FFI Norwegian Defence Research Establishment

21/02531

FFI-RAPPORT

Predicting scalar coupling constants via machine learning

Fredrik Bakken
Lars L Sandberg
Dennis Christensen
Thor Engøy
Hallvar Gisnås
Lars Aurdal

Predicting scalar coupling constants via machine learning

Fredrik Bakken
Lars L Sandberg
Dennis Christensen
Thor Engøy
Hallvar Gisnås
Lars Aurdal

Norwegian Defence Research Establishment (FFI) 21 December 2021

Keywords
Maskinlæring
Nevrale nettverk
Gradient boosting
Kvantekjemi
Skalarkoblingskonstant

FFI report
21/02531

Project number
5410

Electronic ISBN
978-82-464-3382-0

Approvers
Stig Rune Sellevåg, *Research Manager*
Janet M Blatny, *Research Director*
Ann-Kristin Elstad, *Principal Scientist*

The document is electronically approved and therefore has no handwritten signature.

Summary

Over the preceding decade, machine learning techniques have been successfully applied in several fields of research, including the prediction of chemical properties of atoms and molecules. Whereas conventional quantum chemical methods can be very computationally expensive, machine learning algorithms give rise to fast and accurate predictions beyond the known data set, given that they have been trained with a sufficient amount of quality data.

Online platforms, such as Kaggle (kaggle.com), host machine learning competitions with well-defined problem descriptions and a substantial amount of accompanying data. These provide a well-defined objective with a clear-cut deadline, making them ideal for short-term focused research work. In addition, the Kaggle website serves as an interactive learning environment, with a continually updated scoreboard and public discussion forum.

During the summer of 2019, a team of students and scientists at the Norwegian Defence Research Establishment (FFI) participated in the Kaggle competition *Predicting Molecular Properties*, where the task was to predict the scalar coupling constant via machine learning. The scalar coupling constant is an expression of the magnetic interactions between atoms in a molecule and depends on its atomic composition and geometry. We investigated several mathematical representations of molecular data as inputs to various supervised learning algorithms, including deep neural networks and gradient boosting trees. Combining the molecules' distance matrices with angular information provided a flexible data representation, enabling accurate predictions. Our most successful model comprised an ensemble of deep neural networks and gradient boosting trees, resulting in a 308th place among the 2,737 competing teams. A key factor of the team's success was the mixture of relevant domain expertise and machine learning experience.

Sammendrag

I løpet av det foregående tiåret har maskinlæringsteknikker blitt anvendt innen en rekke forskningsområder, inkludert prediksjon av kjemiske egenskaper til atomer og molekyler. Til forskjell fra konvensjonelle kvantekjemiske metoder som kan være meget beregningskrevende, gir maskinlæringsalgoritmer raske og presise prediksjoner utover det kjente datasettet, gitt at de har blitt trent med tilstrekkelig mengde data av god kvalitet.

Online-plattformer, som Kaggle (`kaggle.com`), er vertskap for maskinlæringskonkurranser med klart definerte problembeskrivelser og en betydelig mengde tilhørende data. Disse setter rammer for fokusert forskning over kort tid med veldefinert målsetting og kort tidsfrist. I tillegg fungerer Kaggle-nettsidene som en interaktiv læringsplattform med en kontinuerlig oppdatert rangeringsliste og et åpent diskusjonsforum.

