A regression model for plasma reaction kinetics

OPEN ACCESS
IOP Publishing
Journal of Physics D: Applied Physics
J. Phys. D: Appl. Phys. **56** (2023) 374001 (24pp)
https://doi.org/10.1088/1361-6463/acd390

A regression model for plasma reaction kinetics

Martin Hanicinec[1,2], Sebastian Mohr[1] and Jonathan Tennyson[2,*]

[1] Quantemol Ltd, 320 City Rd, London EC1V 2NZ, United Kingdom
[2] Department of Physics & Astronomy, University College London, Gower St., London WC1E 6BT, United Kingdom

E-mail: j.tennyson@ucl.ac.uk

Received 1 February 2023, revised 3 May 2023
Accepted for publication 9 May 2023
Published 13 June 2023

Abstract
Machine learning (ML) is used to provide reactions rates appropriate for models of low temperature plasmas with a focus on A + B → C + D binary chemical reactions. The regression model is trained on data extracted from the QBD, KIDA, NFRI and UfDA databases. The regression model used a variety of data on the reactant and product species, some of which also had to be estimated using ML. The final model is a voting regressor comprising three distinct optimized regression models: a support vector regressor, random forest regressor and a gradient-boosted trees regressor model; this model is made freely available via a GitHub repository. As a sample use case, the ML results are used to augment the chemistry of a BCl_3/H_2 gas mixture.

Keywords: machine learning, chemical reactions, voting regressor model, plasma chemistry

(Some figures may appear in colour only in the online journal)

1. Introduction

Utilizing the unique properties of the low-temperature plasma has become an integral part of almost every industry sector, spanning over a wide range of applications such as medicine, biotechnology, surface modification, microfabrication, harvesting energy, thrusters, ozone generation or abatement systems, to name just a few. As an example of the importance of the low-temperature plasma technologies for our every day lives, it has been estimated that as much as one-third of steps involved in the manufacturing of microelectronic technologies are plasma-based [1]. While providing desirable properties, the very complex nature of low-temperature plasma systems also poses challenges for describing and understanding plasma phenomena. Understanding the plasma properties is crucial for the optimization of plasma-based processes and technologies and the only way to acquire an insight of any significant depth is through numerical modeling techniques. Therefore, any work aiming to improve modeling of plasma physics phenomena has the potential to carry high impact for the field.

There are many methods available for modeling low-temperature plasma properties and behaviour which vary in both accuracy and complexity. No matter what kind of plasma model of whatever spatial dimensionality is considered, each is built around a chemistry set which describes the volumetric interactions between all the species tracked in the model, and additionally, the interactions between the species and surfaces. A volumetric and surface chemistry set is a very important base for every plasma model, accounting for the majority of sources and sinks of species. Many pre-compiled detailed chemistry sets for various feed gases and applications can be found in the literature, see for example [2–9]. As a consequence of advances in gas kinetics, published chemistry sets are becoming increasingly larger. For plasma physics modeling applications, chemistry sets may routinely include up

* Author to whom any correspondence should be addressed.

to a hundred species and many thousands of reactions. For example, Koelman *et al* [10] provide a chemistry set for the splitting of CO_2 in non-equilibrium plasmas which contains 73 unique species involved in 5724 reactions. In the combustion modeling community, where very large chemistry sets have been used for longer than in the plasma modeling community, some applications require sets that may contain several hundred or even thousands of species and tens of thousands of reactions [11].

A major problem faced by the plasma modelers is the availability of the reaction kinetic data. Each reaction needs to have its kinetic parameters specified and these data are not always readily available. The are a number of online databases of kinetic processes for modeling low-temperature plasmas such as Quantemol Database (QDB) [9], and LXCat [12] which largely contains data on electron collisions, Phys4Entry [13] for modeling atmospheric re-entry plasmas; databases for astrochemical modeling include KIDA [14, 15], UDfA [16] and BASECOL [17]; fusion oriented databases include the Japanese NIFS database [18], the Korean NFRI database [19] and the ALADDIN database maintained by the International Atomic Energy Agency [20]. These databaes provide only a finite and limited set of data. Models of new plasma chemistries (e.g. to model plasma in a novel gas mixture) or which extend an existing chemistry set (e.g. to cover a different range of conditions than those the source chemistry set was compiled for) often cannot find published kinetic data for key reactions.

Under these circumstances plasma modellers often fall back on estimation by analogy or an educated guess. In fact, in a typical published chemistry set, a substantial subset of the reaction kinetics may actually be estimated. For example, Turner [21] performed a review of the state-of-the-art chemistry set for helium–oxygen atmospheric pressure plasmas, carefully tracing the primary sources of kinetic data for all the reactions in the set. He found, that 63 out of the total 373 reactions had estimated kinetic data. This is a high fraction considering that He–O_2 is a fairly simple system and can be expected to have better coverage than more complicated chemistries. The same phenomenon can be observed also in online databases. For example, 1298 reactions in the KIDA database, as well as 869 reactions in the UDfA database share the same reaction rate coefficient $k = 7.5 \times 10^{-8}\,(T/300)^{-0.5}$, and the same reference, pointing to the publication by Harada and Herbst [22]. This paper which, which actually cites Smith *et al* [23] as the data source, only lists a single reaction with this value of k:

$$\mathrm{C_3N^- + C^+ \rightarrow C_3N + C}, \tag{1}$$

All the reactions in KIDA and UDfA pointing to this source are mutual neutralisation reactions of the general form

$$\mathrm{R_1^- + R_2^+ \rightarrow P_1 + P_2\ \ (+P_3 + P_4)}, \tag{2}$$

and their reaction rate coefficients have been estimated by analogy with reaction (1). The practice of making such estimated has a place in plasma modeling, but requires researchers insight and experience which so far has been difficult to algorithmize and automate. An approximate, automated and fast method for estimating of unknown kinetics could be very beneficial; machine learning (ML) offers this possibility.

Given the expense of measuring individual reaction rates, it has been argued, for example by Mason and Tennyson in The 2017 Plasma Roadmap: Low temperature plasma science and technology [24] and by Bartschat and Kushner [25], that the majority of atomic and molecular data required by the plasma modeling community for diverse modeling application is expected to be derived from theoretical calculations. However, such calculations remain expensive and the accuracy needed for reliable quantitative predictions remains a challenge in many cases [25]; theory is therefore still far from providing all the data required by plasma modelers.

ML is already being used very extensively in plasma physics, processing, and modeling, as well as in computational chemistry. A sizable body of research has been done on artificial neural network (ANN) models used as surrogate models for prediction of macroscopic plasma processing outputs (such as etch rate, deposition rate, etc) from the processing reactor control variables, such as RF power, pressure, or feed gas flows. Examples from plasma etch process modeling and real-time process control include, among others, the extensive work of Kim *et al* [26–29], Himmel and May [30], Rietman and Lory [31], Han *et al* [32], Stokes and May [33], or Tudoroiu *et al* [34]. The same is true for other areas of plasma processing. The plasma deposition process control modeling researchers such as Rosen *et al* [35], Bhatikar and Mahajan [36], Chen *et al* [37], or Ko *et al* [38] have also been using ML. ANNs have been further used to model plasma spray processes (e.g. by Guessasma *et al* [39], Jean *et al* [40], and Choudhury *et al* [41]), for modeling of plasma sputtering (e.g. by Krueger *et al* [42] or Kino *et al* [43]), plasma-assisted nanoparticle synthesis (e.g. by Leparoux *et al* [44]), or plasma surface modification (e.g. by Wang *et al* [45], or Abd Jelil *et al* [46]). Finally, there is also a large amount of work dedicated to the utilization of ANNs in any plasma processing generally, such as by Rietman [47], Salam *et al* [48], Molga [49], Kim *et al* [50, 51], or Mesbah and Graves [52].

Apart from modeling plasma processing, control, and diagnostics, ANNs have also been used to augment some traditional quantum chemistry calculation methods. For example, Dral *et al* [53] used ML models to learn the parameters for semi-empirical quantum chemistry calculation methods from molecule structure, while Komp and Valleau [54] used deep ANNs to predict quantum reaction rate constants for simple systems trained on calculated data, to overcome the high cost of *ab initio* calculation. Zhang [55] used ANNs to estimate the standard enthalpies of formation of several kinds of acyclic alkanes, and Hansen *et al* [56] used ML methods for predicting molecular atomization energies. The review paper by Goh *et al* [57] summarizes the use of deep learning in computational chemistry.

Pertinent to the present work, ML techniques were also used in the calculation of chemical kinetics. Ventura *et al* [58] and Galvan *et al* [59] used ANNs for curve-fitting

complex experimental kinetic data, bypassing kinetic models built around chemistry sets altogether. Bas *et al* [60, 61] developed an ANN model for estimating the reaction rates of the catalyzed enzymatic hydrolysis of maltose into glucose, also bypassing a kinetic model. Valeh-e-Sheyda *et al* [62] applied ANN trained on experimental data to estimate the reaction rate of methanol dehydration as a function of temperature, pressure, and the purity of the feed stream. Tumanov and Gaifullin [63] describe ANNs learning the activation energies of reactions of phenyl radicals with hydrocarbons at a single given temperature. Allison [64] trained an ANN to learn to predict rate coefficients of reactions of ·OH radicals from the bonds and bends of the selected set of possible reactants. Choi *et al* [65] discuss the feasibility of activation energy prediction of gas-phase reactions by gradient-boosted trees method from structural and thermodynamical properties of the molecules, as does Grambow *et al* [66] using deep learning. Kuang and Xu [67] showcased the use of a convolutional neural network for the prediction of kinetic triplets for pyrolysis processes from experimental data, more specifically the temperatures at preselected values of conversion degrees. Very similar work has also been done by Huang *et al* [68], and Vieira and Krems [69]. In most cases, a research work intersecting ML and chemical kinetics introduces ANNs and other ML model techniques (or soft computing) as an alternative to the hard kinetic model of a system, which typically integrates the differential equations governing the species densities to calculate the reaction rates. Inputs to such models are typically absorbance, concentration, temperature, pH, etc. This 'soft' approach for chemical kinetics is reviewed nicely in the paper by Amato *et al* [70].

In this work we explore the use of ML to supply unknown reaction rates in plasma chemistries thus allow complete chemistry sets to be generated without resorting to estimation or guesswork.

2. Method

2.1. ML algorithms

In this work we test the use of ML, as implemented in the Scikit-learn [71] Python library, to fill gaps in kinetic data. We concentrate on chemical reactions represented by an Arrhenius form. Almost all regressor classes offered by the Scikit-learn package were tested; three of these regressor classes showed noticeably better performance than the rest. Thus the three regression model classes used here are the Support Vector Machine (SVM) regression model, the Random Forest regression model, and the Gradient-boosted Trees regression model. These are briefly discussed below; the full theory behind these models can be found in standard textbooks on ML such as [72].

A SVM is a class of powerful and versatile algorithms capable of performing linear and non-linear classification and regression. The SVMs were developed by Boser *et al* [73] originally for classification problems. The most common kernels used with SVMs are the linear, polynomial and the Gaussian radial basis function kernels; each kernel has its own set of hyperparameters.

Random forests are among the most powerful and versatile regression and classification ML algorithms available [72]. The simpler decision tree regression model forms a fundamental component of random forests. Decision trees [74, 75] are a class of ML algorithms that can perform both classification and regression. The decision tree recursively splits the dataset into two subsets, building a binary tree of such splits all the way down to the leaf nodes. Each leaf node then corresponds to its range in the feature space and fits all the targets inside this range with a single value y. The decision nodes are built greedily from the root down, and the decision feature and the decision threshold for each decision node are determined by the CART algorithm (Classification and Regression Tree) [74]. For each decision node, the CART algorithm finds the feature and the threshold, which minimizes the weighted mean square error (MSE) for both subsets created by splitting the dataset by that feature and threshold. Instead of training a single decision tree on the whole training dataset, it is possible to train many separate decision tree regressors on random subsets of the training dataset and aggregate the predictions; this is called the random forest [76].

The gradient boosting method was introduced by Brieman [77] and further developed Friedman [78]. Gradient-boosted trees regressor follows a similar idea to random forests, that is, it combine many weak-learning trees to form a single powerful regressor. However, instead of building many trees on different subsets of the training dataset, in the gradient-boosting method the trees are added in a sequence, and each additional tree is trained on the residual errors of the previous tree. The regressors used in present work were trained to estimate the kinetic data from available data belonging to their reactants and products (ranging from trivial, such as charges, to more sparsely available, such as enthalpies of formation).

2.2. Training data

Kinetic parameters for plasma reactions can be found in scientific publications and in online databases. Here we extracted these from various databases. All the data used for training and testing the regression models were automatically scraped from the following databases: QDB [8], NFRI [19], KIDA [14], and UDfA [16]. These four widely-used databases provide a good quantity of kinetic data a for binary heavy-species collisions at or near room-temperature.

We have direct access to the QDB database, so simply queried its underlying relational database structure, which made data extraction simple. The UDfA database provides its raw data as a simple ASCII text file, with a clear structure, documented in the accompanying paper [16], which meant UDfA data could be extracted with a short text parser. The data from NFRI and KIDA databases were extracted using web scraping techniques using python package Scrapy [79], directly from the web user interface. The databases were all scraped in 2020.

The regression model developed here describes binary heavy-species collisions only. In addition, several other data-filtering criteria were established, to further limit the scope of the project. These criteria naturally make the resulting trained

regression model only applicable to a well-defined but fairly narrow set of cases. The full set of criteria for the training/test data set acquisition are summarised as follows.

(i) Only heavy-species reactions are considered. Electron collisions and heavy-species collisions follow completely different dynamics, which would make it impractical to mix them in a single model. Furthermore, electron collisions are usually required for plasma simulations in the form of cross-section, which have a much more complicated form than reaction rate coefficients, which are typically sufficient to represent heavy-species collisions.
(ii) Only binary reactions are considered. This is a practical choice, as the reaction rate coefficient changes units with the number of reactants in the reaction.
(iii) Only reactions with two products are considered as part of the dataset feature-space directly describes the species of the reactions (both reactants and products) and their physical properties; limiting the dataset to only reactions with the same number of reactants and products prevents the problems that arise with datasets that have inherently missing values.
(iv) Reactions involving photons are not considered. All the databases used to source the data support photons as species in their reactions. These, however, make up only a small fraction of the reactions listed, and were therefore excluded from the dataset.
(v) Only reactions involving stateless species are considered. This choice disallows great many reactions from entering the dataset, and limits the applicability of the resulting model considerably. However, the reprentation of the internal states of molecules is beyond the scope of an initial project.

The data collection also ignored any reactions which did not conserve charge, or elemental stoichiometry. Additional considerations for the data collection are discussed below.

The kinetics for heavy-species collisions are represented in QDB by the coefficients of the modified Arrhenius formula, parametrizing the temperature dependence of the reaction rate coefficient by three parameters, α, β, and γ, as

$$k(T) = \alpha \left(\frac{T}{300}\right)^{\beta} \exp\left(-\frac{\gamma}{T}\right). \tag{3}$$

The pre-exponential factor α is mandatory for each reaction, while the parameters β and γ are optional, and are indeed missing for many of the reactions listed. For these reactions, the rate coefficient is simply described as a constant, without any temperature dependence.

