Tracking the Chemical Evolution of Iodine Species Using Recurrent Neural Networks

Jenna A. Bilbrey,* Carlos Ortiz Marrero, Michel Sassi, Andrew M. Ritzmann, Neil J. Henson, and Malachi Schram

ABSTRACT: We apply recurrent neural networks (RNNs) to predict the time evolution of the concentration profile of multiple species resulting from a set of interconnected chemical reactions. As a proof of concept of our approach, RNNs were trained on a synthetic dataset generated by solving the kinetic equations of a system of aqueous inorganic iodine reactions that can follow after nuclear reactor accidents. We examine the minimum dataset necessary to obtain accurate predictions and explore the ability of RNNs to interpolate and extrapolate when exposed to previously unseen data. We also investigate the limits of our RNN by evaluating the robustness of the training initialization on our dataset.

INTRODUCTION

Radioactive iodine-131 ($^{131}$I) is a byproduct of uranium fission. Under normal operation of a nuclear power plant, little to no iodine is released into the environment; it is either contained in the spent fuel or scrubbed by filtration systems. However, if an incident occurs, such as when the Tohoku earthquake and tsunami hit the Fukushima Daiichi Nuclear Power Plant in 2011, radioiodine can be released into the environment. The half-life of $^{131}$I is 8.0262 $\pm$ 0.0006 days, during which it can undergo a number of reactions in the atmosphere, including those in the gaseous and aqueous phases as well as solid-state reactions with particulates in the atmosphere that likely result from the incident. After nuclear accidents, $^{131}$I is commonly observed in the environment. Takeyasu et al. monitored the fallout of radionuclides from the Fukushima accident and found $^{131}$I to be present in the highest concentration among all radionuclides.

Radioiodine presents health problems because it can enter the body through numerous mechanisms. Radioiodine in the aqueous phase can enter local water sources and be imbibed directly or taken up by plants and animals, where it is then passed to the population. In the gaseous phase, radioiodine can enter the bloodstream through inhalation. Once in the body, radioiodine concentrates in the thyroid gland and has the potential to cause thyroid cancer. For example, after the accident at the Chernobyl Nuclear Power Plant in 1986, those that were not impacted by direct radiation exposure were affected by $^{131}$I released from the explosion, mainly through contact with contaminated vegetable and dairy products. After the Fukushima accident, $^{131}$I was detected in human breast milk, even when only mild environmental $^{131}$I pollution was present. Recently, researchers in Japan have detected high rates of thyroid cancer in young individuals, thought to have resulted from radiation exposure after the Fukushima accident.

Radioiodine can react with hydroxyl radicals present in the atmosphere to form an array of species. A number of reports have examined the multitude of species generated in such accidents. In this work, we adopt the library of iodine reactions in containment (LIRIC) created by Wren and Ball, which provides a comprehensive mechanistic model for the behavior of iodine in containment under nuclear reactor accident conditions. Their library consists of nearly 200 reactions and includes chemical reactions, surface interactions, and mass transport processes, along with their respective rate constants. Notably, LIRIC does not contain nuclear decay rates. As our goal is to explore the utility of deep learning methods in tracking complex chemical evolutions, we do not consider the added complexity of nuclear decay when determining the fate of iodine, as the time scale of the studied system of reactions is much shorter than the half-life of $^{131}$I.

Deep learning methods are gaining popularity for the task of forecasting time series data. Recurrent neural networks (RNNs), in particular, are powerful tools for predicting large input sequences of variable lengths, exemplified by complex chemical reactions. Such networks are commonly applied to language and image processing. Long short-term memory (LSTM) is a type of RNN that better captures long-term dependencies within the data by passing a sequence of hidden states. LSTM employs input, forget, and output gates to...
control the information that is passed to the successive cell through the hidden state, allowing important information to pass through cells unchanged. Notably, this gating mechanism overcomes the vanishing and exploding gradient problem encountered when attempting to learn long sequences.19,20

