Explainable Machine Learning Algorithms To Predict Glass Transition Temperature

Article *in* Acta Materialia · April 2020

DOI: 10.1016/j.actamat.2020.01.047

7 authors, including:

Edesio Alcobaça
University of São Paulo
14 PUBLICATIONS **209** CITATIONS
SEE PROFILE

Saulo Martiello Mastelini
University of São Paulo
54 PUBLICATIONS **598** CITATIONS
SEE PROFILE

Tiago Botari
University of São Paulo
34 PUBLICATIONS **1,448** CITATIONS
SEE PROFILE

Bruno Pimentel
Universidade Federal de Alagoas
32 PUBLICATIONS **485** CITATIONS
SEE PROFILE

EXPLAINABLE MACHINE LEARNING ALGORITHMS TO PREDICT GLASS TRANSITION TEMPERATURE

Edesio Alcobaça,* Saulo Martiello Mastelini,* Tiago Botari,* Bruno Almeida Pimentel, Daniel Roberto Cassar, André Carlos Ponce de Leon Ferreira de Carvalho and Edgar Dutra Zanotto
Institute of Mathematics and Computer Sciences
University of São Paulo, São Carlos, Brazil
and
Department of Materials Engineering
Center for Research, Technology, and Education in Vitreous Materials
Federal University of São Carlos, São Carlos, Brazil
{bapimentel, andre}@icmc.usp.br, {edesio, mastelini}@usp.br
{tiagobotari,daniel.r.cassar}@gmail.com, dedz@ufscar.br

February 3, 2020

ABSTRACT

Modern technologies demand the development of new glasses with unusual properties. Most of the previous developments occurred by slow, expensive trial-and-error approaches, which have produced a considerable amount of data over the past 100 years. By finding patterns in such types of data, Machine Learning (ML) algorithms can extract useful knowledge, providing important insights into composition-property maps. A key step in glass composition design is to identify their physical-chemical properties, such as the glass transition temperature, T_g. In this paper, we investigate how different ML algorithms can be used to predict the T_g of glasses based on their chemical composition. For such, we used a dataset of 43.240 oxide glass compositions, each one with its assigned T_g. Besides, to assess the predictive performance obtained by ML algorithms, we investigated the possible gains by tuning the hyperparameters of these algorithms. The results show that the best ML algorithm for predicting T_g is the Random Forest (RF). One of the main challenges in this task is the prediction of extreme T_g values. To do this, we assessed the predictive performance of the investigated ML algorithms in three T_g intervals. For extreme T_g values ($\leq 450\,\mathrm{K}$ and $\geq 1150\,\mathrm{K}$), the top-performing algorithm was the k-Nearest Neighbours, closely followed by RF. The induced RF model predicted extreme values of T_g with a Relative Deviation (RD) of $3.5\,\%$ for glasses with high T_g ($\geq 1150\,\mathrm{K}$), and RD of $7.5\,\%$ for glasses with very low T_g ($\leq 450\,\mathrm{K}$). Finally, we propose a new visual approach to explain what our RF model learned, highlighting the importance of each chemical element to obtain glasses with extreme T_g. This study can be easily expanded to predict other composition–property combinations and can advantageously replace empirical approaches for developing novel glasses with relevant properties and applications.

1 Introduction

Glasses are non-equilibrium, non-crystalline materials that spontaneously relax to the supercooled liquid state [1]. Unlike crystals, glasses do not need to satisfy rigid stoichiometry rules and can be thought of as continuous solutions of chemical elements. There is a huge number of possible compositions for forming glassy materials. Indeed, 80 chemical elements combined in discrete quantities of $1\,\mathrm{mol}\%$ would produce 10^{52} possible glass compositions [2]. Nevertheless,

* These authors contributed equally to this work.

the number of inorganic glasses reported is only around 10^6, which implies an enormous window of opportunity for the discovery of novel glass-forming compositions.

Developing new multicomponent glasses—with relevant properties and applications—has been an empirical endeavor mainly guided by educated guesses via the expensive and time-consuming trial-and-error approach. Using computational tools such as *ab initio* and classical molecular dynamics simulation is limited to simple compositions, typically with less than 5 elements. The current predictive approaches to aid the thoughtful synthesis of new multicomponent glasses consists of predicting their properties by resorting to empirical models with a small confidence range.

In this scenario, Machine Learning (ML) has demonstrated a considerable capability to address complex problems in materials sciences by leveraging existing available knowledge [3, 4]. The existing materials data provide a fertile environment for ML applications, which can be used to harvest and analyze the embedded knowledge providing a valuable source of information for further research and technological development.

To move towards the use of ML algorithms to obtain new glass compositions, the development of models for predicting glass properties with higher accuracy is essential. At the same time, to access the embedded knowledge in materials data, the decision-making process needs to be understood of behind the predictions performed. Towards this aim, there is a category of algorithms that can, in principle, induce interpretable models. By explaining how they arrive at their decisions, explainable models can provide useful insights, aiding in the discovery of new and relevant knowledge.

In order to develop new glasses, knowledge of the glass transition temperature (T_g) is fundamental. T_g can be defined (in short) as the temperature that a glassy material transitions from a hard and brittle substance to a viscous, soft state. Its importance is related, for instance, to the relief of residual stresses and the glass stability against crystallization [5], as well as mechanical stability [6]. The T_g strongly depends on the chemical composition. Chemical elements and combinations that generate glasses of very low T_g are being continually searched to reduce manufacturing costs, whereas element combinations that lead to very high T_g glasses are used to develop glasses for refractory applications.

A previous study [7] reported a successful application of a Multilayer Perceptron (MLP) artificial neural network (ANN) to predict the T_g of multicomponent oxide glasses. The model was trained using more than $50\,000$ glass compositions containing over 46 chemical elements. The estimated accuracy of the inferred model was less than $\pm 6\,\%$ error over $90\,\%$ of the times. Fortunately, the prediction error did not depend on the number of elements in the glass composition. However, for glasses with very high T_g ($\geq 1150\,\mathrm{K}$), or very low T_g ($\leq 450\,\mathrm{K}$), the prediction uncertainty was significantly larger.

Although MLP neural networks have been successfully used in several Materials Science predictive tasks [8, 9, 10, 11, 12, 13, 14, 15, 16], other ML algorithms could provide better estimations due to their different learning biases [17, 18]. Additionally, it is well known that the predictive performance of an ML algorithm depends on the values assigned to its hyperparameters. Choosing a good set of values is not an easy task. For this reason, many software packages with implementations of the ML algorithms suggest default values for their hyperparameters. Even though default values can, in most (but not all) cases, lead to a reasonable performance, the results are usually better if a proper set of values is selected, which is known as hyperparameter tuning [19].

Another essential aspect to be considered when using ML algorithms is how straightforward the interpretation of the induced models is. Models obtained by some popular ML algorithms, such as the ANN and Support Vector Machines (SVM) [20], are often difficult to interpret. For this reason, these models have been called black-box models [21]. Therefore, there is an active movement towards the induction of explainable models [21, 22]. This movement argues that to trust a model induced by a ML algorithm, the model must be easily understood and interpreted by humans. Explainable models can, in principle, provide useful insights regarding the strategy used in the decision-making process.

Considering the previously discussed issues, in this work, we aim to dwell on the following questions:

- **Q.1:** How is the predictive performance of the induced models affected by using tuned instead of default hyperparameter values for the ML algorithms?
- **Q.2:** Is there any statistically significant difference in the predictive performance of models induced by different algorithms for the whole T_g dataset?
- **Q.3:** Is it possible to induce, for our dataset, explainable models with predictive power similar to the predictive power of non-explainable, frequently used models?
- **Q.4:** Is there any statistically significant difference in the predictive performance of models induced by different ML algorithms for extreme values of T_g?