In the QDB object model, the reactants and products of each reaction are instances of the species object, which can hold its own properties. The following data were collected for each reaction stored in QDB: the kinetic coefficients α, β, and γ, and for each reactant and product of the reaction their formula, charge, and their enthalpy of formation at normal temperature, if available. QDB does not provide any range of validity its reactions, so every reaction adhering to the criteria listed above was collected. In some cases, QDB contains multiple data for the same reaction; in such cases, the reaction was added multiple times

In contrast to QDB, the NFRI database does not provide Arrhenius parameters for heavy-species reactions; rather it represents the reaction kinetics for each reaction as discrete points of either the reaction rate coefficient, or the cross-section, as a function of temperature. As this work is focussed on cold plasma applications, only those reactions whose kinetic data range overlapped a temperature of $T = 300\ \mathrm{K} \pm 10\%$ were collected. With this filtering criterion applied, only a handful of reactions remained in the cross-sectional form, which were excluded. NFRI does not provide any additional structure around its reactions' species, therefore only the species names (formulas) were collected, together with the reaction kinetics in the form of one or more $[T, k]$ pairs.

The NFRI data presented an additional challenge over units of the reaction rate coefficient. Two different units for reaction rate coefficient data appear in the database: $\mathrm{cm^3{\cdot}s^{-1}}$, and $\mathrm{cm^3{\cdot}mol^{-1}{\cdot}s^{-1}}$. The distributions of rate coefficient values for the two units should be both fairly similar and only differing from each other Avogadro number; however, plot of the data showed otherwise, implying that some of the reactions in the NFRI database must have incorrect rate coefficients units. Units were (re-)assigned following the simple rules:

- if the value of $k(300\ \mathrm{K}) < 10^{-6}$, then k is in $\mathrm{cm^3{\cdot}s^{-1}}$,
- if the values of $k(300\ \mathrm{K}) > 10^{4}$, the unit is $\mathrm{cm^3{\cdot}mol^{-1}{\cdot}s^{-1}}$,
- if $10^{-6} \leqslant k(300\ \mathrm{K}) \leqslant 10^{4}$, the unit cannot be trusted and the reaction is removed.

Another source of data rejection was reactions involving species with ambiguous formulas and charges. Although considerable work was performed parsing as many species from NFRI reactions as possible, due to the fact that species are only represented by their formula string in this database, many species formulas could not be parsed and correctly identified.

Like QDB, the KIDA database represents the heavy-species kinetics using the three parameters, α, β, and γ. However, KIDA supports three different temperature dependence functions for its reaction rate coefficients: the kinetics are parametrized either by the modified Arrhenius formula equation (3), or by one of the formulas for ion–polar systems, describing the rate coefficients for unmeasured reactions between ions and neutral species with a dipole moment, computed using the Su-Chesnavich capture approach [14, 80, 81]:

$$k(T) = \alpha\beta \left(0.62 + 0.4767\gamma \left(\frac{300}{T}\right)^{0.5}\right), \tag{4}$$

or

$$k(T) = \alpha\beta \left(1 + 0.0967\gamma \left(\frac{300}{T}\right)^{0.5} + \frac{\gamma^2}{10.526}\frac{300}{T}\right). \tag{5}$$

Each of the reactions (4), and (5) is defined for a different temperature range, α represents the branching ration of the reaction, β is the Lanagevin rate, while γ determines the temperature dependence for the given temperature range.

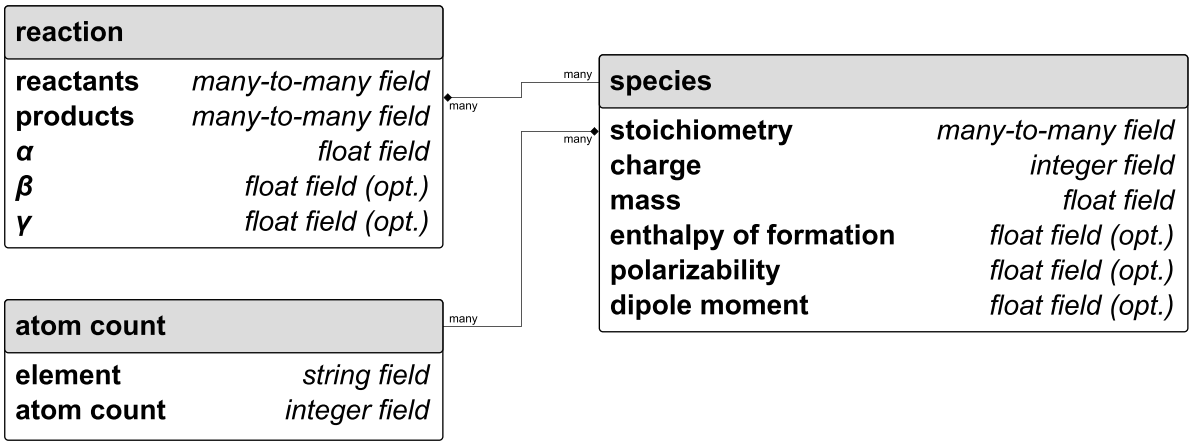

Figure 1. ER diagram showing the relevant attributes of the unified dataset aggregating the data instances collected from all four databases.

The KIDA database also has a species model, and stores additional attributes for each reactant and product of any reaction. For each eligible KIDA reaction, the following data were collected: the kinetic parameters α, β, and γ, the type of reaction rate temperature dependence formula to interpret those parameters, and finally, for each reactant and product of the reaction, mass and charge were collected, and if present, also the enthalpy of formation at the normal temperature, polarizability of the species and its dipole moment.

KIDA also provides 4 tiers of data evaluation, assigning to each reaction one of the following values: not recommended, not rated, valid and recommended. The reactions labeled not recommended were ignored and not added to the dataset, while the reactions with all the other evaluation labels were added and treated equally. KIDA often lists multiple sets of kinetic parameters for a single reaction, and as in the case of QDB, these were all preserved and added into the dataset as individual data instances. Finally, each reaction in KIDA has a valid temperature range attached, and only reactions where this range of validity overlaps with the range of $T = 300\ \mathrm{K} \pm 10\%$ are added to the dataset.

The last database scraped for data was the UMIST Database for Astrochemistry. The kinetics of reactions in the UDfA database is described exclusively by the modified Arrhenius formula of equation (3). Each reaction also has a temperature range of validity, and the criterion for adding the reactions into the datasets was the same as in the case of QDB and KIDA; the temperature range must overlap with a range around the room temperature. No additional species data are provided by UDfA, only strings representing their formulas. This means that the species charges, elemental stoichiometry, and possible states had to be parsed from the formulas.

2.3. Dataset unification

The structure of the final unified dataset, aggregating the data from all the four databases, can be summarized by the entity relationship (ER) diagram given in figure 1 (only the relevant parameters are shown). In this model, every reaction is uniquely identified by two species as reactants, two species as products, and the set of kinetic parameter values, α, and optionally β, and γ. Each species is then uniquely identified by its elemental stoichiometry and a charge.

For simplicity, different isomers having the same elemental composition and charge were collapsed to a single species, characterised by its stoichiometry and charge. As an example, the following three species collected from KIDA with their unique formulas of HNCCC, HCCNC, and HCNCC, were all unified into a single species characterised by the elemental stoichiometry of {'H': 1, 'C': 3, 'N': 1}, and the charge $q = 0$. If the enthalpy of formation $\Delta_\mathrm{f}H^\circ$, the polarizability α, or the dipole moment p was found in KIDA or in QDB for more than one such isomer, the resulting species got assigned the parameters of the isomer with the lowest $\Delta_\mathrm{f}H^\circ$.

Species from all four databases were identified by parsing the (database-specific) species formulas and extracting the elemental stoichiometry and charges from the formula strings. The species were also further validated with the help of the pyvalem python package by Hill [82], and by checking the charge and stoichiometry conservation of the reactions they appear in. The species masses were determined from the elemental stoichiometry and checked against the masses scraped from the databases, adding an additional layer of confidence in correct parsing of the species formulas.

The polarizability and dipole moment species parameters were populated exclusively from the KIDA database, where present. The enthalpy of formation values were being searched for, in order, in the KIDA database, the QDB database, and the NIST-JANAF [83] and ATcT [84] tables, which had previously been scraped by Lu [85].

Finally, in addition to the reaction criteria listed above, two additional criteria for reactions elimination were introduced based on a first analysis of the unified dataset. When creating a training dataset, it helps to eliminate obvious fringe and outlying data instances to increase the data coherence [72]. These additional criteria were:

(i) Only reactions with neutral or singly-ionized species are kept. Doubly ionized species made up only less than 0.3% of all the species in the dataset.

Table 1. The sums of reactions in the final dataset per source database.

Source database	Number of reactions
QDB [8]	1586
NFRI [19]	1171
KIDA [14, 15]	4862
UDfA [16]	1851

(ii) No reactions with free electrons are kept. The associative electron detachment reactions made up only about 1.7% of all the reactions in the dataset, and were therefore eliminated for sake of the dataset coherence.

After removing duplicate reactions the final dataset consists of 9470 reactions involving 1080 distinct species. Table 1 provides the number of reactions in the final dataset sourced from each one of the four databases. The final dataset, following the relational structure depicted in figure 1, is given as `data_final.yaml` file in the project GitHub repository https://github.com/martin-hanicinec-ucl/regreschem.

3. Kinetics regression model

3.1. Targets

First we need to select the outputs ('targets') the model aims to regress. Kinetics for heavy species reactions are usually described by a modified Arrhenius formula equation (3), which parameterizes the temperature dependence of the reaction rate coefficient $k(T)$ by three parameters α, β, and γ. Ideally, these three parameters could be predicted by a multivariate regression model. However, for this to work all the three parameters also need to be present in the training dataset as targets for supervised learning and most of our data sources do not give a full set of Arrhenius coefficients; in practice, the majority of reactions in our dataset are characterized by a single reaction rate constant parameter, α. The kinetic data in the NFRI database [19] are provided as a series of reaction rate coefficient values for different temperatures. In principle, the desired Arrhenius coefficients could be fitted to these data, but this would require at least three data points. However, less than 4% of the NFRI reactions offer 3 or more data points; for more than 90% of NFRI reactions only a single data point is provided which is the reaction rate constant for a temperature within 10% margin around 300 K. For the QDB [8], KIDA [14, 15], and UDfA [16] databases, which offer kinetic data already in Arrhenius form, only about 3% of the reactions selected actually contained all the three Arrhenius parameters.

We therefore decided to limit our regression model to a single-value prediction of a reaction rate constant expressed for $T = 300$ K. In practice we used its logarithm as the rate coefficients need to be well resolved in a range of many orders of magnitude; the trick of target values logarithmization has been used before on a similar topic [54]. Therefore as the targets vector $\vec{y}$, we used the vector of $\log_{10} k(300\ \mathrm{K})$ values expressed for all the reactions in the dataset, with k in $\mathrm{cm^3\ s^{-1}}$.

There were two more uses with the targets which need considering: duplicate reactions, and target capping. The same kinetic data describing a particular reaction appeared in many cases in more than one database. These duplicates were detected based solely on the set of reactants, set of products, and $k(300\ \mathrm{K})$, or the target. While iterating over the dataset, each reaction was removed if it had the same two reactants, the same two products, and the $k(300\ \mathrm{K})$ value within 10% to another reaction present already.

The regression models train to minimize the error measure between the vector of predicted values and the vector of targets, usually the root MSE (RMSE) or the mean absolute error (MAE) [72]. With logarithmic targets, however, it would be a bad strategy to treat all data instances with the identical prediction error equally. Predicting e.g. $k_1^{\mathrm{pred}} = 10^{-5}\ \mathrm{cm^3\ s^{-1}}$ for a data instance with the target of $k_1 = 10^{-7}\ \mathrm{cm^3\ s^{-1}}$ is clearly more significant, than, for example, predicting $k_2^{\mathrm{pred}} = 10^{-25}\ \mathrm{cm^3\ s^{-1}}$ for a data instance with the target $k_2 = 10^{-27}\ \mathrm{cm^3\ s^{-1}}$, even if the two instances will share the same square (and absolute) error in the logarithmic target space. This is because reactions with low rate coefficients will have relatively little impact on the solutions of plasma models. As a workaround we define an effective minimal rate coefficient $k_{\mathrm{min}} = 10^{-20}\ \mathrm{cm^3\ s^{-1}}$. The targets of all reactions with $k < k_{\mathrm{min}}$ were capped to the minimal value of $\log_{10} k_{\mathrm{min}}$. The predicted values were capped the same way, when evaluating different model classes, or when optimizing the model hyperparameters. Figure 2 shows histograms of all the dataset targets before and after capping. The bimodal distribution of k values is discussed in section 3.3.

3.2. Features

We tried to collect data on as many as possible features which might possibly correlate with the reaction rate coefficients being predicted. These data form the raw dataset.

3.2.1. Raw dataset table. The data collected in the raw dataset could be divided into two categories:

(i) **Data describing the individual species:** 26 attributes were collected for each species, totaling 104 columns in the raw dataset table (26 per 2 reactants and 2 products). These attributes are:

- mass m in [amu],
- charge q in [e],
- standard enthalpy of formation $\Delta_{\mathrm{f}}H^{\circ}$ at room temperature $T = 298.15$ K in [$\mathrm{kJ{\cdot}mol^{-1}}$]
- enthalpy of formation of a neutral $\Delta_{\mathrm{f}}H^{\circ}_{n_0}$ describing $\Delta_{\mathrm{f}}H^{\circ}$ of the neutral counterparts to charged species,
- polarizability α in [$\mathrm{Å^3}$],
- dipole moment p in [D],
- number of atoms summed per each block of the periodic table (4 attributes, for 4 blocks: s, p, d, f)
- number of atoms summed per each group of the periodic table (16 attributes, for 16 groups: IA, IB, IIA, …, VIIB, VIIIA, VIIIB).

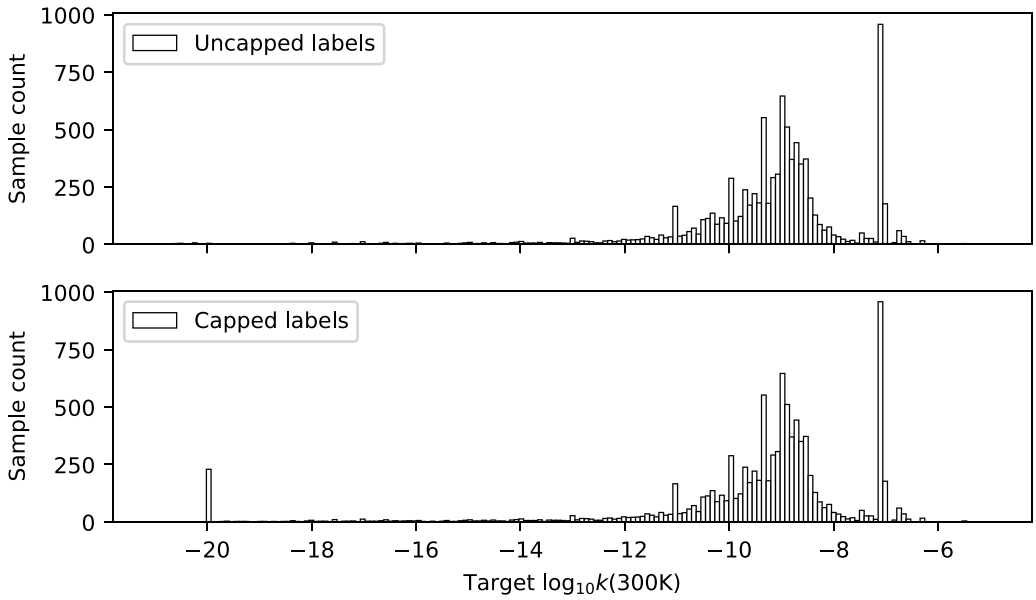

Figure 2. Histograms of all the dataset target values before (top) and after (bottom) capping to $k_{\text{min}} = 10^{-20}\,\text{cm}^3\,\text{s}^{-1}$.

For neutral reactants and products, $\Delta_\text{f}H^\circ_{n_0} = \Delta_\text{f}H^\circ$. The m and q values are naturally fully populated, but the rest of the values are not present in each data instance. The atom counts per block and group are an attempt to encode the elemental composition of the species into the data instances.