When applied to chemistry problems, RNNs are often used for property prediction,21 product prediction,22 or molecular design in a specific region of chemical space,23,24 often using a graph or textual representation of the molecule under study. Here, we apply RNNs to a chemical kinetics problem, namely, predicting the chemical evolution of species as a reaction progresses. We combine kinetic models from two systems previously published in the literature to examine the ability of RNNs to predict the chemical evolution of iodine species in a supposed nuclear reactor incident. We combine a previously published model for the radiolysis of water25,26 to a closed subset of the LIRIC model that involves aqueous-phase reactions of inorganic iodine species. The RNN is agnostic to the actual species under question and instead utilizes concentration data in the form of a time series. The RNN predicts the concentrations of all species as an ensemble, which allows for correlations between the different species to be leveraged when making predictions. In this report, we show that RNNs are able to produce highly accurate predictions of the evolution of multiple chemical species with concentrations inside the bounds of those in the training set, while the predictions quickly worsen when the concentrations fall too far outside of those bounds. We also investigate the network variability and sampling size necessary to achieve good predictions. We find that the variability in the sampling choice increases as the size of the training set decreases, indicating that the careful selection of samples is necessary when training networks on low amounts of data.

## METHODS

### Data Generation

In 2001, Wren and Ball published a comprehensive kinetic model called LIRIC that describes the behavior of iodine under conditions related to incidents at nuclear power plants.12 The same year, Pastina and LaVerne published G values for the radiolysis of water with γ-rays,27 and a few years later, Poinssot et al. published a report for the European Commission with a complete kinetic model for radiolytic reactions in pure water.28 To mimic iodine contamination in a water coolant system at a nuclear power plant, we merged these models, using the full water radiolysis model and a closed subset of the LIRIC model, by first evolving the water radiolysis model and applying the outcome as an input to the iodine model. Note that several of the rate constants in the LIRIC paper were altered; the full list of reactions and rate constants used in this work is given in the Supporting Information (see Table S1).

Table 1. Dataset Notations, Number (N) of Simulations, and Concentration Ranges Used in This Work⁴

| dataset     | N   | [I2]_{min} | [I2]_{max} | [I]_{min} | [I]_{max} |
|-------------|-----|------------|------------|-----------|-----------|
| training set| 1024 | 1.00 × 10⁻⁷ | 5.00 × 10⁻⁷ | 1.00 × 10⁻⁴ | 5.00 × 10⁻⁸ |
| I₁ (I) I* (I) | 961 | 1.06 × 10⁻⁷ | 4.93 × 10⁻⁷ | 1.06 × 10⁻⁴ | 4.93 × 10⁻⁸ |
| I₁ (E) I* (E) | 930 | 1.06 × 10⁻⁷ | 4.93 × 10⁻⁷ | 1.00 × 10⁻⁴/5.13 × 10⁻⁸ | 9.34 × 10⁻⁷/7.00 × 10⁻⁸ |
| I₂ (E) I* (E) | 930 | 1.00 × 10⁻⁷/5.13 × 10⁻⁷ | 9.34 × 10⁻⁷/7.00 × 10⁻⁷ | 1.00 × 10⁻⁴ | 4.93 × 10⁻⁸ |
| I₂ (E) I* (E) | 900 | 1.00 × 10⁻⁷/5.13 × 10⁻⁷ | 9.34 × 10⁻⁷/7.00 × 10⁻⁷ | 1.00 × 10⁻⁴/5.13 × 10⁻⁸ | 9.34 × 10⁻⁷/7.00 × 10⁻⁸ |

⁴I indicates that the initial concentrations are inside the bounds of the training set, while E indicates that the initial concentrations are outside the bounds. Note that the values in the E datasets encompass a range below the values in the training set and a range above the values in the training set. The minimum and maximum for both ranges are given.