To address **Q.1, Q.2**, we used six distinct ML algorithms in the task of predicting T_g values. The algorithms used are the following: MLP [23], Support Vector Regression (SVR) [20], Categorical Boosting (CatBoost) [24], k-Nearest Neighbors (k-NN) [25], RF [26], and Classification and Regression Tree (CART) [27]. Moreover, all ML algorithm hyperparameters were tuned and their accuracy was compared with each other. To answer **Q.4**, we analyzed the performance of the ML algorithms in three ranges of T_g values: low ($T_g \leq 450\,\mathrm{K}$), intermediate ($450\,\mathrm{K} < T_g < 1150\,\mathrm{K}$), and high ($T_g \geq 1150\,\mathrm{K}$) temperatures. Finally, we addressed the **Q.3** by analyzing three ML algorithms that are able to induce interpretable models: RF, CART and CatBoost.

This paper starts with a brief review of related works in Section 2. Next, we discuss the dataset collection and the methodology adopted for the ML experiments (Section 3). Afterwards, we present and discuss the main results, followed by the interpretation of the explainable induced models (Section 4). Finally, in Section 5, we present our final considerations concerning the findings from this research.

2 Related Work

ML has been used in the field of Materials Science and Engineering since the late nineties [9] and has attracted great attention over the last decade. ML algorithms have been used to predict properties of polymers, metallic alloys, and ceramics [8, 11, 10, 12, 14, 15, 28, 29, 30, 31, 32, 33, 34, 35].

In the field of **oxide glasses**, to the best of our knowledge, the first work to use ANNs was that of Brauer et al. [13], which focused on the prediction of the chemical durability of glasses containing P_2O_3, CaO, MgO, Na_2O, and TiO_2. About ten years later, Krishnan and co-authors [36] explored the same property considering other ML algorithms, such as RF and SVM. In their analysis, they discussed a physics-informed ML approach, where the training was done separately, depending on the pH of the solution. This approach improved the predictive power of the algorithms, as it could account for the V-shape dependence of chemical durability on the acidity of the solution.

ANNs were also successfully used to predict the glass transition temperature [7] and to evaluate of the quality of tempered glass [37]. When applied to predict the Young modulus for a small dataset of silicate glasses (up to 105 examples), MLP neural networks were outperformed by Gaussian Process Regression [38]. When applied to predict the Young modulus obtained by molecular dynamics simulations [39], ANNs offered the highest accuracy when compared with polynomial regression, LASSO, and RF. Although it is not a "glass property", the liquidus temperature is essential for the glass making process and it was also focused on studies using ANNs by Dreyfus et al. [10] and Mauro etal. [16].

In related works reporting the application of ML algorithms to glasses, only a small number of algorithms were investigated, and usually without hyperparameter tuning. Moreover, scarce or no effort was made to interpret explainable models. In order to address these gaps, in this work we selected six ML algorithms, including not only those that provide non-explainable solutions, such as the models explored in related works [8, 9, 10, 11, 12, 13, 14, 15, 16, 40], but also ML algorithms that enable model interpretation. Additionally, we performed hyperparameter tuning and investigated the predictive accuracy of the induced models in three T_g regions (low, intermediate, and high). This last investigation was carried out to test how the induced models behave in predicting the extreme values of T_g, for which we expect a low predictive accuracy [7].

3 Experimental methodology

This section describes the methodology used in the ML experiments. Firstly, we present the original dataset used in the experiments and pre-processing techniques we applied to the dataset to improve its use. Next, we describe the strategies used for ML algorithm tuning and validation of ML experiments. It is worth mentioning that in the ML literature, an experiment typically consists of the following steps: the application of an ML algorithm to a training set (inducing a predictive model), followed by the evaluation of the predictive performance of the induced model on a test set.

3.1 Dataset

The T_g dataset used in this work was collected from scientific journals, books, and patents obtained from the *SciGlass* database version 7.12 [41]. We limited our query to oxide glasses. For such, we considered compositions with at least $30\,\%$ of the atomic fraction of oxygen. We excluded all compositions containing chemical elements that could change the balance of oxygen, such as sulfur, hydrogen, carbon, fluorine, chlorine, nitrogen, bromine, iodine, and the noble metals (platinum and gold, which are typically present in small amounts in metallic form).

The collected raw dataset has approximately 51 000 glass compositions, each composition with 2 to 32 different chemical elements from a total of 65 chemical elements. A close analysis of the raw dataset revealed that many entries refer to the same composition. Therefore, we removed and replaced these duplicated instances by a single entry with their median T_g value. A discussion about the effect of replacing the duplicate data by their median or keeping them is provided in Sections 1 and 7 of the supplementary material. The final number of unique glass compositions is 43 238. In this cleaned dataset, T_g values range from 342 K to 1495 K, and they seem to follow a normal distribution with mean 774 K. Figure 1 shows a histogram of the T_g values in the raw dataset (with duplicated compositions) and the cleaned dataset (without duplicated compositions).

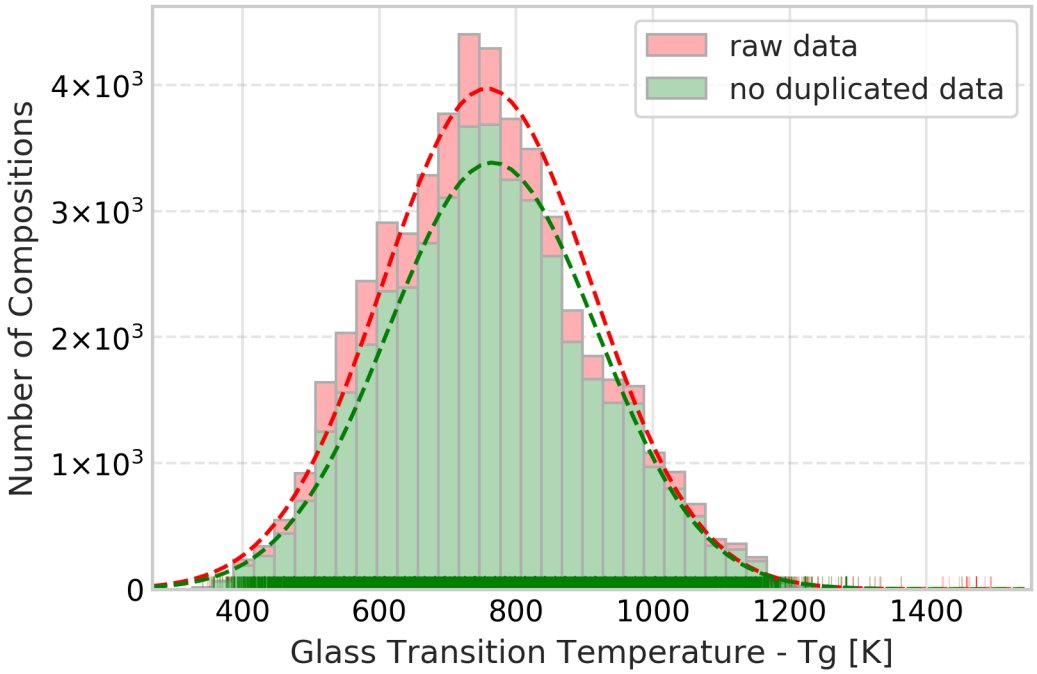


Figure 1: Histogram of T_g values. The red bars represent the raw dataset, and the green bars represent the non-duplicated dataset. The dotted lines are Gaussian fits of the histograms with $R^2 = 0.981$ and $R^2 = 0.984$ for the raw dataset and the non-duplicated dataset, respectively.

