phase equilibrium mole fraction corresponding to x_i, P_i^{sat} is the vapor pressure of component i, P is the gas phase pressure.

The activity coefficient (γ_i) quantifies the deviation of components' properties in a mixture from an ideal solution behavior. In simple terms, a solution is considered ideal if its extensive thermodynamic properties are very close to the sum of its parts. High deviations are expected if the mixture is composed of chemically dissimilar parts, which are usually the case of fragranced mixtures. In this way, the well-known UNIFAC method (Poling et al., 2001) was used here to take into account such deviations.

Eqs. (4) to (8) comprise a set of non-linear coupled partial and ordinary differential equations that has to be solved numerically. The numerical solution was implemented in the gPROMs® (Process Systems Enterprise, 2015) software by dividing the z- domain in 200 elements using a second order finite-difference discretization scheme. The time integration was done using the default DASOLV library available in gPROMs®. The parameters P_i^{sat}, $D_{i,j}$, P, A_{gl} were taken from (M. A. Teixeira et al., 2013).

2.2. Odor intensity model

One of the challenges in the design of perfume products is to find precise ways to map changes in physical quantities (increasing the composition of a given component, for example) to changes in the perception of the human olfactory system. In this sense, psychophysics, a scientific field from psychology that investigates how human perception systems respond to physical stimuli, provides valuable tools to help to overcome such difficulties (Teixeira et al., 2010a). Among the contributions of the field for olfactory perception, two models are emphasized the most in the Perfume field: Power Law (PL) and Odor Value (OV). The OV model establishes a linear relationship between the concentration of a component and the corresponding olfactory perception. The PL is inspired by Stevens Psychological law (Stevens, 1957) in which a linear relationship between odor intensity and concentration is present on a logarithmic scale. The advantages and drawbacks of each model are detailed in (Teixeira et al., 2010a).

The power-law model was used here to map odorant concentration to perceived intensity. It can be written as:

$$\Psi = \left(\frac{C_g}{ODT}\right)^n \tag{9}$$

where C_g is the concentration of an odorant in the gas phase, ODT is the odor detection threshold, which is the minimum amount required for a human nose detects the odorant, and n is the exponent.

The concentration of each odorant at a given distance from the liquid source z, and time t can be obtained by solving the equations described in 2.1.1 which in turn can be mapped to odor intensity Ψ using Eq. (9).

The described PL model defines the odor intensity of pure components in the air. However, a typical perfume formulation incorporates several chemicals that pose a new challenge to define the odor features of perfumes. One approach is characterizing the odor of a mixture using the stronger component model (M. A. Teixeira et al., 2013). It states that the odor intensity of the mixture is the maximum intensity among the components. Furthermore, it provides a qualitative measurement of the odor by stating that the odorant with maximum intensity dominates the odor character and can be recognized by the human nose. It can be mathematically expressed as:

$$\Psi_{\text{mix}} = \max\left(\Psi_i\right),\ i = 1,\ 2, \ldots,\ 3,\ N \tag{10}$$

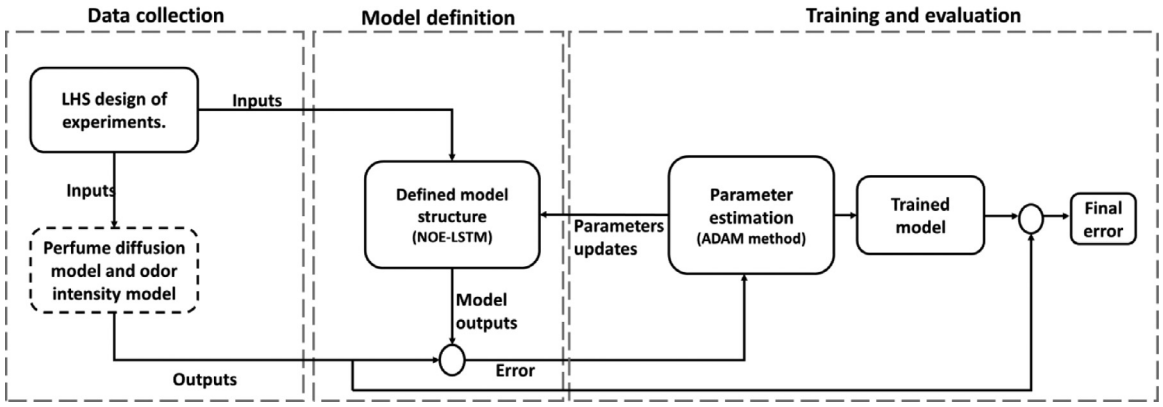

Fig. 2. Diagram of system identification steps.

2.3. System Identification

As previously mentioned, this work proposes the identification of a DNN model to represent with fidelity the perfume release and propagation in space and time. The model identification is a crucial step during the identification of an empirical model. Therefore, it should be carefully done. Thus, the identification process was divided into three steps:

(1) Data acquisition procedure;
(2) Definition of predictor and model structure;
(3) Training and performance evaluation procedure;

The data acquisition step consists of the planning of the experiments that needs to be done to generate the necessary database. The Latin Hypercube Sampling (LHS) method was employed. The details of this step are given in Section 2.2.1. The data set containing the information about the perfume release and propagation were obtained by implementing and simulating first principles models – the diffusion model and psychophysical model for olfactory perception identified in (M. A. Teixeira et al., 2013). The Section 2.1.1 and 2.1.2 detailed the models as well as the perfume release and propagation process under study. Those models will be used as base for the system identification.

The proper definition of the predictor structures to describe dynamic systems is a subject of great interest in the Engineering community. In this work, a non-linear representation of the system dynamics was done using a Non-Linear Output Error (NOE) predictor. The approximating function employed is a Deep Neural Network, specifically, the long short-term memory (LSTM) cell. The NOE concept and LSTM cell are detailed in Sections 2.2.2 and 2.2.3, respectively.

In the third step, the parameters of the model need to be estimated/trained. Here, the least-squares estimation is used. Finally, the final model performance has to be assessed. Section 2.2.4 deals with the training and evaluation procedures. Fig. 2 summarizes the whole system identification procedure.

2.4. Data acquisition procedure

The choice of input training data may have a huge impact on the final model performance. Although a vast literature exists on the design of experiments, none of them offers specific solutions to the huge amount of processes that can be identified by empirical means. Thus, this step requires the analyst to have a specific domain knowledge to choose the most appropriate excitation to the process.

In the present work, we use the Latin Hypercube Sampling (LHS) input design method firstly proposed in (McKay et al., 1979). It is a computer algorithm to generate near-random samples of stratified uni/multivariate data drawn from a specified distribution. According to Helton and Davis, 2003, it incorporates many of the desirable features of random sampling and stratified sampling and also produces more stable analysis outcomes than random sampling. Furthermore, its efficient stratification properties allow covering the physics or explicable part of the process with a smaller sample size than pure random sampling. The parameters of LHS input design are the sampling distribution, the number of random variables, the sample size and the variables' ranges. The variables to be sampled in the present work are the number of moles of odorants per mol of solvent in the liquid. Furthermore, the sample size was chosen here by considering the amount of computational resources to train the model. The choice of variables' ranges considered limits that are useful in the perfume design point of view – pure solvent is an example of meaningless training data in this context.

The other part of data acquisition refers to the collection of outputs, namely which output variables and their sampling rate. In the study of perfume release and propagation, the primary interest is to have samples of odorants intensities over a long-time horizon and at different distances from the liquid source to characterize the perfume performance. Thus, the chosen outputs in the present work are the dynamics of odorants' intensities at several distances from the liquid source. In this work, the sampling rate was set small enough to capture the whole dynamic trajectory of the variables.

The inputs were designed using the *lhsdesign* function from Matlab 2019b. The values were stored in an array and passed as input data to the perfume diffusion and odor intensity models implemented in gPROMs® through the Go:Matlab® communication software. After every 5 minutes of process simulation time, gPROMs® sent back to Matlab the odor intensity of each odorant at several distances from the perfume source.

2.5. Predictor and model structure

The variables in dynamic systems, such as the process of perfume release and propagation, are generally represented as time series. It means that the output observations on a given time depend on past observations of either input or output. However, traditional machine learning regression algorithms such as linear regression, support regression machines, kernel ridge regressors, artificial neural networks seek to find a relation between features and the targets regardless of the order of data appearance. In fact, most of the supervised learning algorithms shuffle the data at every iteration to hasten the training.

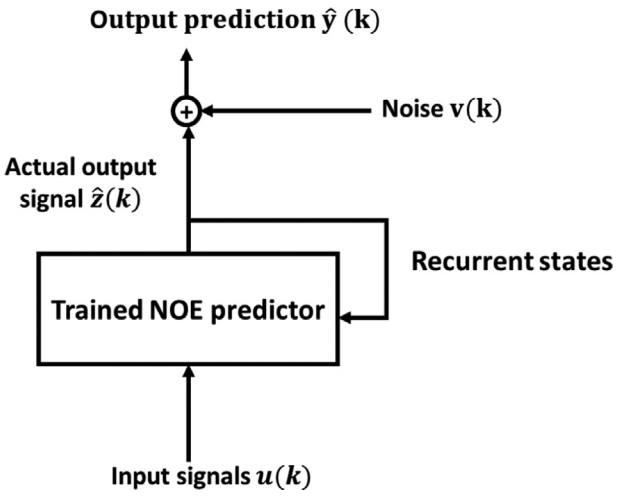

Fig. 3. Illustration of NOE predictor.