When exposed to ionizing radiation, water decomposes into a number of ionic and excited states, which quickly decompose into radical and molecular species. H, OH⁺, H₂, and solvated electrons are formed within a few picoseconds; short-term products, additional radicals, and primary molecular products (H₂, H₂O₂, etc.) are formed within microseconds; and long-term products (secondary molecular products, such as O₂) can take up to an hour to form. Evolving the water radiolysis model over 10,000 s (approximately 2.8 h) starting from bulk water results in 2.42 × 10⁻⁷ M H⁺, 1.35 × 10⁻⁵ M H₂, 1.80 × 10⁻⁸ M H₂O₂, 7.02 × 10⁻⁹ M H₃O₂, 3.09 × 10⁻⁶ M O₂, 2.01 × 10⁻⁷ M O₃⁺, 1.84 × 10⁻¹⁰ M OH⁺, and 4.15 × 10⁻⁸ M OH⁻. Note that concentrations below 0.1 nM were assumed to be 0. These values were input into the iodine kinetic model along with various concentrations of I₂ and I⁺, depending on the dataset being generated. The system was integrated over 99 time points from 1 × 10⁻⁷ to 0.04 s to capture the full dynamics of the system. All kinetics equations were integrated using ChemPy.30

A training set and four test sets were generated for use with our RNN using the integration method described above. Of the 12 species with non-zero concentrations resulting from the integration of the kinetic model of aqueous-phase inorganic iodine reactions, H₂O, H₂O₂, and O₂ were not included because their concentrations did not change significantly during the reaction and thus were unnecessary to track, leaving nine species for our RNN to learn. In the training set, the initial concentration of I₂ ranged linearly between 1.00 × 10⁻⁷ and 5.00 × 10⁻⁷ M, while that of I⁺ ranged linearly between 1.00 × 10⁻⁸ and 5.00 × 10⁻⁸ M. Thirty-two concentrations were used for each species, giving a total of 1024 combinations. These 1024 simulations comprised our training set, 10% of which were used for validation during training of the network.

Our first test set was generated to examine the ability of our network to interpolate within the bounds of the initial concentrations in our training set. For this, concentrations between those in the training set were generated. In the training set, each concentration differed by 0.12 × 10⁻⁷ or 10⁻⁸ M, such that the initial concentrations in the training set were shifted by 0.06 × 10⁻⁷ or 10⁻⁸ M to force the initial concentrations to lie exactly between those in the training set, giving a total of 961 simulations. A second test set was generated to examine the ability of our network to extrapolate outside the bounds of our training data. In this set, the concentrations of I₂ ranged from 1.00 × 10⁻⁹ to 1.00 × 10⁻⁷, while those of I⁺ ranged from 1.00 × 10⁻¹⁰ to 1.00 × 10⁻⁸, again giving a total of 961 simulations. Two more test sets were generated, in which the initial I₂ concentrations were within the limits of the training set, while those of I⁺ were outside those limits, and vice versa. Table 1 shows the ranges of concentrations for each dataset, along with the number of
Network Structure. We apply a feed-forward network in which the simulation sequences of normalized concentrations pass from the input layer through a hidden LSTM layer to a fully connected output layer to produce the normalized concentrations of all species simultaneously at the next time step (Figure 1). In both the LSTM and fully connected layers, the input data undergo a series of linear transformations, the weights of which are learned during training, which are then passed through a nonlinear activation function. The LSTM layer also applies a series of gates designed to selectively retain and forget certain information from previous time steps; this gating is also learned during training. Given that the training data were generated by solving a set of coupled ordinary differential equations (ODEs), the process of training our network is an attempt to approximate the flow map of the set of ODEs. Note that the process of training our network required no a priori knowledge of the ODEs involved in generating the data.

Both the input and output of our network are a series of concentrations, the values of which vary from $10^{-10}$ to $10^{-7}$. Such large variation can decrease the performance of deep learning algorithms, and so we applied min–max scaling to the concentrations before training to ensure that the concentrations of all species were within the same order of magnitude (see Figure 2). During training, the rectified linear unit (ReLU) was used as the activation function in both the LSTM and fully connected layers to provide nonlinearity and prevent the occurrence of concentrations with negative values. ReLU is a non-negative, piecewise linear activation function that has become popular in recent years for its ability to produce sparse gradients during training.