From these 43 238 glass compositions, more than 20 000 are materials containing silicon and boron. These components are among the most commonly used glass forming elements. Some chemical elements appear in very few materials, as is the case of Hg, Rh, Ru, and Pd, which appear in 1, 5, 9, and 10 compositions, respectively. A histogram of the number of compositions containing each chemical element for the final dataset (except oxygen, which is present in every glass) is shown in Figure 2.

Additionally, in the supplementary material, we present histograms of the total number of compositions and the T_g distribution of some duplicated glasses (Figures S1 and S2, respectively). Only the clean dataset was used in the experiments. Besides, a comparison with the results obtained using the original dataset is shown and discussed in the supplementary material.

3.2 Machine Learning Algorithms

In this section, we briefly present the six ML algorithms used in our experiments: MLP [23], SVR [20], CatBoost [24], k-NN [25], RF [42, 26], and CART [27]. We chose these algorithms due to their different inductive biases [18] and because some of them induce explainable models. A detailed description of each of these algorithms can be found in the supplementary material, in Section 2.

We grouped the algorithms into two categories: non-explainable (black box) and explainable (white box). Here, explainable models are those whose explanations about their decision-making process can be easily extracted. Thus, models induced by MLP, SVM, and k-NN are considered non-explainable, whereas those induced by RF, CatBoost, and CART are considered explainable.

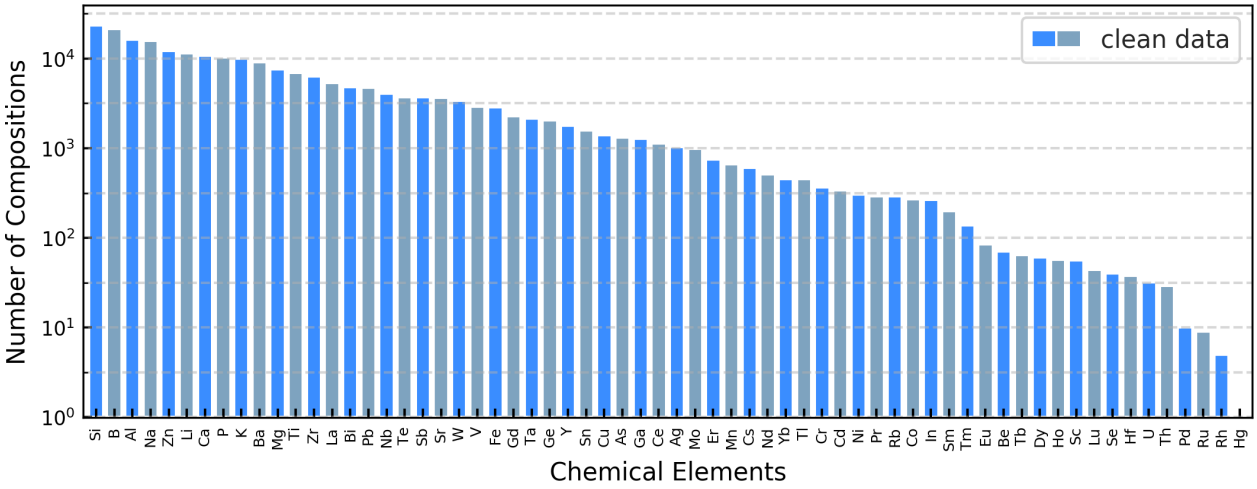


Figure 2: Number of compositions containing each element in the clean dataset. Please note that Oxygen is present in all glasses of this dataset.

3.3 Evaluation measures

To compare the predictive performance obtained by the six regression algorithms, we used four evaluation measures often used in regression tasks: Relative Root Mean Square Error (RRMSE), Root Mean Square Error (RMSE), Relative Deviation (RD), and Determination Coefficient (R^2). These measures use as input the number of test objects N, the expected outcomes y, the predicted values $\hat{y}$, and the target mean value $\bar{y}$. The measures are detailed in Section 3, in the supplementary material.

3.4 Training and Evaluation Setup

We did not perform any feature engineering in our data set, i.e., we only considered the original features (chemical compositions) for training the ML models. The target value used was T_g.

As previously mentioned, the predictive performance of a model induced by an ML algorithm is usually affected by the values assigned to its hyperparameters. Taking this aspect into consideration, in our experiments, we compared the predictive performance of regression models induced by the six ML algorithms with tuned and default hyperparameters. To tune the hyperparameters of the algorithms, we used the Random Search method [19] with a budget of 500 iterations for each algorithm.

For such, we divided the dataset into training and test sets following the k-fold cross-validation (k-CV) strategy, with $k = 10$. This strategy randomly divides the data into 10 folds of similar size, with 9 folds to train a regression model and 1 fold to test the trained model. Thus, we have 10 different training and test sets, and the performance of the regressor can be assessed by taking the average predictive performance in the test folds.

For the hyperparameter tuning, we used the training data by further separating it into 5-CV, with 4 folds used for training and 1 for validating the tuning process. For each training set, 500 different sets of hyperparameter values were evaluated. The hyperparameter sets were ranked accordingly to the median of their predictive performance on the validation sets. We used RRMSE in the tuning process because it compares the performance of the evaluated predictor against a baseline (a model that always predicts the mean value in the evaluation data) [43]. Thus, it can indicate whether tuning the hyperparameter values indeed produced a relevant improvement over a trivial predictor or not. For more information concerning this metric, please refer to Section 3 in the supplementary material.

Finally, to access the predictive performance in the test set, we selected the configuration with the best predictive performance in the 5-CV. We used this configuration to induce a new model with the whole training set and to evaluate its performance in the test set. The final measure used to compare each algorithm, and their best configuration was the average and the standard deviation of the 10 test sets. It is worth noting that we did not ever use the test set for hyperparameter tuning of the ML algorithm or model selection.

We applied the non-parametric version of the Wilcoxon signed-rank test [44] (using $\alpha = 0.05$) to confirm whether there was enough evidence to assume an algorithm performed better than the others, and to check whether the tuning was adequate. We applied this method to assess whether there is a statistically significant difference between the mean

performance of each pair of ML algorithms. If we observe a significant difference, we then reapply the test to evaluate whether the difference is positive or negative.

All experiments were performed using the Python programming language with the *sklearn*, *numpy*, *pandas*, and *catboost* packages. We compared models induced by tuned ML algorithms with their default counterparts, i.e., the models induced by algorithms using the default hyperparameter values suggested by the used Python packages. In Section 4 of the Supplementary Material, we describe the set and range of all hyperparameters used for tuning the regression algorithms, as well as the best values found.

4 Results and Discussion

This section presents the main results from our experiments and our findings from these results. We present a comparison of the predictive performance obtained by the ML algorithms, tuning them, and using their default hyperparameters. Furthermore, we evaluate the performance of extreme T_g values. Afterwards, we interpret explainable models in searching for new insights for glass scientists and engineers. To address the questions introduced in Section 1, first, we answer **Q1** and **Q2** (Subsection 4.1), then we proceed to **Q4** (Subsection 4.2), and finally, we address **Q3** (Subsection 4.3).

