Multi-Ion-Sensing Emulator and Multivariate Calibration Optimization by Machine Learning Models

IVAN NY HANITRA, (Graduate Student Member, IEEE), FRANCESCA CRISCUOLO, SANDRO CARRARA, (Fellow, IEEE), AND GIOVANNI DE MICHELI, (Life Fellow, IEEE)
Integrated Systems Laboratory (IC-IINFOM-LSI1), Swiss Federal Institute of Technology Lausanne (EPFL), Station 14, 1015 Lausanne, Switzerland

Corresponding author: Ivan Ny Hanitra (ivan.nyhanitra@epfl.ch)

This work was supported by the Horizon 2020 CyberCare under Grant ERC 2014 ADG 669354.

ABSTRACT One paramount challenge in multi-ion-sensing arises from ion interference that degrades the accuracy of sensor calibration. Machine learning models are here proposed to optimize such multivariate calibration. However, the acquisition of big experimental data is time and resource consuming in practice, necessitating new paradigms and efficient models for these data-limited frameworks. Therefore, a novel approach is presented in this work, where a multi-ion-sensing emulator is designed to explain the response of an ion-sensing array in a mixed-ion environment. A case study is performed emulating the concurrent monitoring of sodium, potassium, lithium, and lead ions, in a medium representative of sweat samples. These analytes are relevant examples of sweat ion-sensing applications for physiology, therapeutic drug monitoring, and heavy metal contamination. It is demonstrated that calibration datasets output by the emulator explain accurately the experimental response of polymeric solid-contact ion-selective electrodes, where root-mean-squared error of $1.37$, $1.44$, $1.78$, $2\,mV$ are obtained, respectively, for $Na^+$, $K^+$, $Li^+$, $Pb^{2+}$ sensor calibration in artificial sweat. Besides, synthetic datasets of custom size are generated to train, validate, and evaluate different types of multivariate regressors. A Multi-Output Support Vector Regressor (M-SVR) is proposed as a compact, accurate, robust, and efficient multivariate calibration model. It features $13.22\%$ normalized root mean squares, and $20.29\%$ mean root squares improvement compared to a simple linear regression model. It is an unbiased estimator for medium to large datasets, and its average generalization error is of $3.22\%$. Besides, M-SVR models have a lower computational complexity than single-output SVR or neural network models, making them a suitable solution for memory and energy-constrained edge devices used for continuous and real-time multi-ion monitoring.

INDEX TERMS Ion interference modeling, machine learning, multi-ion-sensing emulator, multivariate calibration, multi-output support vector regressor.

I. INTRODUCTION Potentiometric ion-sensors are increasingly applied in many fields, leveraging progress in all-solid-state sensing and technology [1]. Namely, they are used for environmental monitoring of pollutants [2], for agricultural soil analysis [2], for water quality control [3], or for biomedical applications such as physiology and healthcare monitoring [4], [5]. The sensor consists in an Ion-Selective Electrode (ISE) coated with a polymeric membrane that selectively entraps the target analyte. Research efforts are focused on improving sensing technology and transduction mechanisms [6], but ion interference is strongly limiting sensing performances because of ISE intrinsic selectivity bounds [7]. Interference arises from the electrolytes inherently constituting the sample, whose composition could vary in a non-predictable way with the environment. Besides, accurate ion-monitoring becomes more intricate when the target electrolyte is extremely diluted in the sample, the analyte concentration being of the same order as sensor Limit Of Detection (LOD).

As a result, multivariate calibration, that consists in binding the multivariate response of the ion-sensing array to the concentration of the target ions, is becoming less accurate. To overcome ion interference and increase sensing accuracy,
the concept of electronic tongue has been introduced. It is a recent trend for which sensor arrays are coupled with chemometric tools. [8] More precisely, it is defined by IUPAC as “a multisensor system, which consists of a number of low-selective sensors, and uses advanced mathematical procedures for signal processing based on Pattern Recognition and/or Multivariate data analysis—Artificial Neural Networks (ANNs), Principal Component Analysis (PCA), and so forth.” [9] Several works are proposed in literature where ANN architectures are coupled to sensor arrays in order to determine simultaneously multi-ions in samples subject to ion interference [2], [10]–[17]. The quality of results obtained with these black-box chemometric models depends on the size and the representativeness of the training dataset. Unfortunately, the acquisition of big data is time and resource consuming in practice, entailing high labor costs of chemical assays. Indeed, the datasets in the aforementioned works are restricted to 15 up to 100 samples. Some approaches have been proposed to cope with this data scarcity issue. Data fusion techniques are used in [18] to extract relevant features from data-limited ion-sensors, and generate synthetic samples from Cumulative Distribution Functions (CDFs) of the concentration of the target ions, before feeding the training set to different machine learning algorithms implemented for ammonium ion measurement.

In this work, a novel approach is proposed, where a multi-ion-sensing emulator is designed to explain the response of an ion-sensor array subject to ion interference from a mixed-ion sample. A compact version of the phase-boundary potential model [19], that is valid for polymeric ISEs, is built at the core of the emulator. Datasets of different size are generated and fed to linear and non-linear regressors that are implemented to improve sensing accuracy. A case study is carried out, emulating the concurrent monitoring of sodium, potassium, lithium, and lead ions, in an environment representative of sweat samples. These ions are relevant examples of sweat ion-sensing applications for physiology, therapeutic drug monitoring, and trace heavy metal contamination. Namely, the main minerals present in perspiration are sodium, potassium, and calcium [20]. During physical exercise, a depletion of sodium and potassium ions is observed [21]. An excessive loss of these ions could lead to muscle cramps, dehydration, or to hypokalemia and hypotonatremia [22]. Besides, sweat ion-sensing could be applied to therapeutic drug monitoring. For instance, lithium salts are administrated to people suffering from bipolar disorder [23]. The therapeutic window of the drug is narrow (0.8 – 1.5 mM [24]), necessitating an accurate and continuous lithium monitoring. Moreover, some trace heavy metals such as lead are present in sweat, but in diluted amount (below 283 µg/L). Nevertheless, massive lead contamination from food, water, or the environment could result in memory trouble, pain, numbness, or behavioral problems [25], thus requiring an accurate lead monitoring system.

The remainder of the manuscript is organized as follows. First, some related works on multivariate calibration optimization in multi-ion-sensing are discussed in Section II to highlight the challenges in data-constrained systems for accurate and simultaneous monitoring of multi-ions. Then, the multi-ion-sensing emulator is presented in Section III, with the description of the compact phase-boundary potential model that is at its core. The methodology carried out to generate synthetic datasets out of the emulator is described in Section IV. Next, the proposed Multi-output Support Vector Regressor (M-SVR) is detailed in Section V. The results of the case study are discussed in Section VI, and the conclusions are reported in Section VII.

II. CHALLENGES AND RELATED WORKS

One of the main challenge in accurate ion-sensing arises from ion interference, due to the background electrolytes in the sample, that creates non-linearity in sensor response. Moreover, in multi-ion-sensing, cross-interference is not negligible, and monitoring analytes that are diluted in the sample is intricate. As a result, multivariate calibration is commonly optimized by machine learning models. Some related works on multi-ion-sensing are listed in Table I, aiming mainly at environmental, agriculture, and water quality monitoring. Polymeric ISEs are fabricated and characterized, enabling the monitoring of ions of interest, and in some works, the tracking of interfering ions as well. Then, synthetic training datasets are acquired from the sensors. They are of relative small size, considering the huge time and chemical resource needed to acquire big data. Nevertheless, the calibration dataset should be representative enough of real samples. Factorial design, random design in the concentration range of the target ions, or usage of CDFs of the concentration of the ions are strategies implemented to obtain statistically relevant synthetic training sets with a limited amount of samples. The test sets are acquired either from real samples, or with synthetic design procedures. Multivariate calibration is performed mostly with feed-forward Back-Propagation Neural Network (BPNN) of one to three hidden layers, trained with a Bayesian regularization algorithm. More complicated architectures are investigated to incorporate a priori knowledge from the sensors (charge balance, electrical conductivity constraints) [12], or genetic independent component analysis [10], [11].