At the beginning of training, the network parameters are set to an initial state. A poor initialization can impede the training of a highly nonlinear system; therefore, we used the popular initialization scheme outlined by Glorot and Bengio, where the initial values for the weights are drawn from a uniform distribution. Because this initialization scheme is nondeterministic, we set random seeds to ensure reproducible results. Furthermore, to ensure that the final trained network is not an artifact of initialization, we trained the network 16 times with different random seeds. All networks were trained until convergence—that is, we stopped training once no improvement in the loss of the validation set was seen over 100 consecutive epochs or once a predefined number of maximum epochs was reached. All networks were optimized using the Adam algorithm. The initial learning rate was set to 0.001 and allowed to dynamically decrease to a minimum of 0.0001 based on the validation loss; the learning rate was decreased by a factor of 0.2 after the validation loss remained steady for 20 consecutive epochs.

Because the network forecasts only a single time step ahead, predictions are made iteratively to derive the entire chemical evolution. Forecasts from each single time step are added to the growing evolution, which is fed back into the network to make the prediction at the next time step. It is expected that this process may lead to compounding errors.

RESULTS AND DISCUSSION

Network Performance. We first trained a single RNN on all 1024 simulations in our training set. The trained network was then used to predict the chemical evolutions in our four test sets. Figure 3 (left) shows an example prediction on the interpolation test set. One can see that the predictions, represented by dashed lines, almost exactly overlay the integrated model values, represented by solid lines. The error in the predictions over the entire interpolation test set is shown in Figure 3 (right). The errors for all species over all time steps are approximately two orders of magnitude smaller than the values of the concentrations, indicating the excellent interpolative ability of our RNN.
We next tested the extrapolative ability of our RNN. The mean error begins an order of magnitude higher than that of the interpolation test set and precipitously increases in the negative direction starting at \(3 \times 10^{-2}\) s. However, this large error is not consistent across the test set. The extremes of the extrapolation test set (i.e., very high and very low concentrations of both species) give much higher errors than the rest of the set. This is to be expected given that the network was not trained on data that represent this region. Removing the 200 simulations with the highest concentration and the 200 with the lowest, leaving 500 simulations in the extrapolation test set, prevents the error from exploding as time progresses (see Figure S2). Therefore, our RNN has the ability to extrapolate up to a point, after which the predictions catastrophically fail.

To gain a holistic understanding of the error in our predictions, we use the symmetric mean absolute percentage error (sMAPE), which was previously shown to be a good descriptor for time series problems, though it should be noted that large positive errors are penalized more than large negative errors

\[
sMAPE = 100\% \times \frac{2}{\sum_{k=1}^{k} \left| Y_k - \tilde{Y}_k \right| / \left( |Y_k| + |\tilde{Y}_k| \right)}
\]

where \(k\) is the forecasting horizon (98 in this case), \(Y_k\) are the concentrations at time \(t\) produced by integrating the kinetics equations (i.e., the ground truth), and \(\tilde{Y}_k\) are the predictions at time \(t\). The sMAPE expressed in this way has a lower bound of 0% and upper bound of 200%.

Table 2 gives the sMAPEs of the predictions of each species in all four test sets. All but two species have low sMAPEs. Notably, the sMAPEs of \(H^+\) and \(IO^-\) are above 50%. Both species have very low maximum concentrations of less than \(1.8 \times 10^{-10}\) over the entire training set. The comparatively low concentrations are why the comparatively large errors do not