4.1 Tuned hyperparameter values versus default hyperparameter values

Table 1 summarizes the main experimental results obtained by the six ML algorithms with the default and tuned hyperparameter values for the four previously mentioned measures (Section 3.3). The results are divided into two groups: non-explainable models and explainable models. For the RMSE, RRMSE, and RD measures, the lower the value, the better, whereas for R^2 it is the opposite. The best results per regression algorithm are shown in **bold**, whereas the best results per performance measure are underlined. In the following discussion, we mainly focus on the RRMSE and RD measures. RRMSE, as pointed out in Section 3.4, assesses the obtained predictive gains against a naive predictor. RD, by presenting a percentage of the prediction errors, is of easy and straightforward interpretation.

Among all the evaluated algorithms, the worst case (SVR) predicts T_g values with approximated $16\,\%$ of RD for the models induced by algorithms with default hyperparameters, and $3.8\,\%$ for the models induced by algorithms with tuned hyperparameters. On the other hand, the best performer (RF) obtained RD rates of approximately $2.5\,\%$ for the default algorithms and $2.4\,\%$ for the tuned algorithms. An RD of $3\,\%$ means that, for instance, a T_g prediction of $1000\,\mathrm{K}$ has in average $30\,\mathrm{K}$ of error. After tuning, R^2 also showed high values, more than 0.90 for all cases.

The RRMSE always presented values less than one, indicating that even if we have not tuned the algorithms, the models generated would be better than the mean baseline. This can be observed because RRMSE $= 0$ indicates a perfect fit while RRMSE $= 1$ indicates the mean of the test target value (a baseline that always predicts the mean).

In the same table, we also show in the same table results for the paired Wilcoxon tests comparing the standard version of the regressors with their tuned counterparts. As can be seen, for all the cases except for CART, the tuned algorithms performed statistically better than their default versions for most of the performance measures (except for k-NN with RRMSE). These observations answer our research question **Q.1**.

The difference between tuned and non-tuned (default) ML algorithms can be better visualized by comparing the spread of the true T_g values versus the observed predictions. To this end, we present scatter plots for each regression algorithm investigated. In these figures, the x-axis represents the measured values of T_g, whereas the y-axis represents the value predicted by the model. We also added a straight line representing the identity function, i.e., an ideal setting where the models predict the expected responses precisely. Therefore, the farther the points are from this line, the worse the predictions. Finally, we colored the points according to the observed RD values per instance. We show this comparison for the RF and SVR regressors in Figure 3. For the other algorithms, we show the comparisons in Section 5 of the supplementary material.

The sole observation of the mean error values or prediction scattering cannot give enough evidence that an algorithm indeed performed statistically better than its competitors. We address this question by analyzing the performance of the compared algorithms considering the whole T_g distribution in the Wilcoxon test, as shown in Table 2. As it can be easily seen, RF was statistically better than all other algorithms. Next, k-NN was superior to all other algorithms, except RF. Both CatBoost and MLP tied in this analysis, being either worse than RF and k-NN. In the last positions came CART and SVR, once again tied.

In fact, the RF algorithm is known to be robust and performs well, even without tuning its hyperparameters [45]. Despite the fact that the k-NN algorithm is be the simplest ML algorithm used, it presented the second best predictive performance. The main reason for this good performance is related to the fact that glasses with similar composition are

Table 1: Experimental results: tuned hyperparameters versus default hyperparameters. The best results per regressor are shown in **bold**, and the best results per measure are underline/underlined. The upward arrows indicate that the tuned algorithms are statistically better than their default counterparts, the downward arrows indicate the opposite, and the circles indicate ties.

Panel A: Non-explainable models

Measure	MLP			k-NN			SVR		
	Default	Tuned		Default	Tuned		Default	Tuned	
RMSE	60 ± 2	$\mathbf{35 \pm 2}$	↑	35 ± 1	$\mathbf{33 \pm 1}$	↑	147.4 ± 0.9	$\mathbf{44 \pm 2}$	↑
RRMSE	0.40 ± 0.01	$\mathbf{0.23 \pm 0.01}$	↑	0.23 ± 0.01	$\mathbf{0.22 \pm 0.01}$	●	0.98 ± 0.01	$\mathbf{0.29 \pm 0.01}$	↑
RD	5.9 ± 0.2	$\mathbf{3.0 \pm 0.2}$	↑	2.75 ± 0.06	$\mathbf{2.5 \pm 0.1}$	↑	16.0 ± 0.1	$\mathbf{3.8 \pm 0.3}$	↑
R^2	0.84 ± 0.01	$\mathbf{0.95 \pm 0.01}$	↑	0.95 ± 0.01	$\mathbf{0.95 \pm 0.01}$	↑	0.41 ± 0.01	$\mathbf{0.92 \pm 0.01}$	↑

Panel B: Explainable models

Measure	CatBoost			CART			RF		
	Default	Tuned		Default	Tuned		Default	Tuned	
RMSE	43.9 ± 0.8	$\mathbf{36 \pm 1}$	↑	44 ± 1	$\mathbf{43 \pm 2}$	●	32 ± 1	$\underline{\mathbf{30 \pm 1}}$	↑
RRMSE	0.30 ± 0.01	$\mathbf{0.24 \pm 0.01}$	↑	0.29 ± 0.01	$\mathbf{0.29 \pm 0.01}$	●	0.21 ± 0.01	$\underline{\mathbf{0.20 \pm 0.01}}$	↑
RD	4.07 ± 0.07	$\mathbf{3.2 \pm 0.1}$	↑	$\mathbf{3.3 \pm 0.1}$	3.4 ± 0.1	↓	2.49 ± 0.06	$\underline{\mathbf{2.38 \pm 0.06}}$	↑
R^2	0.92 ± 0.01	$\mathbf{0.94 \pm 0.01}$	↑	0.92 ± 0.01	$\mathbf{0.92 \pm 0.01}$	●	0.96 ± 0.01	$\underline{\mathbf{0.96 \pm 0.01}}$	↑

Table 2: Statistical test between tuned (line) and not tuned (column) algorithms considering RRMSE. The upward arrows indicate the line is statistically better than the column and the downward arrows the opposite. The circles indicate the line and column are statistically equal.

	CatBoost	CART	k-NN	SVR	MLP	RF
CatBoost	●	↑	↓	↑	●	↓
CART	↓	●	↓	●	↓	↓
k-NN	↑	↑	●	↑	↑	↓
SVR	↓	●	↓	●	↓	↓
MLP	●	↑	↓	↑	●	↓
RF	↑	↑	↑	↑	↑	●