III. EMULATOR DESIGN FOR MULTI-ION-SENSING

The approach proposed in this work to overcome the data-limited constraints in the training of chemometric tools deployed for multivariate calibration is to design an emulator of synthetic datasets, explaining the responses of polymeric ISEs exposed to mixed-ions samples. First, the compact modeling of ISE response is described, then the design of the emulator is detailed.

A. MODELING OF INTERFERENCE IN ION-SENSING

In potentiometric sensing, ISEs are functionalized with an ion-selective membrane that entraps the target ion. The electrode transduces the thermodynamic activity of the target
ion into an electric potential. The Open Circuit Potential (OCP) between the sensing electrode and an inert reference electrode, which has a stable potential, is measured under quasi-zero current conditions [6]. Sensor calibration consists in mapping the measured OCP with the target ion activity. The latter refers to the ability of the analyte to react with other ions, and is related to the effective analyte concentration in the sample by the activity coefficient. In the following, ion activity is the chemical property of interest in the multivariate calibration.

ISE response is usually described by the semi-empirical Nicolsky-Eisenman model [26]. However, it is not very accurate when ions of different charges are present in the sample. Namely, large deviations from acquired sensor data are observed in ion mixtures [27]. In this work, the phase-boundary potential model is used to describe ion-sensor response of a polymeric ISE in mixed-ion solutions. A comprehensive description of the model is presented in [19]. The hereunder analytical derivation is carried out to get a compact ion-sensing model, highlighting the key parameters explaining sensor output distortion due to ion interference. An exhaustive derivation of the model is reported in Appendix A. The phase-boundary potential model is based on ion-exchange considerations at the sample/membrane interface. Transmembrane ion fluxes are ignored in this model. When the target ion is entrapped in the polymeric membrane, an electrical potential is built up at the interface to counter-balance the ion fluxes from the sample phase to the membrane phase, and to preserve electroneutrality. The phase-boundary potential at equilibrium for any ion j is shown in (1).

\[ E_{PB}^{j} = \Phi(m) - \Phi(aq) \]

Next, it is assumed that the membrane ionic strength is unaltered during ion exchange. By applying membrane electroneutrality, with multiple ions interacting with the membrane phase (interfering ions),

\[ 10^{-z_j E_{PB}^{j}/s} \sum_j a_j(aq) 10^{z_j E_0^{j}/s} = 1, \]

where \( E_0^{j} = \epsilon_j^0 + \frac{1}{z_j} s \log \left( \frac{1}{a_j(m)} \right) \) is the apparent standard potential of ion j. All interactions of ion j with the ionophore of the membrane are embedded in \( a_j(m) \). Then, the membrane selectivity is quantified by the selectivity coefficient

\[ \log K_{pot}^{I,j} = \frac{z_J}{s} (E_{0}^{I} - E_{0}^{j}). \]

It represents how much the membrane is selective towards a target ion I than an interferent ion J. Substituting (3) in (2), the compact equation (4), as shown at the bottom of the next page, is obtained, considering monovalent and divalent ions in the sample. The contributions of the target ion I, monovalent, and divalent interfering ions are put in evidence. The selectivity coefficients could be seen as weighting factors of the activity of interfering ions. Equation (4) is solved for \( E_{PB}^{j} \), and yields (5) and (6), as shown at the bottom of the next page, for a monovalent and divalent target ion, respectively.

An ideal Nernstian response is recovered if all interfering electrolytes have a null activity (no interferent), or if

\[ \forall j, \log K_{pot}^{I,j} \rightarrow -\infty \text{(perfectly selective membrane)}. \]
constant $E_I^0$ that could be seen as an offset potential independent of $a_I(aq)$. The parameters to model ion interference are reduced to the selectivity coefficients $\log K_{I,j}^\text{pot}$, and the activity of the interfering ions $a_j(aq)$. Eventually, the OCP between an ISE for target ion I, and the reference electrode is

$$E_I = K_{cell} + E_{PB},$$

where $K_{cell}$ takes into account the constant potentials in the galvanic cell, including reference electrode potential and electrode contact potential drops.

**B. MULTI-ION-SENSING EMULATOR**

A Graphical Unit Interface (GUI) implementation of an ion-sensor calibration curves emulator has been presented in [28]. Its main features are summarized in the following. The tool allows for selecting generic monovalent or divalent primary ions and interfering ions. The selectivity coefficients of the ISEs are tunable, as well as the offset parameter that is fitted from real ISE calibration curves. The phase-boundary potential model and Nicolsky-Eisenman model could be used to output the calibration data, where a Gaussian noise equivalent to sensor signal-to-noise ratio is added. The investigation tool features parametric analysis of the impact of interferent ion activity, and selectivity coefficient of an ISE towards an interferent, in the distortion of the calibration curves due to ion interference. Database of calibration curves from in-vitro measurements could be loaded to compare the emulator outputs to real calibration data.

The investigation tool described previously is modified into a synthetic dataset generator in order to produce the training dataset for the multivariate calibration models. The building blocks of the emulator are pictured in Fig. 1. The compact phase-boundary potential model takes as input parameters the selectivity coefficients of each target ion against each interferent, and the offset parameter encompassing membrane composition. Both parameters are experimentally extracted from real ISE calibration curves. The synthetic dataset is obtained following a factorial design where the activity of each constituent in the sample is discretized in $L$ levels in the detection range of the analyte, for the given application. The constituents include the primary ions and the interfering ones. The output of the dataset generator consists of calibration curves of each sensor at discrete level points.

$$\begin{align*}
10^{\frac{z_I(E_I^0-E_{PB})}{s}}a_I(aq) + 10^{\frac{z_I^2(E_I^0-E_{PB})}{s}} & \sum_{j \text{ s.t. } z_j=1} (K_{I,j}^\text{pot})^{\frac{1}{2}} a_j(aq) + 10^{\frac{z_I(E_I^0-E_{PB})}{s}} & \sum_{j \text{ s.t. } z_j=2} (K_{I,j}^\text{pot})^{\frac{1}{2}} a_j(aq) = 1. 
\end{align*}$$

$$E_{PB} = E_I^0 + s \log \left( \frac{1}{2} \left( a_I(aq) + \sum_{j \text{ s.t. } z_j=1} K_{I,j}^\text{pot} a_j(aq) \right) + \frac{1}{4} \left( a_I(aq) + \sum_{j \text{ s.t. } z_j=2} K_{I,j}^\text{pot} a_j(aq) \right)^2 + \sum_{j \text{ s.t. } z_j=2} (K_{I,j}^\text{pot})^2 a_j(aq) \right).$$

$$E_{PR} = E_I^0 + s \log \left( \frac{1}{2} \sum_{j \text{ s.t. } z_j=1} (K_{I,j}^\text{pot})^{\frac{1}{2}} a_j(aq) + \frac{1}{4} \left( \sum_{j \text{ s.t. } z_j=1} (K_{I,j}^\text{pot})^{\frac{1}{2}} a_j(aq) \right)^2 + a_I(aq) + \sum_{j \text{ s.t. } z_j=2} K_{I,j}^\text{pot} a_j(aq) \right).$$
mixed-ion samples, in an automated way. Four sensors for \( \text{Na}^+ \), \( \text{K}^+ \), \( \text{Li}^+ \), and \( \text{Pb}^{2+} \) monitoring are emulated in presence of \( \text{NH}_4^+ \), \( \text{Ca}^{2+} \), and \( \text{Mg}^{2+} \), that are the major ions constituting sweat samples [31]. The determination of ion mixture combinations requires a multi-factorial design of synthetic samples. In this work, the physiological range of activity of each constituent is divided into six levels. It is a good trade-off between capturing enough variability in the activity of the ions, and avoiding to generate exponential number of samples. Indeed, a full-factorial design involves \( 6^{7} \sim 280'000 \) ion combinations. This amount is not reasonable. Moreover, a full-factorial design will produce a redundant dataset. Therefore, Taguchi method is implemented to generate a representative subset of all constituent combinations, leveraging orthogonal arrays [32]. The \( L_{18}(6^4 \times 3^6) \) orthogonal array used is reported in Table 2, where the factor \( C_0 \) comprises 6 levels, and the 6 other factors have 3 levels. The columns \( C_1 \) are orthogonal, and orthogonality is preserved through column permutation. Thus, \( C_0 \) could be permuted over the seven constituents. Row \( R_0 \) is redundant while permuting columns, so it is removed. The activity of the seven constituents are quantized over the physiological or therapeutic range of the considered ions in sweat. The discrete levels of activities are reported in Table 3. As for selectivity coefficients, they are obtained from electrochemical characterization of polymeric solid-contact ISEs. Fixed interference method is carried out to compute the coefficients reported in Table 4, and the full characterization of \( \text{Na}^+ \), \( \text{K}^+ \), \( \text{Li}^+ \), and \( \text{Pb}^{2+} \) sensors is published at [33]. The entire factorial-designed set is needed to train the multivariate calibration model since it includes all the statistical variability in the data that the regressor should learn. Therefore, cross-validation procedure or splitting of the training set is avoided. Consequently, validation and test sets are designed independently. Random sampling in the activity window of the ions of interest is the usual approach adopted [2], [15]. In this work, the Weibull distribution is used. Its Probability Distribution Function (PDF) is plotted in Fig. 2, and it is defined as