**Table 2. sMAPE of the Predictions of Each Species for Four Distinct Test Sets Made by the RNN Trained on 1024 Simulations**

| species | \(I\) | \(\ell\) | \(I\) | \(\ell\) | \(I\) | \(\ell\) |
|---------|-------|-------|-------|-------|-------|-------|
| \(H^+\) | 52.41 ± 5.87 | 53.77 ± 7.07 | 100.43 ± 34.41 | 105.73 ± 38.92 |
| \(HOI\) | 0.69 ± 0.98 | 1.34 ± 2.40 | 42.63 ± 53.71 | 51.60 ± 55.22 |
| \(I^-\) | 0.32 ± 0.38 | 0.46 ± 0.79 | 32.91 ± 46.86 | 38.13 ± 48.35 |
| \(IO^-\) | 1.53 ± 2.70 | 3.84 ± 8.72 | 74.51 ± 74.46 | 80.40 ± 72.92 |
| \(I^+\) | 51.63 ± 6.94 | 52.56 ± 7.58 | 103.35 ± 34.41 | 109.48 ± 39.74 |
| \(O_2^-\) | 5.17 ± 5.32 | 5.32 ± 4.27 | 17.52 ± 18.48 | 24.28 ± 31.21 |
| \(OH^-\) | 1.29 ± 0.81 | 1.83 ± 1.82 | 39.55 ± 35.47 | 47.45 ± 37.48 |
| \(OH^+\) | 4.74 ± 2.58 | 5.31 ± 3.11 | 30.78 ± 35.47 | 35.05 ± 38.29 |

*I indicates that the initial concentrations are inside the bounds of the training set, while \(E\) indicates that the initial concentrations are outside the bounds.

We next tested the extrapolative ability of our RNN. The mean error begins an order of magnitude higher than that of the interpolation test set and precipitously increases in the negative direction starting at \(3 \times 10^{-2}\) s. However, this large error is not consistent across the test set. The extremes of the extrapolation test set (i.e., very high and very low concentrations of both species) give much higher errors than the rest of the set. This is to be expected given that the network was not trained on data that represent this region. Removing the 200 simulations with the highest concentration and the 200 with the lowest, leaving 500 simulations in the extrapolation test set, prevents the error from exploding as time progresses (see Figure S2). Therefore, our RNN has the ability to extrapolate up to a point, after which the predictions catastrophically fail.

To gain a holistic understanding of the error in our predictions, we use the symmetric mean absolute percentage error (sMAPE), which was previously shown to be a good descriptor for time series problems, though it should be noted that large positive errors are penalized more than large negative errors.
show up in strict analysis of the errors, but do appear when analyzed by the sMAPE.

Because the extrapolative ability of our RNN was poor compared to its interpolative ability, we next examined combined interpolation/extrapolation test sets in which the initial concentration of one species was within the range of those in the training set, while that of the other was outside of the range of the training set. As shown in Table 2, when the initial concentrations of I$_2$ were within the bounds of the training set, even if those of I$^*$ were out of bounds, the sMAPE for each species remained comparable to that when both species were within the bounds. In the opposite situation, when I$^*$ was within the bounds and I$_2$ was out of bounds, the sMAPE for each species was comparable to that when both species were out of bounds, though slightly lower in all cases. Figure 4 shows the distribution of the average sMAPE over all species for the four test sets. The interpolative test set (denoted I$_2$ (I) I$^*$ (I)) shows the tightest distribution with a mean sMAPE of 13.60 ± 1.66%. This is closely followed by the test set in which the initial concentrations of I$_2$ were within the bounds and those of I$^*$ were out of bounds (I$_2$ (I) I$^*$ (E)), with a mean sMAPE of 14.40 ± 2.92%. The mean sMAPE for the contrasting test set (I$_2$ (E) I$^*$ (I)) greatly increased to 52.57 ± 30.95%, which is similar to the mean sMAPE of the extrapolative test set (I$_2$ (E) I$^*$ (E)) of 58.76 ± 35.57%. This examination indicates that the network is more sensitive to the concentration of I$_2$, likely owing to its order-of-magnitude larger concentration compared with that of I$^*$. The sMAPEs of the I$_2$ (E) I$^*$ (I) and I$_2$ (E) I$^*$ (E) test sets range from 10 to 130%, displaying the large variability in the predictions of simulations in these two test sets.