(ii) **Data describing the exchanged fragment:** This category of raw dataset table columns regards species fragments exchanged between reactants, in order to create the products. As an example, in the reaction

$$\text{H} + \text{NH} \rightarrow \text{H}_2 + \text{N}, \tag{6}$$

a single H atom is exchanged. The attributes encoding the exchange fragments are the mass, the number of atoms, and the number of atoms per block and group of the periodic table, as in the previous point. This makes in total 22 columns. In some cases, a single fragment is not enough to turn reactants into products, and the values simply sum all the fragments exchanged. In most cases, multiple ways exist to turn reactants into products, and the fragments exchanged with the lowest total mass are picked. So in the reaction

$$\text{OH}^- + \text{CCl}_4 \rightarrow \text{OCl}^- + \text{CHCl}_3, \tag{7}$$

the fragments Cl (passed from CCl_4 to OH^-), and H (passed from OH^- to CCl_4) are selected in favour of fragments O, and CCl_3.

The entire raw dataset table is available in the project repository https://github.com/martin-hanicinec-ucl/regreschem as `dataset_raw.csv`. Apart from the columns described already, several additional columns exist, containing extra metadata about, such as reaction strings (e.g. `'SF4 + SF6- -> SF5 + SF5-'`), the name of the database the reaction instance belonged to (`'qdb'`, `'kida'`, `'umist'`, or `'nfri'`), the doi identifier of the primary source, where available in the database, or the names and source databases for the individual species in each reaction (data instance) line. These columns are not used to construct features in the regression models.

3.2.2. Data imputation. ML algorithms typically can not accept missing data [72]. This is a problem, there are many instances were at least one of the $\Delta_\text{f}H^\circ$, $\Delta_\text{f}H^\circ_{n_0}$, α, or p values is missing for at least one reactant or product. Limiting the dataset to only instances with all the values present would decrease the dataset size considerably. Figure 3 gives an overview of how many data instances are missing which attributes. As an example, well over half of the instances are missing e.g. α at least one of its species, but hardly any instances are missing α for every one of its species. To prevent decreasing the dataset size to less than a half, the missing values must be imputed.

The `IterativeImputer` class, available in the `sklearn.impute` python module [71], was use to regress the missing data in a dataset from all the other attributes. In each iteration, a single column containing some missing data gets filled by an imputation regression model, which is trained on all the other completely populated columns. In this way, the imputation model is just another regression model, which is trained to predict the missing values, in order to produce a complete features matrix. The `IterativeImputer` model can use different regression model classes to perform the imputation; we used the default Bayesian Ridge regressor. The Scikit-learn implementation, the `BayesianRidge` regression model, is based on an algorithm described by Tipping [86] and

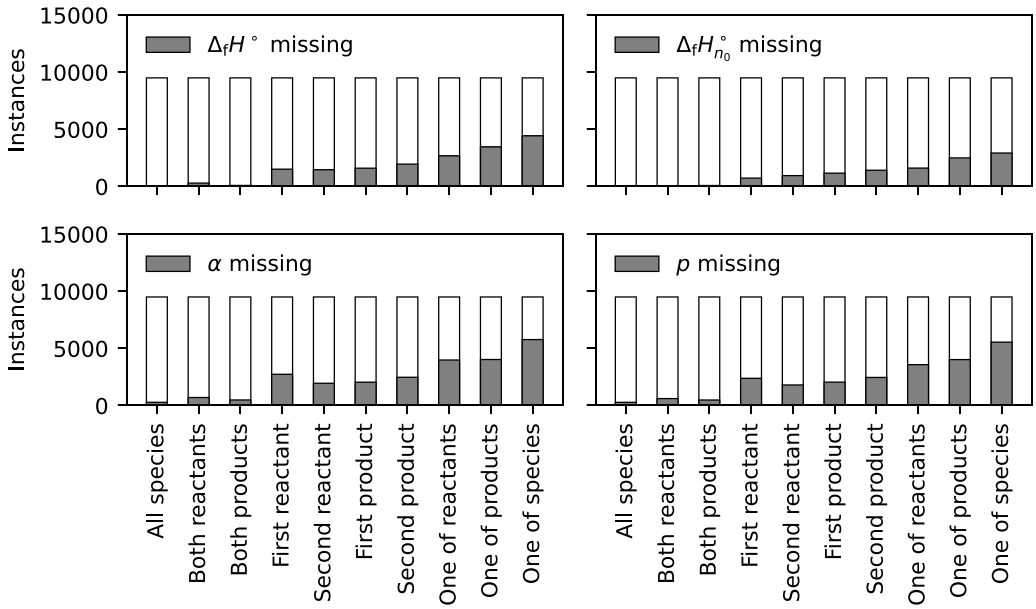

Figure 3. Bar plots showing the fraction of instances in the dataset with missing $\Delta_f H^\circ$, $\Delta_f H^\circ_{n_0}$, α, or p for its reactants and products.

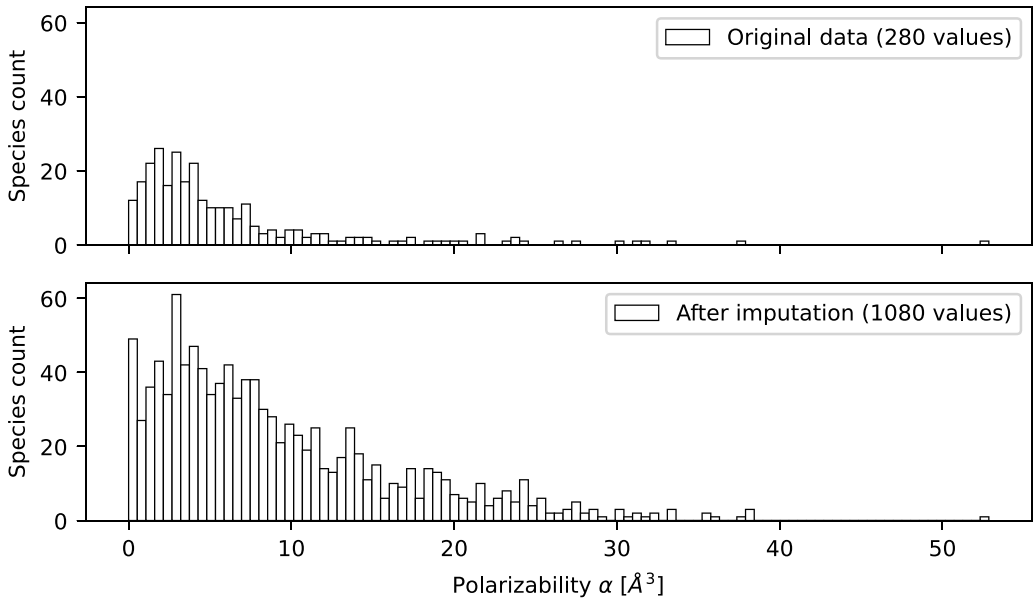

Figure 4. Histograms showing the distribution of α values in the dataset before and after the imputation of missing values.

MacKey [87]. As an illustration, figure 4 shows histograms of the polarizability α before and after imputation of the missing values.

All the data instance attributes described in section 3.2.1 do not yet form the feature matrix for the regression models. Typically, it makes sense to manipulate the values in a way so that the final features utilize some heuristics already known about the system, or some more sensible representations [72]. This manipulation is referred to as feature engineering and is the key to a successful ML model [88].

3.2.3. Feature engineering. As with model selection and hyperparameters tuning, feature engineering is domain-specific and the features matrix $\vec{X}$ needs to be optimized, often iteratively by trial and error [88]. Here we describe our final set of features obtained from a lengthy process of optimization for the lowest prediction errors.

As the order of reactants and products in any reaction is purely a matter of chance or convention, the features encoding attributes of reactants and products should be symmetric with respect to swapping the two reactants (or products).

The features encoding the reaction species were engineered as follows:

- **Masses** m of both reactants were replaced by the reduced mass μ of the left-hand-side (LHS) of the reaction. For a generic reaction

$$\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} + \mathrm{D}, \tag{8}$$

$$\mu_{\mathrm{LHS}} = \frac{m_{\mathrm{A}} m_{\mathrm{B}}}{m_{\mathrm{A}} + m_{\mathrm{B}}}. \tag{9}$$

  The same was done for the products, and the right-hand-side (RHS) of any reaction.
- **Charges** q of both reactants were replaced by a series of one-hot encoded charge combinations. For the reaction left-hand-side, this resulted in three boolean-valued features: Q^{00}_{LHS}, Q^{+0}_{LHS}, and Q^{+-}_{LHS}. As an example, for the generic reaction (8),

$$Q^{+0}_{\mathrm{LHS}} = \begin{cases} 1 \text{ if } (q_{\mathrm{A}} = 0 \text{ and } q_{\mathrm{B}} = 1) \text{ or } (q_{\mathrm{A}} = 1 \text{ and } q_{\mathrm{B}} = 0) \\ 0 \text{ otherwise.} \end{cases} \tag{10}$$

  The reactant charges are converted by the same token in the features Q^{00}_{RHS}, and Q^{+0}_{RHS}. The charge combinations characterized by the features Q^{-0}_{LHS} and Q^{-0}_{RHS}, were dropped from the dataset as there were very few collisions between neutrals and negative ions present in the dataset so nearly all values for these features were zero.
- **Enthalpy of formation** $\Delta_{\mathrm{f}}H^{\circ}$ values for both reactants and products were turned into the enthalpy of formation of each side of the reaction. For the generic reaction (8), this made two features

$$\Delta_{\mathrm{f}}H^{\circ}_{\mathrm{LHS}} = \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{A}} + \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{B}}, \tag{11}$$

$$\Delta_{\mathrm{f}}H^{\circ}_{\mathrm{RHS}} = \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{C}} + \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{D}}. \tag{12}$$

  Additionally, the total enthalpy of formation for the whole reaction was explicitly added as a feature

$$\Delta_{\mathrm{f}}H^{\circ}_{\mathrm{total}} = \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{RHS}} - \Delta_{\mathrm{f}}H^{\circ}_{\mathrm{LHS}}. \tag{13}$$

  The $\Delta_{\mathrm{f}}H^{\circ}_{n_0}$ values were manipulated exactly the same way.
- **Polarizability** values α were turned into 2 distinct features

$$F^{\alpha}_{\mathrm{LHS}} = \alpha_{\mathrm{A}}|q_{\mathrm{B}}| + \alpha_{\mathrm{B}}|q_{\mathrm{A}}|, \tag{14}$$

$$F^{\alpha}_{\mathrm{RHS}} = \alpha_{\mathrm{C}}|q_{\mathrm{D}}| + \alpha_{\mathrm{D}}|q_{\mathrm{C}}|, \tag{15}$$

  following the species naming convention from the generic reaction (8). The choice of these features is motivated by the fact that the electrostatic force between a charged and a polar particle will, in the first approximation, be proportional to the product of the charge and the polarizability of the particles.
- **Dipole moment** values p of all the reactants and products were turned into the features F^{p}_{LHS} and F^{p}_{RHS}, following the same reasoning used for the polarizability: the electrostatic force between a charged particle and a particle with a dipole moment will be roughly proportional to the product of p and square of the charge, therefore

$$F^{p}_{\mathrm{LHS}} = |p_{\mathrm{A}}|q^2_{\mathrm{B}} + |p_{\mathrm{B}}|q^2_{\mathrm{A}}, \tag{16}$$

$$F^{p}_{\mathrm{RHS}} = |p_{\mathrm{C}}|q^2_{\mathrm{D}} + |p_{\mathrm{D}}|q^2_{\mathrm{C}}. \tag{17}$$

- Finally, the species attributes describing the **elemental composition** of the reactants and products were all collapsed into just 7 features: $N^{\mathrm{bl.=s}}$, $N^{\mathrm{bl.=p}}$, $N^{\mathrm{gr.=IA}}$, $N^{\mathrm{gr.=IVA}}$, $N^{\mathrm{gr.=VA}}$, $N^{\mathrm{gr.=VIA}}$, $N^{\mathrm{gr.=VIIA}}$. For an explanation by example, $N^{\mathrm{bl.=s}}$ is the number of atoms appearing on the LHS of the reaction, which belong to the s block of the periodic table of elements. There are very few species in the dataset made of elements belonging to the d block and none of elements belonging to the f block. Therefore, only the features describing the s and p blocks were retained in the features matrix. Similarly, the vast majority of species in the dataset are composed of elements belonging to one of the IA, IVA, VA, VIA, VIIA groups of the periodic table. All the other groups were dropped from the features space. All the 7 features described are evaluating the numbers of atoms found on the LHS of any reaction only. As each reaction conserves the species stoichiometry, the features belonging to RHS is identical so are not needed.

Apart from the features encoding the reactants and products, there are 9 more features describing the elements exchanged between the two reactants in order to create the two products. Following the same nomenclature as in the list above, these features are fairly self-evident: m_{X}, N_{X}, $N^{\mathrm{bl.=s}}_{\mathrm{X}}$, $N^{\mathrm{bl.=p}}_{\mathrm{X}}$, $N^{\mathrm{gr.=IA}}_{\mathrm{X}}$, $N^{\mathrm{gr.=IVA}}_{\mathrm{X}}$, $N^{\mathrm{gr.=VA}}_{\mathrm{X}}$, $N^{\mathrm{gr.=VIA}}_{\mathrm{X}}$, $N^{\mathrm{gr.=VIIA}}_{\mathrm{X}}$. Here, X refers to a hypothetical particle made of the exchanged elements (see section 3.2.1), and N_{X} is simply a number of atoms of X, no matter which block or group.

Table 2 shows the final list of features forming the features matrix $\vec{X}$ in this work. Also shown are the feature names consistent with the code in the project repository, and the features data types. In total, 33 features were used.

Finally, scale sensitivity is typically handled by applying a scaling to all the numeric features [72]; this was done by adding the `StandardScaler` instance from `sklearn.preprocessing` module [71] into the data transformation pipeline. The standard scaler subtracts the mean from each feature column and scales all the values to unit variance. Figure 5 shows the distribution of the final $\Delta_{\mathrm{f}}H^{\circ}_{\mathrm{total}}$ feature (using the $\Delta_{\mathrm{f}}H^{\circ}$ values after imputation) with different horizontal axes belonging to the original and standard-scaled feature data.

3.3. Dataset analysis

As mentioned above, duplicate reactions were identified as identical reactions with very close reaction rate coefficients

Table 2. The final list of features; also given are the feature names used in the project repository code and to the data types of all the features values.