\[
f(x; k, b) = \begin{cases} b k x^{k-1} \exp(-b x^k), & x \geq 0, \\ 0, & x < 0, \end{cases}
\]

where the scale and shape of the distribution are \( \lambda = b^{-1/k} \) and \( k \), respectively. This PDF is chosen so as to generate independent and identically distributed ion mixtures representative of sweat samples for which the activity of each constituent is around its nominal value, or rather in excess in the sample (e.g. due to physical exercise for physiology). In order to increase the robustness of the comparison between multivariate regressors, five validation splits, of one-fifth of the size of the training set each, are randomly generated to emulate a cross-validation. Thus, chemometric models are evaluated on each validation split, and the average and standard deviation of the metrics are computed. The final

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection range</td>
<td>[7.95] mM</td>
<td>[3.8] mM</td>
<td>[2.4; 4.5] mM</td>
<td>[1.4; 8.5] ( \mu )M</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Nominal concentration</td>
<td>39.1 mM</td>
<td>5.12 mM</td>
<td>0.1 mM</td>
<td>0.24 ( \mu )M</td>
<td>3.4 mM</td>
<td>0.37 mM</td>
<td>53.5 ( \mu )M</td>
</tr>
</tbody>
</table>

| \( \text{L}_0 \) | 1 | 0.1 | 0.1 | 1e-3 | 0.1 | 1e-2 | 1e-3 |
| \( \text{L}_1 \) | 5 | 0.5 | 0.25 | 5e-3 | 0.5 | 5e-2 | 5e-3 |
| \( \text{L}_2 \) | 10 | 1 | 0.5 | 1e-2 | 1 | 0.1 | 1e-2 |
| \( \text{L}_3 \) | 25 | 2.5 | 1.25 | 2.5e-2 | 2.5 | 0.25 | 2.5e-3 |
| \( \text{L}_4 \) | 50 | 5 | 2.5 | 5e-2 | 5 | 0.5 | 5e-2 |
| \( \text{L}_5 \) | 100 | 10 | 5 | 0.1 | 10 | 1 | 0.1 |

<table>
<thead>
<tr>
<th>( I )</th>
<th>( \text{Na}^+ )</th>
<th>( \text{K}^+ )</th>
<th>( \text{Li}^+ )</th>
<th>( \text{Pb}^{2+} )</th>
<th>( \text{NH}_4^+ )</th>
<th>( \text{Ca}^{2+} )</th>
<th>( \text{Mg}^{2+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Na}^+ )</td>
<td>-</td>
<td>-2.2</td>
<td>-2.58</td>
<td>n.a.</td>
<td>-1.89</td>
<td>-2.57</td>
<td>-3.56</td>
</tr>
<tr>
<td>( \text{K}^+ )</td>
<td>-1.8</td>
<td>-1.91</td>
<td>n.a.</td>
<td>-1.68</td>
<td>-4.6</td>
<td>-3.32</td>
<td></td>
</tr>
<tr>
<td>( \text{Li}^+ )</td>
<td>-2.55</td>
<td>-2.56</td>
<td>n.a.</td>
<td>-3.08</td>
<td>-4.01</td>
<td>n.a.</td>
<td></td>
</tr>
<tr>
<td>( \text{Pb}^{2+} )</td>
<td>-2.89</td>
<td>-1.64</td>
<td>-2.9</td>
<td>-2.76</td>
<td>-4.28</td>
<td>-3.76</td>
<td></td>
</tr>
</tbody>
</table>

The Weibull distribution (\( k = 1.25 \), \( b = 6 \)) (blue), and discrete levels of each constituent scaled to its respective activity range(red).
validation of the model is performed on an additional test set of similar size as a validation split. Furthermore, datasets of different size are generated to assess the effect of dataset scarcity on the chemometric models. One notices that if the upper bound of the ion activity \( L^5 \) in Table 3 is discarded, there are 10 combinations of 3-level ion activity to configure the orthogonal Table 2, allowing to tune the training set size. As a result, training sets of 68 up to 680 samples are generated following the aforementioned procedure.

V. MULTIVARIATE CALIBRATION MODELS

In this section, the multivariate calibration models used in this work are presented. In a multivariate calibration procedure, the signals output by an array of sensors are processed by a multivariate model that estimates the activity of selected analytes constituting the sample. The chemometric model is trained beforehand with OCP/activity observations, and is then inferred to predict the activity of the analytes in unknown samples [34]. For \( N \) observations, let \( \mathbf{X} = \{ \mathbf{x}_n \}_{n=1}^{N} \), with \( \mathbf{x}_n \in \mathbb{R}^P \), denote the tensor of OCP signals coming from \( P \) ISEs, and \( \mathbf{Y} = \{ \mathbf{y}_n \}_{n=1}^{N} \), with \( \mathbf{y}_n \in \mathbb{R}^M \), denote the tensor of activity of the \( M \) electrolytes of interest. The multivariate calibration problem can be formulated as

\[
\mathbf{Y} = \mathbf{XW} + \mathbf{E},
\]

where \( \mathbf{W} \) is the tensor of regression coefficients, and \( \mathbf{E} \) is the error made in the prediction of the activity of the analytes. It is an inverse calibration problem since the controlled variables (activity of the ions) are estimated taking the dependent variables (OCP signals) as regressors.

A. TRADITIONAL LINEAR REGRESSION MODELS

Inverse least-squares regressor is the simplest first-order model used in multivariate calibration. Contrary to classical least-squares regression, the activity of all electrolytes constituting the sample do not need to be known during the calibration phase. As a result, the tensor of activity of the ions \( \mathbf{Y} \) could be of any dimension \( M \). The minimization objective is

\[
\gamma \ m \in \mathbb{N}, \min_{\mathbf{w}_{a,m}, \mathbf{b}_m} \| \mathbf{Xw}_{a,m} - \mathbf{y}_{a,m} \|_2^2,
\]

where \( \mathbf{w}_{a,m} \) and \( \mathbf{y}_{a,m} \) are columns of \( \mathbf{W} \) and \( \mathbf{Y} \), respectively. Namely, the activity of each target ion is estimated independently, ignoring cross-correlations. Hence, this multivariate calibration model consists in multiple independent Ordinary Least-Squares (OLS) regressions, or Multiple Linear Regression (MLR). \( \mathbf{W} \) could be computed by least-squares minimization of (10), or in a closed-form solution with the normal equation

\[
\hat{\mathbf{W}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y},
\]

where \( \mathbf{X}^T \mathbf{X} \) must be invertible. This holds if \( \mathbf{X} \) is a full-rank matrix, or the columns of \( \mathbf{X} \) are uncorrelated. The application of PCA to \( \mathbf{X} \) will generate a matrix \( \mathbf{T} \) that complies with the latter constraint. Namely, OCP signals are decomposed in a set of successive orthogonal components that explain a maximum amount of their variance. The resulting Principal Component (PC) scores, denoted as \( \mathbf{T} \), are the new input tensor variables. Combination of PCA and MLR refers to Principal Components Regression (PCR). Furthermore, Partial Least-Squares (PLS) regression is another popular chemometric tool [35]. It constructs latent variables for \( \mathbf{X} \) and \( \mathbf{Y} \) by taking into account the variance of \( \mathbf{X} \) and the correlation between factors excerpted from both tensors.