**Effect of Training Data Size.** The RNNs discussed above were trained on 1024 simulations. This number of simulations, while common for deep learning applications in other domains (for comparison, Makridakis et al. used a set of 1045 series for a comprehensive comparison of deep learning versus statistical forecasting methods), is often not feasible when, for instance, one simulation corresponds to data from one experiment. Therefore, we examined the effect of minimizing the number of simulations used to train the network.

To create smaller training sets, we sampled from the initial training set consisting of 1024 simulations. Because the training of a network is highly dependent on the data structure, we prepared 16 resamplings of the larger dataset for each dataset of reduced size. We created five equal-sized bins based on the initial concentrations of I$_2$. We then sampled proportionately from those bins to ensure that the training data were not skewed around certain initial conditions. We created reduced datasets consisting of 512, 256, 128, 96, 64, and 32 simulations. For comparative purposes, the reported error is on the interpolative test set.

Figure 5 shows the variation in the average sMAPE over all species as the training set size decreases. The error bars represent the standard deviation in the average sMAPE for each resampling. As expected, the sMAPE increases as the training set size decreases; however, this decrease is not linear. The average sMAPE from the network trained on 512 simulations is 17%, compared to 14% for the network trained on 1024 simulations. Decreasing the training set size fourfold to 128 simulations increases the average sMAPE only by 4%. However, decreasing this set by a further 25% to 96 simulations increases the average sMAPE by another 5%, after which the increase in the average sMAPE becomes increasingly larger. In addition, the variation ranges between 1.8 and 3.0% until the training set is reduced to 96 simulations, after which the variation increases to above 5%. This indicates that as the number of examples in the training set decreases, one must be more discerning of how the samples are chosen to train the network.

**Network Variance.** Because RNNs use their learned internal representations, rather than exact templates from the training data, to make predictions, the final network parameters depend on the initialization conditions. This variability can lead to different trained networks, the variability of which can be quantified by training the network multiple times with different initial weights. Therefore, we retrained our RNN using the largest training set (1024 simulations) with different initial weights, set according to the NumPy and TensorFlow seeds. In each instance, the network converged to a loss on the order of 10$^{-5}$, but at different rates (see Figure S1 for plots of the loss during training). Notably, the same learning rate did not lead to convergence for every initial set of weights. The default learning rate for the Adam optimizer of 0.001 did not lead to convergence for 8 of the 16 seeds. In 6 cases, convergence was achieved by increasing the initial learning rate to 0.005, indicating that the initial weights were...
far from ideal. In the remaining 2 cases, convergence was achieved by decreasing the initial learning rate to 0.0005, indicating that these seeds provided good initial guesses for the network weights and taking too large of an initial step moved the network out of the ideal state. In all instances, an adaptive learning rate was used, where after 20 epochs of no improvement in the loss of the validation set, the learning rate was steadily decreased to a minimum of 0.0001. The sMAPEs for each species for the 16 different initial weight configurations were very similar (for a full listing of the sMAPEs for each initial configuration, see Table S2), with averages of 51.98 ± 4.20% for H+, 1.50 ± 1.08% for HOI, 0.36 ± 0.19% for I−, 4.72 ± 3.70% for I0, 53.25 ± 3.53% for IO3−, 4.98 ± 2.24% for I2, 1.59 ± 0.60% for O22−, 6.26 ± 3.52% for OH−, and 4.76 ± 1.45% for OH+. Thus, if the network is converged to the lowest possible loss, the predictions will be similar, though not identical.

# CONCLUSIONS

We have demonstrated that neural networks can be used to predict the chemical evolution of multiple species undergoing a series of interconnected aqueous inorganic iodine reactions that can result from an incident at a nuclear power plant. We showed that LSTM can capture the dynamics of multiple species, over multiple time steps. We also validated the usefulness of our network by considering the inherent network variance and various training set sizes. We believe that neural networks give rise to good surrogate models for chemistry forecasting problems in which data are available but the underlying mechanisms are not known. In practice, we envision that this technique will be useful in a real-world scenario in which scientists have the ability to capture data in real time via atmospheric sensors to examine the environmental transfer of iodine after a nuclear reactor incident. In such a situation, the identification of species may not be certain, but their concentrations can be tracked over time. Neural networks could then be used to perform accurate short-term forecasting of the unknown species. However, working with real-world sensor data presents its own challenges, such as anomalous readings, sensor failures, and noisy data. Neural networks open a path to the data-driven prediction of chemical evolutions that do not rely on explicit knowledge of the chemical species or mechanisms.