Thus, before using a machine learning model, it is necessary to consider a data structure, which will introduce time dependencies in such models. It can be done considering lagged input and/or output signals as regressors and additive noise. Two classes of non-linear predictors are emphasized the most in the field of system identification: Non-Linear AutoRegressive Model with Exogeneous Inputs (NARX) and Non-Linear Output Error (NOE) (Nogueira et al., 2018). An extensive discussion about each of them can be found in (Vardanega, 2011). In general lines, the NOE predictor is more appropriate for simulation purposes, which is the case of the present work. Simulation here means that after training, the model is capable to output accurate values of a process dynamic behavior without relying on measured past observations of the output signal. It means that predictions in the unknown future solely depends on input signals and past predictions. This is also known as a recurrent structure. In mathematical terms, this can be represented as:

$$\begin{aligned}\hat{\mathbf{z}}(\mathrm{t}) &= \mathbf{f}\left[\hat{\mathbf{z}}(\mathrm{k}-1),\ldots,\hat{\mathbf{z}}(\mathrm{k}-1-\mathrm{n_a}),\ \mathbf{u}(\mathrm{k}-\mathrm{d}),\ldots,\mathbf{u}(\mathrm{k}-\mathrm{d}-\mathrm{n_b}+1),\theta\right]\\ \hat{\mathbf{y}}(k) &= \hat{\mathbf{z}}(\mathrm{k})+\mathrm{v}(\mathrm{k})\end{aligned} \tag{11}$$

where $\hat{\boldsymbol{z}}(k)$ is the model output prediction, $\hat{\boldsymbol{y}}(\mathrm{k})$ is the output, $u(k)$ is the input signal, $\mathrm{v}(k)$ is the white noise at time k. $\boldsymbol{f}$ is a non-linear parametrized, θ, function, d is the time delay, $\mathrm{n_b}$ and $\mathrm{n_a}$ are the order regressors. Fig. 3 illustrates it.

2.6. Long short-term memory

The recurrent non-linear function approximator here used is the long short-term memory (LSTM), which is used in a deep neural network (DNN) model structure. This structure, first proposed by (Hochreiter and Schmidhuber, 1997), is able to learn long-time-lag sequence tasks. It was conceived to solve the vanishing gradient problems by enforcing constant error flow through multiplicative gates.

The LSTM computational cell has two variables that incorporate past information in new predictions. These variables are called the hidden states. At first, the hidden states are initialized at random or with zero values and, at each time-step computation, they are updated using trainable weight matrices and non-linear functions. Three gates control the updates of these two variables: the update gate, forget gate and output gate. The gates also have trainable weight matrices that define how much information is forward passed over the time steps computations.

To illustrate the LSTM structure, the equations for a single forward-time computation of one example can be written as:

$$\tilde{c}(k) = \boldsymbol{f_c}(W_{ca}a(k-1) + W_{cx}X(k) + b_c) \tag{12}$$

$$\Gamma_f = \sigma\left(W_{fa}a(k-1) + W_{fx}X(k) + b_f\right) \tag{13}$$

$$\Gamma_u = \sigma\left(W_{ua}a(k-1) + W_{ux}X(k) + b_u\right) \tag{14}$$

$$c(k) = \Gamma_u * \tilde{c}(k) + \Gamma_f * c(k-1) \tag{15}$$

$$\Gamma_o = \sigma\left(W_{oa}a(k-1) + W_{ox}X(k) + b_o\right) \tag{16}$$

$$a(k) = \Gamma_o * \boldsymbol{f_a}(c(k)) \tag{17}$$

where $*$ is element-wise matrix multiplication, $X(k)$ is the input signal at the time step k. $c(k)$ and $a(k)$ are the hidden states at time step k. The hidden terminology means they generally do not carry the information of main interest but are intermediate values in the computational process. $\tilde{c}(k)$ can be interpreted as the variable that brings fresh information from the current time step k. The update and forget gates Γ_u, Γ_f use the sigmoid function (σ) to decide how much past and new information is incorporated in the new value of $c(k)$ by using Eq. (15). The output gate Γ_o updates the a state by incorporating information from the updated c and past a with Eq. (17). W_{ij} and b_i are the trainable weight matrices and biases respectively, $\boldsymbol{f_i}$ are the non-linear activation functions.

The open-source deep learning programming framework TensorFlow (Abadi et al., 2016) was used in this work. It operates under the high-level Keras API (Chollet, 2015) and counts with highly optimized implementations of recurrent structures, including LSTM.

Choosing an artificial neural network architecture is an iterative process. Even though guidelines exist, the most successful applications of neural networks generally involve trial and error. A common practice is starting with the simplest possible architecture and embodying complexity as more accuracy is needed and both data and computational resources become available. Another important aspect is choosing an appropriate predictor for a problem. In this sense, using a recurrent structure for time series simulation is a reasonable choice. Thus, the present work proposes a two-layer DNN structure. The first layer is the LSTM structure with hyperbolic tangent activation. The second is a fully-connected layer with linear activation.

As the main interest is knowing the dynamics of odorants' intensities at different distances from the liquid source, several two-layer neural networks were trained in a way that each can learn the dynamics at a given distance from the perfume source. Thus, it was built as a multiple- sensor structure to assess the evaluation through time and space of the perfume diffusion. The overall sensors' structure is illustrated in Fig. 4.

2.7. Training and evaluation

As it is possible to see from Eqs. (12)-(17), the DNN model has several parameters to be estimated. The estimation of those parameters constitutes the artificial neural network training. Using the outputs of the system, namely the odorants' intensities, which are measured in a continuous numerical scale it is possible to apply the Euclidean distance as fitting criteria in a least-squares estimation function. Thus, the model is fitted when the optimization method finds a set of trainable parameters that minimize the average of the loss computed across the training set. This average loss

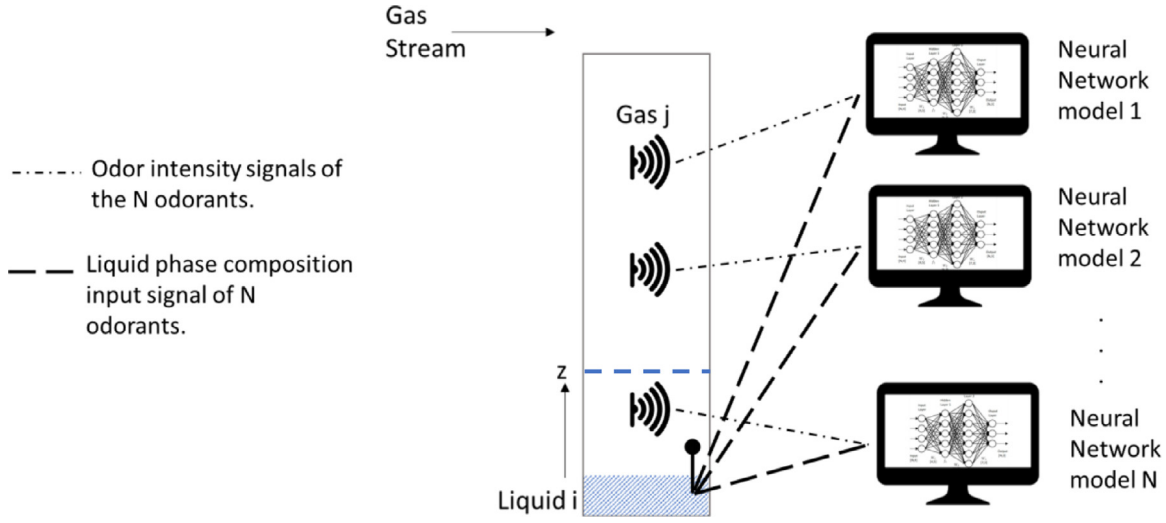

Fig. 4. Overall predictor structure of perfume release and propagation.

is also named the cost function, in this case, the mean squared error. In this case, the cost function is given by:

$$J(W_{ij}, b_i) = \frac{1}{2m}\sum_{i=1}^{m}\sum_{j=1}^{n_y}\sum_{k=1}^{T_y}\left(\hat{y}_{i,j,k}(W_{ij}, b_i, X_{i,k}) - y_{i,j,k}\right)^2 \tag{18}$$

In Eq. (18), i varies from 1 to the number of training examples m. j varies from 1 to the number of output features n_y and k varies from 1 to the number of output time-steps T_y. $W_{i,j}$ and b_i are the trainable parameters and $X_{i,k}$ is the i^{th} input signal at time step k.

The majority of the popular deep learning frameworks like TensorFlow train the models with gradient-based optimizers by using back-propagation. Among the optimizers, the adaptive moment estimation (Adam) (Kingma and Ba, 2015) have gained great acceptance in the deep learning community by speeding up several fold the learning process. Thus, this method was used here.

Several parameters of the optimization algorithm control the final fitting such as the learning rate, learning rate decay, number of epochs and mini-batch size. All the mentioned parameters are not trained during cost the function optimization but guessed before training. They are usually called hyperparameters.