Linear regressors are traditionally used for univariate calibration systems. However, in multivariate systems, and above all, in presence of non-linearity due to ion interference, non-linear regressors are privileged to increase multivariate calibration accuracy. They are described in the following.

B. SINGLE-OUTPUT SUPPORT VECTOR REGRESSION

Support Vector Machines (SVMs) are prominent tools for linear and non-linear input/output modeling, as it is the case of multivariate calibration subject to ion interference [36]. Standard SVR models perform unidimensional regression of the input tensor \( \mathbf{X} \) to each output channel \( m, \mathbf{y}_{a,m} \). For non-linear SVRs, \( \mathbf{X} \) is implicitly mapped to a higher dimensional feature space by a kernel function [37]. Thus, the model learns a linear function in the space induced by the kernel. Gaussian Radial Basis Function (RBF) is a non-linear kernel commonly used for which \( \mathbf{X} \) is mapped to an infinite dimensional Hilbert space [38]. The primal formulation of SVR minimization objective is

\[
\forall \ m \in \{1; M\}, \min_{\mathbf{w}_{a,m}, \mathbf{b}_m} \frac{1}{2} \| \mathbf{w}_{a,m} \|^2 + C \sum_{n=1}^{N} L_c(u_{n,m}),
\]

with \( u_{n,m} = \mathbf{y}_{n,m} - (\Phi(\mathbf{x}_n))^T \mathbf{w}_{a,m} + \mathbf{b}_m \).

\[
L_c = \text{Vapnik } \epsilon \text{-insensitive loss function defined as } L_c(u) = \max\{0, |u| - \epsilon\}. \Phi \text{ is the non-linear function defining the kernel function } \kappa(\mathbf{x}_i, \mathbf{x}_j) \equiv \Phi(\mathbf{x}_i)^T \cdot \Phi(\mathbf{x}_j). \ C \text{ is a hyper-parameter trading-off margin violations and minimization of the distance } \epsilon \text{ from the support vectors to the SVR model. In practice, a dual formulation of the SVR minimization is used, where quadratic programming leveraging kernel formulation yields a problem faster to solve.}

C. PROPOSED MULTI-OUTPUT SUPPORT VECTOR REGRESSION

In single-output SVR problems, the loss function \( L \) is an \( L_1 \)-based norm, hence, it needs to take into account each of the \( M \) dimensions independently which makes the solution complexity grow linearly with \( M \). Moreover, such unidimensional model does not consider correlations between the output columns. Hence, it is less robust to noise and non-linearity in the dataset. In this work, a multivariate multi-output SVR is implemented to consider correlations and non-linear cross-relations between \( \mathbf{X} \) and \( \mathbf{Y} \), enabling a concurrent optimization of the regressor of each target ion activity.
The objective function to minimize is

\[ \mathcal{L}(W, b) = \frac{1}{2} \sum_{m=1}^{M} \left\| \mathbf{w}_{s,m} \right\|^2 + C \sum_{n=1}^{N} L(u_n), \]

with \( u_n = \| \mathbf{e}_n \|^2 \) s.t. \( \mathbf{e}_n = y_n - (W^T \Phi(x_n) + b) \). (13)

Iterative Reweighted Least-Squares (IRWLS) procedures permit the use of an arbitrary cost function \( L \). \( L \)-based norm cost functions are appealing since the constraints for all \( M \) output dimensions could be considered into a single error vector, contrary to \( L_1 \)-based Vapnik loss function described previously that need to take into account all \( M \) dimensions separately. As a result, IRWLS procedures yield one-dimension support vectors, instead of \( M \)-dimension for single-output SVRs. In this work, an IRWLS procedure is implemented with the quadratic and differentiable cost function

\[ L(u) = \begin{cases} 0, & u < \epsilon, \\ a^2 - 2au + \epsilon^2, & u \geq \epsilon. \end{cases} \] (14)

The proof of convergence of such SVM is demonstrated in [39]. The IRWLS method is constructed by performing a first-order Taylor approximation of the error function (14) over the previous solution \((W, b)\). A quadratic approximation of the former expansion gives the weighted least-squares minimization problem

\[ \mathcal{L}'(W, b) = \frac{1}{2} \sum_{m=1}^{M} \left\| \mathbf{w}_{s,m} \right\|^2 + \frac{1}{2} \sum_{n=1}^{N} a_n u_n^2 + \text{const}, \] (15)

where \( a_n = \frac{C d L(u_n)}{du_n} \bigg|_{u_n=0} = \begin{cases} 0, & u_n^k < \epsilon, \\ 2C(u_n^k - \epsilon)/u_n^k, & u_n^k \geq \epsilon. \end{cases} \)

The latter function is minimized with respect to \((W, b)\). It yields a linear system represented in matrix form as

\[ \forall m \in \{1; M\}, \left[ \begin{array}{c} \mathbf{K} + D_a^{-1} \mathbf{a}^T \mathbf{K} \\
\mathbf{a}^T \end{array} \right] \left[ \begin{array}{c} \mathbf{w}_{s,m} \\
\mathbf{b}_{s,m} \end{array} \right] = \left[ \begin{array}{c} \mathbf{y}_{s,m} \\
\mathbf{a}^T \mathbf{y}_{s,m} \end{array} \right]. \] (16)

where \((\mathbf{K})_{ij} = \kappa(x_i, x_j), \mathbf{a} = [a_1, \ldots, a_n]^T, (D_a)_{ij} = a_i a_j \delta(i-j)\). It is important to notice that the matrix \( \mathbf{H} \) is independent of \( m \), so it does not depend on the output channel. The kernel function used is the Gaussian RBF. The pseudocode of the IRWLS procedure is reported hereunder.

IRWLS procedure is implemented in a quasi-Newton approach, for which the errors \( u_n^k \) and \( a_n \) are recomputed at each iteration \( k \), so as the solution \((\mathbf{W}^{sol}, \mathbf{b}^{sol})\) of the reweighted least-squares problem. Each iteration has the complexity of an OLS problem.

\section*{D. Multi-Layer Perceptrons Model}

Multi-Layer Perceptrons (MLP) models are commonly used in complex multivariate calibration systems [2], [10]–[17]. MLP networks are built for the multivariate calibration of potentiometric sensors subject to ion interference in order to benchmark the proposed M-SVR algorithm. The feed-forward NN architecture implemented is illustrated in Fig. 3. The input layer consists of four pass-through neurons that relay the input features \( x_n \) to the subsequent layers. The features are either the OCP signals of the \( P = 4 \) ion-sensors, or their PC scores after applying PCA. Two hidden layers of \( N1 \) and \( N2 \) units produce non-linear transformations of the dataset, and the output layer predicts the activity of the \( M = 4 \) target ions. The model parameters (weights and bias) are trained with a back-propagation training algorithm minimizing the mean-squared error loss function [40]. The hyper-parameters of the MLP models are optimized with a grid-search procedure. They are summarized in Table 5.