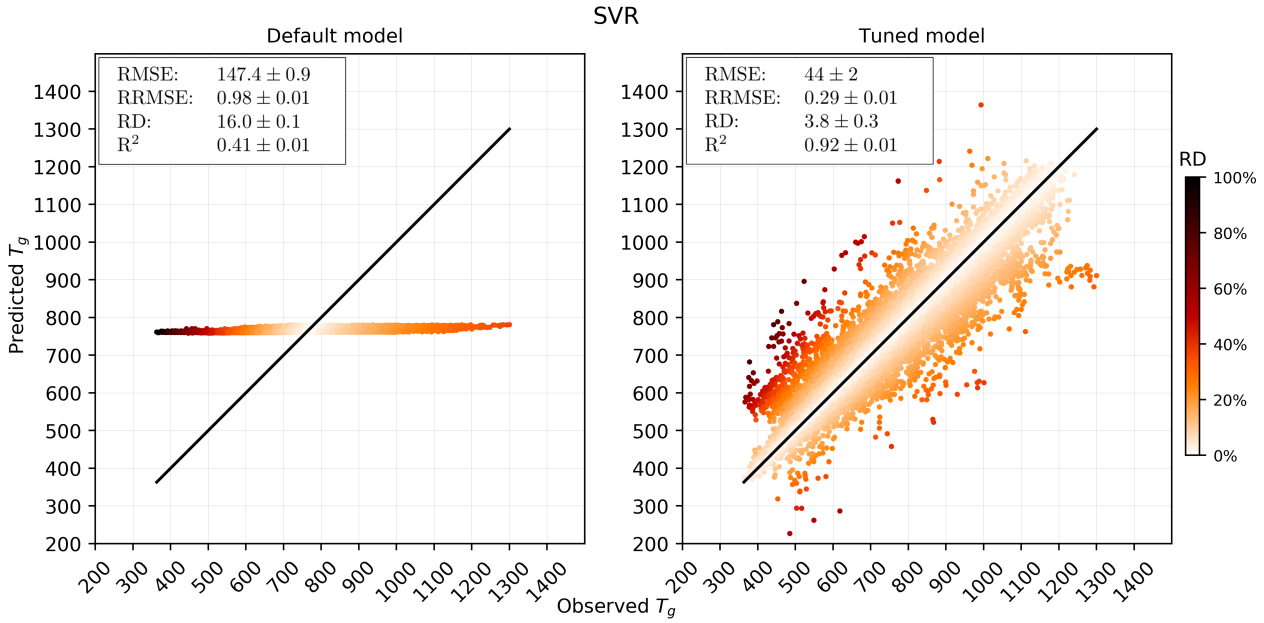
expected to have similar T_g values [7], which is the main idea behind the k-NN algorithm. With these observations, we thus recommend using the RF algorithm as a regressor when aiming at predicting all regions of the T_g space. These analyses answer our research question **Q.2**.

4.2 Comparing the ML algorithms in the extremes of the T_g distribution

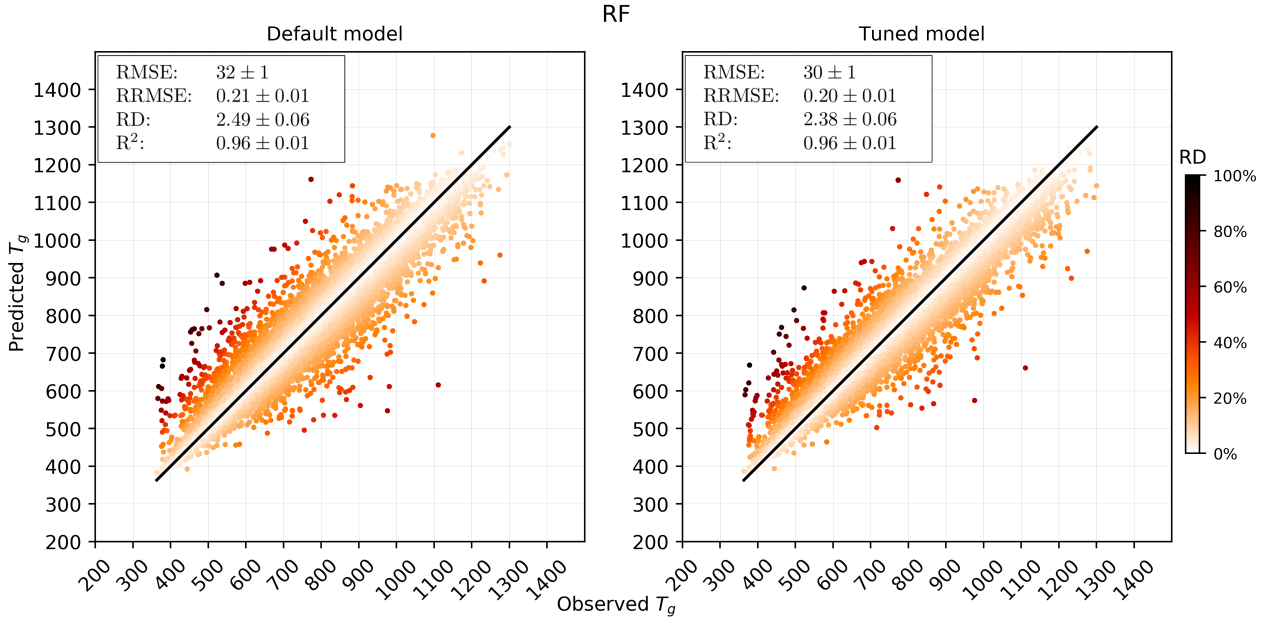
The previous discussion does not address the performance of the algorithms for two regions of interest for glass makers and scientists: the extremes of the T_g distribution. Hence, we address **Q.4** by analyzing how the algorithms performed for glasses with low ($\leq 450\,\mathrm{K}$), intermediate ($450\,\mathrm{K} < T_g < 1150\,\mathrm{K}$), and high ($\geq 1150\,\mathrm{K}$) T_g values, respectively. The referred analysis is shown in Table 3. Here we discovered an interesting behavior: k-NN performed better than the RF in both extremes of the T_g distribution, except for the intermediate region, where RF was the top contender.

Considering the best performing algorithms, RF and k-NN, the accuracy considering different ranges of T_g value are very similar, as shown in Table 4. For both low and high ranges of T_g, k-NN outperformed RF by $\approx 5\,\mathrm{K}$. While for the intermediate T_g range, RF outperformed k-NN by $\approx 4\,\mathrm{K}$. These differences between the RF and k-NN algorithms are negligible considering the nature of T_g, which again emphasizes the use of RF as the most reliable regression algorithm for predicting this glass property.

We also analyzed the performance of RF and k-NN (using the best hyperparameter found) on the inclusion of duplicated instances against replacing them by their median (i.e., the approach used so far). This analysis can be found in Section 7 of the supplementary material.



(a)



(b)

Figure 3: Prediction scattering for SVR (a) and RF (b): default vs. tuned hyperparameters.

4.3 Interpreting the RF results: contribution of the different elements to the value of T_g

Overall, the RF algorithm presented the best predictions for the test examples and it has been considered an explainable algorithm [46]. Therefore, an analysis of the trees in the RF model could, in principle, provide valuable insights into how the percentage of each chemical element in a glass affects its T_g, which is particularly interesting for extreme T_g values. Figure 4 illustrates, using Violin plots [47], the distribution of compositions having very low and very high T_g, whose presence in the dataset is superior to a certain threshold (10 glasses).

In particular, Figure 4a shows a violin plot for each of the 22 most abundant chemical elements present in the 400 oxide glasses with $T_g \leq 450\,\mathrm{K}$. Figure 4b is a related figure for the 17 most abundant elements contained in the

Table 3: Statistical test between tuned (line) and tuned (column) algorithms considering RRMSE. Results for both low ($T_g \leq 450\ K$), intermediate ($450\ K < T_g < 1150\ K$), and high ($T_g \geq 1150\ K$) T_g ranges are presented, in this order. The upward arrows indicate the line is statistically better than the column and the downward arrows the opposite. The circles indicate the line and column are statistically equal.

	CatBoost	CART	k-NN	SVR	MLP	RF
CatBoost	● ● ●	● ↑ ●	● ↓ ↓	↑ ↑ ↑	● ● ●	● ↓ ●
CART	● ↓ ●	● ● ●	● ↓ ●	↑ ● ↑	● ↓ ●	● ↓ ●
k-NN	● ↑ ↑	● ↑ ●	● ● ●	↑ ↑ ↑	↑ ↑ ↑	↑ ↓ ↑
SVR	↓ ↓ ↓	↓ ● ↓	↓ ↓ ↓	● ● ●	↓ ↓ ↓	↓ ↓ ↓
MLP	● ● ●	● ↑ ●	↓ ↓ ↓	↑ ↑ ↑	● ● ●	● ↓ ●
RF	● ↑ ●	● ↑ ●	↓ ↑ ↓	↑ ↑ ↑	● ↑ ●	● ● ●

Table 4: Results for the best performing ML algorithms: RF and k-NN for low, intermediate, and high T_g values. The best results by range of T_g is in bold.