Sommeren 2019 deltok et team bestående av studenter og forskere ved Forsvarets forskningsinstitutt (FFI) i Kaggle-konkurransen *Predicting Molecular Properties* hvor oppgaven var å predikere den skalare koplingskonstanten ved hjelp av maskinlæring. Den skalare koplingskonstanten er et uttrykk for den magnetiske vekselvirkningen mellom atomer i et molekyl og avhenger av dets atomære sammensetning og geometri. Vi undersøkte flere matematiske representasjoner av molekyldata som input til ulike veiledede læringsalgoritmer, inkludert dype nevrale nett og gradient boosting-trær. En kombinasjon av molekylenes distansematriser og angulær informasjon gav en fleksibel datarepresentasjon som muliggjorde presise prediksjoner. Vår mest vellykkede modell inneholdt et ensemble av nevrale nett og gradient boosting-trær og resulterte i en 308. plass av de 2737 deltakende teamene. En nøkkelfaktor for teamets suksess var å kombiner og utnytte relevant domenekunnskap og erfaring med maskinlæringsoppgaver fra ulike forskningsgrupper på FFI.

Contents

1 Introduction

In this report, we describe our contribution to the 2019 Kaggle competition *Predicting Molecular Properties* hosted by members of CHemistry and Mathematics in Phase Space (CHAMPS) at the University of Bristol, Cardiff University, Imperial College London and the University of Leeds [1]. This activity was conducted as part of a summer project at the Norwegian Defence Research Establishment (FFI) from June to August 2019. The aim of the competition was to predict the scalar coupling constant for a large collection of molecules via machine learning. To this end, we investigated several mathematical representations of molecular data and supervised learning machines, including deep neural networks and gradient boosting trees. In this report, we cover our most successful attempts and describe our progress in the competition. Further details such as Python code installation guides for replicability and a quick set up of the environment can be found in the accompanying Github repository [2].

1.1 Motivation

The nature of Kaggle competitions differs from that of the research usually conducted at FFI in a number of ways. For instance, Kaggle competitions have more well-defined goals and operate on a shorter time frame than most FFI research projects, which typically last around three years. Kaggle competitions usually span only a few months and have a clear-cut deadline, making them ideal for student summer projects. The Kaggle website serves as an interactive learning environment throughout the scope of the competition, with a continually updated scoreboard and public discussion forum. Additionally, the main prize (usually a cash award or the opportunity to publish ones results) serves as motivation for the students. The hope of advancing on the scoreboard continually encourages exploration of new techniques. On the other hand, in many FFI research projects, it can be difficult to pursue ideas which diverge from the original project goals, particularly if they span more than one scientific discipline and require work by multiple existing research groups.

Due to its interdisciplinary content, the CHAMPS Kaggle competition gained the interest of researchers in two different FFI divisions with backgrounds in mathematics, computational chemistry and machine learning who supported the team effort. The core of the team comprised Bakken and Sandberg (our two summer students) who conducted the programming, regularly evaluated different models and submitted the results to Kaggle, continually monitoring the team's progress.

For a research team to succeed at a Kaggle competition, a combination of both domain expertise and a solid foundation in machine learning is required. For instance, in the CHAMPS competition, which was based in computational chemistry, chemists and physicists first use their intuition and background knowledge to deduce which input variables should be the most relevant, guiding the search for useful features in the dataset. The machine learning experts then combine this insight with their experience to construct a useful predictive model. This process naturally encourages discussion across scientific domains, since one can rarely know in advance whether it is a change in feature engineering or model architecture which will lead to the next improvement in the model's predictive accuracy.

1.2 Brief scientific background

Over the preceding decade, there has been a growing interest in the applications of machine learning in quantum chemical calculations [3, 4, 5, 6, 7, 8, 9, 10]. Conventionally, such calculations rely on solving approximations of the Schrödinger equation, like those constructed from density functional theory (DFT). Although DFT yields relatively low computational costs when compared to other traditional alternatives like the Hartree-Fock method, it usually takes a substantial amount of computing time to calculate molecular quantities (like energy) for even a single moderately sized molecule. Even worse, computing time grows rapidly as the number of atoms involved increases.