\section*{E. SOFTWARE AND HARDWARE}

Data pre-processing pipeline and multivariate models are implemented within a Python 3.7 environment. PCA algorithm is implemented through a Singular Value Decomposition (SVD) where LAPACK routine [42] is used. MLR and OLS minimization problems are solved using SVD on

\begin{algorithm}
\caption{Iterative Reweighted Least-Squares Procedure}
\begin{algorithmic}[1]
\Input OCP signal \( X \), ions log-activity \( Y \), kernel function \( \Phi \), kernel parameters \( \text{param} \), convergence tolerance \( \text{tol} \), maximum number of iterations \( \text{max\_iter} \)
\Output Trained weights \( W \), trained bias \( b \), number of iterations \( k \)

// Initialization */
1 \( (W, b) \leftarrow (0, 0) \); \( k \leftarrow 0; \)
2 \( K \leftarrow X \text{ mapped to kernel } \Phi \);

3 \For each \( n \) \do
4 \hspace{1em} \( \mathbf{e}_n \leftarrow y_n - (W^T K_n, n + b); \) \hspace{0.5em} \Comment{residual error}
5 \hspace{1em} \( u_n^b \leftarrow \|\mathbf{e}_n\|; \)
6 \hspace{1em} \{ i \}^k \leftarrow \{ n \text{ s.t. } u_n^b \geq \epsilon \}; \hspace{0.5em} \Comment{support vectors}
7 \EndFor
8 \hspace{1em} \( L(u_k^b) \leftarrow u_k^b - 2u_k^b \epsilon + \epsilon^2; \) \hspace{0.5em} \Comment{to compute \( \mathcal{L} \)}
9 \hspace{1em} \( a_i \leftarrow \frac{2C(u_k^b - \epsilon)}{u_k^b}; \) \hspace{0.5em} \Comment{to compute \( \mathcal{L}' \)}

// Iterative procedure */
10 \While convergence tolerance is not reached and \( k < \text{max\_iter} \) \do
11 \hspace{1em} \( (W^k, b^k) \leftarrow (W, b) \); \hspace{0.5em} \( E^k \leftarrow E; \)
12 \hspace{1em} \( u^k \leftarrow u; \) \{ i \}^k \leftarrow \{ i \}; \)
13 \hspace{1em} \text{build } H \text{ tensor and solve } \mathcal{L}' \to (W^{sol}, b^{sol});
14 \hspace{1em} \text{update } u, \{ i \}, L, \text{ and } \mathcal{L};
15 \hspace{1em} \text{minimize } \mathcal{L} \text{ with descending direction step } \eta^k;
16 \hspace{1em} \text{compute next step solution }
17 \hspace{1em} \left[ [W_k + 1] \right] = 
18 \hspace{1em} \left. \begin{array}{c}
19 \hspace{1em} \left[ (b^{k+1})^T \right] = 
20 \hspace{1em} \left. \begin{array}{c}
21 \hspace{1em} \left[ W^k \right] \\
22 \hspace{1em} \left[ (b^k)^T \right] \\
23 \hspace{1em} \left[ W^{sol} - W^k \right] \\
24 \hspace{1em} \left[ W^{sol} - b^k \right] \end{array} \right] \end{array} \right); \)
25 \hspace{1em} \text{recompute } a_i \text{ and update least-squares problem } \mathcal{L}'^k;
26 \hspace{1em} k \leftarrow k + 1;
27 \EndWhile
\end{algorithmic}
\end{algorithm}
FIGURE 3. Feed-forward neural network architecture for simultaneous monitoring of sodium, potassium, lithium, and lead ions: the input features (OCP signals or their PC scores) are fed to the passthrough neurons of the input layer, then conveyed to two fully-connected hidden layers of $N_1$ and $N_2$ number of units, respectively. The output layer consists of four units with no activation function, providing the activity of the target ions.

TABLE 5. Configuration and hyper-parameters of MLP model, optimizer, and training procedure.

<table>
<thead>
<tr>
<th>MLP model</th>
<th>Input layer: 4 pass-through neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden layer 1: N1 neurons ($\sigma = \text{ELU}$)</td>
<td></td>
</tr>
<tr>
<td>Hidden layer 2: N2 neurons ($\sigma = \text{ELU}$)</td>
<td></td>
</tr>
<tr>
<td>Output layer: 4 neurons</td>
<td></td>
</tr>
<tr>
<td>Weights initializer: Glorot (normal distribution) [41]</td>
<td></td>
</tr>
<tr>
<td>Regularizer: L2 regularizer ($\alpha = 10^{-3}$)</td>
<td></td>
</tr>
<tr>
<td>Loss function</td>
<td>Mean-squared error</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Type: stochastic gradient descent with momentum optimization</td>
</tr>
<tr>
<td></td>
<td>Learning rate scheduling: $\eta(t) = \frac{\eta_0}{1 + 10^{\beta(t - 1)}}$</td>
</tr>
<tr>
<td>Training</td>
<td>Epochs: 500 (early-stopping)</td>
</tr>
<tr>
<td></td>
<td>Batch size: 32</td>
</tr>
</tbody>
</table>

zero-mean and unit-variance $X$ with LAPACK routine [42], while PLS is implemented with Non-Linear Iterative Partial Least-Squares (NIPALS) algorithm [43]. Single-output SVR is constructed leveraging LIBSVM library that supports quadratic programming [44]. As for MLP models, they are implemented through Keras high-level API [45], with TensorFlow 2.0 deep learning library as computational backend. The experiments are run on a 2x Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz with 256 GB RAM.

VI. RESULTS AND DISCUSSION

In this section, the results of the case study are presented. The performance of the multi-ion-sensing emulator is assessed and compared with calibration curves obtained from polymeric solid-contact ISEs. Then, the synthetic datasets generated by the tool are shown. Next, the multivariate calibration optimization with linear and non-linear regressors is discussed.

A. MULTI-ION-SENSING EMULATOR AND ION INTERFERENCE MODELING

First, the ion-sensing emulator is used as investigation tool to evaluate the effect of ion interference on ISE response. Fig. 4 illustrates the impact of the two model parameters on calibration curve distortion. Namely, the activity of the interfering ion $a_J$, and the selectivity coefficient of the primary ion $I$ towards the interfering ion $J$, $\log K_{pot}^{I,J}$. The background potential offset in diluted analyte ($\log a_I \leftarrow -\infty$) increases with the activity of the interfering electrolyte, and this effect is enhanced for a poorly-selective sensor. Moreover, the elbow of the calibration curve highlighting the change of sensor regime from flat OCP response (not detecting) to a Nernstian behavior is extrapolated as the sensor lower LOD [46]. It is shown that sensor lower LOD increases, the more the severity of ion interference is high. This is critical if the lower LOD reaches the physiological/therapeutic range of the target analyte. In practice a margin of 1 : 10 is needed between the sensor lower LOD and the lowest activity of the primary ion to be measured.