Measure	**RF**			**k-NN**		
	Low	**Intermediate**	**High**	**Low**	**Intermediate**	**High**
RMSE	60 ± 10	$\mathbf{28.8 \pm 0.8}$	70 ± 20	$\mathbf{50 \pm 20}$	33 ± 1	$\mathbf{60 \pm 20}$
RRMSE	2.9 ± 0.7	$\mathbf{0.20 \pm 0.01}$	3 ± 1	$\mathbf{2.6 \pm 0.9}$	0.23 ± 0.01	$\mathbf{3 \pm 1}$
RD (%)	8 ± 1	$\mathbf{2.32 \pm 0.06}$	3.5 ± 0.7	$\mathbf{6 \pm 2}$	2.5 ± 0.1	$\mathbf{2.9 \pm 0.6}$
R^2	0.9 ± 0.1	$\mathbf{0.04 \pm 0.01}$	0.93 ± 0.06	$\mathbf{0.8 \pm 0.2}$	0.05 ± 0.01	$\mathbf{0.91 \pm 0.08}$

329 oxide glasses with $T_g \geq 1150\,\mathrm{K}$. The elements are ordered from left to right according to their abundance in the respective dataset of very high or very low T_g.

Similar to a box plot, a violin plot shows the data distribution using a kernel density estimation. Each violin has 2 sides. The right side shows the distribution of the element in the training data. The left side represents how often the element is present in the forest's trees, with its color being an indirect measure of the confidence of predictions for the given element. It is important to observe that this measure is different from the built-in RF's Feature Importance measure [26], which, despite being well-known in the ML community, does not take into account specific ranges of interest in the target variable (T_g in our case).

These figures also show additional information for each element. The value on the top of each violin plot refers to the number of glasses that contain this element in the specific dataset (very high or very low T_g). Above each value, on the top of the rectangle, there is another value, the number of glasses containing this element in the dataset of 43 238 glasses obtained after removing of the duplicated examples.

Thus, the left side of the violin plots is of statistical nature (amount of the element in the respective T_g range dataset). The right side shows how important the RF algorithm considered this element for the induction of RF trees, which is part of the explanation provided by the RF models on how it predicts T_g values for a new, previously unknown, glass. Thus, the right side bears two types of information:

- Shape: defined by the glass composition rules (path from tree root to a tree leaf) of the RF trees. Thus, they reflect concentration ranges of each element in the trees.
- Color: represents the frequency the element appeared in the paths present in the RF trees. It can be interpreted as how important the element was for the RF predictions, when that element appears in the glass composition. The higher the importance the darker the shade of blue.

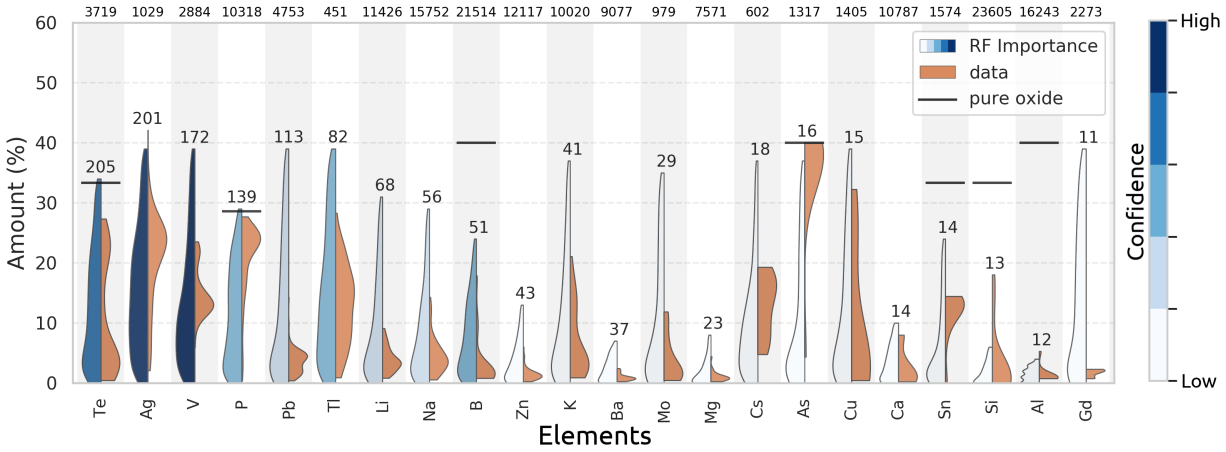
Next, we discuss these results separately for the high T_g and low T_g glasses.

4.3.1 Statistics for high and low T_g glasses

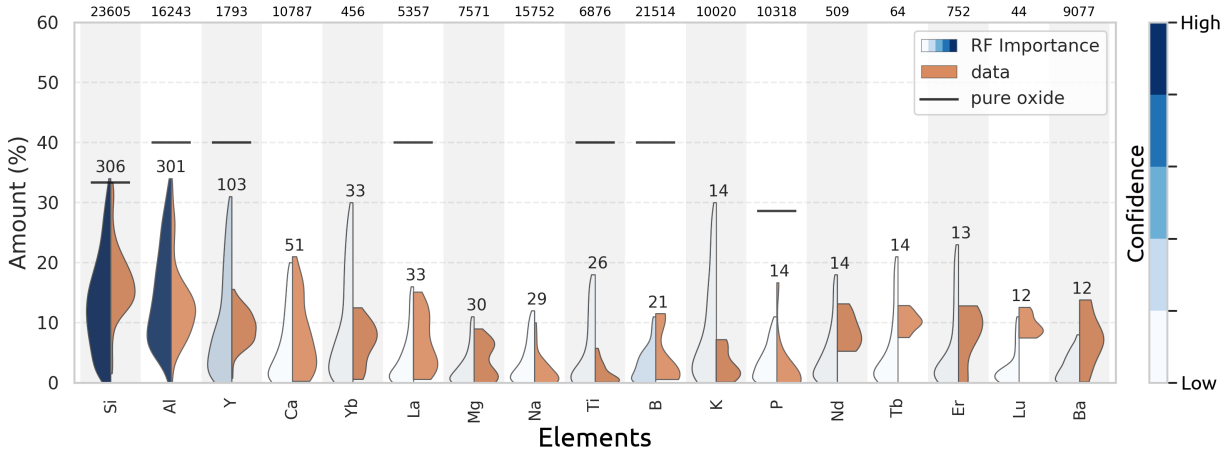
Within the set of 329 glasses that have $T_g \geq 1150\,\mathrm{K}$, 306 contain silicon, 301 contain aluminum, and 103 contain yttrium in their composition. Their amounts in these glasses vary from 0 % to 33 %, 0 % to 33 % and 0 % to 15 %, respectively. These 3 elements are indeed the basis of glass network forming refractory oxides that form high T_g glasses. Other components, including some fluxing agents, such as the alkali and alkaline earth elements, are present in smaller amounts in other glasses of this dataset, as expected. Calcium is an exception because it improves the glass-forming ability of aluminate and yttriate glasses, which can explain its high content in some of these high T_g glasses.

Similar statistics are shown for the 400 low T_g compositions. In this low T_g dataset, 205 formulations contain tellurium, 201 contain silver, and 172 contain vanadium. They are the defining components in this group, only scarcely appearing in the previous set for high T_g glasses (see the full plot with all the elements in the supplementary material). Conversely, the contents of the major components of high T_g glasses (silicon, aluminium, and yttrium) are very small, as expected.