The computational challenges mentioned above have inspired an investigation into the applications of machine learning in calculating quantities like molecular energies [3, 4, 5, 6, 7, 8, 9, 10], Mulliken charges [11] and force fields [9, 10, 12]. Considerable efforts have also been made in constructing faithful representations of molecular data for such tasks [9, 13]. In the machine learning approach, instead of constructing an explicit approximation, one instead thinks of solving the Schrödinger equation as a black box function, in which a molecular description maps to the desired molecular quantity. This function is then approximated by a suitable machine learning method (like a neural network) and trained on pre-existing calculated data. Regarding calculating energies, Schutt et al. have shown that deep tensor and continuous-filter convolutional neural networks achieve the same accuracy as DFT for a large collection of small molecules (a mean average error of 0.03 eV) [9, 10].

This project investigates ways of predicting the scalar coupling constant via machine learning. Also known as the J-coupling constant, the scalar coupling constant measures (in Hz) an indirect magnetic interaction between two atomic nuclei with non-zero spin. This interaction arises from a series of hyperfine interactions involving the two nuclei and their local electrons, like those between (i) the magnetic moments of the nuclei and the electrons due to their spin, (ii) the nuclear magnetic moments and the magnetic field due to orbital angular momentum of the electrons, and (iii) the nuclei due to their magnetic moments [14]. The hyperfine splitting of the molecular energy levels is experimentally used to identify molecules [15]. Scalar couplings are characterised by which two atoms are involved in the magnetic interaction, and the number of bonds between these two atoms. The Kaggle competition data set included eight different coupling types, namely 1JHC, 1JHN, 2JHH, 2JHC, 2JHN, 3JHH, 3JHC and 3JHH. Here, the two last letters describe which atoms are coupled and the integer before the J denotes the number of bonds between the two atoms [16].

1.3 Kaggle competition problem description

The problem description for the competition reads as follows [1]:

> In this competition, you will be predicting the `scalar_coupling_constant` between atom pairs in molecules, given the two atom types (e.g., C and H), the coupling type (e.g., 2JHC), and any features you are able to create from the molecule structure (xyz) files.
>
> For this competition, you will not be predicting all the atom pairs in each molecule rather, you will only need to predict the pairs that are explicitly listed in the train and test files. For example, some molecules contain Fluorine (F), but you will not be predicting the scalar coupling constant for any pair that includes F.
>
> The training and test splits are by molecule, so that no molecule in the training data is found in the test data.

1.4 Acknowledgements

Bakken and Sandberg would like to thank their co-student Niels Aase for help with calculating Mulliken charges and magnetic dipole moments, which were used in the final feature set.

2 Methods

2.1 Data description and feature engineering

We now cover how we extracted useful features and data representations from the input data provided in the Kaggle competition. In its raw form, this comprised geometric characterisations of the molecules in question, along with various key physical properties, listed in Section 2.1.2. During the course of the project, we discovered the importance of extracting useful feature representations of this data, to enable successful training of our machine learning algorithms. One of the most important tasks was to extract a meaningful mathematical representation of the geometric structures of the molecules.

2.1.1 Molecular structures

The dataset file `structures.csv` contained the geometric structures of the molecules used for training, specifying the Cartesian coordinates of all the atoms in each molecule along with their atomic numbers. For a molecule consisting of n atoms, we write its coordinates as $\{(x_i, y_i, z_i) \mid i = 1, \dots, n\}$. In order to obtain a representation which is invariant of isometries of $\mathbb{R}^3$ (that is, rotations, translations and reflections), we calculated the distance matrix

$$D = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{pmatrix} \tag{2.1}$$

for each molecule, where

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{2.2}$$

This representation for molecular structures has successfully been utilised in deep-tensor neural network models for predicting molecular energies [9]. Another closely related representation is the Coulomb matrix, whose merit for this kind of task has also been illustrated [3].

Along with atomic distances, the scalar coupling constant also depends on the angles between the atoms in the coupling and the other atoms in the molecule. Our most successful model used both atomic distances and angular data as input. The inspiration for calculating the angles was taken from the open-source kernel "Keras Neural Net for CHAMPS" [17]. Intuitively, one would expect the effect on the coupling constant to be strongest by those atoms closest to the coupling pair. This was confirmed by a gradient boosting feature importance test we performed (see Section 2.2.2) [18]. We found that our models improved after removing all distances and angles except those involving the coupling pair.