Next, the calibration curves generated by the emulator are compared with OCP responses acquired from batch of polymeric solid-contact ISEs in artificial sweat [33]. The results are displayed in Fig. 5, where the measured calibration curves exhibit inter-sensor variability, mainly for $\text{Li}^+$ and $\text{Pb}^{2+}$ sensors. The simulated OCP response is computed by taking the same electrolyte background composition as the artificial sweat, and with the selectivity coefficients of the corresponding polymeric solid contact ISEs against the interfering ions. The offset parameter for each primary ion is fitted to the in-vitro calibration data. Then, the RMSE between the modeled and the experimental data is computed, and yields, 1.37, 1.44, 1.78, 2 mV, for $\text{Na}^+$, $\text{K}^+$, $\text{Li}^+$, $\text{Pb}^{2+}$ sensors, respectively. As a result, the ion-sensing model explains accurately polymeric ISE responses subject to ion interference from artificial sweat background electrolyte.
### TABLE 6. Prediction accuracy obtained by applying linear regressors to datasets of different size.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( \text{log } a_{\text{Na}^+} ) NRMSE</th>
<th>MRE</th>
<th>( \text{log } a_{\text{K}^+} ) NRMSE</th>
<th>MRE</th>
<th>( \text{log } a_{\text{Li}^+} ) NRMSE</th>
<th>MRE</th>
<th>( \text{log } a_{\text{Pb}^{2+}} ) NRMSE</th>
<th>MRE</th>
<th>Total NRMSE</th>
<th>MRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset_1</td>
<td>3.64</td>
<td>3.04</td>
<td>9.72</td>
<td>8.19</td>
<td>3.11</td>
<td>2.76</td>
<td>6.68</td>
<td>5.82</td>
<td>23.15</td>
<td>19.81</td>
</tr>
<tr>
<td>Dataset_2</td>
<td>4.32</td>
<td>3.61</td>
<td>11.22</td>
<td>7.23</td>
<td>2.84</td>
<td>2.61</td>
<td>5.63</td>
<td>4.96</td>
<td>21.71</td>
<td>18.41</td>
</tr>
<tr>
<td>Dataset_3</td>
<td>3.76</td>
<td>3.40</td>
<td>8.74</td>
<td>7.15</td>
<td>3.00</td>
<td>2.52</td>
<td>5.44</td>
<td>4.36</td>
<td>20.94</td>
<td>17.43</td>
</tr>
<tr>
<td>Dataset_4</td>
<td>3.72</td>
<td>3.07</td>
<td>7.33</td>
<td>6.56</td>
<td>3.18</td>
<td>2.47</td>
<td>5.66</td>
<td>4.71</td>
<td>19.88</td>
<td>16.81</td>
</tr>
<tr>
<td>Dataset_5</td>
<td>3.17</td>
<td>2.59</td>
<td>8.89</td>
<td>7.85</td>
<td>2.60</td>
<td>2.21</td>
<td>5.17</td>
<td>3.85</td>
<td>19.86</td>
<td>16.49</td>
</tr>
<tr>
<td>Dataset_6</td>
<td>3.02</td>
<td>2.47</td>
<td>8.24</td>
<td>6.81</td>
<td>2.72</td>
<td>2.28</td>
<td>5.31</td>
<td>4.11</td>
<td>19.29</td>
<td>15.68</td>
</tr>
<tr>
<td>Dataset_7</td>
<td>3.55</td>
<td>2.99</td>
<td>7.21</td>
<td>5.95</td>
<td>3.68</td>
<td>2.20</td>
<td>4.98</td>
<td>4.10</td>
<td>19.41</td>
<td>15.23</td>
</tr>
<tr>
<td>Dataset_8</td>
<td>3.51</td>
<td>2.94</td>
<td>8.21</td>
<td>5.31</td>
<td>2.21</td>
<td>1.80</td>
<td>5.08</td>
<td>4.11</td>
<td>19.92</td>
<td>14.16</td>
</tr>
<tr>
<td>Dataset_9</td>
<td>3.14</td>
<td>2.63</td>
<td>6.98</td>
<td>5.15</td>
<td>3.36</td>
<td>1.98</td>
<td>5.94</td>
<td>4.31</td>
<td>18.52</td>
<td>13.87</td>
</tr>
<tr>
<td>Dataset_10</td>
<td>3.35</td>
<td>2.87</td>
<td>7.16</td>
<td>5.30</td>
<td>3.42</td>
<td>1.77</td>
<td>4.13</td>
<td>3.46</td>
<td>18.06</td>
<td>13.40</td>
</tr>
</tbody>
</table>

**FIGURE 5.** Comparison between the calibration curves output by the ion-sensing emulator and the calibration curves acquired from batch of polymeric solid-contact ISEs in artificial sweat. RMSE of 1.37, 1.44, 1.78, and 2 mV are obtained for \( \text{Na}^+ \), \( \text{K}^+ \), \( \text{Li}^+ \), and \( \text{Pb}^{2+} \) sensors, respectively.

**B. GENERATION OF SYNTHETIC DATASETS**

The multi-ion-sensing emulator has been demonstrated to model well ISE responses in nominal sweat background electrolyte. Thus, in this section, the tool is used to generate synthetic datasets following experimental design procedures described in Section IV. Ten datasets Dataset_n of training set size \( N_{\text{train}} = 68 \cdot n \) (\( n \in [1; 10] \)) are generated. The scarcest dataset is displayed in Fig. 6. The six levels of the training set are clearly visible since it has been generated through a factorial design. The composition of the validation and test sets is randomly designed in the same activity range as the training set, resulting in samples intermingled in between the factor levels. Emulated \( \text{Na}^+ \)-ISE calibration data exhibits a Nernstian response since the physiological range in sweat is large (above tens of mM). Conversely, \( \text{Pb}^{2+} \) is detected as metal traces (tens of \( \mu \text{M} \)), so it is diluted in the sample. The emulated potential response shows more potential dispersion. As for \( \text{Li}^+ \)-ISE, the sensor has a good selectivity towards the constituents (see Table 4), so it is less affected by ion interference. It is not the case of \( \text{K}^+ \)-ISE that is poorly selective, mainly towards \( \text{Na}^+ \) that is a predominant ion in the samples. This results in major OCP dispersion, accentuated at lower \( a_{\text{K}^+} \). The same results are observed with larger datasets. The apparent linear correlation between \( E_X \) and \( \log a_X \) that is observed in Fig. 6 suggests to compute the correlation coefficients between the two variables, for each ion, for the training set of the ten datasets. The results are displayed in Fig. 7. The lower correlation in \( \text{K}^+ \) and \( \text{Pb}^{2+} \) channels is due to ion interference that is explained by the larger potential dispersion at lower ion activity. The correlation between sensor OCP and ion activity tends to increase with larger datasets.

**C. LINEAR REGRESSION MODELS**

Prior to multivariate calibration, PCA is applied to the input tensor \( X \). Fig. 8 shows the scatter plots of the PC scores.
FIGURE 7. Pearson correlation coefficient between $E_X$ and log $a_X$ for $Na^+$ (green), $K^+$ (blue), $Li^+$ (red), and $Pb^{2+}$ (black) channels.

FIGURE 8. Scatter matrices of PCA scores for Dataset_1.

The whitened PC scores are uncorrelated and do not present higher order dependence. Moreover, the explained variance per PC yields $34.23\%$, $24.51\%$, $21.39\%$, and $19.87\%$, for $PC1$ to $PC4$, respectively. This indicates that all the PCs are needed to explain the variance present in the input tensor. It is important to underline that PCA is employed as a pre-processing step, and not as a dimensionality reduction technique ($P = 4$). Moreover, two metrics are introduced to assess the accuracy of the regressors. The Normalized Root Mean-Squared Error (NRMSE) and the Mean Relative Error (MRE) are defined as

$$NRMSE = \frac{100}{\bar{y}} \sqrt{\frac{1}{N_{test}} \sum_{n=1}^{N_{test}} (\bar{y}_n - \hat{\bar{y}}_n)^2},$$

$$MRE = \frac{100}{\bar{y}} \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \frac{|\bar{y}_n - \hat{\bar{y}}_n|}{\bar{y}_n},$$

where $\bar{y}$ is the mean of the test set labels, $\bar{y}_n$ and $\hat{\bar{y}}_n$ are the true and predicted log-activity of the primary ions, respectively. Normalized metrics are required to balance error contributions from analytes highly concentrated (sodium ions), and diluted analytes in the sample (lead ions). In addition, the two metrics do not over-penalize outliers. NRMSE and MRE are computed for each primary ion, but total NRMSE and MRE of all four primary ions yield two compact model performance measure that are used during training.

An analysis is carried out by applying MLR to $\tilde{X} = \{\{PC1\}, \{PC1, PC2\}, \{PC1, PC2, PC3\}, \{PC1, PC2, PC3, PC4\}\}$. The regressors are fitted to the training set and evaluated on the validation splits. The total NRMSE decreases with the dimension of the latent space. Likewise, PLS algorithm is applied to the input tensor $X$, and the same factorial analysis is performed, suggesting to use the four-dimension latent space. Next, the fitted models are evaluated on the external test set. It results that factorial decomposition with PCR and PLS yield identical multivariate calibration accuracy when the full latent space is used. When a reduced latent space is used, PLS performs slightly better, as expected, since it is a more compact model than PCR. The metrics achieved by applying linear regressors on the ten datasets are reported in Table 6. It could be seen that $K^+$ and $Pb^{2+}$ calibration are the less accurate ones due to the non-linear distortion added by severe ion interference. Besides, the metrics are improving steadily with larger datasets, for the four target ions. The metrics obtained with linear regression models are used as benchmark for the subsequent non-linear multivariate regressors implemented.

### D. MULTIVARIATE CALIBRATION OPTIMIZATION

In this section, the non-linear multivariate calibration models are compared to simple MLR model. Namely, SVR, the proposed M-SVR, and MLP models are evaluated on the ten synthetic datasets. A cross-validation grid-search on the model hyper-parameters is carried out to identify the best models, while limiting over-fitting. More explicitly, the models are trained on the entire orthogonal training set, and evaluated on five validation splits. The best model is retained and evaluated on the external test set. For MLP models, the hyper-parameters space being wide, a coarse search is done before refining the grid-search on relevant hyper-parameters space. The results of multivariate calibration are displayed in Fig. 9.
FIGURE 9. Total NRMSE and total MRE obtained by evaluating the best multivariate calibration models on the test set of the ten synthetic datasets.