For these two categories of glasses, low and high T_g, it is revealing to note the most frequent amount of each element. We will consider this information in the discussion below.



(a) $T_g \leq 450\,\mathrm{K}$



(b) $T_g \geq 1150\,\mathrm{K}$

Figure 4: Composite violin plot of the (a) 17 most frequent elements in high T_g glasses and (b) 22 most frequent elements in low T_g glasses. Please see text for more information.

4.3.2 Teachings of the RF algorithm — high T_g glasses

According to the analysis of frequency of chemical elements in the RF trees, i.e., the intensity of the color of the left-hand side (LHS), silicon and aluminium are considered by RF to be the most important elements in determining if a glass will have high T_g. The importance for the other glass formers – yttrium and boron – is considered lower than that of silicon and aluminium, but higher than for the rest. It is important to stress that "importance" of an element means that the presence or lack of the element played a significant role in the creation of the RF trees.

The importance given by RF for the remaining elements (Ca, P, Mg, Na, Ti, K, Ba, and rare earths Yb, Nd, Tb, Er, and Lu) is lower, i.e., they appear with low frequency in the RF tree pathways that lead to high T_g glasses. Therefore, one cannot draw a firm conclusion about them with the current analysis. Despite this fact, one can see that RF learned

that only small amounts of these elements should be added to produce high T_g glasses, i.e., the mode of the distributions on the LHS is close to zero.

According to the LHS distribution shown in Figure 4b, the composition range learned by the RF trees for silicon, aluminium, and yttrium is very extensive, from 0 to about $30\,\%$ atomic fraction, with modes of $11\,\%$, $8\,\%$, and $3\,\%$, respectively. For the other glass-forming elements, such as boron, titanium and phosphorous, the learned ranges are more restricted, up to $20\,\%$, with a mode close to zero. This information learned by RF is in line with knowledge in the field.

When comparing the right-hand side (RHS) and the LHD distributions of the split violins, it can be observed that the modes of the respective distributions are dissimilar for Nd, Tb, Er, Lu, and Ba. Thus, there are glasses with higher amounts of these elements than those learned by RF. While this seems to go against the RF analysis, it is important to keep in mind that the confidence of the prediction for these elements is low; and it is expected that some unusual glasses might have peculiar compositions.

4.3.3 Teachings of the RF algorithm- low T_g glasses

At first glance, according to what the models learned by the RF algorithm, low T_g glasses accept higher amounts of network modifier elements (more than $20\,\mathrm{at}\%$) when compared with high T_g glasses. This makes sense as, from a topological point of view, all high T_g glasses must have a well-connected oxygen network. Indeed, it is well known that high T_g glasses should contain only small amounts of network modifiers, whereas the opposite is usual for low T_g glasses.

The models induced by the RF algorithm suggest a range between zero and $33\,\%$ for tellurium, with a mode of $5\,\%$. Indeed, it is known by the glass community that tellurite glasses have low T_g. Another glass network former known for its low T_g is phosphorous. This knowledge is also reflected in the analysis of the models induced by the RF algorithm. The same conclusion can be drawn for the other glass formers of the dataset: vanadium, thallium, and boron.

Alkali ions, such as lithium, sodium, potassium, and cesium seem to have mid-lower to low importance (lighter color) for the induced model. Despite this fact, according to the analysis of the models induced by RF, these elements can be added up to about $30\,\%$, whereas alkaline earth elements, such as calcium, barium, and magnesium, can be added up to about $10\,\%$. This is in line with knowledge in the field.

However, Figure 4a shows intriguing results for another glass former, arsenic. The RF models suggest small additions of this element to make low T_g glasses. However, in the database used, there are 16 low T_g glasses with high arsenic content, usually above $30\,\%$. This is again a reflection of the lower importance given by the model to glasses having this element. On the other hand, despite the fact that the confidence level is low, the induced model suggests that low T_g glasses could host higher amounts of V, Pb, Li, Na, B, Zn, K, Ba, Mo, Cs, Sn, Gd. Finally, the range of Si and Al suggested by the model is close to zero, reflecting that these refractory elements are good for high T_g glasses, corroborating existing knowledge.

5 Final Considerations

In this work, we carried out a large number of experiments evaluating six popular ML algorithms to analyze a dataset of over $43\,240$ oxide glass compositions and their respective glass transition temperatures, T_g. We investigated the performance of these algorithms when used for the prediction of T_g values with default and with tuned hyperparameter values.

We also investigated the importance of using *explainable* models to better understand glass property-composition relationships. The results obtained in this work are supported by statistical tests and pointed out that, as expected, the models induced using tuned hyperparameter values performed better than those induced using default hyperparameter values. The tuned version of the RF algorithm presented the best overall predictive performance for the test examples. For extreme values of T_g, the k-NN algorithm was slightly better than RF. RF also produced an explainable model, which shed light on the individual importance of the chemical elements for developing glasses with very low or very high T_g.

This study can be easily expanded to predict other composition-property combinations, such as thermal expansion coefficient, elastic modulus, hardness, viscosity, and density, to successfully replace empirical approaches for developing novel glasses with useful properties and applications.

Acknowledgement

This study was funded by the São Paulo State Research Foundation, FAPESP, grants 2013/07375-0, (CERTEV), 2013/07793-6 (CEMEAI), 2018/14819-5 (EA), 2018/07319-6 (SMM), 2017/06161-7 (TB), 2017/20265-0 (BAP), and 2017/12491-0 (DRC), the Coordination for the Improvement of Higher Education Personnel (CAPES) and the National Council for Scientific and Technological Development (CNPq), Brazilian funding agencies.

Data availability statement

The glass transition temperature data used in this work comes from the SciGlass database. This database was recently published under an ODC Open Database License (ODbL) at https://github.com/epam/SciGlass. The raw data used in this work is available as supplementary data.

Competing interest statement

The Authors declare no Competing Financial or Non-Financial Interests.

Author contribution statement

DRC collected the data; EA, SMM, and TB trained and tuned the machine learning algorithms, assessed the statistical significance of the experimental results with statistical hypothesis tests and produced the graphical interpretation of the Random Forests. All authors helped in designing, analyzing the results, and writing up the manuscript.

References

[1] Zanotto, E. D. & Mauro, J. C. The glassy state of matter: Its definition and ultimate fate. *Journal of Non-Crystalline Solids* **471**, 490–495 (2017).

[2] Zanotto, E. & Coutinho, F. How many non-crystalline solids can be made from all the elements of the periodic table? *Journal of Non-Crystalline Solids* **347**, 285–288 (2004). URL `http://www.sciencedirect.com/science/article/pii/S0022309304005101`.

[3] Tshitoyan, V. *et al.* Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95 (2019).

[4] Huo, H. *et al.* Semi-supervised machine-learning classification of materials synthesis procedures. *npj Computational Materials* **5**, 62 (2019).

[5] Nascimento, M. L. F., Souza, L., Ferreira, E. B. & Zanotto, E. D. Can glass stability parameters infer glass forming ability? *Journal of Non-Crystalline Solids* **351**, 3296–3308 (2005).

[6] Varshneya, A. K. & Mauro, J. C. *Fundamentals of Inorganic Glasses* (Elsevier, 2019), 3 edition edn.

[7] Cassar, D. R., de Carvalho, A. C. P. L. F. & Zanotto, E. D. Predicting glass transition temperatures using neural networks. *Acta Materialia* **159**, 249–256 (2018). URL `http://www.sciencedirect.com/science/article/pii/S1359645418306542`.