2.1.2 Other features

The training set included not only geometric descriptors for the molecules' structures, but also other relevant physical properties, namely dipole moments, magnetic shielding tensors, Mulliken charges, potential energies and the contributions from the interactions listed in Subsection 1.2. Our most successful model included two of these: dipole moments and Mulliken charges. A full list of the data types of these physical properties and their definitions can be found on the Kaggle competition's web page [1].

2.2 Training of algorithms and validation of results

In this section, we describe the machine learning models used. According to the rules of the Kaggle competition, their performance were scored and ranked using the following score function [1]

$$\text{score} = \frac{1}{T}\sum_{t=1}^{T} \log\left(\frac{1}{n_t}\sum_{i=1}^{n_t} |y_i - \hat{y}_i|\right), \tag{2.3}$$

where

- T is the number of scalar coupling types,
- n_t is the number of scalar couplings for each type,
- y_i is the actual scalar coupling constant,
- $\hat{y}_i$ is the predicted scalar coupling constant.

Note that a negative score is possible (and indeed desirable) due to taking logarithms.

Any machine learning algorithm F is trained with respect to an empirical estimate of a specified loss function L. Typically, the loss to be minimised can then be expressed as

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i), \tag{2.4}$$

where y_i are the output data and $\hat{y}_i = F(x_i)$ are the predictions, for $i = 1\ldots, n$. If possible, it is often desirable to choose a loss function which is mathematically related to the score function. However, the score function (2.3) does not take the same form as an empirical estimate like in (2.4). Hence, in our first attempts, where all data was passed through the same model regardless of their coupling type, we could not train our models with respect to the score function directly. For these models, we found that using the mean absolute error as loss function served as a good proxy. However, in later attempts, we passed data into distinct models based on their coupling type. These models were then trained separately, again with respect to the mean absolute error as loss function. Due to the separation by coupling type, this is in fact mathematically equivalent to training with respect to the score function (2.3) directly.

2.2.1 Deep neural networks

Deep neural networks (DNN) are a model used to approximate some complicated mathematical function. A DNN can for example map input images to discrete output categories such as cat or dog. Such networks are referred to as image classifiers. Another example is a DNN mapping input facial images to the output age of the person photographed. In this case, the output of the network is continuous. Since predicting the scalar coupling constant is a regression problem, continuous-output networks are needed. The term *neural network* refers to the fact that DNNs are typically represented by composing many functions iteratively, loosely mirroring the interaction between neurons in the human brain. The input data is typically propagated through many functions, known as layers, and the number of layers is referred to as the depth of model. In each layer, a constant term called the bias is also added. Figure 2.1 shows the concept of the input data propagating through the network to produce the output. Each circle can be thought of as representing an artificial neuron which activates depending on the input it receives [19]. The coloured nodes denote the bias terms.

Initially, the parameters of a DNN are set randomly, and so the model will have no predictive power. However, during training, in which the model is exposed to large amounts of input and