Although statistical methods for hyperparameter tuning exist, the most common approach is to use random grid search. It consists of selecting a finite number of values for each hyperparameter, finding all possible combinations, randomly choosing a subset, training the model and finding the corresponding values with the best performance. This can be done several times up to a point one has to manually decide among a few options. The random grid search was used in the present work to search the initial learning rate, learning rate decay and mini-batch size that achieve the lowest mean squared error. It is important to mention that the proposed hyperparameter tuning assesses which combination has best performance in a data set not used during the cost function optimization step – the validation set.

The number of neurons is a parameter that defines the artificial neural network architecture and has a significant influence on the model performance. It deserves a separate explanation for not being part of the optimization algorithm but of the model architecture. In a similar way, it needs to be defined before the identification method. In non-linear representations like artificial neural networks, increasing the number of neurons of a layer means increasing the number of learnable parameters. Ideally, one can increase it up to the number of training examples which would ideally lead to a perfect model fitting. However, fitting a model with too many parameters would cause it to lose predicting capacity on data out of the training set (overfitting). Thus, the tuning of the number of neurons in layers must consider the existence of a performance gap between training data and non-training data, which is an indication of overfitting. Regularization is the most common method to tackle overfitting problems and was used here. Specifically, the early-stopping regularization. The optimal number of neurons in the hidden layer was embodied as part of the random grid search described in the previous paragraph. Finally, the final model performance was assessed by calculating the mean squared error on the same data set in which the hyperparameters were tuned.

The predictor structure, the training process and the hyperparameters tuning were implemented using TensorFlow on the Google Colaboratory environment with Python kernels. Google Colaboratory is a free serveless jupyter notebook environment for prototyping machine learning models on powerful hardware options such as graphical process units (GPUs) and tensor process units (TPUs) (Bisong and Bisong, 2019).

2.8. Perfume performance optimization

The present work uses a model-based optimization with a surrogate model-based approach using a trained DNN to provide accurate prediction of the system. It seeks to find the optimal composition of a perfume that will result in a specific fragrance while minimizing the solvent odor. The optimal composition is defined by the optimization problem (objective function and constraints), which is solved by the optimization algorithm. The optimized variables are the weighted sum of odor intensities across time and space. In other words, it is the accumulated effect of each odorant odor intensity over time.

In line with the idea of maximizing the odor effect of desired odorants, the objective function that we seek to minimize has the following form:

$$\min_{n} F_{obj}\ (n_1, n_2, n_3, \ldots, n_N)$$

$$F_{obj}\ (n_1, n_2, n_3, \ldots, n_N) = \frac{1}{(\Delta x \Delta t)}\sum_{\substack{i,\,j=1 \\ i \neq j}}^{N_o}\int_{t_{min}}^{t_{max}}\int_{x_{min}}^{x_{max}} w_{i,i}\Psi_i(t, x, n_1, n_2, \ldots n_N)dxdt + w_{i,j}\left|\Psi_i\ (x, t, n_1, n_2, \ldots n_N) - \Psi_j(x, t, n_1, n_2, \ldots n_N)\right|dxdt \tag{19}$$

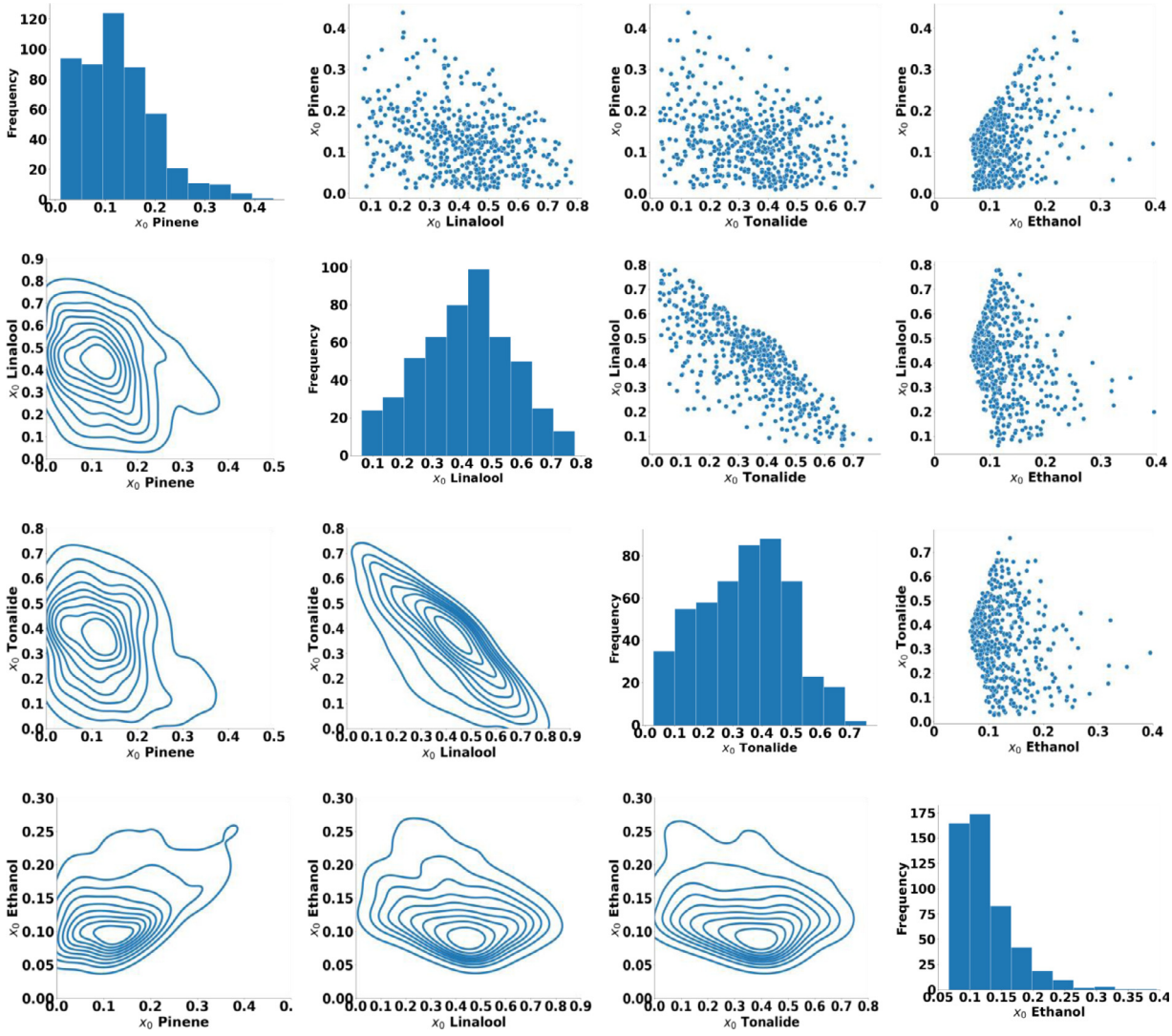

Fig. 5. Distributions of mole fraction inputs.

Subject to:

$$n_1^l < n_1 < n_1^u; n_2^l < n_2 < n_2^u; \ldots; n_N^l < n_N < n_N^u$$

The objective function is a weighted sum of odor intensity Ψ_i integrals in the spatial dimension z and time t of each odorant i which is sought to compose the perfume aura, N is the total number of components, N_o is the number of odorants, n_k is the number of moles per mol of the solvent of component k. (N_o can be lower than N since perfume formulations can contain odorless components). The integrals in the objective function are numerically approximated by using the trapezoidal rule.

It is important to note in Eq. (19) that $w_{i,i}$ and $w_{i,j}$ are weight matrices that can assume integer values. They work as a quantitative representation of how desirable or undesirable is the odor of a given component i. They will also ponder how desirable is an odor intensity to be equal to another, which is represented in the absolute deviation between the odorants j and i. Finally, if $w_{i,i}$ is high and positive, for instance, the optimization algorithm will avoid formulations that cause an intense smell of the corresponding component i across time and space, which is the case of the alcohol for example. On the other hand, if $w_{i,i}$ is high and negative, the optimization algorithm search for formulations that lead to an intense odor of the corresponding component i across time and space.

The constraints n_i^l and n_i^u are the lower and upper limits of the decision variables. They coincide with the ranges on which the artificial neural networks were trained. Allowing an optimization algorithm to search on regions outside these limits may cause inaccurate evaluations of the odorants' intensities and the objective function.

The employed algorithm used to maximize the objective function (19) was the Particle Swarm Optimization (PSO). The implementation of the optimization using PSO was done using the research toolkit PySwarms (James and Miranda, 2018) on the Google Colaboratory environment, using Python language. It offers a black-box approach for solving optimization tasks with few lines of code yet allows a flexible API for non-standard swarm models. Here we used the classic global PSO.