[8] Joyce, S. J., Osguthorpe, D. J., Padgett, J. A. & Price, G. J. Neural network prediction of glass-transition temperatures from monomer structure. *Journal of the Chemical Society, Faraday Transactions* **91**, 2491 (1995). URL `http://xlink.rsc.org/?DOI=ft9959102491`.

[9] Bhadeshia, H. K. D. H. Neural Networks in Materials Science. *ISIJ International* **39**, 966–979 (1999).

[10] Dreyfus, C. & Dreyfus, G. A machine learning approach to the estimation of the liquidus temperature of glass-forming oxide blends. *Journal of Non-Crystalline Solids* **318**, 63–78 (2003). URL `http://linkinghub.elsevier.com/retrieve/pii/S0022309302018598`.

[11] Zhang, Z. & Friedrich, K. Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology* **63**, 2029–2044 (2003). URL `http://linkinghub.elsevier.com/retrieve/pii/S0266353803001064`.

[12] Afantitis, A. *et al.* Prediction of high weight polymers glass transition temperature using RBF neural networks. *Journal of Molecular Structure: THEOCHEM* **716**, 193–198 (2005). URL `http://linkinghub.elsevier.com/retrieve/pii/S0166128004009510`.

[13] Brauer, D. S., Rüssel, C. & Kraft, J. Solubility of glasses in the system P_2O_5–CaO–MgO–Na_2O–TiO_2: Experimental and modeling using artificial neural networks. *Journal of Non-Crystalline Solids* **353**, 263–270 (2007). URL http://linkinghub.elsevier.com/retrieve/pii/S0022309306013135.

[14] Chen, X., Sztandera, L. & Cartwright, H. M. A neural network approach to prediction of glass transition temperature of polymers. *International Journal of Intelligent Systems* **23**, 22–32 (2008). URL http://doi.wiley.com/10.1002/int.20256.

[15] Liu, W. & Cao, C. Artificial neural network prediction of glass transition temperature of polymers. *Colloid and Polymer Science* **287**, 811–818 (2009). URL http://link.springer.com/10.1007/s00396-009-2035-y.

[16] Mauro, J. C., Tandia, A., Vargheese, K. D., Mauro, Y. Z. & Smedskjaer, M. M. Accelerating the Design of Functional Glasses through Modeling. *Chemistry of Materials* **28**, 4267–4277 (2016). URL https://doi.org/10.1021/acs.chemmater.6b01054.

[17] Bishop, C. M. *Pattern recognition and machine learning* (springer, 2006).

[18] Caruana, R. & Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, 161–168 (ACM, 2006).

[19] Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281–305 (2012).

[20] Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Statistics and computing* **14**, 199–222 (2004).

[21] Gilpin, L. H. *et al.* Explaining explanations: An overview of interpretability of machine learning. In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, 80–89 (2018). URL https://doi.org/10.1109/DSAA.2018.00018.

[22] Nature Publishing Group. Towards trustable machine learning. *Nature Biomedical Engineering* **2**, 709–710 (2018). URL https://doi.org/10.1038/s41551-018-0315-x.

[23] Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural networks* **2**, 359–366 (1989).

[24] Dorogush, A. V., Ershov, V. & Gulin, A. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

[25] Weinberger, K. Q. & Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* **10**, 207–244 (2009).

[26] Breiman, L. Random forests. *Machine learning* **45**, 5–32 (2001).

[27] Breiman, L. *Classification and regression trees* (Routledge, 2017).

[28] Cai, A.-h. *et al.* Artificial neural network modeling for undercooled liquid region of glass forming alloys. *Computational Materials Science* **48**, 109–114 (2010). URL http://www.sciencedirect.com/science/article/pii/S0927025609004613.

[29] Steiner, N. Y., Hissel, D., Moçotéguy, P. & Candusso, D. Diagnosis of polymer electrolyte fuel cells failure modes (flooding & drying out) by neural networks modeling. *International journal of hydrogen energy* **36**, 3067–3075 (2011). URL http://www.sciencedirect.com/science/article/pii/S0360319910021853.

[30] Cai, A. H. *et al.* Prediction of critical cooling rate for glass forming alloys by artificial neural network. *Materials & Design (1980-2015)* **52**, 671–676 (2013). URL http://www.sciencedirect.com/science/article/pii/S0261306913005451.

[31] Mannodi-Kanakkithodi, A., Pilania, G., Huan, T. D., Lookman, T. & Ramprasad, R. Machine Learning Strategy for Accelerated Design of Polymer Dielectrics. *Scientific Reports* **6**, 20952 (2016). URL https://www.nature.com/articles/srep20952.

[32] Tripathi, M. K., Chattopadhyay, P. P. & Ganguly, S. A predictable glass forming ability expression by statistical learning and evolutionary intelligence. *Intermetallics* **90**, 9–15 (2017).

[33] Sun, Y. T., Bai, H. Y., Li, M. Z. & Wang, W. H. Machine Learning Approach for Prediction and Understanding of Glass-Forming Ability. *The Journal of Physical Chemistry Letters* **8**, 3434–3439 (2017).

[34] Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. The face of crystals: insightful classification using deep learning. *arXiv:1709.02298 [cond-mat]* (2017). URL http://arxiv.org/abs/1709.02298. ArXiv: 1709.02298.

[35] Ren, F. *et al.* Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Science Advances* **4**, eaaq1566 (2018). URL http://advances.sciencemag.org/content/4/4/eaaq1566.

[36] Anoop Krishnan, N. M. *et al.* Predicting the dissolution kinetics of silicate glasses using machine learning. *Journal of Non-Crystalline Solids* **487**, 37–45 (2018). URL http://www.sciencedirect.com/science/article/pii/S0022309318300905.

[37] Ruusunen, J. *Deep Neural Networks for Evaluating the Quality of Tempered Glass*. M.Sc Dissertation, Tampere University of Technology, Tampere (2018).

[38] Bishnoi, S. *et al.* Predicting young's modulus of glasses with sparse datasets using machine learning. *arXiv preprint arXiv:1902.09776* (2019).

[39] Yang, K. *et al.* Predicting the young's modulus of silicate glasses using high-throughput molecular dynamics simulations and machine learning. *Scientific Reports* **9**, 8739 (2019).

[40] Liu, H., Fu, Z., Li, Y., Sabri, N. F. A. & Bauchy, M. Balance between accuracy and simplicity in empirical forcefields for glass modeling: Insights from machine learning. *Journal of Non-Crystalline Solids* **515**, 133 – 142 (2019). URL http://www.sciencedirect.com/science/article/pii/S002230931930239X.

[41] Mazurin, O. V. & Priven, A. I. *SciGlass - Glass Information System - Glass Database - Glass Properties* (ITC, Inc., 2017). URL http://www.sciglass.info/.

[42] Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, 278–282 (IEEE, 1995).

[43] Borchani, H., Varando, G., Bielza, C. & Larrañaga, P. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **5**, 216–233 (2015).

[44] Siegel, S. & Castellan, N. J. *Nonparametric statistics for the behavioral sciences*, vol. 7 (McGraw-hill New York, 1956).

[45] Bernard, S., Heutte, L. & Adam, S. Influence of hyperparameters on random forest accuracy. In *International Workshop on Multiple Classifier Systems*, 171–180 (Springer, 2009).

[46] Lundberg, S. M. *et al.* Explainable AI for trees: From local explanations to global understanding. *CoRR* **abs/1905.04610** (2019). URL http://arxiv.org/abs/1905.04610. 1905.04610.

[47] Hintze, J. L. & Nelson, R. D. Violin plots: A box plot-density trace synergism. *The American Statistician* **52**, 181–184 (1998).