FIGURE 10. Total NRMSE obtained while assessing multivariate calibration models on five random validation splits and on a test set.

23.92%, for SVR, M-SVR, and MLP, respectively. The accuracy improvement obtained with the proposed M-SVR is satisfactory given the lower complexity of the proposed model. Namely, M-SVR optimizes concurrently the calibration of the four channels with four-dimension support vectors. Conversely, single-output SVR optimizes independently one ion channel at a time with one-dimension support vectors, therefore four times more support vectors than with M-SVR. Besides, M-SVR is more robust to noise and correlation in the dataset since it considers the four ion channels when constructing the model regression hyperplane. It is also observed that with large datasets (more than 540 training samples), M-SVR performs better than SVR and MLP regressors.

Next, the quality of the models is assessed by comparing the scores obtained during the validation stage (metrics calculated with five validation splits of random samples) and the scores obtained with the test set. The results are displayed in Fig. 10. We observe that the generalization error is large for the scarcest dataset, for which the models are more likely subject to over-fitting. Overall, the generalization error is the smallest for M-SVR with an average of 3.22%, while it is of 4.43% and 4.79%, for SVR and MLP models, respectively.

A further analysis is performed on the importance of PCA pre-processing. The scores obtained while applying PCA before the multivariate calibration, and without applying PCA are reported in Fig. 11. It is added that the input tensor $X$ is scaled to zero-mean and unit-variance in both cases. We observe that PCA does not influence the calibration accuracy for MLR and M-SVR, while improvement is observed with SVR and MLP models. MLP regressors are feature-based models, so the pre-processing of the input tensor is essential to feed uncorrelated data to the network model. SVR and M-SVR models are distance-based models, so scaling the input tensor is the essential pre-processing step.

The predicted ion activity vs target values when applying the optimal M-SVR model on Dataset_1 and Dataset_10 are plotted in Fig. 12 and 13, respectively. The scatter points fall along the 1:1 lines, with more dispersion for $K^+$ and $Pb^{2+}$ ions that are subject to severe ion interference. For Dataset_1, the 95% confidence interval on the slopes of the model fit contains 1 for $Li^+$ and $Pb^{2+}$ channels, and the intercepts contain 0 except for $Pb^{2+}$. These results are consistent with the previous conclusion that M-SVR models are less accurate with a scarcer dataset since the model is slightly biased. As for Dataset_10, the 95% confidence interval on the slopes of the model fit contains 1, and the intercepts contain 0 for the four channels. Therefore, M-SVR models applied to larger datasets are accurate and unbiased estimators. The computed coefficients of determination provide reliable information.
TABLE 7. Training parameters of the multivariate calibration models.

<table>
<thead>
<tr>
<th>N_{\text{train}}</th>
<th>MLR</th>
<th>SVR</th>
<th>M-SVR</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># trainable parameters</td>
<td># support vectors</td>
<td># IRWLS iterations</td>
<td># trainable parameters</td>
</tr>
<tr>
<td>68</td>
<td>16</td>
<td>251</td>
<td>68</td>
<td>5</td>
</tr>
<tr>
<td>136</td>
<td>16</td>
<td>543</td>
<td>125</td>
<td>5</td>
</tr>
<tr>
<td>204</td>
<td>16</td>
<td>814</td>
<td>204</td>
<td>12</td>
</tr>
<tr>
<td>272</td>
<td>16</td>
<td>1024</td>
<td>272</td>
<td>9</td>
</tr>
<tr>
<td>340</td>
<td>16</td>
<td>1360</td>
<td>340</td>
<td>4</td>
</tr>
<tr>
<td>408</td>
<td>16</td>
<td>1632</td>
<td>408</td>
<td>10</td>
</tr>
<tr>
<td>476</td>
<td>16</td>
<td>1904</td>
<td>476</td>
<td>17</td>
</tr>
<tr>
<td>544</td>
<td>16</td>
<td>2176</td>
<td>544</td>
<td>5</td>
</tr>
<tr>
<td>612</td>
<td>16</td>
<td>2446</td>
<td>612</td>
<td>4</td>
</tr>
<tr>
<td>680</td>
<td>16</td>
<td>2720</td>
<td>680</td>
<td>4</td>
</tr>
</tbody>
</table>

E. ANALYSIS OF MODEL COMPLEXITY

Model complexity is a paramount feature to consider prior to deploying the multivariate calibration models onto memory and energy-constrained edge nodes, for real-time ion prediction. The trainable parameters of the multivariate calibration models are reported in Table 7. MLR model has \( M \cdot P = 16 \) trainable weights, independently of the size of the training set, and OLS minimization has a complexity of \( O(N M^2) \). The complexity of SVR models is indicated by the number of support vectors needed to build the regression hyperplane. More accurate models are obtained with small \( \epsilon \), so most of the training instances are used as support vectors. However, M-SVR has four-dimension support vectors for each training instance since the four ion channels are considered simultaneously. On the contrary, four independent models are constructed with single-output SVR, explaining the four times larger amount of one-dimension support vectors. Besides, the complexity of IRWLS procedure is equivalent to an OLS minimization per iteration. The number of iterations required to reach convergence is reasonable, being less than 20. As for MLP models, the complexity of the model is set by the number of neurons in the model. The amount of weights and bias to estimate during the training phase is quite large, and tends to grow with larger training sets. The number of epochs before early-stopping is also reported. It does not exceed 400 epochs for a training batch size of 32 samples.

The training runtime of the different models is measured to compare the model complexity quantitatively. The models are trained 100 times with the best hyper-parameters found.
I. N. Hanitra et al.: Multi-Ion-Sensing Emulator and Multivariate Calibration Optimization by Machine Learning Models

I. N. Hanitra

acquired in artificial sweat from Na
couple of in-vitro calibration curves acquired from polymeric
parameters of the tool are experimentally determined from a
interference are explained by an ion-sensing emulator. The
size. Namely, the response of ion-sensing arrays subject to ion

tion depends on the training dataset representativeness and
This work presents a novel approach for multi-ion-sensing
ning applications.

ded latency (scale divided by ten to fit onto the graph),
single-output SVR embed M dense models, explaining the
µ
for MLR models. M-SVR models are compact models, so the
in Fig. 14.b. The prediction latency is obviously the shortest
models directly onto edge devices for real-time ion moni-

t models could be trained offline and we could deploy the trained
models onto edge devices, paving the way for continuous and real-
time multi-ion monitoring.

Eventually, the multi-ion-sensing emulator and multivi-
rate calibration models are designed in this work for biomedical
and healthcare monitoring in sweat. But the flexibility
of the tool enables its use for broader applications such as
environmental or water quality monitoring.