3. Results and discussion

3.1. Data acquisition

A quaternary perfume formulation with pinene, linalool, tonalide, and ethanol was considered here as the study case, whose formulation is based on the one presented in Teixeira et al., 2013. They formed a simple set of components that play the role of top, middle, base notes and solvent, respectively. This formula-

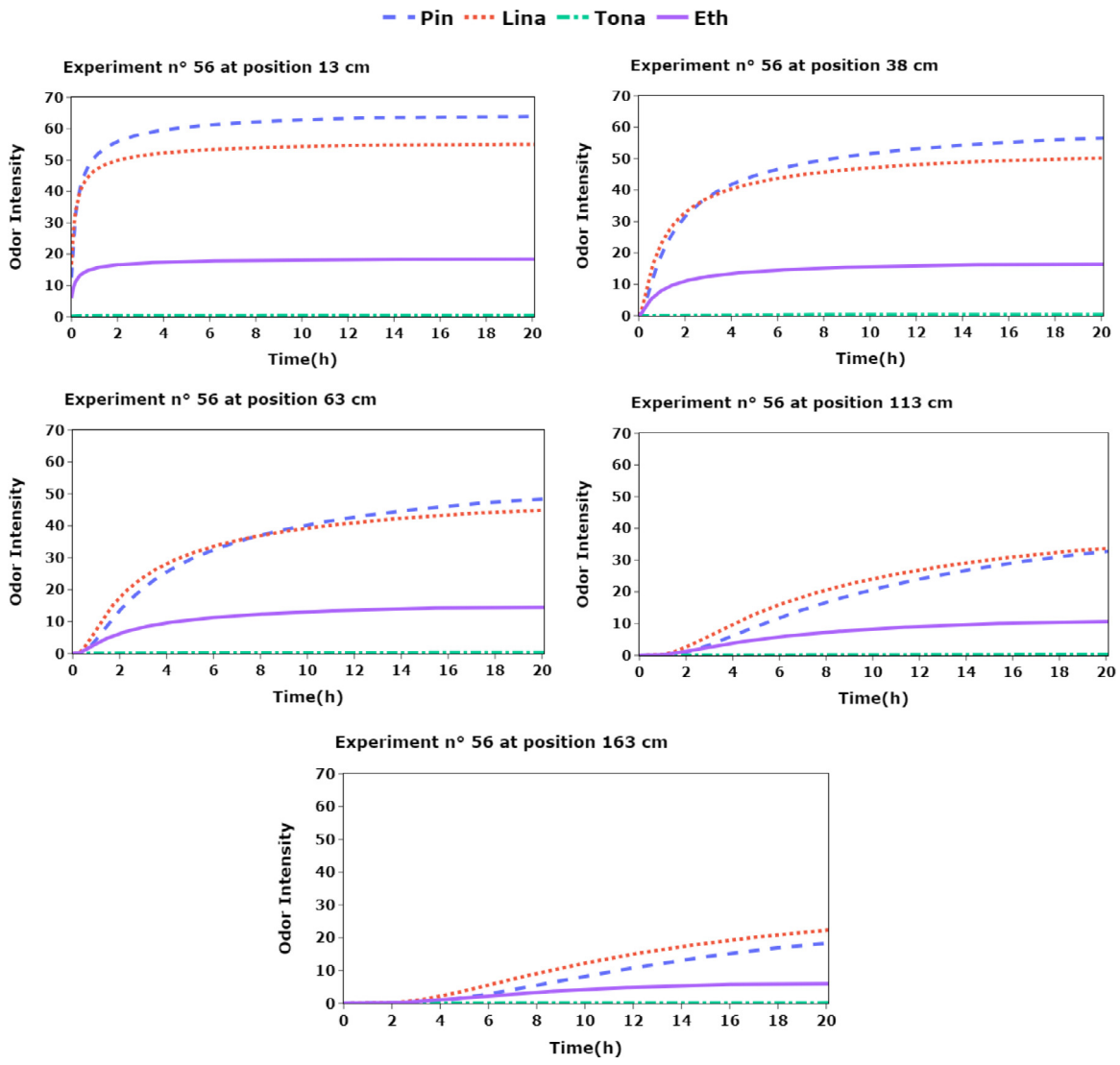

Fig. 6. Time evolution of odorant's intensities at five distances from the liquid perfume source for experiment number 56.

Table 1
Mole fraction ranges.

Component	Class	Minimum	Maximum
Pinene	Top note	0.01	0.44
Linalool	Middle note	0.06	0.78
Tonalide	Base note	0.03	0.76
Ethanol	Solvent	0.07	0.40

tion allows to evaluate the methodology proposed in this work while comparing the optimized formulation with Teixeira et al., 2013 study. with some margin to make the model flexible for optimization.

Table 1 shows the mole fraction ranges of pinene, linalool and tonalide in the liquid phase used in the input design. The ranges were chosen respecting a heuristic for perfume formulation in which top note should comprise 15% — 25%, middle 30% — 40% and base 45% —55% (Mata et al., 2005a) with some margin to make the model flexible for optimization.

An experiment consists of a set of initial mole fraction of the liquid phase. For the present case, 500 sets of moles fractions were generated using Matlab *lhsdesign* function with ranges displayed in Table 1. Therefore, the LHS defined 500 experiments to be done.

The univariate and pairwise distributions of the mole fractions are displayed in Fig. 5. The diagonal contains the histograms. The upper corner contains pair-wise scatter plots, and the lower contains contours pair-wise kernel density plots. It shows that some of the mole fractions distributions are skewed, especially the ethanol. Furthermore, the mole fractions are correlated at a certain extent, which is expected since the sum of mole fractions in each sample must result in one.

After doing the design of experiments, the samples were taken using the phenomenological model as a virtual plant, implemented in gPROMs®, which was accessed through Matlab in a software-in-the-loop (SIL) approach. Here, our analysis considered the evolution of odor intensities over time in five distinct positions along the tube: (13, 38, 63, 113 and 163) centimeters. These values coincide with the positions that Teixeira et al. (2013) used to collect samples from their experimental setup. Furthermore, 600 samples of odorants' intensities were collected at a 5-minute measurement frequency for each of the 500 experiments done. It resulted in a total of 300,000 data points generated for system identification.

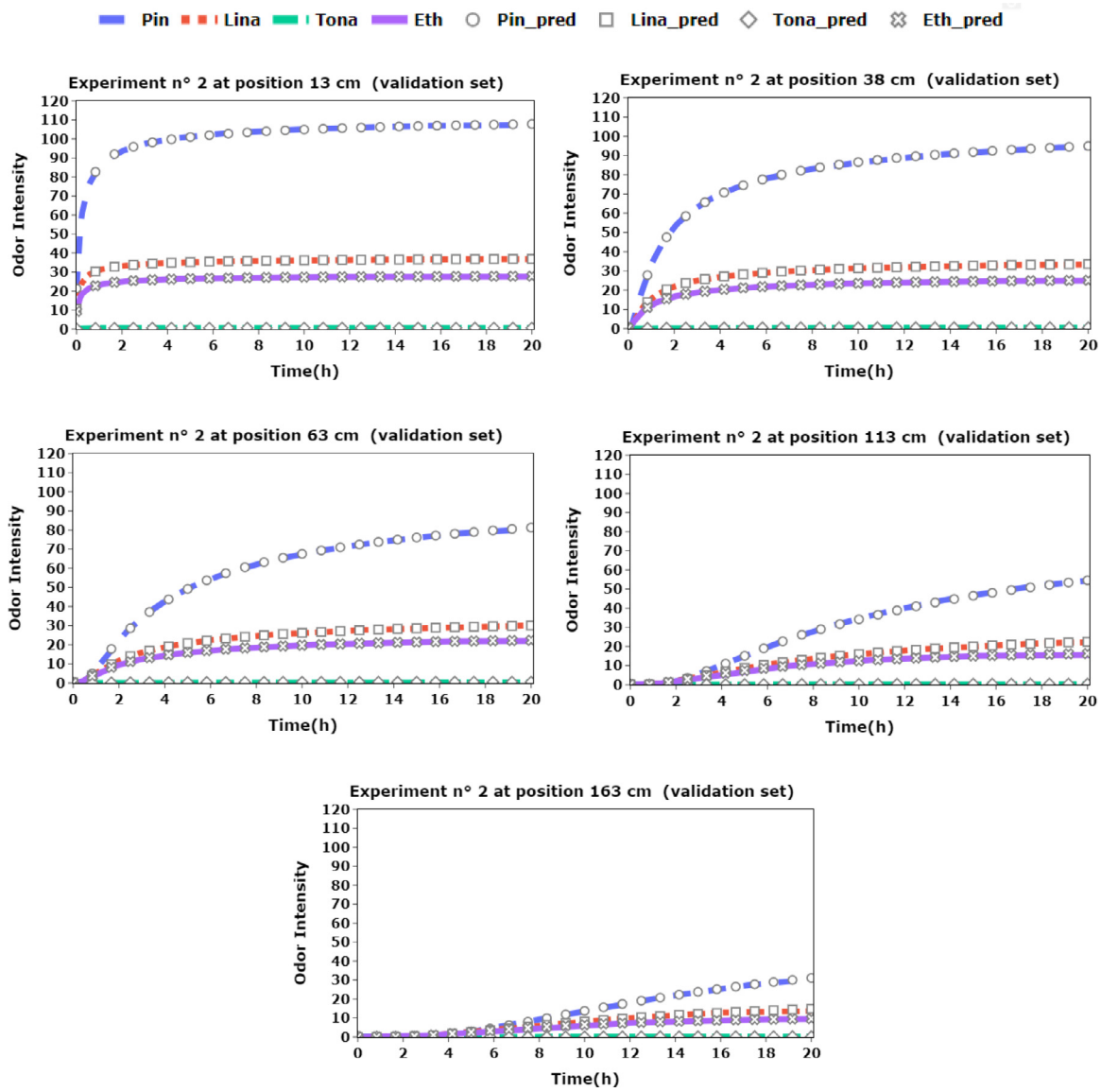

Fig. 7. DNN models validation.