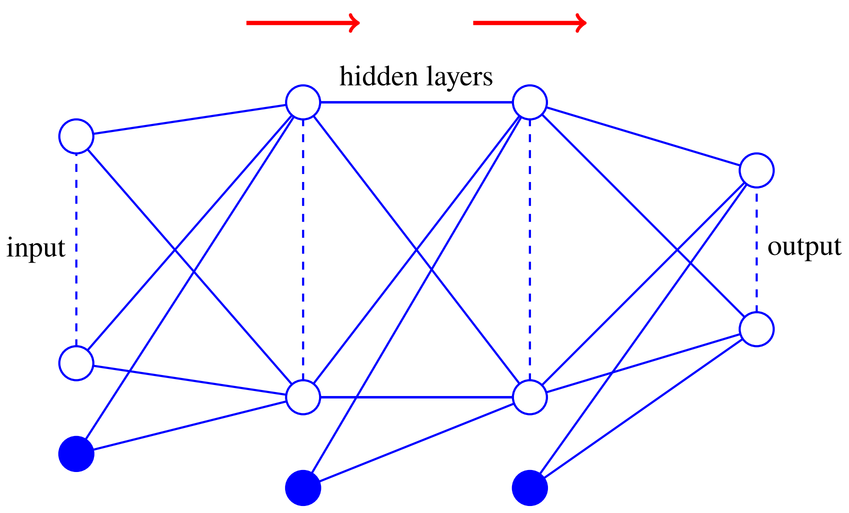


Figure 2.1 *Geometric diagram illustrating a typical fully connected neural network architecture with two hidden layers. The circles represent the artificial neurons; the filled circles represent the bias neurons. The arrows indicate the direction of propagation of inputs.*

output data, the parameters are tuned accordingly, resulting in a (hopefully) better performance. This tuning is done with two mathematical optimisation tools, namely automatic differentiation and stochastic gradient descent [19], both of which are easy to implement in TensorFlow [20, 21]. Rather than using stochastic gradient descent directly as optimiser, we instead chose to use Adam, which is known for its stable performance [22].

To establish a benchmark result, our first attempts in the competition involved simple DNNs. However, we quickly discovered that an approach based on neural networks alone did not produce competitive results. We instead shifted our focus to models comprising an ensemble of smaller neural networks and gradient boosting algorithms, whose accuracy far exceeded those of DNNs in spite of having fewer trainable parameters.

2.2.2 Gradient boosting

Gradient boosting (GB) based methods have been shown to be successful in similar Kaggle competitions with tabular data input, yielding faster training and simpler model architectures than neural networks. Following the original construction of GBs [23, 24], we assume the data is produced by some true algorithm F_{true} which we want to estimate. We use the term *learner* to mean a function from input to target space attempting to approximate F_{true}. The main idea is to sequentially construct a single strong learner as a linear combination of weak learners. These weak learners are usually taken as easily computable, simple functions, like decision trees. After M steps, we may thus write

$$F_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x),$$

where $h_m(x)$ are weak learners and γ_m are coefficients to be chosen.

We go through the main steps of choosing h_m and γ_m inductively. Say we have observed input data x_i and output data y_i, $i = 1, \ldots, n$. To start off, we choose a weak learner (like the mean of the observed data) as h_1, and set $\gamma_1 = 1$. Now, suppose we have found $F_m = \sum_{k=1}^{m} \gamma_k h_k$ where $m \geq 1$. In order to find a suitable learner h_{m+1} and coefficient γ_{m+1}, we would like to minimise the loss

$$h_{m+1} = \arg\min_{h} \sum_{i=1}^{n} L\left(h, F_m(x_i) - y_i\right). \tag{2.5}$$

However, this optimisation problem is computationally intractable since the space $\mathcal{H}$ of real differentiable functions (in which h lives) is too large. Therefore, one usually only chooses h from a finite set $\mathcal{H}_0$ of candidate functions. In any case, we now employ the principle of steepest descent in order to guide our choice of learner. That is, we calculate the negative functional gradient of the cost function, evaluated at the observed data:

$$r_{mi} = -\left.\frac{\partial L(y_i, F)}{\partial F}\right|_{F=F(x_i)}, \quad i = 1, \ldots, n. \tag{2.6}$$

The values r_{mi} are referred to as the pseudo-residuals. The next learner h_{m+1} is then fitted to the pseudo-residuals, so that the choice of γ_{m+1} is equivalent to a line-search along the axis of steepest descent for the loss function. That is, having found h_{m+1}, we determine the value of γ_{m+1} by solving the one-dimensional optimisation problem

$$\gamma_{m+1} = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_m(x_i) + \gamma h_m(x_i)\right). \tag{2.7}$$

Then we update our learner by letting $F_{m+1} = F_m + \gamma_{m+1} h_{m+1}$.