Symbol	Feature name	Data type
$\Delta_f H^\circ_{total}$	`delta_hform`	`real`
$\Delta_f H^\circ_{n_0,total}$	`delta_hform_neutral`	`real`
Q^{00}_{LHS}	`lhs_charge_00`	`boolean`
Q^{+0}_{LHS}	`lhs_charge_+0`	`boolean`
Q^{+-}_{LHS}	`lhs_charge_+-`	`boolean`
μ_{LHS}	`lhs_mu`	`real`
$\Delta_f H^\circ_{LHS}$	`lhs_hform`	`real`
$\Delta_f H^\circ_{n_0,LHS}$	`lhs_hform_neutral`	`real`
F^α_{LHS}	`lhs_polarizability_factor`	`real`
F^p_{LHS}	`lhs_dipole_moment_factor`	`real`
$N^{bl.=s}$	`lhs_block_s`	`integer`
$N^{bl.=p}$	`lhs_block_p`	`integer`
$N^{gr.=IA}$	`lhs_group_IA`	`integer`
$N^{gr.=IVA}$	`lhs_group_IVA`	`integer`
$N^{gr.=VA}$	`lhs_group_VA`	`integer`
$N^{gr.=VIA}$	`lhs_group_VIA`	`integer`
$N^{gr.=VIIA}$	`lhs_group_VIIA`	`integer`
Q^{00}_{RHS}	`rhs_charge_00`	`boolean`
Q^{+0}_{RHS}	`rhs_charge_+0`	`boolean`
μ_{RHS}	`rhs_mu`	`real`
$\Delta_f H^\circ_{RHS}$	`rhs_hform`	`real`
$\Delta_f H^\circ_{n_0,RHS}$	`rhs_hform_neutral`	`real`
F^α_{RHS}	`rhs_polarizability_factor`	`real`
F^p_{RHS}	`rhs_dipole_moment_factor`	`real`
m_X	`exchanged_mass`	`integer`
N_X	`exchanged_atoms`	`integer`
$N_X^{bl.=s}$	`exchanged_block_s`	`integer`
$N_X^{bl.=p}$	`exchanged_block_p`	`integer`
$N_X^{gr.=IA}$	`exchanged_group_IA`	`integer`
$N_X^{gr.=IVA}$	`exchanged_group_IVA`	`integer`
$N_X^{gr.=VA}$	`exchanged_group_VA`	`integer`
$N_X^{gr.=VIA}$	`exchanged_group_VIA`	`integer`
$N_X^{gr.=VIIA}$	`exchanged_group_VIIA`	`integer`

and were filtered out of the dataset. However, the dataset still contains many reactions which share the same reactants and products but have significantly different reaction rate coefficients. Those might be sourced either from different databases, or from the same database, but from different source publications. In some cases, the reaction rate coefficients for identical reactions differ vastly across different data samples. Figure 6 shows different target values found in the dataset for each one of three chosen reactions. As the data samples belonging to a single reaction will share the features vector, those form conflicting data samples in the dataset.

It can be seen from figure 6, that the difference in reaction rate coefficient k between two conflicting training data samples can in some instances be over 10 orders of magnitude. The difference between the maximal and minimal value was evaluated for each reaction having conflicting reaction rate coefficients in the dataset. Out of the total of 1916 reactions with multiple k values, the vast majority of reactions have fairly consistent k values within a single order of magnitude. There are, however, a significant number of samples with a much wider spread. This naturally has implications for the limits of how well any regression model can actually perform.

Another thing worth analyzing is the distribution of target values. Figure 2 showed the overall distribution of k values across the dataset with a distinct bimodal appearance. The individual peaks of the bimodal distribution correlate with the charge combinations of reactants or with the features Q^{00}_{LHS}, Q^{+0}_{LHS}, Q^{+-}_{LHS}, as can be seen in figure 7. The rate coefficients belonging to reactions of two neutrals ($Q^{00}_{LHS} = 1$) and to neutral–ion reactions ($Q^{+0}_{LHS} = 1$) together form the first, broader peak, while the reactions between positive and negative ions ($Q^{+-}_{LHS} = 1$) form the second, tighter peak. Most of the anion–cation collisions in the dataset are mutual neutralization reactions.

A closer look at the anion–cation collisions samples reveal two populous reaction rates. First, as discussed above, 953 reactions share the same target value, corresponding to the reaction rate coefficient

$$k(T) = 7.5 \times 10^{-8}\ (T/300)^{-0.5}\ \mathrm{cm^3\ s^{-1}},$$

and all appear to be a generalization of a single mutual neutralization reaction (1) sourced from Harada and Herbst [22]. Second, 166 reactions, all acquired from QDB, share the same value of $k = 1.0 \times 10^{-7}\,\mathrm{cm^3\,s^{-1}}$.

3.4. Training the model

The performance of our trained ML model is measured as a prediction error scored on a set of data using the error functions are RMSE and MAE functions. As the predicted target values were capped to y_{min}, corresponding to $k_{min} = 1 \times 10^{-20}\,\mathrm{cm^3\,s^{-1}}$, the error functions are defined by:

$$\mathrm{RMSE}\left(\vec{y}, \vec{y^{pred}}\right) = \sqrt{\frac{\sum_{i=1}^{N}\left[y_i - \mathrm{cap}\left(y_i^{pred}\right)\right]^2}{N}}, \tag{18}$$

$$\mathrm{MAE}\left(\vec{y}, \vec{y^{pred}}\right) = \frac{\sum_{i=1}^{N}\left|y_i - \mathrm{cap}\left(y_i^{pred}\right)\right|}{N}, \tag{19}$$

where

$$\mathrm{cap}\,(y) = \begin{cases} y_{min} = -20 & \text{if } y < y_{min}, \quad (20) \\ y & \text{otherwise.} \quad (21) \end{cases}$$

There, $\vec{y}$ and y_i refer to the known target values, while $\vec{y^{pred}}$ and y_i^{pred} are the values predicted by the model. N is the number of data samples the prediction error is evaluated on. Note that the known target values $\vec{y}$ are already capped at y_{min}.

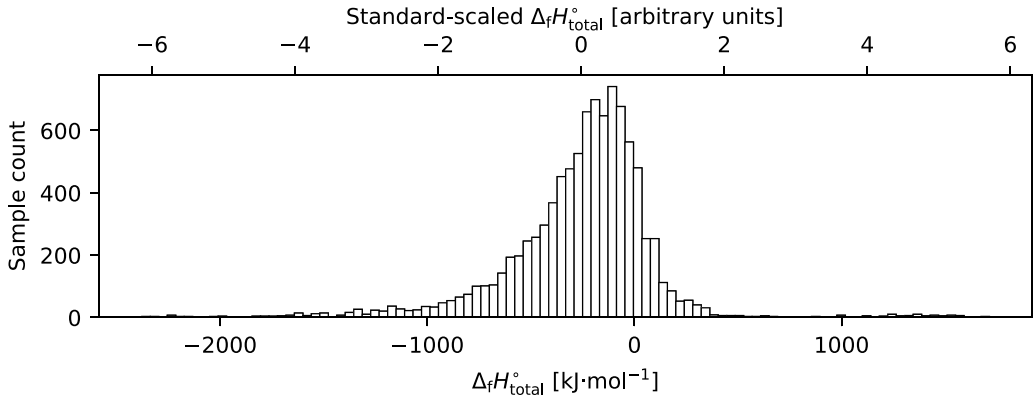

Figure 5. Histogram showing the distribution of the $\Delta_f H^\circ_{\text{total}}$ feature values in the original unit space (bottom horizontal axis), and in the rescaled space (top horizontal axis).

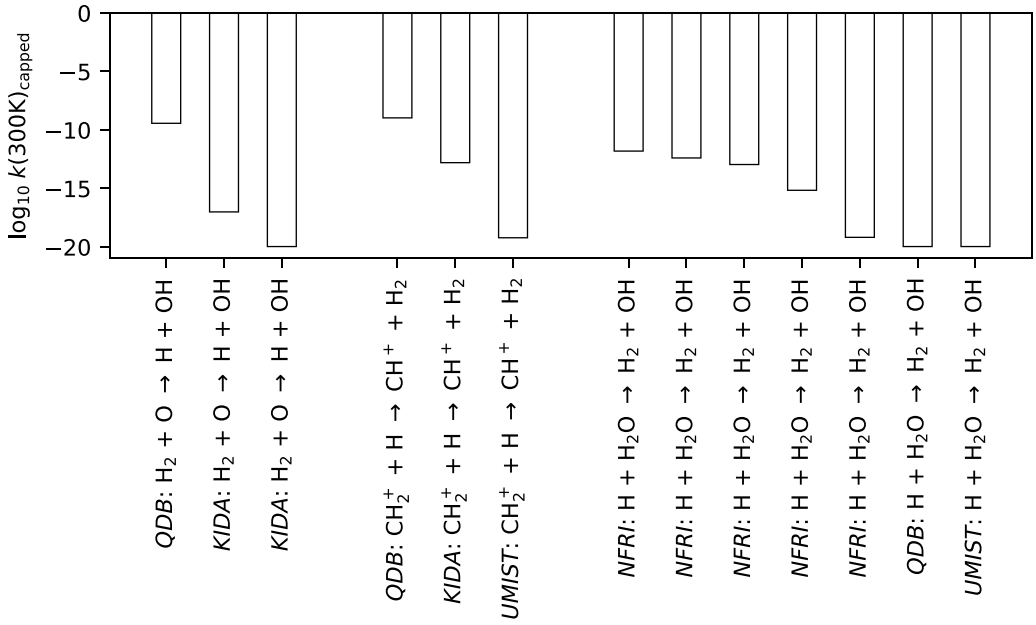

Figure 6. Different target values found in the dataset for three chosen reactions.

We followed the standard practice of splitting our dataset into training and a test set with a training error for the training set, and a generalization or out-of-sample error [72] evaluated using the test set. A low training error and high generalization error is indicative of over-fitting [72]. Here we withheld 20% of a randomly selected samples as the test set giving a training set of 7576 reactions and a test set of 1894 reactions.

To overcome potential biases an n-fold cross-validation technique [89] is very often used. In n-fold cross-validation, the training set is split into n non-overlapping subsets, and each subset is used both as a training and validation set, in a sequence of n trials. We optimized the hyperparameters for the selected model classes by minimizing the mean validation error of a 5-fold cross-validation. Two techniques were used predominantly: grid search and randomized search. Hyperparameters were optimized for all three shortlisted regression models (support vector regressor, random forest regressor, and gradient-boosted trees regressor). The optimization was carried out using the mean MAE error given in equation (19) over 5-fold cross-validation, with some attention to the difference between training and validation errors. We chose to optimize for MAE, as the RMSE of equation (18) is more sensitive to outlier samples, which are definitely present in the dataset, as shown in figure 6. Finally, the three optimized models were combined into a single voting regression model, with the optimized vector of weights of the constituent models, as the only hyperparameter. For the full reproducibility, the optimized regression models and their hyperparameters as a code snippet in figure 8.

The mean cross-validation MAE errors μ_{MAE} for each model are listed in table 3, together with the standard deviations σ_{MAE} over the 5-fold cross-validation trials.

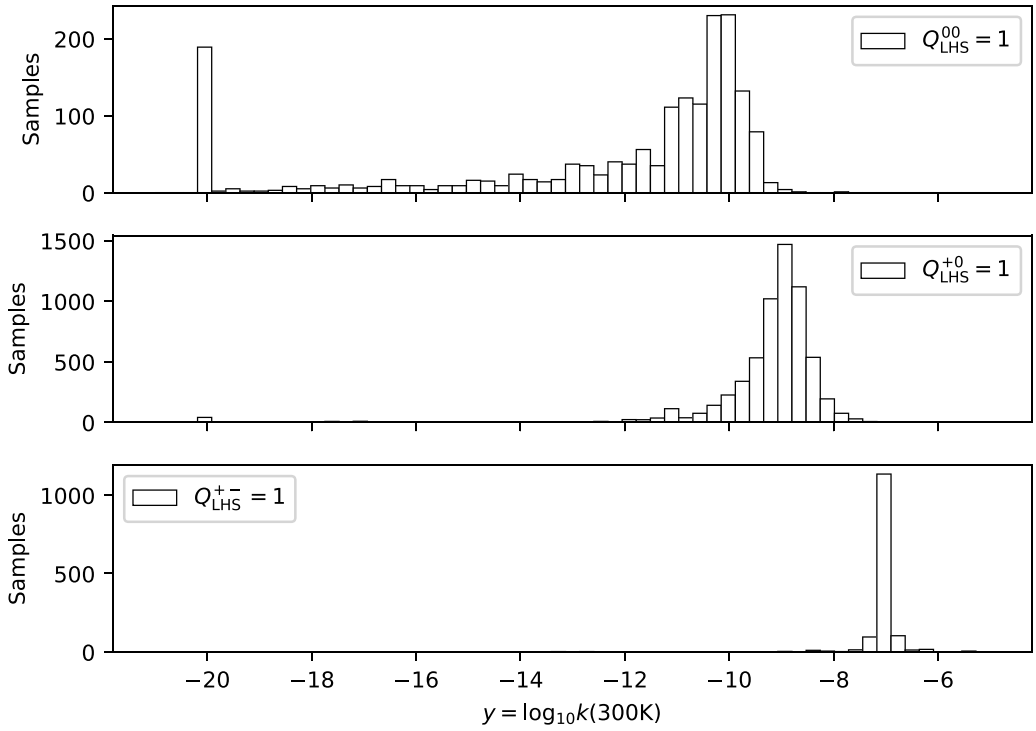

Figure 7. Distributions of k values in the dataset shown on three separate histograms for neutral–neutral collisions ($Q^{00}_{\mathrm{LHS}} = 1$), neutral–cation collisions ($Q^{+0}_{\mathrm{LHS}} = 1$), and finally anion–cation collisions ($Q^{+-}_{\mathrm{LHS}} = 1$).

Results for two more models are listed as benchmarks. `LinearRegression` model [71] simply performs a linear regression in the features space. `MedianEstimator` model is an extremely naive custom estimator, which simply assigns each sample from the validation set (or test set) with the unknown target value the value of the median of all the known target values from the training set (while completely ignoring the features matrix). The results in table 3 were obtained with `scikit-learn` version 0.24.2 and with random (but repeatable) train/test, and train/validation splits. All the code is available as a Jupyter notebook [90] in the project repository https://github.com/martin-hanicinec-ucl/regreschem for full reproducibility.

4. Results and discussion

Table 4 shows the $\mathrm{MAE}^{\mathrm{test}}$ error evaluated on the test set obtained using regression model optimized in its final form (figure 8). For comparison, the error measures of the two benchmark estimators are listed in the table. Our final model is the `VotingRegressor` instance with the optimal hyperparameters. Also shown for comparison next to the $\mathrm{MAE}^{\mathrm{test}}$ values are the mean cross-validation errors $\mu_{\mathrm{MAE}^{\mathrm{val}}}$ from table 3. The value of $\mathrm{MAE}^{\mathrm{test}} = 0.593$ means that there is a bit more than half order of magnitude average difference between the reaction rate coefficients for the reactions of the test set predicted by the final model, and their known target values.

The test errors are slightly but significantly lower than the mean cross-validation errors; this is surprising and in general improbable. Thorough hyperparameters tuning will typically overfit to the training subset data instances, making the cross-validation errors (evaluated on the training set) typically lower than the test set error [72]. It is a hallmark of a well-trained and optimized model, that the test error is very close to the validation error (while both being as low as possible), but typically the test error is higher than the validation error. This anomaly can be explained by looking at not just the mean cross-validation error $\mu_{\mathrm{MAE}^{\mathrm{val}}}$, but at the individual validation errors of the cross-validation folds $\mathrm{MAE}^{\mathrm{val}}_1$—$\mathrm{MAE}^{\mathrm{val}}_5$. The individual folds validation errors are shown in figure 9, together with the test error $\mathrm{MAE}^{\mathrm{test}}$.

The MAE errors for individual cross-validation folds differ considerably between folds, which are trained on subsets randomly drawn from the same training set. It is possible, that the hyperparameters of the final voting regressor (and its constituent models) were optimized conservatively enough not to cause over-fitting to the training set, and at the same time, the withheld test just by chance consists of data instances responding to the final trained model exceptionally well. As discussed below, the whole test (and training) dataset can be split into various subsets, each with significantly different own test (and validation) errors. For a concrete example, the neutral–neutral reactions subset of the test set has much higher MAE than the cation–anion reactions subset, reactions of which get predicted

```
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import VotingRegressor

model_svr = SVR(
    C=5,
    epsilon=0.2
)
model_rfr = RandomForestRegressor(
    max_depth=14,
    max_leaf_nodes=512,
    min_samples_split=2,
    min_samples_leaf=10,
    n_estimators=200,
    random_state=42
)
model_gbr = GradientBoostingRegressor(
    max_depth=8,
    max_leaf_nodes=32,
    max_features=9,
    min_samples_split=6,
    min_samples_leaf=15,
    n_estimators=50,
    random_state=42
)
# ------------------------------------------------- #
model_final = VotingRegressor(
    estimators=[('svr', model_svr),
                ('rfr', model_rfr),
                ('gbr', model_gbr)],
    weights=[5, 2, 4]
)
```

Figure 8. Python code snippet showing instantiation of the three regression model classes with their optimized hyperparameters, together with the final voting regressor combining them into a single regression model. Where not stated explicitly, the default values of hyperparameters were used.

by the final model relatively precisely. In this case, the final MAE error measure might be quite sensitive to the ratio of neutral–neutral reactions and cation–anion reactions among the data instances, which can be more favorable for the test set than for the training set, just by the act of the random train/test split.