Fig. 6 illustrates the time evolution of odorants' odor intensities at five distinct positions in the tube for one of the 500 experimental conditions generated by LHS. In this experiment, the initial compositions of the liquid phase were: 0.02 - pinene, 0.70 - linalool, 0.17 - tonalide and 0.11 - ethanol. It is possible to see that tonalide has the lowest odor intensity among odorants. This is verified in all 500 experiments since tonalide has a considerably smaller vapor pressure than the other components. However, it plays an important role in the formulation by acting as a fixative.

3.2. Training and evaluation

As mentioned, identifying a machine learning model is an iterative process. According to the described methodology, five DNNs were trained to predict the dynamic of odor intensities at each distance from the perfume source. First, the available data from data acquisition were split into two separate sets – 400 experiments from training and 100 for hyperparameter estimation and performance evaluation.

One common and useful step before training is data preprocessing. The inputs (also named features) in typical machine learning applications are measured in different magnitude order, this difference can affect the training performance and model quality. They were scaled by using maximum and minimum values from the training set. In this way, in the worst case, scaling does not affect the training speed. Similarly, the outputs were also scaled to lie between 0 and 1 by estimating maximum and minimum values from the training set. As Fig. 6 shows, the odor intensities of the components at a given distance may have very distinct ranges. In this case, it can cause vanishing or exploding gradients and make the training unfeasible. The methodology previously described was followed in order to find a combination of hyperparameters that result in the minimum mean squared error. Table 2 shows the overall search space of the hyperparameters: initial learning rate, learning rate decay factor, neurons and mini-batch size.

This small set of values leads to a total of 750 distinct combinations. Evaluating all the possibilities is time-consuming and ineffective – one would end up training several models to discover only in the end that just a few combinations lead to satisfactory results. In this way, only 5% of all combinations were equally likely drawn without replacement. This resulted in 40 combinations of hyperparameters for each of the five DNNs – 200 train procedures in total.

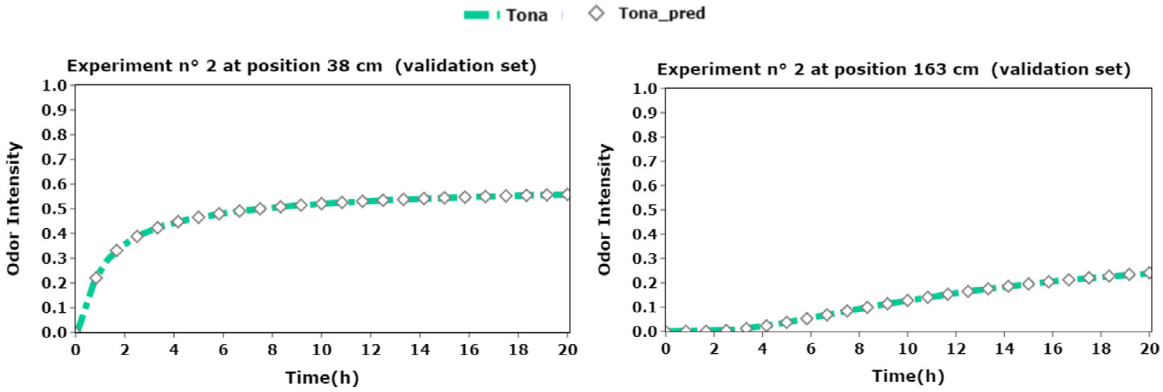

Fig. 8. DNN models validation for tonalide.

Table 2
Hyperparameter search space.

Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size
5.00×10^{-5}	5.00×10^{-1}	20	1
1.58×10^{-4}	3.33×10^{-1}	40	2
5.00×10^{-4}	2.50×10^{-1}	60	8
1.58×10^{-3}	1.67×10^{-1}	80	32
5.00×10^{-3}	–	–	–

Table 3
Results of hyperparameter tuning with random grid search.

DNN for distance 13 cm					
Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size	Validation mse	Rank
5.00×10^{-3}	2.50×10^{-1}	80	2	1.19×10^{-6}	1
5.00×10^{-3}	1.67×10^{-1}	80	1	1.85×10^{-6}	2
5.00×10^{-3}	2.00×10^{-1}	60	1	2.04×10^{-6}	3
DNN for distance 38 cm					
Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size	Validation mse	Rank
5.00×10^{-3}	2.50×10^{-1}	80	2	1.20×10^{-6}	1
5.00×10^{-3}	2.00×10^{-1}	60	1	2.06×10^{-6}	2
5.00×10^{-3}	1.67×10^{-1}	80	1	2.10×10^{-6}	3
DNN for distance 63 cm					
Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size	Validation mse	Rank
1.58×10^{-3}	2.50×10^{-1}	40	2	1.54×10^{-6}	1
1.58×10^{-3}	2.00×10^{-1}	40	2	1.63×10^{-6}	2
5.00×10^{-4}	5.00×10^{-1}	60	2	1.79×10^{-6}	3
DNN for distance 113 cm					
Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size	Validation mse	Rank
5.00×10^{-4}	3.33×10^{-1}	80	2	1.65×10^{-6}	1
1.58×10^{-3}	5.00×10^{-1}	60	1	1.80×10^{-6}	2
5.00×10^{-3}	2.50×10^{-1}	80	2	1.97×10^{-6}	3
DNN for distance 163 cm					
Initial learning rate	Learning rate decay ratio	Neurons in the hidden layer	Mini-batch size	Validation mse	Rank
5.00×10^{-3}	2.50×10^{-1}	80	2	1.43×10^{-6}	1
1.58×10^{-3}	5.00×10^{-1}	60	1	1.50×10^{-6}	2
5.00×10^{-3}	1.67×10^{-1}	80	1	1.65×10^{-6}	3

*mse is the mean squared error.

The networks were trained for 350 epochs with early-stopping regularization and patience 30 to prevent overfitting. Furthermore, the learning rate scheduler comprised a maximum of 10 discrete drops with the several decay ratios shown in Table 2.

Table 3 shows the results of hyperparameters tuning for combinations with three best performances for each of the five networks. A considerable improvement was attained with a maximum mean squared error of 1.65×10^{-6} for the five DNNs. Furthermore, the results showed that 80 neurons in the LSTM layer and the mini-batch size 2 comprise the best combination for these two parameters. The best combination of initial learning rate and learning rate decay ratio is less clear, but 5×10^{-3} for the first and 2.50×10^{-1} for the second seems the most promising ones.

Once the optimal number of neurons and hyperparameters were found, all the DNNs were retrained with the best hyperparameters to be used in the perfume performance optimization. Fig. 7 shows a visual representation of the fitting quality achieved by the DNNs for a given experiment in the validation set. The lines represent the predictions of perfume diffusion and odor intensity models implemented in gPROMs® and markers represent the out-