In our implementation of GBs, we let the weak learners be decision trees. Since the depth of a decision tree corresponds to the number of interdependencies of variables in the model, we found it necessary to enforce a bound on its size in order to prevent overfitting. Overall, GBs were found to be a quick and effective method, and were used throughout the project.

2.2.3 Ensembling

An effective way to improve results is to combine several successful models into one. One of the most popular ways of doing so is via ensembling, which refers to taking a convex combination of the existing models. In the case of two models F_1, F_2, this would mean constructing a new model F by letting

$$F(x) = \alpha F_1(x) + (1 - \alpha) F_2(x), \tag{2.8}$$

where $0 \leq \alpha \leq 1$ is a constant, referred to as the weighting proportion of the convex combination. Most commonly, α is estimated via trial and error. Ensembling rarely yields a substantial improvement unless the Pearson correlation (see Reference [25]) for the individual models' validation scores is sufficiently low.

In our case, we used ensembling to combine a neural network with a gradient boosting algorithm. Rather than ensembling these two models directly, we instead performed this procedure separately for each coupling type, which substantially improved our results. That is, for each coupling type in the data set, we trained a neural network and a gradient boosting algorithm, and then ensembled them. The weighting proportions α were selected from the values $\{0.1, 0.2, \ldots, 0.9\}$ so as to maximise the validation score. In Table 2.1, we summarise our findings for 3 different coupling types, listing the score achieved for the neural network, the gradient boosting algorithm, their Pearson correlation, the optimal weighting proportion α and the score for the ensemble. We note that the Pearson correlations are relatively low for all three coupling types and that the ensembled models achieved a better score than any of the isolated DNN or GB models.

Table 2.1 Comparison of validation errors for the DNN and GB models, and the ensemble model. Validation errors for the DNN and GB models, their Pearson correlation, the relative weighting proportion and the error of the ensemble model.

Coupling type	DNN error	GB error	Correlation	α	Ensemble error
1JHC	0.645	0.639	0.686	0.5	0.560
2JHC	-0.554	-0.516	0.613	0.6	-0.627
3JHC	-0.614	-0.662	0.650	0.4	-0.730

Overall, we found ensembling to be an effective way of combining our most successful models to improve results.

3 Results

During the course of this summer project, we experimented with various machine learning models and choices of features. We now summarise this exploration, concluding with how our best score was attained. We remind the reader that a lower score means better performance.

In order to establish a benchmark result, we first trained a DNN with plain Cartesian coordinates as features, which achieved a score of -0.048. As described in section 2.1.1, Cartesian coordinates do not provide a faithful representation of molecular data, since they are not invariant under isometries of $\mathbb{R}^3$. We therefore explored using distance matrices as input to our DNN. This improved the score substantially, to -0.457. Inspired by their success in previous similar competitions, we next tried using a GB algorithm rather than a DNN. This improved the score even further, to -0.771. Including Mulliken charges and magnetic dipole moments as additional features and separating our GB algorithms by coupling type improved the score again to -0.922. Finally, we performed ensembling as described in Section 2.2.3, combining a DNN with a GB algorithm, separately for each coupling type. Our best score of -1.008 was achieved when using distance matrices, Mulliken charges and magnetic dipole moments as features. This put us in 308th place out of 2,737 submissions (12th percentile). All code is available on GitHub, and can provide a starting point for tabular data based regression [2].