# ASSOCIATED CONTENT

## Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.9b04104.

Subset of aqueous inorganic iodine reactions used in this work; plot of training loss of models with different seeds; and sMAPE of each species as the size of the training set decreases with different samplings of the full dataset (PDF)

## AUTHOR INFORMATION

### Corresponding Author

**Jenna A. Bilbrey** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States; orcid.org/0000-0002-4199-7244; Phone: (509)372-4256; Email: jenna.bilbrey@pnnl.gov

### Authors

**Carlos Ortiz Marrero** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States

**Michel Sassi** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States; orcid.org/0000-0003-2582-3735

**Andrew M. Ritzmann** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States; orcid.org/0000-0002-9291-7208

**Neil J. Henson** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States; orcid.org/0000-0002-1842-7884

**Malachi Schram** — Pacific Northwest National Laboratory, Richland, Washington 99352, United States

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.9b04104

### Notes

The authors declare no competing financial interest.

### ACKNOWLEDGMENTS

The authors thank Robert Rallo, Jan Strube, and Mathew Thomas for insightful discussions. The research described in this paper was conducted under the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory (PNNL), a multi-program national laboratory operated by Battelle for the U.S. Department of Energy, Release no. PNNL-SA-148824.

### REFERENCES

(1) Miyake, Y.; Matsuzaki, H.; Fujiwara, T.; Saito, T.; Yamagata, T.; Honda, M.; Muramatsu, Y. Isotopic ratio of radioactive iodine (129I/131I) released from Fukushima Daiichi NPP accident. Geochim. J. 2012, 46, 327–333.

(2) Lebel, L. S.; Dickson, R. S.; Glowa, G. A. Radioiodine in the atmosphere after the Fukushima Dai-ichi nuclear accident. J. Environ. Radioact. 2016, 151, 82–93.

(3) Khazov, Y.; Mitropolsky, I.; Rodionov, A. Nuclear Data Sheets for A = 131. Nucl. Data Sheets 2006, 107, 2715–2930.

(4) Takeyasu, M.; Nakano, M.; Fujita, H.; Nakada, A.; Watanabe, H.; Sumiya, S.; Furuta, S. Results of environmental radiation monitoring at the nuclear fuel cycle engineering laboratories, JAEA, following the Fukushima Daiichi nuclear power plant accident: Fukushima NPP accident related. J. Nucl. Sci. Technol. 2012, 49, 281–286.

(5) Likhtarev, I. A.; Shandala, N. K.; Gulkov, G. M.; Kairo, I. A.; Chepuny, N. I. Ukrainian thyroid doses after the Chernobyl accident. Health Phys. 1993, 64, 594–599.

(6) Robbins, J.; Schneider, A. B. Radiiodine-induced thyroid cancer: studies in the aftermath of the accident at Chernobyl. Trends Endocrinol. Metab. 1998, 9, 87–94.

(7) Robbins, J.; Schneider, A. B. Thyroid cancer following exposure to radioactive iodine. Rev. Endocr. Metab. Disord. 2000, 1, 197–203.

(8) Zvonova, I.; Balonov, M. The Chernobyl Papers. Doses to the Soviet Population and Early Health Effects Studies, Research Enterprises Inc., 1993; Vol. 1.

(9) Unno, N.; Minakami, H.; Kubo, T.; Fujimori, K.; Ishiwata, I.; Terada, H.; Saito, S.; Yamaguchi, I.; Kunugita, N.; Nakai, A.; Yoshimura, Y. Effect of the Fukushima nuclear power plant accident on radioiodine (131I) content in human breast milk. J. Obstet. Gynecol. Res. 2012, 38, 772–779.