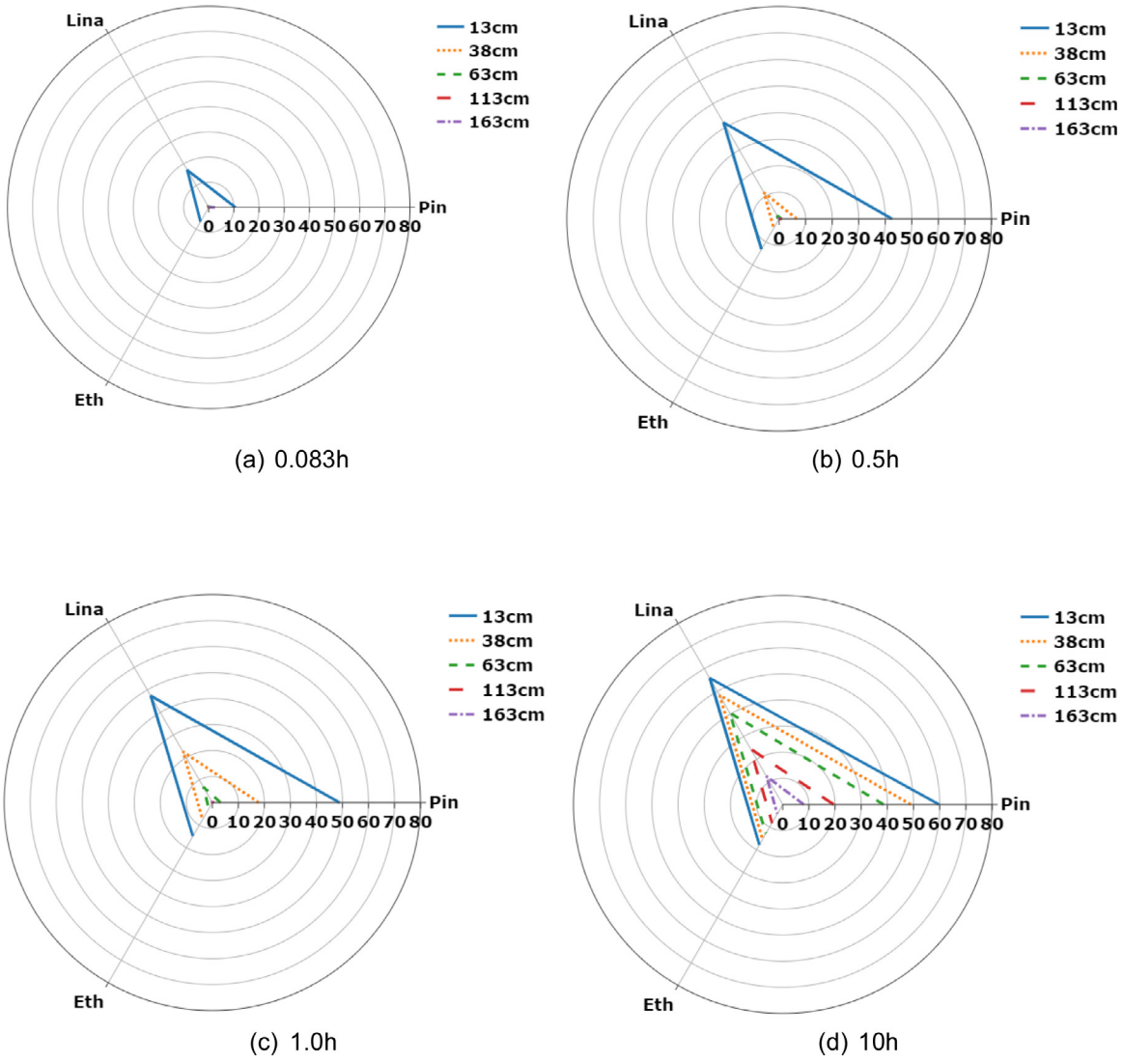

Fig. 9. Odor intensity radar plot for optimized perfume performance.

put of the DNNs. The fitting quality was evaluated with the validation set (data used for hyperparameter tunning), and a separate test set would be required for a truly independent model evaluation. However, if the data come from a distribution similar to the one it was trained, the performance would probably be as high as the presented here and good enough for the optimization purpose.

In Fig. 7, it can be seen a good fitting for pinene, linalool and ethanol. However, the fitting quality is not as clear for tonalide due to the very distinct scales. Thus, a separate visual is displayed in Fig. 8 for two of the distances (38 and 163 cm) to show that a similar predicting performance is achieved for tonalide.

One of the advantages of using a DNN as a surrogate model in the optimization step is the required CPU time to simulate the whole dynamic of odorants' intensities. While gPROMs® takes 6s to simulate one experiment, TensorFlow 2.3 takes about 0.16 ms on an Intel i7-7500U, 2.7 GHz CPU with little sacrifice on prediction accuracy (the validation mean squared error is around 1.00×10^{-6}). For instance, running 500 experiments in the phenomenological model takes 50 min, while using the DNN models it takes 0.08 s. Considering an optimization scenario where tens of thousands of experiments are necessary it becomes a noticeable difference. If the optimization were done with the high-fidelity model, it would take about 22 h, while with the surrogate model, it took about 20 min. Therefore, if the goal is designing fragrances on-demand in real-time, as it is in the present work, using the high-fidelity model would be unfeasible. In contrast, the data-driven approach is very time-demanding at the training stage, but predictions are delivered quickly and accurately afterward.

3.3. Perfume performance optimization

As explained in the methodology, the proposed objective function and constraints in Eqs. (19) require input weights that reflect how desirable is the odor of the fragrances and their relative discrepancy. In the present work, one of the main objectives is reducing ethanol odor intensity so it does not dominate the odor characteristic, since it is irritant and not appreciated by customers (M. A. Teixeira et al., 2013). Moreover, it is also desired to maximize the odor intensities of both top and middle notes (pinene and linalool), yet not allowing too much discrepancy between them. Even though it is an easy task for experienced perfumers to balance fragrances with ethanol, we need to explicitly define that the ethanol odor intensity does not surpass fragrances intensities in our proposed approach. This is also a way to introduce the expert knowledge into a mathematical problem, as a constraint of the optimization, leading to a flexible methodology that provides a synergy between experts and engineering.

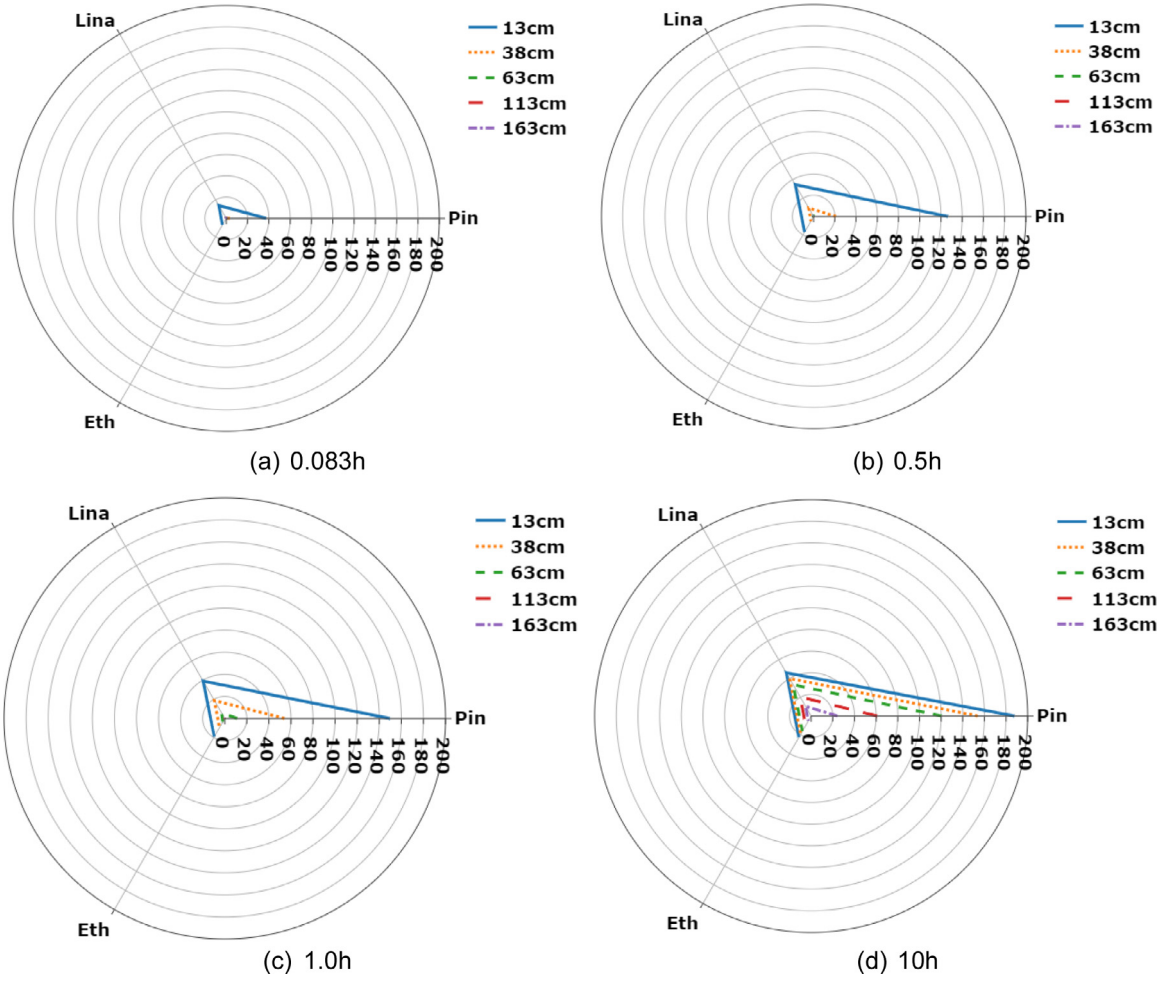

Fig. 10. Odor intensity plot for (M. A. Teixeira et al., 2013) work perfume formulation.

Table 4
Weights for the proposed objective function to optimize perfume performance.

	Pinene	Linalool	Tonalide	Ethanol
Pinene	−1	+10	0	0
Linalool	–	−1	0	0
Tonalide	–	–	−1	0
Ethanol	–	–	–	+1

Table 4 shows the weights that reflect our objectives, which can be expressed as: develop a perfume that will provide an equivalent odor of pinene and linalool through the space and time, while minimizing as much as possible the solvent odor. Even though it is an easy task for experienced perfumers to balance fragrances with ethanol, we need to explicitly define that the ethanol odor intensity does not surpass fragrances intensities in our proposed approach. This is also a way to introduce the expert knowledge into a mathematical problem, as a constraint of the optimization, leading to a flexible methodology that provides a synergy between experts and engineering.

In terms of the proposed optimization problem, it seeks to maximize the odor value of pinene and linalool, keeping their values close to each other and minimizing the odor value of Ethanol. Therefore, producing a perfume with a pine forest and floral scents. By using the objective function in Eq. (19) and the weights in Even though it is an easy task for experienced perfumers to balance fragrances with ethanol, we need to explicitly define that the ethanol odor intensity does not surpass fragrances intensities in our proposed approach. This is also a way to introduce the expert knowledge into a mathematical problem, as a constraint of the optimization, leading to a flexible methodology that provides a synergy between experts and engineering.