Table 3. table of mean MAE cross-validation errors and their standard deviations for all the three shortlisted optimized models, as well as the final voting regressor model and two naive benchmark models.

Model	μ_{MAE}	σ_{MAE}
`MedianEstimator`	1.329	0.031
`sklearn.linear_model.LinearRegression`	0.992	0.021
`sklearn.svm.SVR`	0.673	0.021
`sklearn.ensemble.RandomForestRegressor`	0.679	0.017
`sklearn.ensemble.GradientBoostingRegressor`	0.668	0.020
`sklearn.ensemble.VotingRegressor`	0.646	0.018

Table 4. Final MAE evaluated on the test set shown for the final regression model together with two basic models shown as benchmarks. The values are shown in comparison to the mean cross-validation errors, listed in table 3 already.

Model	$\mu_{\mathrm{MAE^{val}}}$	$\mathrm{MAE^{test}}$
Median estimator	1.329	1.278
Linear regression	0.992	0.955
Final model	0.646	0.592

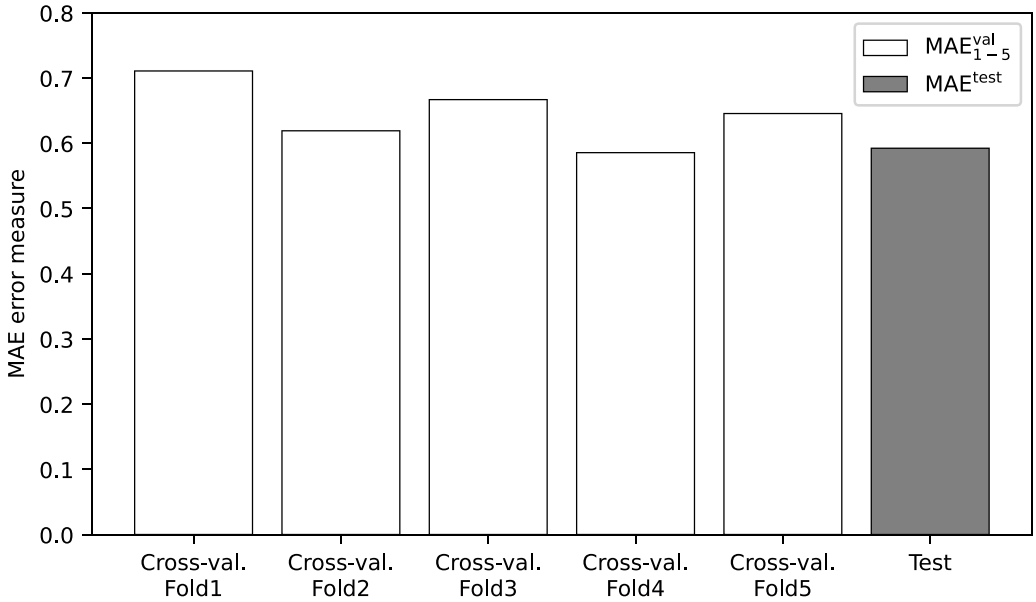

Figure 9. Comparison of the final generalization MAE error measure on the withheld test set with errors of the individual cross-validation folds.

Table 5. MAEs evaluated on different subsets of the test set showing that the combination of charges among the two reactants has a great influence on the mean absolute prediction error.

Test subset	Instances	MAE
All	1849	0.592
Neutral–neutral	336	1.289
Ion–neutral	1271	0.509
Cation–anion	287	0.143

4.1. Analysis of reactants charge combinations

Table 5 gives average prediction errors for different charge combinations of the reactants. Figure 10 illustrates the distributions of prediction errors plotted for each of the subsets from table 5. It is evident that different charge combinations among the reactants translated into different mean prediction errors. The very low MAE error measure for the cation–anion reactions is hardly a surprise. This subset had a very tight distribution of target values in the first place, tightly centered around a single value, as discussed in section 3.3 and shown in figure 7. The final regression model evidently recovered this tight distribution fairly well. The fact that neutral–neutral reaction rate coefficients span a larger range than the predominantly fast ion–neutral collisions is probably the reason that they are predicted by the model with much higher errors.

As the neutral–neutral, ion–neutral, and ion–ion collisions have such obviously different prediction error distributions, as well as target values distributions (figure 7), it was worth exploring the idea of training a dedicated regression model for each of those subsets. Unfortunately, this did not lead to any lower prediction errors. Tests showed that the distribution of

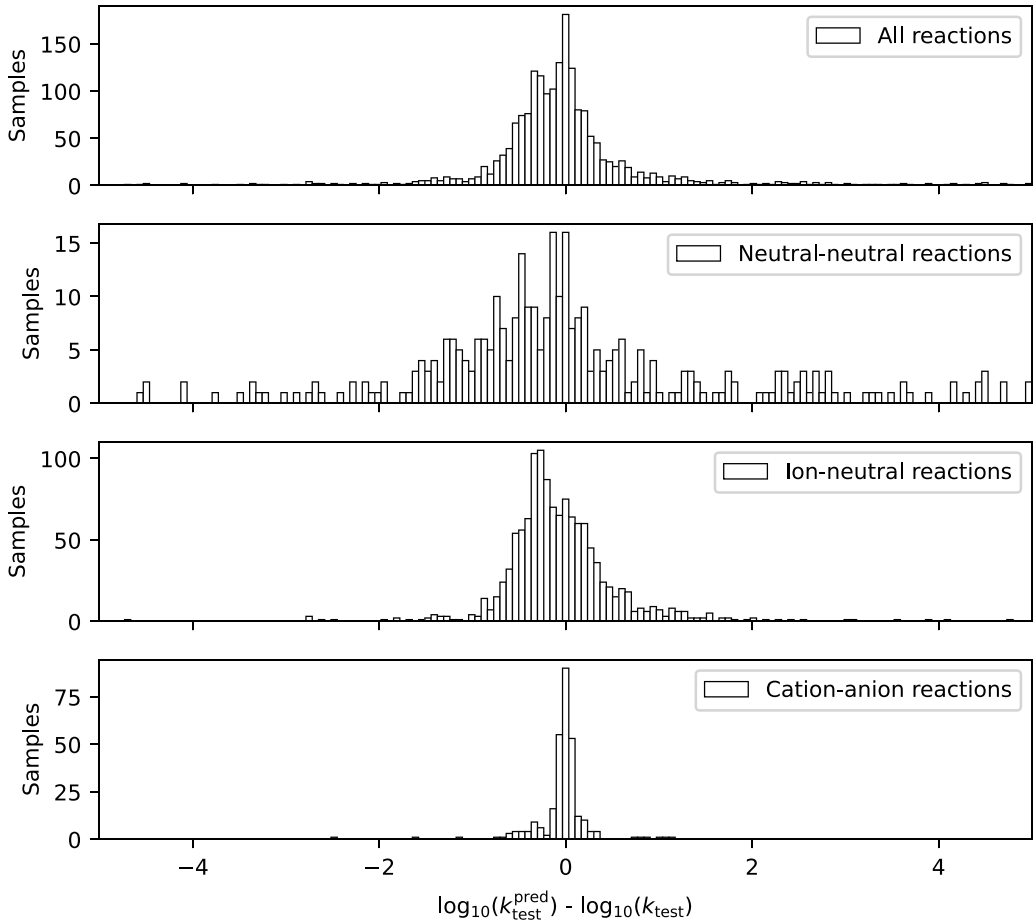

Figure 10. The distribution of prediction errors plotted for different subsets of the test set. It can be seen that the combination of charges among the two reactants has a great influence on the prediction errors.

target values in the dataset as a whole and in the test set were similar.

4.2. Analysis of feature importance

The regression models based on decision trees (random forest and gradient-boosted trees in our case) offer a measure of feature importance as the average depth any particular feature appears as a decision node across all the constituent trees of the ensemble. Figure 11 shows the feature importance measure for every single feature in the dataset, as assessed by two parts of the final voting regressor: the random forest regressor and the gradient-boosted trees regressor.

Three interesting facts can be noted about the feature importance values shown in figure 11. Firstly, the features encoding the properties of reactants (`lhs_` prefix) appear to be more relevant for predicting the reaction rate coefficients, than the features encoding the properties of products (`rhs_` prefix). Notably, the `rhs_polarizability_factor` ($F^{\alpha}_{\mathrm{RHS}}$) and the `rhs_dipole_moment_factor` ($F^{p}_{\mathrm{RHS}}$) features, see equations (15), (17), both appear to be completely irrelevant for predicting the rate coefficients, while their reactants counterparts ($F^{\alpha}_{\mathrm{LHS}}$, F^{p}_{LHS}) proved to be somewhat important. Furthermore, the features designed to encode the fragments exchanged between reactants (`exchanged_` prefix, see section 3.2.3 and table 2) all appear to be almost completely ignored by the model when predicting rate coefficients. Lastly, it can be seen that the boolean features explicitly encoding the charge combinations among reactants and products, `lhs_charge_00`, `lhs_charge_+0`, `lhs_charge_+-`, `rhs_charge_00`, `rhs_charge_+0` (or Q^{00}_{LHS}, Q^{+0}_{LHS}, Q^{+-}_{LHS}, Q^{00}_{RHS}, Q^{+0}_{RHS} respectively), are being completely ignored by the random forest and gradient-boosted trees regressors. And yet, the distinct distributions of rate coefficients for categories represented by different values of those features were correctly recovered in the predicted rate coefficient values. This implies, that the same information (distinguishing the $Q^{00}_{\mathrm{LHS}} = 1$, $Q^{+0}_{\mathrm{LHS}} = 1$, and $Q^{+-}_{\mathrm{LHS}} = 1$ cases) must have been encoded implicitly by other features, assessed as more important by the final regression model, such as $F^{\alpha}_{\mathrm{LHS}}$ in equation (15) or F^{p}_{LHS} in equation (17).

4.3. Analysis of the biggest outliers

Figure 10 clearly shows that some of the test set instances (mainly belonging to the neutral–neutral category) were predicted by the model with some significant prediction errors. Tables 6 and 7 show the ten reactions with the most

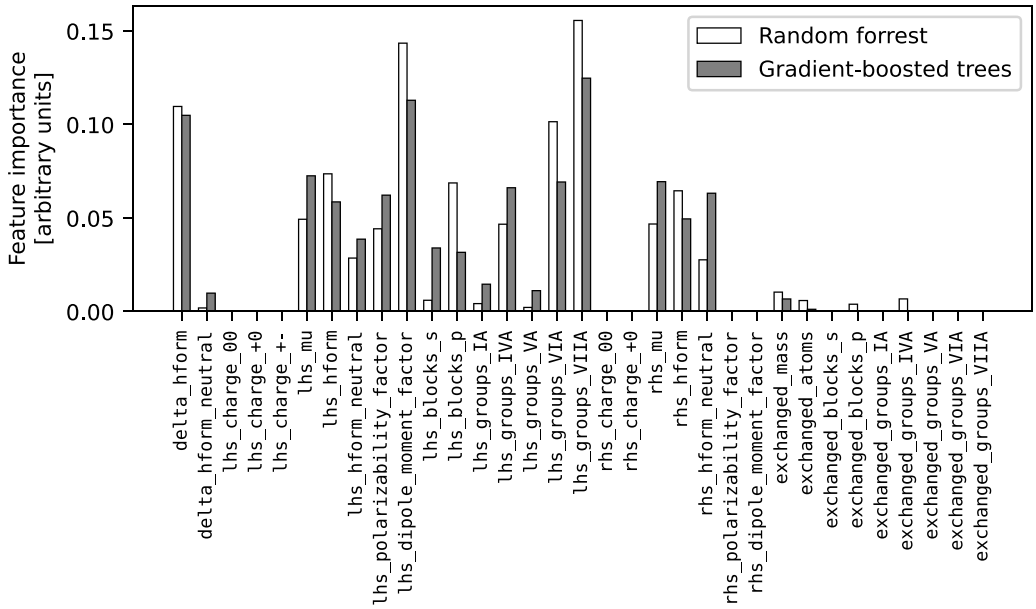

Figure 11. Feature importances for all the model features, pulled out of the random forest and gradient-boosted trees regressors (as two constituent models of the final voting regressor model). For clarity the feature names from the project GitHub repository are used, see table 2 for a list of features used.

Table 6. Top ten reactions with the most overestimated predicted rate coefficients. For each reaction, the prediction error is listed as well as the predicted value of the reaction rate coefficient and the target value. If there exist alternative rate coefficient targets in the dataset, those are also shown.

		Prediction error	Prediction	Target		
ID	Reaction	$\log k_{\text{pred}} - \log k_{\text{target}}$	$k_{\text{pred}}(\text{cm}^3\ \text{s}^{-1})$	$k_{\text{target}}(\text{cm}^3\ \text{s}^{-1})$	$k^{(2)}_{\text{target}}(\text{cm}^3\ \text{s}^{-1})$	$k^{(3)}_{\text{target}}(\text{cm}^3\ \text{s}^{-1})$
7800	$C_4H_3^+ + HCN \rightarrow C_4H_2 + HCNH^+$	10.72	5.19×10^{-10}	1.00×10^{-20}		
213	$H^+ + HCN \rightarrow H^+ + HCN$	10.57	3.72×10^{-10}	1.00×10^{-20}	1.00×10^{-09}	1.47×10^{-08}
2569	$CH_3 + HCO^+ \rightarrow CH_4^+ + CO$	10.06	1.14×10^{-10}	1.00×10^{-20}		
5105	$C_3H_2 + HCNH^+ \rightarrow C_3H_3^+ + HCN$	9.84	6.92×10^{-11}	1.00×10^{-20}	1.96×10^{-09}	
9044	$C_8H_2 + HCNH^+ \rightarrow C_8H_3^+ + HCN$	9.68	4.77×10^{-11}	1.00×10^{-20}		
5041	$N_2 + NH_4^+ \rightarrow N_2H^+ + NH_3$	9.56	3.61×10^{-11}	1.00×10^{-20}		
3540	$H_2O + HCNH^+ \rightarrow H_3O^+ + HCN$	9.55	3.51×10^{-11}	1.00×10^{-20}	1.00×10^{-20}	8.80×10^{-13}
5403	$HCO^+ + N_2 \rightarrow CO + N_2H^+$	9.50	3.13×10^{-11}	1.00×10^{-20}	6.70×10^{-10}	2.00×10^{-09}
8560	$C_6H_2 + HCNH^+ \rightarrow C_6H_3^+ + HCN$	9.45	2.79×10^{-11}	1.00×10^{-20}		
5192	$C_2H_4 + C_3H_3^+ \rightarrow C_5H_5^+ + H_2$	9.39	2.73×10^{-10}	1.10×10^{-19}	5.50×10^{-10}	1.10×10^{-09}

overestimated and underestimated rate coefficient predictions, respectively; both tables show reaction instances from the full dataset, not only the test set. The prediction error in the tables refers to the test errors for instances of the test set and training errors for the instances of the training set. Apart from the prediction errors and predicted and target values of reaction rate coefficients, also alternative rate coefficient target values are shown for each reaction where they exist in the dataset.