The competition's winning team, whose score was -3.24530, were from Bosch Research and consisted of both machine learning experts and domain experts [26]. For the task, they employed a novel type of neural network, called a graph transformer [27], with a particular choice of activation function involving the distance matrix. As feature set, they developed a hierarchical embedding with different levels of specificity for the different atoms and bonds. Depending on how many atoms were involved in a certain level, different geometric descriptors such as bond length and angles between atoms were used. Finally, they combined 13 slight variations of the same model via ensembling. Their work illustrates the importance of collaboration between different domains of expertise for such tasks. It also serves as motivation to stay on top of the existing machine learning literature in order to be able to explore novel techniques and model architectures.

4 Conclusion

In this report, we have presented our work over the summer of 2019 as a participating team in the Kaggle competition *Predicting molecular properties*, hosted by CHAMPS. The competition's objective was to employ machine learning techniques to predict the scalar coupling constant from molecular data. During the course of the project, we experimented with different feature representations and machine learning models. Our choices were initially motivated by the domain expertise of the broader team of FFI scientist, before being implemented and tested by our two summer students Bakken and Sandberg. This involved regularly testing and combining different techniques, continually monitoring the team's progress on the scoreboard. New ideas for further experimentation were guided by studying relevant publications and the Kaggle discussion forum. Our best submission attained a score of -1.008, putting us in 308th place out of $2,737$ submissions.

This project illustrates the merit of participating in competition-based research in FFI summer projects. The fact that Kaggle competitions provide a well-defined objective with a clear-cut deadline makes them ideal for short-term work. The continually updated scoreboard provides a concise way to evaluate ones own work, and serves as motivation for improving it. In addition, such machine learning based competitions usually span more than one scientific domain, and thus naturally encourage interdisciplinary collaboration. This creates a particularly interactive learning environment for the students, through discussions within the research team as well as on the Kaggle discussion forum.

4.1 Lessons learned

Finally, based on their experience, here are our summer students Bakken's and Sandberg's top five tips for competing in Kaggle competitions:

1. Follow the popular discussions, especially those where competitors with good results meet, to get pointers on features and architectures to try out.
2. Download good kernels posted and reproduce their results. This can give a boilerplate code and yield insights with little effort.
3. Create a smaller mock dataset at an early stage for quick and effective experimentation. This allows work on both feature engineering and architectures to proceed concurrently.
4. Evaluate your choice of features systematically via a feature importance test. This can be done in several ways. For this competition, we utilised Breiman's variable importance test [18].
5. Avoid the temptation to copy the work of others without understanding it yourself, although this seems to be common in Kaggle competitions. We decided to constrain ourselves with a handful of rules (write all code, generate all data, train everything from scratch etc.) which is recommended for maximum learning output.

References

[1] Kaggle competition: Predicting molecular properties. `https://www.kaggle.com/c/champs-scalar-coupling`, 2019.

[2] Lars Sandberg and Fredrik Bakken. Project GitHub repository. `https://github.com/FredrikBakken/predicting-molecular-properties`, 2019.

[3] Matthias Rupp et al. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[4] John C Snyder et al. Finding density functionals with machine learning. *Physical review letters*, 108(25):253002, 2012.

[5] Grégoire Montavon et al. Learning invariant representations of molecules for atomization energy prediction. *Advances in Neural Information Processing Systems*, pages 440–448, 2012.

[6] Grégoire Montavon et al. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[7] Katja Hansen et al. Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation*, 9(8):3404–3419, 2013.

[8] Katja Hansen et al. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters*, 6(12):2326–2331, 2015.

[9] Kristof T Schütt et al. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8:13890, 2017.

[10] Kristof T Schütt et al. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in Neural Information Precessing Systems*, pages 991–1001, 2017.

[11] Boris Dorado. Predicting Mulliken charges with ACSF descriptors. `https://www.kaggle.com/borisdee/predicting-mulliken-charges-with-acsf-descriptors`, 2019.

[12] Stefan Chmiela et al. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, 9(1):3887, 2018.

[13] Michael Gastegger et al. wACSF — weighted atom-centered symmetry functions as descriptors in machine learning potentials. *The Journal of chemical physics*, 148(24):241709, 2018.