Table 4, the optimization was carried out using the PSO algorithm, with a swarm size of 150, particle velocity scaling factor 0.5, scaling factor to search away from particle's best position 0.5, scaling factor to search away from swarm's best position 0.5 and 90 iterations. The values of the decision variables at the optimal point, namely, the number of moles per mole of ethanol are 0.17, 7.00, 0.86 for pinene, linanool and tonalide, respectively. This corresponds to a mole fraction of 0.02, 0.77, 0.09, 0.11 for pinene, linalool, tonalide and ethanol, respectively.

To show the time evolution of optimized perfume performance, Fig. 9 displays several frames of the perfume radar. Each frame is a snapshot of odor intensities over time. The radial axis encodes the odor intensity while the angular axis corresponds to the component names. The polygons connect the components' odor intensities for a given distance – the higher the perimeter of the polygon at a given time, the smallest is the distance from the perfume source. At the beginning of the release (0.083 h), the components have a detectable smell only at distance 13 cm with smell predominance of linalool. At about 0.5 h, only at the distances 13 and 38 cm the odorants have a significant odor intensity – the first with well-balanced linalool-pinene smell and the later with a slight linalool predominance. At 1 h, at distance 63 cm, a blend of linalool, pinene and ethanol smell appears but with dominance

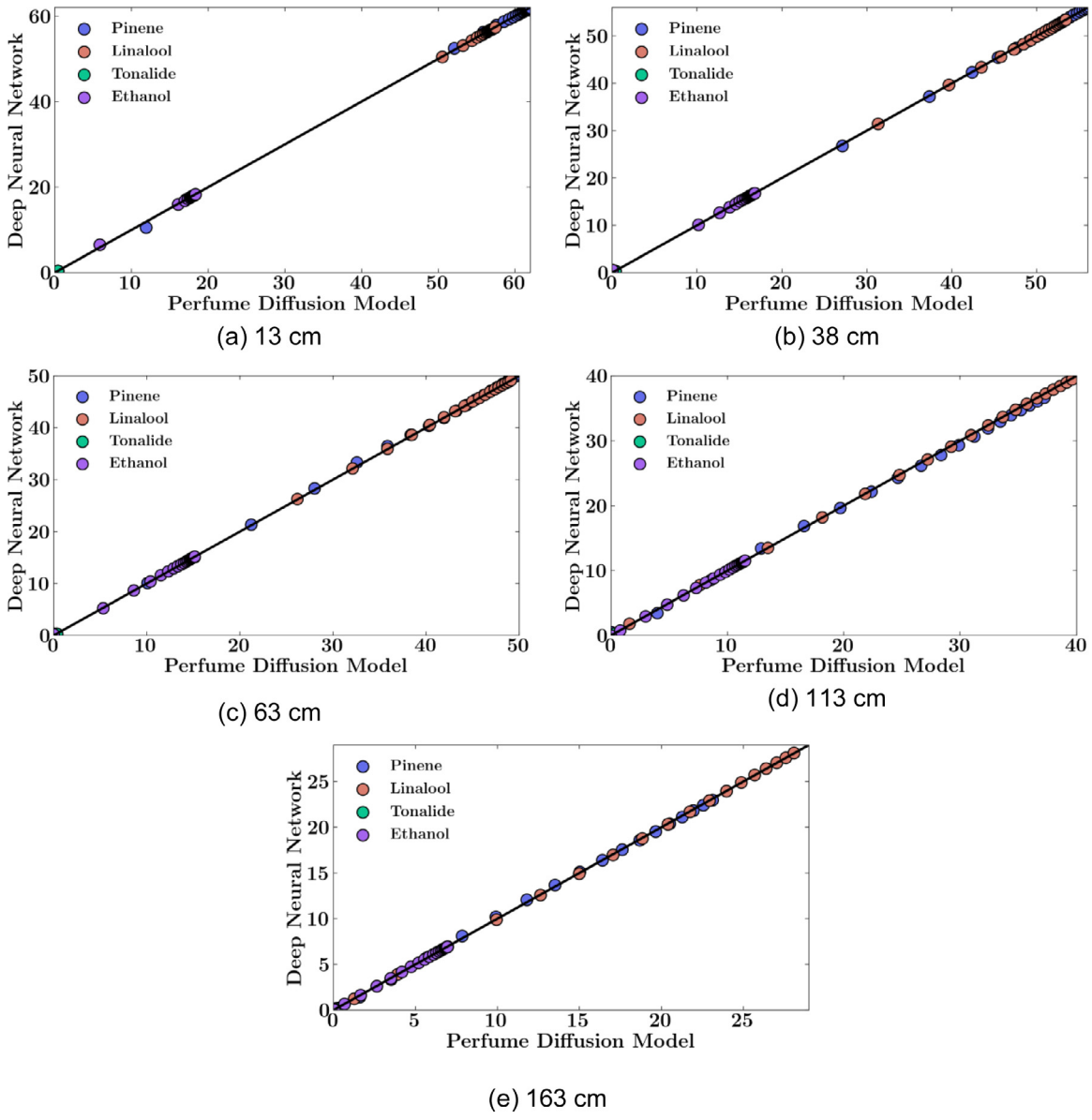

Fig. 11. Different points obtained from the optimization procedure tested in the phenomenological model vs their prediction with the DNN model.

of linalool smell. At distance 38 cm, linalool smell dominance attenuates and at 13 cm, pinene slightly dominates the odor characteristics. At 10 h, all distances have been affected by the perfume with well-balanced blend of pinene and linalool. Fig. 9 shows that the proposed objective function was capable of setting the perfume performance as desired – the ethanol odor intensity is kept far below pinene and linalool which have very close intensities across time and space. Even with relatively large ranges for the algorithm to search, it found the optimal composition at reasonable values. For example, in the optimal formulation found, the composition of tonalide was given at 0.09 molar fraction, within the expected range, even though the problem did not explicitly limit it to that range. Furthermore, it is far below the experimentally tested composition of 0.24 mol fraction in the literature (M. A. Teixeira et al., 2013). The proposed optimization method, however, could easily incorporate constraints for the compositions to meet other requirements such as solubility, stability, among others.

The formulations from Teixeira et al. (2013) with the same four components were used as a reference to compare with the formulation resulting from the present optimization. The referred work aimed to assess the fixative effect of tonalide. Thus, it is important to highlight that this comparison is a mere reference. The mole fractions are 0.27, 0.38, 0.24 and 0.11 for pinene, linalool, tonalide and ethanol, respectively. This composition was given to the DNN models in order to predict the perfume evolution. It is important to note that these mole fractions are within the training set input distribution that makes DNNs' predictions accurate.

Fig. 10 shows the radar for Teixeira's formulation. It can be seen that ethanol odor intensity does not predominate at any of the distances and time frames – pinene has by far the most intense but highly unbalanced compared to linalool odor intensity.

To evaluate the robustness of the optimal point, and consequently, the consistency of the DNN model prediction, different points obtained from the optimization procedure were tested in the phenomenological model, and their prediction was compared with the DNN prediction (Fig. 11). It was observed a negligible difference between the models' predictions, less than 0.1% for each ODT for all components and positions. As it is possible to see in Fig. 11, the predictions of the DNN model with the optimization points are in good agreement with the phenomenological model

prediction. Given that these points are a set of values not included in either the training or test dataset, it is possible to conclude that the result is consistent and the DNN predictions are reliable and robust.

4. Conclusions

Defining, predicting, and designing the performance of fragranced products is still a challenge. Few tools were tested and uncovered in the scientific literature to address such difficulties. In this context, the present work proposes a Deep Learning assisted simulation-optimization approach, identifying a surrogate model that is able to learn the behavior of perfume release and propagation while correlating it with the perception. The identified model was then used for the performance optimization of a perfume formulation, composed by a quaternary mixture. The developed DNN models demand considerably low CPU time to predict the full dynamic behavior of perfume release – 0.16 ms against 6s for the first-principles model.

In this way, a novel systematic methodology for optimal perfume formulation was presented. A single objective function was constructed in a way to encode the peculiarities of a fragrance construction, that comprises the question: Which is the perfume composition to attain the desirable odor spectrum of the perfume across time and space? To find the optimal compositions, the particle swarm optimization (PSO) method was used. It is a flexible algorithm in which calculations are fast and is only limited to the objective function evaluations. Thus, our work proposes the use of a trained recurrent neural network surrogate model since it demands considerably lower CPU time than the first principles model. The PSO method was able to maximize the proposed objective function and found the perfume formulation that precisely reflected our desire: minimize ethanol odor intensity and allow the top and middle notes (pinene and linalool) to be the predominant odors with little discrepancy across time and space provide an equivalent odor of pine forest and floral through the space and time In future works, the perfume description can go to deeper levels since classification as pine, forest, and floral is still very broad in perfumery. Each of these classes may be further sub-divided, as for example, rose-like, jasmin-like, hyacint-like, muguet-like. For that purpose, the perfumer's knowledge needs to be incorporated. Therefore, the methodology presented here is made in a "plug-and-go" strategy, which easily allows the addition of this knowledge in the system.