It is very encouraging to see, that in all the cases where any alternative target values exist, they agree with the predicted value much closer than the most diverging target value responsible for flagging these predictions as outliers. Even without inspecting the sources and credibility of the data instances, it could be argued that the data instances with the very low target values of $k < 10^{-20}\,\text{cm}^3\,\text{s}^{-1}$ from table 6 are very probably erroneous, as the same reactions can in many cases be found in the dataset with rate coefficients about ten orders of magnitude lower. The cases from table 6 could then be considered erroneous data samples, rather than erroneous predictions.

Taking as an example the elastic reaction

$$H^+ + HCN \rightarrow H^+ + HCN \qquad (22)$$

Table 7. Top ten reactions with the most underestimated predicted rate coefficients. For each reaction, the prediction error is listed as well as the predicted value of the reaction rate coefficient and the target value. If there exist alternative rate coefficient targets in the dataset, those are also shown.

		Prediction error	Prediction	Target		
ID	Reaction	$\log k_{\text{pred}} - \log k_{\text{target}}$	k_{pred}(cm^3 s^{-1})	k_{target}(cm^3 s^{-1})	$k^{(2)}_{\text{target}}$(cm^3 s^{-1})	$k^{(3)}_{\text{target}}$(cm^3 s^{-1})
7041	$CH_3CHCH_2 + H_3^+ \rightarrow C_3H_7^+ + H_2$	−17.89	9.36×10^{-09}	7.26×10^{09}		
3837	$C_3H_2 + H_3O^+ \rightarrow C_3H_3^+ + H_2O$	−8.78	1.66×10^{-09}	1.00×10^{00}	4.60×10^{-09}	3.00×10^{-09}
7495	$C_4H + S^+ \rightarrow C_3H^+ + CS$	−8.59	1.28×10^{-09}	4.98×10^{-01}		
7494	$C_4H + S^+ \rightarrow C_4S^+ + H$	−8.24	2.85×10^{-09}	4.98×10^{-01}	3.17×10^{-09}	1.00×10^{-09}
5114	$C_2H_4 + H \rightarrow C_2H_3 + H_2$	−7.28	4.70×10^{-17}	9.00×10^{-10}	1.00×10^{-20}	
97	$H + OCN \rightarrow CN + OH$	−6.69	2.04×10^{-17}	1.00×10^{-10}		
4427	$C_2H_2 + H \rightarrow C_2H + H_2$	−6.39	4.11×10^{-17}	1.00×10^{-10}	1.00×10^{-20}	1.00×10^{-20}
949	$C + H_2 \rightarrow CH + H$	−5.87	2.00×10^{-16}	1.50×10^{-10}	1.00×10^{-20}	
2690	$H_2 + O \rightarrow H + OH$	−5.87	4.63×10^{-16}	3.44×10^{-10}	9.16×10^{-18}	1.00×10^{-20}
7848	$CH_2F_2 + H \rightarrow CHF_2 + H_2$	−5.56	4.11×10^{-16}	1.49×10^{-10}		

labeled with ID **213**, it can be seen in table 6 that it has 3 data samples in the dataset:

(i) The first comes from KIDA [15] with $k = 10^{-9} \exp(-7850/T)$ cm^3 s^{-1}, which gives 4.32×10^{-21} at $T = 300$ K and gets capped to the value of $k = 10^{-20}$. This value is very low and a careful examination of the entry in KIDA database reveals the reason: KIDA lists it as the $H^+ + HCN \rightarrow H^+ + HNC$ reaction, therefore the low rate coefficient belongs to a reaction changing the isomer from hydrogen cyanide HCN, to hydrogen isocyanide HNC, rather than the elastic reaction (22) appearing in the dataset, which does not resolve different species isomers, as discussed in section 2.3. The rate coefficient of $k = 10^{-20}$ therefore does not apply for the reaction (22) and the model was right to regress much higher value. KIDA cites Harada *et al* [91] as the data source, but we could not find this reaction explicitly in the cited publication.

(ii) The second was sourced from UDfA [16], which lists the reaction with the coefficient value of $k^{(2)} = 10^{-9}$ cm^3 s^{-1}, without any temperature dependence (and without any citation).

(iii) The third available data sample is also from KIDA, which lists it with rate coefficient in the form of one of the formulas for ion–polar systems see equation (4), giving $k^{(3)} = 1.47 \times 10^{-8}$ cm^3 s^{-1}. UDfA cites work by Woon and Herbst [80] for this coefficient value, where the authors performed quantum-chemical calculations for neutral molecules, among others for HCN. This data sample could be considered the most reliable out of the three as it actually contains the citation to a paper relevant for the reaction. The prediction error compared to this data instance is still about 1.6 orders of magnitude, but this is well within the main peak of the prediction errors distribution shown in figure 10.

Similar conclusions could be drawn about the most diverging data samples from table 7. For example, in the case of the first four reactions in table 7, it is obvious that the target values responsible for such high prediction errors are way too high and clearly incorrect. In the case of reactions with IDs **3837** and **7494** in table 7, the alternative target values are very close to the predicted one, validating the predictions. And in the case of the first reaction (ID **7041**) in table 7, with the highest prediction error of the whole dataset, the data sourced from KIDA is also very obviously wrong. KIDA cites Hickson *et al* [92] as the source publication for the value of the reaction rate coefficient described by the functional dependence given in equation (5). KIDA lists the parameter β (which corresponds to the Langevin rate in cm^3 s^{-1}) as $\beta = 3.5 \times 10^9$ when it clearly should have been $\beta = 3.5 \times 10^{-9}$. This typo was corrected in February 2021 which was after the training data for this project was scraped. With the correct β coefficient, the rate coefficient for this sample evaluates to $k = 3.76 \times 10^{-9}$ cm^3 s^{-1}, which is fairly close to the predicted value.

4.4. Analysis of missing features

Figure 11 shows that the features $\Delta_f H^\circ_{\text{total}}$, F^α_{LHS}, and F^p_{LHS} appear on average fairly high in the decision trees of the random forest and the gradient-boosted trees regression models, which signals their relatively high importance for the prediction of reaction rate coefficients. These three features are also among those derived from values that were not present for all the data samples. More specifically, dipole moment p and polarizability α of both reactants were used to evaluate these features, as were the enthalpies of formation $\Delta_f H^\circ$ of all the reactants and products in any reaction. The missing species data had to be imputed and it could be interesting to see how the prediction error correlates with features data availability. Such a correlation is shown as a bar plot in figure 12.

At the first glance, there appears to be a negative correlation between the number of values missing relevant for the features discussed, and the MAE error measure, where one might expect (if any) a positive correlation. After all, the imputation process will likely be fairly crude in guessing missing values of a species from its other attributes. However, this correlation is in fact caused by a bias in the dataset, as shown in figure 13. The subsets of samples with more species attributes

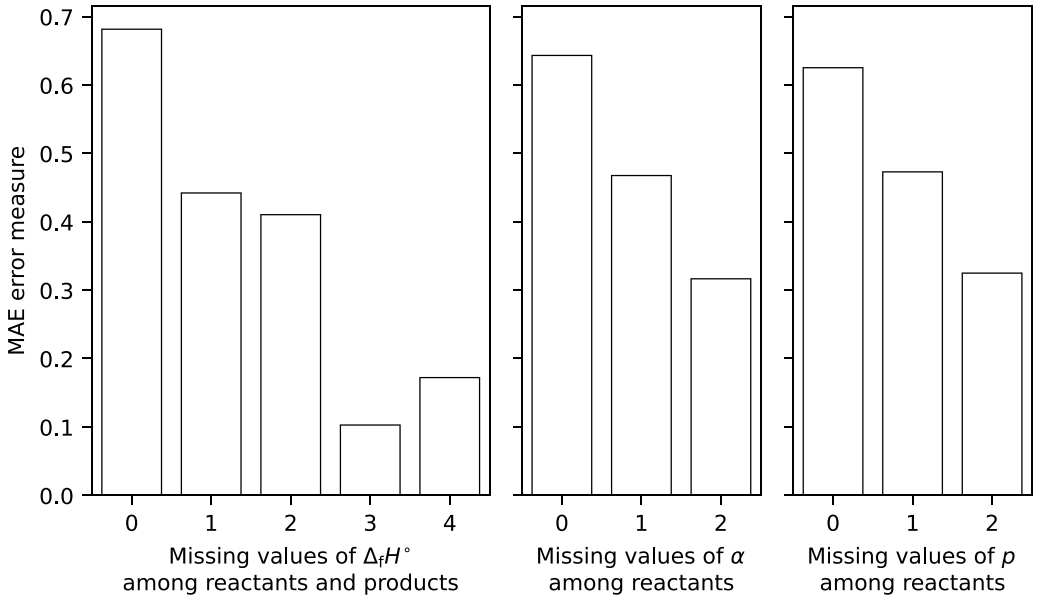

Figure 12. Correlation of MAEs and a number of missing values of enthalpy of formation $\Delta_f H^\circ$ among reactants and products, and polarizability α and dipole moment p among reactants. These values are used to evaluate the features $\Delta_f H^\circ_{\text{total}}$, F^α_{LHS} and F^p_{LHS}, that were found to be important for the prediction of rate coefficients (figure 11).

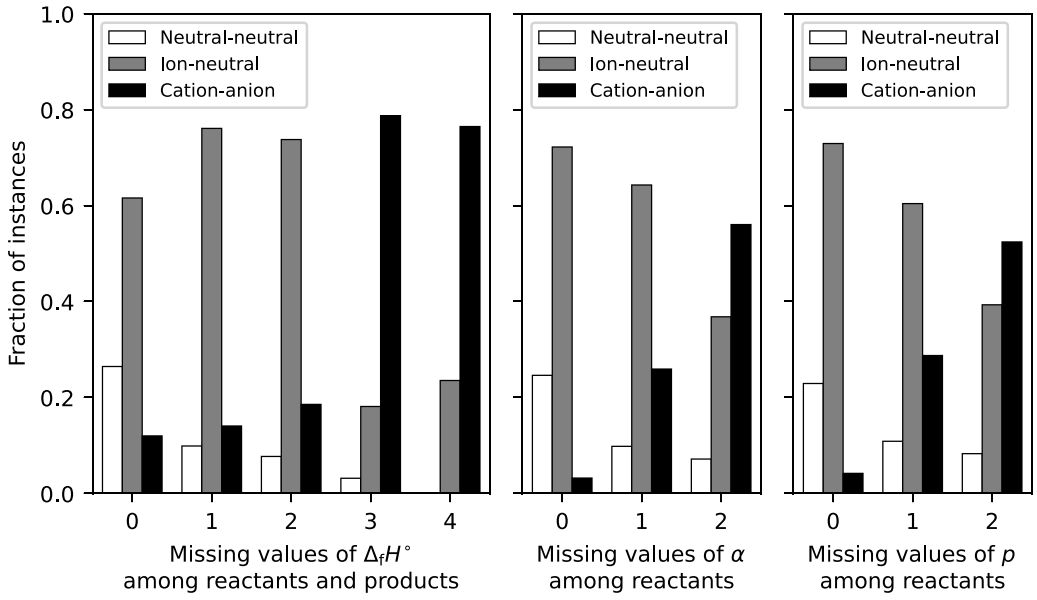

Figure 13. Correlation of fraction of instances with a particular combination of reactant charges, and the number of missing values of enthalpy of formation $\Delta_f H^\circ$ among reactants and products, and polarizability α and dipole moment p among reactants. Reactant charges combination greatly affects the prediction error (figure 10, table 5).

values missing tend to have a higher ratio of cation–anion reactions, and a lower ratio of neutral–neutral reactions. This greatly affects the prediction errors, as the neutral–neutral reactions get predicted with much lower accuracy than the cation–anion reactions. If data similar to those in figure 12 are plotted for e.g. ion–neutral reactions only, the negative correlation disappears and the MAE error measure appears to be independent of the number of missing values. This implies the imputation process is fairly effective in recovering the missing data.

4.5. Packaging of the final regression model

As the final step in the regression model development, we wrapped the final optimized regression model into a custom `Regressor` class, which takes care of capping the predicted data to the minimal value $y_{\text{min}} = \log_{10} k_{\text{min}} = -20$ after prediction, and recovers the reaction rate coefficients in the original units of $\text{cm}^3\ \text{s}^{-1}$ by exponentiating the predicted values back to $k^{\text{pred}} = 10^{y^{\text{pred}}}$. We chained this custom regressor behind the data transformation pipeline, which takes care of missing data imputation,

```
[1]: # Imports:
     import pandas as pd
     import numpy as np
```

```
[2]: example_input = pd.read_csv('sample_input.csv', index_col=0, header=0)
     # (the left-most index values are unique ids identifying the reaction rows)

     example_input  # (truncated view)
```

```
[2]:      reactant_1_name  reactant_1_mass  ...  exchanged_groups_VIA  exchanged_groups_VIIA
     1824              CH           13.019  ...                   0.0                    0.0
     409               H+            1.008  ...                   0.0                    0.0
     4506            C2H2           26.038  ...                   0.0                    0.0
     4012             C2+           24.022  ...                   0.0                    0.0
     3657            H2O+           18.015  ...                   0.0                    0.0
     2286             CH2           14.027  ...                   0.0                    0.0
     1679              C+           12.011  ...                   0.0                    0.0
     8935              C8           96.088  ...                   0.0                    0.0
     1424              C+           12.011  ...                   0.0                    0.0
     6912              Ar+          39.950  ...                   0.0                    0.0

     [10 rows x 72 columns]
```

```
[3]: np.array(example_input.columns)
```

```
[3]: array(['reactant_1_name', 'reactant_1_mass', 'reactant_1_charge', 'reactant_1_hform'
            'reactant_1_hform_neutral', 'reactant_1_polarizability', 'reactant_1_dipole_moment',
            'reactant_2_name', 'reactant_2_mass', 'reactant_2_charge', 'reactant_2_hform',
            'reactant_2_hform_neutral', 'reactant_2_polarizability', 'reactant_2_dipole_moment',
            'product_1_name', 'product_1_mass', 'product_1_charge', 'product_1_hform',
            'product_1_hform_neutral', 'product_1_polarizability', 'product_1_dipole_moment',
            'product_2_name', 'product_2_mass', 'product_2_charge', 'product_2_hform',
            'product_2_hform_neutral', 'product_2_polarizability', 'product_2_dipole_moment',
            'exchanged_mass', 'exchanged_atoms', 'lhs_mu', 'rhs_mu',
            'lhs_charge_00', 'lhs_charge_+0', 'lhs_charge_+-', 'rhs_charge_00', 'rhs_charge_+0',
            'reactant_1_blocks_s', 'reactant_1_blocks_p', 'reactant_2_blocks_s', 'reactant_2_blocks_p',
            'product_1_blocks_s', 'product_1_blocks_p', 'product_2_blocks_s', 'product_2_blocks_p',
            'exchanged_blocks_s', 'exchanged_blocks_p',
            'reactant_1_groups_IA', 'reactant_1_groups_IVA', 'reactant_1_groups_VA',
            'reactant_1_groups_VIA', 'reactant_1_groups_VIIA',
            'reactant_2_groups_IA', 'reactant_2_groups_IVA', 'reactant_2_groups_VA',
            'reactant_2_groups_VIA','reactant_2_groups_VIIA',
            'product_1_groups_IA', 'product_1_groups_IVA', 'product_1_groups_VA',
            'product_1_groups_VIA', 'product_1_groups_VIIA',
            'product_2_groups_IA', 'product_2_groups_IVA', 'product_2_groups_VA',
            'product_2_groups_VIA','product_2_groups_VIIA', 'exchanged_groups_IA',
            'exchanged_groups_IVA', 'exchanged_groups_VA',
            'exchanged_groups_VIA', 'exchanged_groups_VIIA'],
           dtype=object)
```

1

Figure 14. A jupyter notebook python code snippet detailing the required form of input for the trained regression model. The code in the figure shows how an example input `DataFrame` is instantiated from the `csv` table provided by the project repository. The regression model requires a `DataFrame` with 72 columns, given as the output of the cell `[3]`.

features engineering, and scaling (and which is described in section 3.2.3). The resulting final regression model pipeline was trained on the whole dataset and persistently saved as `final_regression_pipeline.joblib` by the `joblib` module from Python standard library. This ready-to-use trained regression model can be easily imported to any python code from the `utils` module in the project repository, by calling the `get_final_regression_pipeline` function.