APPENDIX A
PHASE-BOUNDARY POTENTIAL MODEL
At equilibrium, the electrochemical potentials for any ion j,
of valence \( z_j \), in the sample and membrane phase are

\[
\begin{align*}
\mu_j^{aq} &= \mu_j^{0}(aq) + RT \ln(a_j^{aq}) + \frac{z_j}{F} \Phi(aq), \\
\mu_j(m) &= \mu_j^{0}(m) + RT \ln(a_j(m)) + \frac{z_j}{F} \Phi(m),
\end{align*}
\]

(A.1)

where \( \mu_j^{0} \) is the standard chemical potential, \( a_j \) is
the ion activity, and \( \Phi \) is the electrical potential. Assuming
partition equilibrium of ion \( j \) between both membrane and
sample phases, \( \mu_j(aq) = \mu_j(m) \), one obtains the phase-
boundary potential defined as \( E_{PB} \),

\[
E_{PB} = \frac{\epsilon_j^{0} + 1}{z_j} - \frac{1}{z_j} \log\left( \frac{a_j^{aq}}{a_j(m)} \right),
\]

(A.2)

where \( \epsilon_j^{0} = \frac{\mu_j^{0}(aq) - \mu_j^{0}(m)}{RT} \) is the standard potential, constant
for a given ion. \( s = \frac{RT}{F} \ln 10 \) is the Nernst slope. Then,
let us consider the ion-selective membrane composed of an
ionophore L of charge \( z_L \) and concentration \( L_T \), and an
ion-exchanger R of charge \( z_R \) and molconcentration \( R_T \). The
ionophore includes ion carriers that selectively transport ions
across the membrane. The ionophore forms complexes with the
target ion I as

\[
F^{\text{(m)}}(m) + nL \xrightarrow{\beta_n} jL_n,
\]

(A.3)

where \( \beta_n \) is the complex formation constant. The interactions
between the ion-exchanger and the co-ions are neglected. The
charge balance in the ion-selective membrane yields

\[
z_R R_T + z_L L_T + \sum_n (nz_L + z_j) c_j L_n + z_j c_j(m) = 0,
\]

(A.4)

where each contributions are, from the ion-exchanger,
the ionophore, the complex formed with the target ion, and

FIGURE 14. (a) Training runtime and (b) prediction latency of multivariate calibration models. * values divided by 1.
the free form of the target ion in the membrane, respectively. c stands for compound concentration. The label (m) is added to emphasize that it refers to the concentration of ion j in the membrane and not in the sample phase. Inserting complex formation constants $\beta_n$ and activity coefficients $\gamma_j$,

$$z_R R_T + z_L \frac{a_L}{\gamma_L} + z_j \frac{a_j(m)}{\gamma_j(m)} \left( 1 + \sum_n \frac{n z_L + z_j \gamma_j(m)}{z_j} \beta_n a_L^n \right) = 0. \tag{A.5}$$

Next, let us consider that multiple ions from the sample phase (interfering ions) could interact with the membrane. It is assumed that the activity of the uncomplicated ionophore and the activity coefficient of the extracted ions remain unaltered. This is valid for a polymeric membrane with an ionophore in excess with respect to the ion-exchanger, and considering that the membrane ionic strength is not changing during ion exchange. (A.5) is summed over the ions extracted from the sample phase into the membrane, and yields

$$\sum_j z_j \frac{a_j(m)}{\gamma_j(m)} \left( 1 + \sum_n \frac{n z_L + z_j \gamma_j(m)}{z_j} \beta_n a_L^n \right) = -z_R R_T - z_L \frac{a_L}{\gamma_L}. \tag{A.6}$$

Then, an apparent standard potential for each ion j, $E_j^0$, is introduced such that $E_{PB} = E_j^0 + \frac{z_j}{2} \log(a_j(aq))$. The apparent standard potential is expressed as

$$E_j^0 = \varepsilon_j^0 + \frac{s}{z_j} \log \left( - \frac{1}{\gamma_j(m)} \left( 1 + \sum_n \frac{n z_L + z_j \gamma_j(m)}{z_j} \beta_n a_L^n \right) \right). \tag{A.7}$$

where the expression in the log is equal to $-\frac{1}{a_j(m)}$. (A.7) is rewritten as

$$z_j \frac{k_j}{\gamma_j(m)} \left( 1 + \sum_n \frac{n z_L + z_j \gamma_j(m)}{z_j} \beta_n a_L^n \right) 10^{-\varepsilon_j^0/s} = -z_R R_T - z_L \frac{a_L}{\gamma_L}. \tag{A.8}$$

where the free energy of ion transfer $k_j$ is defined as $\varepsilon_j^0 = \frac{z_j}{2} \log(k_j)$. Moreover, $a_j(m)$ is extracted from (A.2) and substituted in (A.6), yielding

$$\sum_j \frac{z_j}{\gamma_j(m)} k_j \left( 1 + \sum_n \frac{n z_L + z_j \gamma_j(m)}{z_j} \beta_n a_L^n \right) 10^{-\varepsilon_{PB}^0/s} a_j(aq) = -z_R R_T - z_L \frac{a_L}{\gamma_L}. \tag{A.9}$$

Inserting (A.8) in (A.9), one finds

$$10^{-\varepsilon_{PB}^0/s} \sum_j a_j(aq) 10^{\varepsilon_j^0/s} = 1. \tag{A.10}$$

The selectivity coefficient of a membrane with target ion I, with respect to an interfering ion J, is defined as

$$\log K_{I,J}^{pot} = \frac{z_I}{s} (E_I^0 - E_J^0). \tag{A.11}$$

Equation (A.11) is substituted in (A.10), and gives the compact equation

$$10^{z_I (E_I^0 - E_{PB})/s} a_I(aq) + 10^{z_J (E_J^0 - E_{PB})/s} \sum_{j \neq I} (K_{I,J}^{pot})^{z_J/2} a_j(aq) + 10^{z_J (E_J^0 - E_{PB})/s} \sum_{j \neq J} (K_{J,I}^{pot})^{z_J/2} a_j(aq) = 1, \tag{A.12}$$

limiting the sum expansion to monovalent and divalent ions. Equation (A.12) is solved for $E_{PB}$, with a second-order polynomial equation formulation. For $z_J = 1$,

$$E_{PB} = E_I^0 + s \log \left( \frac{1}{2} \left( a_I(aq) + \sum_{j \neq I} K_{I,J}^{pot} a_j(aq) \right) \right) + \left[ \frac{1}{4} \left( a_I(aq) + \sum_{j \neq I} K_{I,J}^{pot} a_j(aq) \right)^2 + \sum_{j \neq I} (K_{I,J}^{pot})^{z_J/2} a_j(aq) \right]. \tag{A.13}$$

while for $z_J = 2$,

$$E_{PB} = E_I^0 + s \log \left( \frac{1}{2} \sum_{j \neq I} (K_{I,J}^{pot})^{z_J/2} a_j(aq) \right) + \left[ \frac{1}{4} \left( \sum_{j \neq I} (K_{I,J}^{pot})^{z_J/2} a_j(aq) \right)^2 + a_I(aq) + \sum_{j \neq I} (K_{I,J}^{pot})^{z_J/2} a_j(aq) \right]. \tag{A.14}$$

APPENDIX B
SYNTHETIC DATASETS

The synthetic datasets generated are available at http://ieeexplore.org/documents/multi-ion-sensing-emulator.

REFERENCES


FRANCESCO CRISCUOLO received the B.Sc. and M.Sc. degrees (cum laude) in materials engineering and nanotechnology from the Politecnico di Milano, Italy, and the Ph.D. degree from the Integrated System Laboratory, EPFL, Lausanne, Switzerland, in April 2020, with her thesis “Wearable multi-electrode platform for ion-sensing”. During her studies, she participated in several exchange programs in France, Czech Republic, The Netherlands, and Belgium. She spent one year with TU Delft, where she specialized in bio-nanomaterials and in nanoelectronics. In 2015, she worked with the Energy Storage Group, IMEC, Leuven, Belgium. She is currently a Postdoctoral Fellow with EPFL, where she focuses on novel electrochemical biosensors for healthcare and diagnostics.

SANDRO CARRARA (Fellow, IEEE) is currently a Faculty with the EPFL, Lausanne, Switzerland, and a former Professor with the University of Genoa, Italy, and the University of Bologna, Italy. Along his carrier, he published seven books, with prestigious publishers like Springer/Nature and Cambridge University Press. He published more than 270 papers and 14 patents. He was a member of the BoG of the IEEE CASS and also a member of the IEEE Sensors Council. He was a recipient of the IEEE Sensors Council Technical Achievement Award. He is also the Editor-in-Chief of the IEEE SENSORS JOURNAL and an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.

GIOVANNI DE MICHELI (Life Fellow, IEEE) is currently a Professor and the Director of the Institute of Electrical Engineering, EPFL, Lausanne, Switzerland. His research interest includes several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips, and bio-sensing/systems. He was a recipient of the 2016 IEEE/CS Harry Goode Award, the 2016 EDAA Lifetime Achievement Award, and the 2012 IEEE/CAS Mac Van Valkenburg Award among other recognitions. He is also a Fellow of ACM and a member of the Academia Europaea and an International Honorary member of the American Academy of Arts and Sciences.