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.

CRediT authorship contribution statement

Vinícius V. Santana: Formal analysis, Investigation, Software, Writing - original draft, Writing - review & editing. **Márcio A.F. Martins:** Supervision, Resources, Validation. **José M. Loureiro:** Supervision, Resources, Validation. **Ana M. Ribeiro:** Project administration, Supervision, Validation. **Alírio E. Rodrigues:** Supervision, Resources, Validation. **Idelfonso B.R. Nogueira:** Conceptualization, Methodology, Supervision, Writing - review & editing, Validation.

Acknowledgments

This work was financially supported by: Brazilian research agency FAPESB under grant BOL0731/2020, EDUFI Fellowships program, Base Funding - UIDB/50020/2020 of the Associate Laboratory LSRE-LCM - funded by national funds through FCT/MCTES (PIDDAC); FCT – Fundação para a Ciência e Tecnologia under CEEC Institucional program.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. OSDI, p. 2016.

Almeida, R.N., Costa, P., Pereira, J., Cassel, E., Rodrigues, A.E., 2019. Evaporation and permeation of fragrance applied to the skin. Ind. Eng. Chem. Res. 58, 9644–9650. doi:10.1021/acs.iecr.9b01004.

Bird, R.B., Lightfoot, E.N., Stewart, W.E., 1961. Transport phenomena. John Wiley and Sons, Inc., New York AIChE Journal doi:10.1002/aic.690070245.

Bisong, E., Bisong, E., 2019. Google Colaboratory, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform. https://doi.org/10.1007/978-1-4842-4470-8_7.

Bushdid, C., Magnasco, M.O., Vosshall, L.B., Keller, A., 2014. Humans can discriminate more than 1 trillion olfactory stimuli. Science (80-.) 343, 1370–1372. doi:10.1126/science.1249168.

Chollet, F., 2015. Keras: The Python Deep Learning library. Keras.Io.

FC. Cuadros Bohorquez, J., Plazas Tovar, L., Wolf Maciel, M.R., Melo, D., Maciel Filho, R., 2020. Surrogate-model-based, particle swarm optimization, and genetic algorithm techniques applied to the multiobjective operational problem of the fluid catalytic cracking process. Chem. Eng. Commun. 207, 612–631. doi:10.1080/00986445.2019.1613230.

Frazier, P.I., 2018. A tutorial on bayesian optimization. arXiv 1–22.

Heinzelmann, F.J., Wasan, D.T., Wilke, C.R., 1965. Concentration profiles in Stefan diffusion tube. Ind. Eng. Chem. Fundam. doi:10.1021/i160013a009.

Helton, J.C., Davis, F.J., 2003. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab. Eng. Syst. Saf. 81, 23–69. doi:10.1016/S0951-8320(03)00058-9.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural. Comput. doi:10.1162/neco.1997.9.8.1735.

IBM Research and Symrise are working together - Symrise [WWW Document], 2018. URL https://www.symrise.com/newsroom/article/breaking-new-fragrance-ground-with-artificial-intelligence-ai-ibm-research-and-symrise-are-workin/ (accessed 3.19.21).

James V. Miranda, L., 2018. PySwarms: a research toolkit for particle swarm optimization in python. J. Open Source Softw. doi:10.21105/joss.00433.

Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Lee, C.Y., Wilke, C.R., 1954. Measurements of vapor diffusion coefficient. Ind. Eng. Chem. doi:10.1021/ie50539a046.

Leffingwell & Associates, 2018. Flavor & Fragrance Industry - Top 10 [WWW Document]. 2018.

Mata, V.G., Gomes, P.B., Rodrigues, A.E., 2005a. Engineering perfumes. AIChE J. 51, 2834–2852. doi:10.1002/aic.10530.

Mata, V.G., Gomes, P.B., Rodrigues, A.E., 2005b. Perfurmery ternary diagrams (PTD): a new concept applied to the optimization of perfume compositions. Flavour Fragr. J. 20, 465–471. doi:10.1002/ffj.1590.

McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21, 239–245. doi:10.1080/00401706.2000.10485979.

Miguel, L., Oliveira, C., Koivisto, H., Iwakiri, I.G.I., Loureiro, J.M., Ribeiro, A.M., Nogueira, I.B.R., 2020. Modelling of a pressure swing adsorption unit by deep learning and artificial Intelligence tools 224. https://doi.org/10.1016/j.ces.2020.115801.

Nogueira, I.B.R., Martins, M.A.F., Requião, R., Oliveira, A.R., Viena, V., Koivisto, H., Rodrigues, A.E., Loureiro, J.M., Ribeiro, A.M., 2019. Optimization of a true moving bed unit and determination of its feasible operating region using a novel sliding particle swarm optimization. Comput. Ind. Eng. doi:10.1016/j.cie.2019.06.020.

Nogueira, I.B.R., Ribeiro, A.M., Requião, R., Pontes, K.V., Koivisto, H., Rodrigues, A.E., Loureiro, J.M., 2018. A quasi-virtual online analyser based on an artificial neural networks and offline measurements to predict purities of raffinate/extract in simulated moving bed processes. Appl. Soft Comput. J. 67, 29–47. doi:10.1016/j.asoc.2018.03.001.

Nogueira, I.B.R., Viena, V., Rodrigues, A.E., Loureiro, J.M., Ribeiro, A.M., 2020. Chemical engineering & processing : process Intensi fication dynamics of a true moving bed reactor : synthesis of n-propyl propionate and an alternative optimization method. Chem. Eng. Process. Process Intensif. 148, 107821. doi:10.1016/j.cep.2020.107821.

Poling, B.E., Prausnitz, J.M., O'Connell, J.P., 2001. The properties of gases and liquids, Fifth Edition. AIChE Journal doi:10.1002/aic.690240634.

Process Systems Enterprise, 2015. https://www.psenterprise.com/products/gproms [WWW Document].

Sanchez-Lengeling, B., Wei, J.N., Lee, B.K., Gerkin, R.C., Aspuru-Guzik, A., Wiltschko, A.B., 2019. Machine learning for scent: learning generalizable perceptual representations of small molecules [WWW Document]. URL http://arxiv.org/abs/1910.10685 (accessed 9.19.20).

Schwaab, M., Biscaia, E.C., Monteiro, J.L., Pinto, J.C., 2008a. Nonlinear parameter estimation through particle swarm optimization. Chem. Eng. Sci. doi:10.1016/j.ces.2007.11.024.

Schwaab, M., Biscaia, E.C., Monteiro, J.L., Pinto, J.C., 2008b. Nonlinear parameter estimation through particle swarm optimization. Chem. Eng. Sci. 63, 1542–1552. doi:10.1016/j.ces2007.11.024.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2016. Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE 104, 148–175. doi:10.1109/JPROC.2015.2494218.

Stevens, S.S., 1957. On the psychophysical law. Psychol. Rev. doi:10.1037/h0046162.

Teixeira, M., Rodríguez, O., Gomes, P., Mata, V., Rodrigues, A.E., 2013. Perfume engineering: design, performance & classification. https://doi.org/10.1016/B978-0-08-099399-7.00006-7.

Teixeira, M.A., Rodríguez, O., Mata, V.G., Rodrigues, A.E., 2009. The diffusion of perfume mixtures and the odor performance. Chem. Eng. Sci. 64, 2570–2589. doi:10.1016/j.ces.2009.01.064.

Teixeira, M.A., Rodríguez, O., Rodrigues, A.E., 2010a. The perception of fragrance mixtures: a comparison of odor intensity models. AIChE J. doi:10.1002/aic.12043.

Teixeira, M.A., Rodríguez, O., Rodrigues, A.E., 2010b. Perfumery radar: A predictive tool for perfume family classification. Ind. Eng. Chem. Res. 49, 11764–11777. doi:10.1021/ie101161v.

The new AI tool that represents the future of fragrance formulations | Givaudan [WWW Document], 2019. URL https://www.givaudan.com/fragrance-beauty/perfumery-school/carto-the-future-of-fragrance-formulations (accessed 3.19.21).

Vardanega, T., 2011. Lecture Notes in Computer Science: Preface. Springer, Verlag Berlin, pp. 312–321 Heidelberg (Ed.), Lecture Notes in Computer Science.

von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C., Garcke, J., 2020. Combining machine learning and simulation to a hybrid modelling approach: current and future directions. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 12080. LNCS 548–560. doi:10.1007/978-3-030-44584-3_43.

Wakayama, H., Sakasai, M., Yoshikawa, K., Inoue, M., 2019. Method for predicting odor intensity of perfumery raw materials using dose-response curve database. Ind. Eng. Chem. Res. 58, 15036–15044. doi:10.1021/acs.iecr.9b01225.

Zhang, L., Mao, H., Liu, L., Du, J., Gani, R., 2018. A machine learning based computer-aided molecular design/screening methodology for fragrance molecules. Comput. Chem. Eng. 115. doi:10.1016/j.compchemeng.2018.04.018.

Zhang, X., Zhou, T., Ng, K.M., 2020. Optimization-based Cosmetic Formulation: integration of mechanistic model, surrogate model, and heuristics. AIChE J. doi:10.1002/aic.17064.