The model needs to be fed by a `pandas.DataFrame` [93] instance, with rows representing the reactions for which the rate coefficients should be estimated. The project repository provides a sample input as `sample_input.csv` which can be read by `pandas` into the `DataFrame` required as the input for the trained model. Figure 14 shows a jupyter notebook python snippet detailing how the `example_input` `DataFrame` is instantiated from the `sample_input.csv` table and showing all the `DataFrame` columns required by the

```
[4]: # import the trained regression model:
     from utils import get_final_regression_pipeline
     regression_model = get_final_regression_pipeline()

     # predict the reaction rate coefficients for the 10 reactions
     # described by the example_input DataFrame:
     k_predicted = regression_model.predict(example_input)

     k_predicted

[4]: array([1.18797400e-09, 5.29729997e-09, 9.87500861e-11, 3.15630581e-10,
            1.89232973e-09, 8.37593678e-10, 6.48672484e-08, 1.77991274e-09,
            9.38216266e-10, 1.66151833e-10])
```

Figure 15. A continuation of the jupyter notebook python code snippet from figure 14, showing how the ready-to-use trained regression model can be imported from the `utils` python module. The regression model is then fed by the previously built `example_input` `DataFrame` instance (see figure 14) to predict the reaction rate coefficients in $\mathrm{cm}^3\ \mathrm{s}^{-1}$.

model. All the columns required are given as the header in the `sample_input.csv` table, while the first column of the table indexes the rows by their unique IDs. All the `DataFrame` values are required in units described in section 3.2.1.

Finally, figure 15 gives a continuation of the code from figure 14, showing how the ready-to-use trained regression model can be imported from the `utils` python module. To obtain the predicted rate coefficients for all the reactions described by the input `DataFrame`, the `predict` method of the regression model instance must be called with the input table as a single parameter, as detailed by figure 15. This returns a `NumPy` array of reaction rate coefficients in $\mathrm{cm}^3\ \mathrm{s}^{-1}$.

5. Example use case

In this section, we present how the algorithm can be used to fill data gaps in chemistry sets and how this might impact the results of a plasma simulation. As a gas mixture, we chose BCl_3/H_2; note, that neither the gas mixture nor the process parameters used in this example are meant to replicate any specific plasma process but were rather arbitrarily chosen to have missing reactions to be filled with the ML algorithm and to show significant differences in the result when comparing the simulations with and without the additional reaction data. A case which can be compared to experimental data would be preferable; this would, however, necessitate bespoke experiments, which are beyond the scope of this work.

A basic BCl_3/H_2 set was created with data from QDB using its set generator described in [9]. Cross-sectional data and rate coefficients for this set are taken from [94–136]. In this basic set, reactions between the various BCl_x species and H are missing. For some candidates, rate coefficients have been reported; [137, 138] report data for the reaction

$$BCl_3 + H \rightarrow BCl_2 + HCl \tag{23}$$

and [139] gives a rate coefficient for the reaction

$$BCl + H \rightarrow HB + Cl. \tag{24}$$

However, both of these reactions are highly endothermic yielding rate coefficients on the order of $10^{-15}\ \mathrm{cm}^3\ \mathrm{s}^{-1}$ or smaller at 300 K. Hence, they are unlikely to have a significant impact at the temperature the ML algorithm was trained for. As additional candidates we consider

$$BCl_2 + H \rightarrow BCl + HCl \tag{25}$$

and

$$BCl + H \rightarrow B + HCl. \tag{26}$$

Reaction (25) is exothermic while (26) is endothermic; therefore, we add reaction (25) but neglect (26). Using the ML algorithm for reaction (25) yields a rate coefficient of $1.3 \times 10^{-11}\ \mathrm{cm}^3\ \mathrm{s}^{-1}$. It should be noted, that there are also candidates for ion-neutral charge exchange and ion-ion recombination reactions; however, these either had no significant impact on the results or their rate coefficients as determined by the ML algorithm are close to common estimates. Therefore, they are not discussed for the sake of brevity.

To show the impact of adding the generated data to a chemistry set might have, we conducted global plasma simulations using the `pygmol` plasma model as detailed in [9]. We tested two sets with and without reaction (25) with (arbitrary) process parameters:

- A pressure of 10 Pa
- An absorbed power of 10 W
- A chamber radius of 0.1 m
- A chamber height of 0.1 m
- A total flow of 100 sccm
- The relative flow of BCl_3 was varied between 10% and 90% with the remainder as H_2 flow.

Figure 16 shows the densities of the species involved in reaction (25) as a function of the relative BCl_3 flow for the process parameters discussed above. We observe significant differences between the basic set and the one with reaction (25) added, namely:

- The density of BCl is between half an order of magnitude and a whole order of magnitude larger with the additional data; qualitatively it increases somewhat slower than in the basis set.

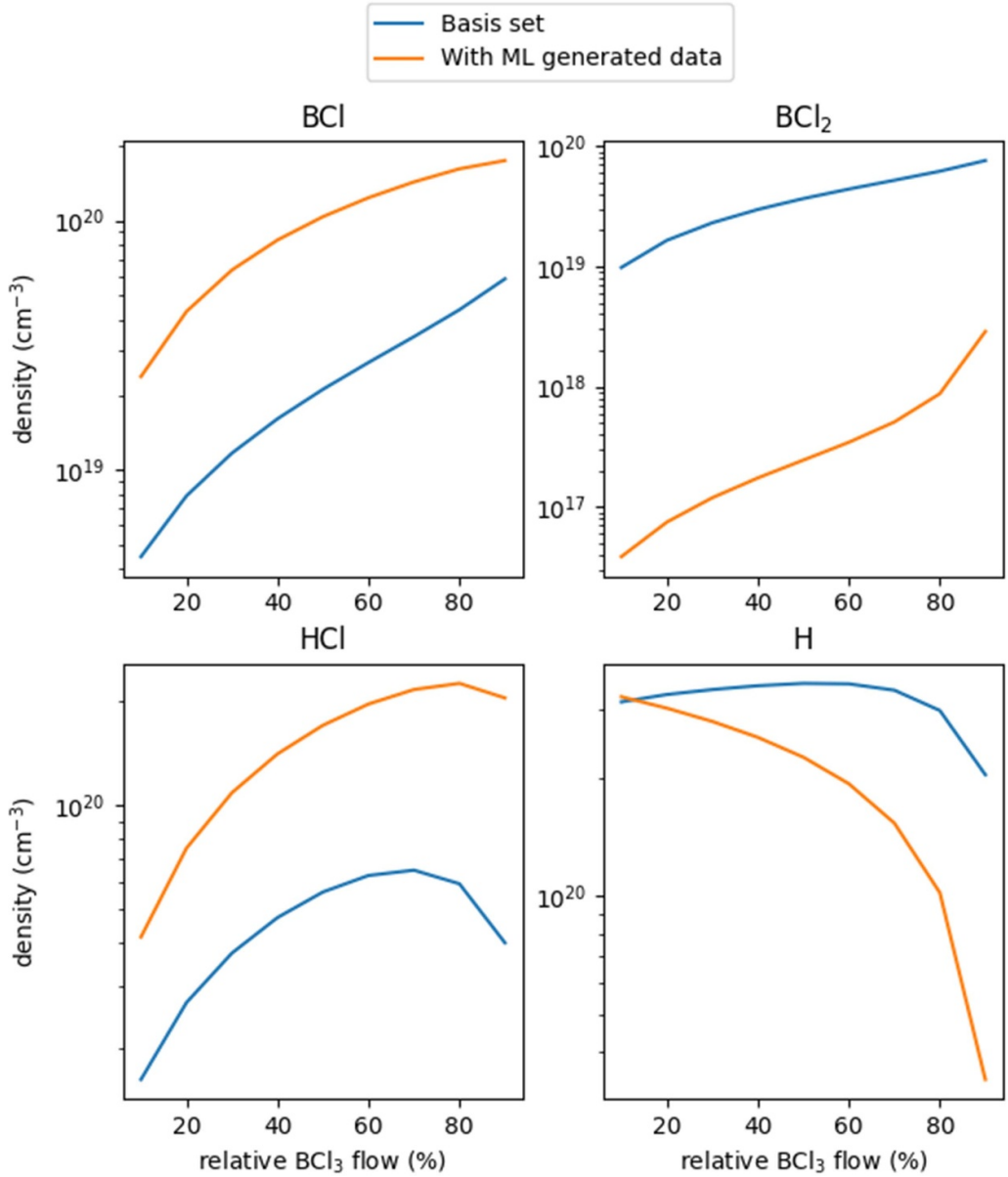

Figure 16. Densities of the neutrals involved in reaction (25) as a function of the relative BCl_3 flow as calculated by the `pygmol` global plasma model [9].

- BCl_2 shows densities which are one to two orders of magnitude smaller than in the basis set. Qualitatively, its density increases faster as a function of the BCl_3 flow than in the basis set.
- HCl shows the smallest difference with a similar qualitative behaviour and higher absolute densities up to a factor of about 5.
- H displays a very different behaviour; in the basis set it stays relatively constant and only drops off for large BCl_3 flows, while with the added reactions it is monotonically decreasing. In absolute terms, the densities are roughly the same for small BCl_3 contents, but differ by an order of magnitude for large ones.

Overall, the example shows how adding a reaction with a hitherto unknown rate coefficients has a significant impact on the chemical composition of the discharge with densities of neutrals differing by up to two orders magnitude. This highlights the importance of taking such reactions into consideration for which our ML algorithm likely gives better estimates than intuitive guesses, while being faster than precise measurements, calculations, or calibrations. This and similar chemistries can constructed within the QDB environment using a combination of the assembled data and the in-house ML algorithm.

6. Conclusions and outlook

Here we present a ML-based regression model for fast approximation of unknown plasma reaction rate constants at $T = 300$ K from commonly available reaction and species data.

The model was restricted to binary reactions of heavy species. The room-temperature rate coefficients are regressed from features built from masses, charges, enthalpies of formation, polarizabilities, dipole moments, and elemental data of both reactants and products. The final model is a voting regressor consisting of three distinct optimized regression models: a support vector regressor [73], random forest regressor [75], and a gradient-boosted trees regressor [77]. This work forms a natural counterpart to our previous study [140] which presented a method of ranking the species in a chemistry set according to their importance for modeling densities of a pre-defined set of species of interest.

The model was trained on a training set of over 7500 data instances acquired from four popular databases of plasma processes. The final generalization MAE error of the reaction rate coefficient prediction, evaluated on a withheld test set of around 1900 instances, was just under 0.6 orders of magnitude, compared to about 0.95 orders of magnitude for a benchmark case of a simple linear regression. The overall distribution of the reaction rate coefficients from the test set was recovered very well in the predicted values, as were the distributions for distinct subsets of neutral–neutral, ion–ion, and neutral–ion reactions.

The ability of the model to flag instances of erroneous data instances is demonstrated: we found that out of the ten most underestimated and overestimated predicted rates, the majority could be attributed to erroneous training targets, rather than incorrect predictions.

The most obvious potential for future work is the expansion of the model applicability space. The model presented was trained exclusively on reactions of two heavy-species reactants, and two heavy-species products, without any excited states, which means that the model applies only to such reactions. As a subject of future research, the model could be expanded to also handle three-body collisions, dissociative or associative processes with different numbers of reactants and products, ionization processes with electrons among the products, and similar. Furthermore, the model could be expanded to handle vibrational and/or electronic excited states among reactants and products. Such an expansion will require redesigning the feature space of the model to capture state-specific properties of species, and to allow features built from more than two species per reaction side.

Electron collisions form a somewhat distinct category of plasma reactions, as full collisional cross-sections are usually required to model low-temperature plasma phenomena. This, with the fact that electron collisions are driven by complicated underlying physics, would probably make the task of expanding the current model to also handle electron collisions challenging. Instead, the development and training of a separate model for electron collisions might be a more sensible approach and another topic of possible future research.

The model presented only regresses rate coefficients expressed for the room temperature, while a much more valuable output of the model would be a temperature dependence $k(T)$ in some form such as the triplet of parameters for the modified Arrhenius formula of equation (3). Training such a model would, however, require training data instances with temperature-dependent target values of rate coefficients, for which there was limited data in our data sources.

Building a higher-quality training dataset would undoubtedly improve the predictions provided by our model. We found that our training dataset contained a number of reactions with multiple data instances with diverging target values, sometimes differing by many orders of magnitude. Curating the training dataset such that it contains only trustworthy data instances might be a very costly effort, but one which would undoubtedly also improve the quality and performance of any regression model trained on such a curated dataset.

Rejecting untrustworthy data would reduce the size of the training dataset meaning that exploring additional training data sources makes for another significant direction of future work. The training dataset could also benefit from an analysis of the biases present, as the selection of data sources inevitably influences which reaction classes, or which rate coefficient calculation methods, experimental techniques, etc, are dominant in the dataset. A careful bias analysis helps to understand how the biases in the training dataset translate to the expected prediction accuracies for, e.g. different reaction classes. Finally, testing the regression model on various well-established published chemistry sets by replacing a fraction of real rate coefficients with data synthesized by the model and comparing the model results, presents itself perhaps as another logical next step in this research.

Data availability statement

The datasets used to undertake this study and their documentation are openly available form the project repository https://github.com/martin-hanicinec-ucl/regreschem. The same repository also contains the ready-to-use trained regression model.

Acknowledgment

Martin Hanicinec would like to thank EPSRC for a CASE studentship under Grant EP/N509577/1.