(10) Yamashita, S.; Suzuki, S.; Suzuki, S.; Shimura, H.; Saeiko, V. Lessons from Fukushima: latest findings of thyroid cancer after the Fukushima nuclear power plant accident. Thyroid 2018, 28, 11–22.

(11) Vikis, A. C.; MacFarlane, R. Reaction of iodide with ozone in the gas phase. J. Phys. Chem. 1985, 89, 812–815.
(12) Wren, J. C.; Ball, J. M. LIRIC 3.2 an updated model for iodine behaviour in the presence of organic impurities. Radiat. Phys. Chem. 2001, 60, 577−596.

(13) Guentay, S.; Cripp, R. C.; Jäckel, B.; Bruchertseifer, H. Iodine behaviour during a severe accident in a nuclear power plant. CHIMIA 2005, 59, 957−965.

(14) Bosland, L.; Funke, F.; Girault, N.; Langrock, G. PARIS project: Radiolytic oxidation of molecular iodine in containment during a nuclear reactor severe accident: Part 1. Formation and destruction of air radiolysis productsExperimental results and modelling. Nucl. Eng. Des. 2008, 238, 3542−3550.

(15) Bosland, L.; Funke, F.; Langrock, G.; Girault, N. PARIS project: Radiolytic oxidation of molecular iodine in containment during a nuclear reactor severe accident: Part 2. Formation and destruction of iodine oxides compounds under irradiation—Experimental results modelling. Nucl. Eng. Des. 2011, 241, 4026−4044.

(16) Graves, A. Generating sequences with recurrent neural networks. 2013, arXiv:1308.0850. arXiv preprint.

(17) Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K. Google’s neural machine translation system: Bridging the gap between human and machine translation. 2016, arXiv:1609.08144. arXiv preprint.

(18) Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015; pp 3156−3164.

(19) Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735−1780.

(20) Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A Field Guide to Dynamical Recurrent Networks; IEEE, 2001.

(21) Bini, R.; Chiappe, C.; Duce, C.; Micheli, A.; Solaro, R.; Starita, A.; Tiné, M. R. Ionic liquids: prediction of their melting points by a recursive neural network model. Green Chem. 2008, 10, 306−309.

(22) Luschi, A.; Pollastri, G.; Baldi, P. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. J. Chem. Inf. Model. 2013, 53, 1563−1575.

(23) Liu, B.; Ramsundar, B.; Kavthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic reaction prediction using neural sequence-to-sequence models. ACS Cent. Sci. 2017, 3, 1103−1113.

(24) Fooshee, D.; Mood, A.; Gutman, E.; Tavakoli, M.; Urban, G.; Liu, F.; Huynh, N.; Van Vranken, D.; Baldi, P. Deep learning for chemical reaction prediction. Mol. Syst. Des. Eng. 2018, 3, 442−452.

(25) Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative recurrent networks for de novo drug design. Mol. Inf. 2018, 37, 1700111.

(26) Müller, A. T.; His, J. A.; Schneider, G. Recurrent neural network model for constructive peptide design. J. Chem. Inf. Model. 2018, 58, 472−479.

(27) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. Science 2018, 361, 360−365.

(28) Poinssot, C.; Ferry, C.; Kelm, M.; Cavedon, J.; Corbel, C.; Jegou, C.; Lovera, P.; Miserque, F.; Poulesquen, A.; Grambow, B. Spent Fuel Stability under Repository Conditions—Final Report of the European Project, 2005.

(29) Pastina, B.; LaVerne, J. A. Effect of molecular hydrogen on hydrogen peroxide in water radiolysis. J. Phys. Chem. A 2001, 105, 9316−9322.

(30) Dahlgren, B. ChemPy: A package useful for chemistry written in Python. J. Open Source Software 2018, 3, 565.

(31) Hirsch, M. W.; Smale, S.; Devaney, R. L. Differential Equations, Dynamical Systems, and an Introduction to Chaos; Academic press, 2012.

(32) Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011; pp 315−323.