[14] Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Quantum mechanics: non-relativistic theory*, volume 3. Elsevier, 1977.

[15] Hans J Reich. Chem 605 - structure determination using spectroscopic methods online course notes. `https://www.chem.wisc.edu/areas/reich/nmr/Notes-05-HMR-v26-part2.pdf`, 2017.

[16] Elizabeth Ter Sahakyan. Predicting scalar coupling constants using machine learning. https://medium.com/@liztersahakyan/predicting-scalar-coupling-constants-using-machine-learning-c213af14e862.

[17] Tod Newman. Keras neural net for CHAMPS. https://www.kaggle.com/todnewman/keras-neural-net-for-champs, 2019.

[18] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[19] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[20] Guido Van Rossum and Fred L Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[21] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[24] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

[25] Marlos Michailidis. Investigating machine learning methods in recommender systems (Thesis). University College London, 2017.

[26] Kaggle competition: Predicting molecular properties - winner's solution. https://www.kaggle.com/c/champs-scalar-coupling/discussion/106575, 2019.

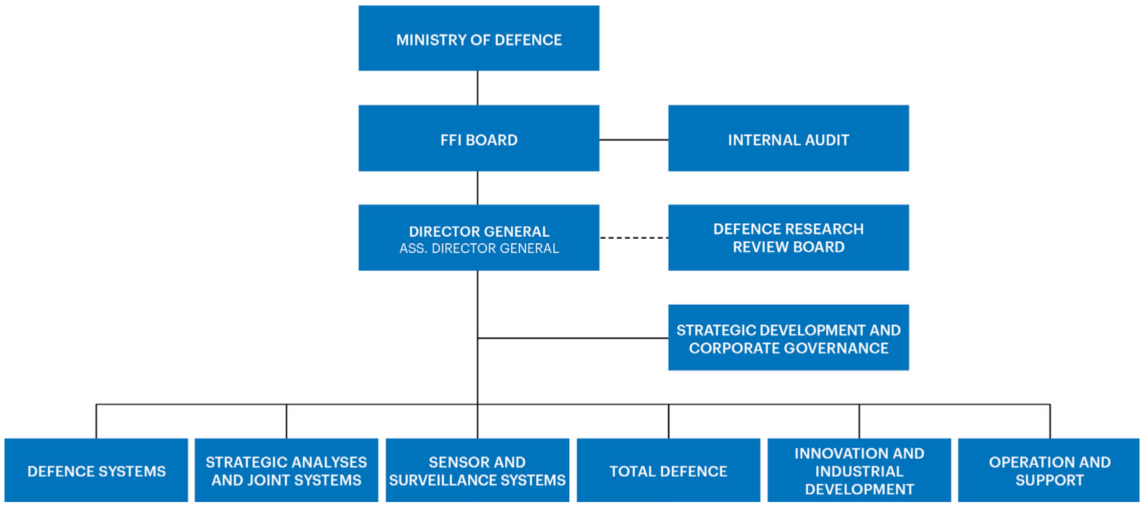
[27] Seongjun Yun et al. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32:11983–11993, 2019.

About FFI
The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's mission
FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's vision
FFI turns knowledge and ideas into an efficient defence.

FFI's characteristics
Creative, daring, broad-minded and responsible.


MINISTRY OF DEFENCE
FFI BOARD
INTERNAL AUDIT
DIRECTOR GENERAL
ASS. DIRECTOR GENERAL
DEFENCE RESEARCH REVIEW BOARD
STRATEGIC DEVELOPMENT AND CORPORATE GOVERNANCE
DEFENCE SYSTEMS
STRATEGIC ANALYSES AND JOINT SYSTEMS
SENSOR AND SURVEILLANCE SYSTEMS
TOTAL DEFENCE
INNOVATION AND INDUSTRIAL DEVELOPMENT
OPERATION AND SUPPORT

Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: post@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: post@ffi.no