ORCID iDs

Martin Hanicinec https://orcid.org/0000-0003-3415-2996
Jonathan Tennyson https://orcid.org/0000-0002-4994-5238

References

[1] d'Agostino R, Favia P, Oehr C and Wertheimer M R 2005 *Plasma Process. Polym.* **2** 7–15
[2] Berthelot A and Bogaerts A 2016 *Plasma Sources Sci. Technol.* **25** 045022
[3] Guerra V, Tejero-del-Caz A, Pintassilgo C D and Alves L L 2019 *Plasma Sources Sci. Technol.* **28** 073001
[4] Hong J, Pancheshnyi S, Tam E, Lowke J J, Prawer S and Murphy A B 2017 *J. Phys. D: Appl. Phys.* **50** 154005
[5] Gaens W V and Bogaerts A 2013 *J. Phys. D: Appl. Phys.* **46** 275201

[6] Sieck L W, Heron J T and Green D S 2000 *Plasma Chem. Plasma Process.* **20** 235–58
[7] Herron J T and Green D S 2001 *Plasma Chem. Plasma Process.* **21** 459–81
[8] Tennyson J *et al* 2017 *Plasma Sources Sci. Technol.* **26** 055014
[9] Tennyson J *et al* 2022 *Plasma Sources Sci. Technol.* **31** 095020
[10] Koelman P, Heijkers S, Tadayon Mousavi S, Graef W, Mihailova D, Kozak T, Bogaerts A and van Dijk J 2017 *Plasma Process. Polym.* **14** 1600155
[11] Nagy T and Turanyi T 2009 *Combust. Flame* **156** 417–28
[12] Pitchford L C *et al* 2017 *Plasma Process. Polym.* **14** 1600098
[13] Celiberto R *et al* 2016 *Plasma Sources Sci. Technol.* **25** 033004
[14] Wakelam V *et al* 2012 *Astrophys. J. Suppl. Ser.* **199** 21
[15] Wakelam V *et al* 2015 *Astrophys. J. Suppl. Ser.* **217** 20
[16] McElroy D, Walsh C, Markwick A J, Cordiner M A, Smith K and Millar T J 2013 *Astron. Astrophys.* **550** A36
[17] Dubernet M L *et al* 2013 *Astron. Astrophys.* **553** A50
[18] Murakami I, Kato D, Kato M, Sakaue H A and Kato T 2007 *Fusion Sci. Technol.* **51** 138–40
[19] Park J H, Choi H, Chang W S, Chung S Y, Kwon D C, Song M Y and Yoon J S 2020 *Appl. Sci. Converg. Technol.* **29** 5–9
[20] Hulse R A 1990 The ALADDIN atomic physics database system *AIP Conf. Proc.* **206** 63–72
[21] Turner M M 2015 *Plasma Sources Sci. Technol.* **24** 035027
[22] Harada N and Herbst E 2008 *Astrophys. J.* **685** 272–80
[23] Smith D, Church M J and Miller T M 1978 *J. Chem. Phys.* **68** 1224–9
[24] Adamovich I *et al* 2017 *J. Phys. D: Appl. Phys.* **50** 323001
[25] Bartschat K and Kushner M J 2016 *Proc. Natl Acad. Sci.* **113** 7026–34
[26] Kim B and May G 1994 *IEEE Trans. Semicond. Manuf.* **7** 12–21
[27] Kim B, Lee D W, Park K Y, Choi S R and Choi S 2004 *Vacuum* **76** 37–43
[28] Kim B and Kwon M 2009 *J. Mater. Process. Technol.* **209** 2620–6
[29] Kim B and Kim S 2010 *Surf. Eng.* **26** 224–8
[30] Himmel C and May G 1993 *IEEE Trans. Semicond. Manuf.* **6** 103–11
[31] Rietman E and Lory E 1993 *IEEE Trans. Semicond. Manuf.* **6** 343–7
[32] Han S, Ceiler M, Bidstrup S, Kohl P and May G 1994 *IEEE Trans. Compon. Packag. Manuf. Technol.* A **17** 174–82
[33] Stokes D and May G 2000 *IEEE Trans. Semicond. Manuf.* **13** 469–80
[34] Tudoroiu N, Patel R and Khorasani K 2006 *Neurocomputing* **69** 786–802
[35] Rosen I, Parent T, Cooper C, Chen P and Madhukar A 2001 *IEEE Trans. Control Syst. Technol.* **9** 271–84
[36] Bhatikar S and Mahajan R 2002 *IEEE Trans. Semicond. Manuf.* **15** 71–78
[37] Chen W, Lee A, Deng W and Liu K 2007 *Expert Syst. Appl.* **32** 1148–53
[38] Ko Y D, Moon P, Kim C E, Ham M H, Myoung J M and Yun I 2009 *Expert Syst. Appl.* **36** 4061–6
[39] Guessasma S, Montavon G and Coddet C 2004 *Comput. Mater. Sci.* **29** 315–33
[40] Jean M D, Lin B T and Chou J H 2008 *J. Am. Ceram. Soc.* **91** 1539–47
[41] Choudhury T, Berndt C and Man Z 2015 *Eng. Appl. Artif. Intell.* **45** 57–70
[42] Krüger F, Gergs T and Trieschmann J 2019 *Plasma Sources Sci. Technol.* **28** 035002
[43] Kino H, Ikuse K, Dam H C and Hamaguchi S 2021 *Phys. Plasmas* **28** 013504
[44] Leparoux M, Loher M, Schreuders C and Siegmann S 2008 *Powder Technol.* **185** 109–15
[45] Wang C, Wang X and He X 2007 Neural Networks Model of Polypropylene Surface Modification by Air Plasma *2007 IEEE Int. Conf. on Automation and Logistics* (Jinan: IEEE) pp 20–24
[46] Abd Jelil R, Zeng X, Koehl L and Perwuelz A 2013 *Eng. Appl. Artif. Intell.* **26** 1854–64
[47] Rietman E A 1996 *J. Vac. Sci. Technol.* B **14** 504
[48] Salam F, Piwek C, Erten G, Grotjohn T and Asmussen J 1997 *IEEE Trans. Control Syst. Technol.* **5** 598–613
[49] Molga E 2003 *Chem. Eng. Process.* **42** 675–95
[50] Kim B and Park S 2001 *Chemometr. Intell. Lab. Syst.* **56** 39–50
[51] Kim B and Bae J 2005 *Solid-State Electron.* **49** 1576–80
[52] Mesbah A and Graves D B 2019 *J. Phys. D: Appl. Phys.* **52** 30LT02
[53] Dral P O, von Lilienfeld O A and Thiel W 2015 *J. Chem. Theory Comput.* **11** 2120–5
[54] Komp E and Valleau S 2020 *J. Phys. Chem.* A **124** 8607–13
[55] Zhang Y 2009 *Chemometr. Intell. Lab. Syst.* **98** 162–72
[56] Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld O A, Tkatchenko A and Müller K R 2013 *J. Chem. Theory Comput.* **9** 3404–19
[57] Goh G B, Hodas N O and Vishnu A 2017 *J. Comput. Chem.* **38** 1291–307
[58] Ventura S, Silva M, Perez-Bendito D and Hervas C 1995 *Anal. Chem.* **67** 1521–5
[59] Galván I, Zaldívar J, Hernández H and Molga E 1996 *Comput. Chem. Eng.* **20** 1451–65
[60] Baş D, Dudak F C and Boyaci I H 2007 *J. Food Eng.* **79** 622–8
[61] Baş D, Dudak F C and Boyaco I H 2007 *J. Food Eng.* **79** 1152–8
[62] Valeh-e Sheyda P, Yaripour F, Moradi G and Saber M 2010 *Ind. Eng. Chem. Res.* **49** 4620–6
[63] Tumanov V E and Gaifullin B N 2015 Evaluation of the Rate Constants of Reactions of Phenyl Radicals with Hydrocarbons with the Use of Artificial Neural Network *Current Approaches in Applied Artificial Intelligence* (*Lecture Notes in Computer Science*) vol 9101, ed M Ali, Y S Kwon, C H Lee, J Kim and Y Kim (Cham: Springer) pp 394–403
[64] Allison T C 2016 *J. Phys. Chem.* B **120** 1854–63
[65] Choi S, Kim Y, Kim J W, Kim Z and Kim W Y 2018 *Chem. Eur. J.* **24** 12354–8
[66] Grambow C A, Pattanaik L and Green W H 2020 *J. Phys. Chem. Lett.* **11** 2992–7
[67] Kuang D and Xu B 2018 *Thermochim. Acta* **669** 8–15
[68] Huang Y W, Chen M Q and Li Q H 2019 *J. Therm. Anal. Calorimetry* **138** 451–60
[69] Vieira D and Krems R V 2017 *Astrophys. J.* **835** 255
[70] Amato F, González-Hernández J L and Havel J 2012 *Talanta* **93** 72–78
[71] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825–30
[72] Géron A 2019 *Hands-on Machine Learning With Scikit-Learn, Keras and Tensorflow: Concepts, Tools and Techniques to Build Intelligent Systems* 2nd edn (Sebastopol, CA: O'Reilly Media, Inc)
[73] Boser B E, Guyon I M and Vapnik V N 1992 A training algorithm for optimal margin classifiers *Proc. 5th Annual Workshop on Computational Learning Theory - COLT '92* (Pittsburgh, PA: ACM Press) pp 144–52
[74] Breiman L, Friedman J, Olshen R and Stone C J 1984 *Classification and Regression Trees* (New York: Routledge)

[75] Breiman L 1998 *Classification and Regression Trees* 1st edn (Boca Raton, FL: CRC Press)
[76] Breiman L 2001 *Mach. Learn.* **45** 5–32
[77] Breiman L 1997 Arcing the edge *Technical Report* 486 Department of Statistics, University of California, Berkeley
[78] Friedman J H 2001 *Ann. Stat.* **29** 1189–232
[79] Kouzis-Loukas D 2016 *Learning Scrapy* (Birmingham: Packt Publishing Ltd)
[80] Woon D E and Herbst E 2009 *Astrophys. J. Suppl. Ser.* **185** 273–88
[81] Wakelam V *et al* 2010 *Space Sci. Rev.* **156** 13–72
[82] Hill C 2020 Pyvalem: Open source python package for parsing, validating, manipulating and interpreting the chemical formulas, quantum states and labels of atoms, ions and small molecules (available at: https://github.com/xnx/pyvalem)
[83] Chase M 1998*NIST-Janaf Thermochemical Tables*4th edn (College Park, MA: American Institute of Physics)
[84] Ruscic B, Pinzon R E, Morton M L, von Laszevski G, Bittner S J, Nijsure S G, Amin K A, Minkoff M and Wagner A F 2004 *J. Phys. Chem.* A **108** 9979–97
[85] Lu B 2020 Databases for plasma modelling *Master's Thesis* University College London United Kingdom
[86] Tipping M E 2001 *J. Mach. Learn. Res.* **1** 211–44
[87] MacKay D J C 1992 *Neural Comput.* **4** 415–47
[88] Domingos P 2012 *Commun. ACM* **55** 78
[89] Berrar D 2019 Cross-validation *Encyclopedia of Bioinformatics and Computational Biology* (Amsterdam: Elsevier) pp 542–5
[90] Kluyver T *et al* 2016 Jupyter notebooks – a publishing format for reproducible computational workflows *Positioning and Power in Academic Publishing: Players, Agents and Agendas* ed F Loizides and B Schmidt (Amsterdam, NL: IOS Press) pp 87–90
[91] Harada N, Herbst E and Wakelam V 2010 *Astrophys. J.* **721** 1570–8
[92] Hickson K M, Wakelam V and Loison J C 2016 *Mol. Astrophys.* **3** 1–9
[93] McKinney W *et al* 2010 Data structures for statistical computing in python *Proc. 9th Python in Science Conf. (Austin, TX)* vol 445 pp 51–56
[94] Eckstrom D J, Nakano H H, Lorents D C, Rothem T, Betts J A, Lainhart M E, Triebes K J and Dakin D A 1988 *J. Appl. Phys.* **64** 1691
[95] Kirkpatrick M, Dodet B and Odic E 2007 *Int. J. Plasma Environ. Sci. Technol.* **1** 96–101
[96] Gougousi T, Johnsen R and Golde M F 1995 *Int. J. Mass Spectrom. Ion Process.* **149–150** 131–51
[97] Vriens L 1964 *Phys. Lett.* **8** 260–1
[98] Kushner M J 1988 *J. Appl. Phys.* **63** 2532
[99] Janev R K, Langer W D and Evans K J 1987 *Elementary Processes in Hydrogen-Helium Plasmas Cross Sections and Reaction Rate Coefficients* (*Springer Series on Atomic, Optical, and Plasma Physics* vol 4) (Berlin: Springer)
[100] Méndez I, Tanarro I and Herrero V J 2010 *Phys. Chem. Chem. Phys.* **12** 4239
[101] Méndez I, Gordillo-Vazquez F J, Herrero V J and Tanarro I 2006 *J. Phys. Chem.* A **110** 6060–6
[102] Subramonium P and Kushner M J 2002 *J. Vac. Sci. Technol.* A **20** 325
[103] Brian J 1990 *Phys. Rep.* **186** 215–48
[104] Hayashi M 1979 *J. Phys. Colloq.* **40** C2-661–C2-662
[105] Vroom D A and de Heer F J 1969 *J. Chem. Phys.* **50** 580
[106] Chan C F, Burrell C F and Cooper W S 1983 *J. Appl. Phys.* **54** 6119
[107] Banks P 1966 *Planet. Space Sci.* **14** 1085–103
[108] Buchelnikova I S 1959 *Sov. Phys. JETP* **8** 783–91
[109] Nagpal R and Garscadden A 1994 *Appl. Phys. Lett.* **64** 1626
[110] Marriott J and Craggs J D 1957 *J. Electron. Control* **3** 194–202
[111] Marchalant P J and Bartschat K 1997 *J. Phys. B: At. Mol. Opt. Phys.* **30** 4373–82
[112] Aydil E S and Economou D J 1993 *J. Electrochem. Soc.* **140** 1471
[113] Hayashi M 1987 Electron collision cross-sections for molecules determined from beam and swarm data Sw*arm Studies and Inelastic Electron-Molecule Collisions* ed L C Pitchford, B V McKoy, A Chutjian and S Trajrnar (New York: Springer) pp 167–87
[114] van Gaens W and Bogaerts A 2013 *J. Phys. D: Appl. Phys.* **46** 275201
[115] Bardsley J N and Wadehra J M 1983 *J. Chem. Phys.* **78** 7227
[116] Phelps A V 1985 Tabulations of collision cross sections and calculated transport and reaction coefficients for electron collisions with O2 *Technical Report* University of Colorado
[117] Rogoff G L, Kramer J M and Piejak R B 1986 *IEEE Trans. Plasma Sci.* **14** 103–11
[118] Cohen N and Westberg K R 1983 *J. Phys. Chem. Ref. Data* **12** 531
[119] Tian C and Vidal C R 1998 *J. Phys. B: At. Mol. Opt. Phys.* **31** 895–909
[120] Mao M and Bogaerts A 2010 *J. Phys. D: Appl. Phys.* **43** 205201
[121] Yoon J S, Song M Y, Han J M, Hwang S H, Chang W S, Lee B and Itikawa Y 2008 *J. Phys. Chem. Ref. Data* **37** 913
[122] Choi S J and Kushner M J 1993 *Appl. Phys. Lett.* **62** 2197
[123] Shuman N S, Wiens J P, Miller T M and Viggiano A A 2014 *J. Chem. Phys.* **140** 224309
[124] Janev R K and Reiter D 2003 Collision processes in low-temperature hydrogen plasmas *Technical Report* Forschungszentrum Jülich GmbH
[125] Celiberto R, Janev R K, Laporta V, Tennyson J and Wadehra J M 2013 *Phys. Rev.* A **88** 062701
[126] Lowke J J and Morrow R 1995 *IEEE Trans. Plasma Sci.* **23** 661–71
[127] Meeks E, Ho P, Ting A and Buss R 1998 *J. Vac. Sci. Technol.* A **16** 2227
[128] Ganas P S 1988 *J. Appl. Phys.* **63** 277
[129] Efremov A, Kim Y, Lee H W and Kwon K H 2011 *Plasma Chem. Plasma Process.* **31** 259
[130] Gul B, Ahmad I, Zia G and Aman-ur-Rehman 2016 *Phys. Plasmas* **23** 093508
[131] Harada N and Herbst E 2008 *Am. Astron. Soc.* **685** 272–80
[132] Pradhan A and Dalgarno A 1994 *Am. Phys. Soc.* **49** 960–4
[133] Smith D and Adams N G 1985 *Am. Astron. Soc.* **298** 827
[134] Blake G A, Anicich V G and Huntress Jr W T 1986 *Am. Astron. Soc.* **300** 415
[135] Harada N, Herbst E and Wakelam V 2010 *Astrophys. J.* **721** 1570
[136] Epée Epée M D Mezei J Z, Motapon O, Pop N and Schneider I F 2016 *Mon. Not. R. Astron. Soc.* **455** 276–81
[137] Jourdain J L, Laverdet G, Le Bras G and Combourieu J 1981 *J. Chim. Phys.* **78** 253–7
[138] Zhang S, Zhang Y, Wang C Y and Li Q S 2003 *Chem. Phys. Lett.* **373** 1–7
[139] Garrett B C and Truhlar D G 1979 *J. Am. Chem. Soc.* **101** 5207–17
[140] Hanicinec M, Mohr S and Tennyson J 2021 *Plasma Sources Sci. Technol.* **29** 125024