A machine learning platform for the discovery of materials

Carl E. Belle1*, Vural Aksakalli2 and Salvy P. Russo1

Abstract
For photovoltaic materials, properties such as band gap $E_g$ are critical indicators of the material’s suitability to perform a desired function. Calculating $E_g$ is often performed using Density Functional Theory (DFT) methods, although more accurate calculation are performed using methods such as the GW approximation. DFT software often used to compute electronic properties includes applications such as VASP, CRYSTAL, CASTEP or Quantum Espresso. Depending on the unit cell size and symmetry of the material, these calculations can be computationally expensive. In this study, we present a new machine learning platform for the accurate prediction of properties such as $E_g$ of a wide range of materials.

Keywords: Machine learning, Deep learning, Materials prediction, Band gap

Introduction
Opportunities to harness the continued pace of computer processing capabilities as well as new and refined data processing techniques exist for those wishing to investigate and predict material properties computationally.

Using a Machine Learning (ML), Deep Learning (DL), and High Throughput (HT) computing techniques can provide an efficient robust data processing platform for the prediction and discovery of new materials.

ML techniques involve processing large datasets in order to generate highly accurate modelling algorithms that can be used to find relationships within the data and predict outcomes.

HT computing techniques involve aggregating the results of computations that have already been executed from many disparate data sources. Quantum chemical calculations and atomic scale calculations are often time consuming and CPU expensive, requiring hundreds of hours of super-computer processing time. Using pre-calculated results from these operations will greatly reduce processing time, allowing for a greater throughput on much more modest hardware.

The combination of ML with HT will allow for rapid and exhaustive exploration of materials properties within a computational environment, at a scale and speed that simply cannot be matched in a laboratory.

In this paper we present a bespoke software platform (codename: Hadoken) for the discovery of materials, as well as 5 models derived from ML techniques that can be used to accurately predict material properties (such as the band gap of a compound), and a newly developed website that provides the basis for a materials prediction platform.

Deep learning
Data preparation for deep learning
A dataset containing information about $250 \times 10^3$ simulations calculated via the Perdew-Burke-Ernzerhof (PBE [1, 2]) DFT functional using the projector augmented wave (PAW [3, 4]) method was sourced via the Hadoken platform and downloaded for processing.

Feature composition
The stoichiometry $S$ value is a string which is split into its constituent parts (a form of one-hot encoding) and...
subsequently used to compose new features, comprising of the element and the count of the instance of that element. The one-hot encoding process involves decomposing categorical values into a binary representation.

\[
S(x_H, x_{He}, \ldots, x_{Og})
\rightarrow \{H = x_H, He = x_{He}, \ldots, Og = x_{Og}\}
\]

(1)

To encode $H_2O$:

\[
S(H_2O) \rightarrow S(H = 2, O = 1)
\rightarrow \{H = 1, O = 2, \ldots, Og = 0\}
\]

To encode copper indium selenide:

\[
S(CuInSe_2) \rightarrow S(Cu_1In_1Se_2)
\rightarrow \{H = 0, \ldots, Cu = 1, In = 1, Se = 2, \ldots, Og = 0\}
\]

Isomers have the same stoichiometric $S$ value, yet have differing physical structures ($C_3H_4$ for example). These isomers will produce identical encoding.

In this paper, the definition $S$ (1) refers to this equation.

The gap type $GT$ feature represents values that indicate the category (one-hot encoded) of gap type present in the compound.

\[
GT(x_{GT_{HM}}, x_{GT_{TD}}, \ldots, x_{GT_{M}}) \rightarrow
\{GT_{HM} = x_{GT_{HM}}, \ldots, GT_{M} = x_{GT_{M}}\} x \in 0, 1
\]

(2)

Table 1 details the possible gap type values with corresponding definitions.

* Given a band gap, this keyword describes if the system is a metal, a semi-metal, an insulator with direct or indirect band gap [5].

The geometry $G$ feature is decomposed using cell parameters (the unit cell’s lengths and angles) into 6 features:

\[
G \rightarrow \{a\vec{A}, b\vec{A}, c\vec{A}, \alpha^\circ, \beta^\circ, \gamma^\circ\}
\]

(3)

Space group $SG$ which defines one of the possible 230 symmetry groups of the crystal lattice is a categorical scalar that requires transformation into appropriate binary features (one-hot encoding). As an example, space group represents one of 230 possible categories with the use of a single integer: this scalar is transformed into 230 binary features:

\[
SG(x_{SG_1}, x_{SG_2}, \ldots, x_{SG_{230}}) \rightarrow
\{SG_1 = x_{SG_1}, SG_2 = x_{SG_2}, \ldots, SG_{230} = x_{SG_{230}}\} x \in 0, 1
\]

(4)

To encode the space group 37:

\[
SG(37) \rightarrow \{SG_1 = 0, \ldots, SG_{37} = 1, \ldots, SG_{230} = 0\}
\]

In the final stages of data preparation, constant features (features that contain the same value for each record) were dropped from the dataset, as well as any rows that contained null feature values. The dataset is now ready for use.

**Aggregated feature set**

Data obtained from all databases (AFLOW [6], Materials Project [7]) is normalised and aggregated into a single, functional form. This process results in the aggregation of maximum number of homogeneous features from consumed data sources. Table 2 details the features obtained from the AFLOW and Materials Project databases along with example values.

Table 3 details the attributes collected, along with names, example values, and original data source.

Table 2 details the feature set with names and example values.

**Additional feature set derivation**

Additional features useful for ML can be derived from existing features and also user input. Deriving these features frees the user from the necessity of performing these calculations and expedites workflow. In some instances, derivation of these additional features has been undertaken purely for experimental purposes, with the expectation that further refinement in the future will yield less theoretical results.

---

Table 1 Possible gap type values with definitions

| Value      | Definition        |
|------------|-------------------|
| NULL       | No definition     |
| HalfMetal  | *                 |
| InsulatorDirect | *             |
| InsulatorDirectSpinPolarised | *             |
| InsulatorIndirect | *             |
| InsulatorIndirectSpinPolarised | *             |
| Metal      | *                 |

---

| Name                  | Example          |
|-----------------------|------------------|
| Stoichiometry         | A1B2C2D2E2F2G2H2I2J2K2L2M2N2O2P2Q2R2S2T2U2V2W2X2Y2Z2          |
| Band Gap              | 4.8022           |
| Density               | 2.25761          |
| Energy                | −151.631         |
| Energy per Atom       | −7.22053         |
| Fermi energy          | 0.4748           |
| Geometry A, B, C      | 5.296, 5.296, 11.448 |
| Geometry $\alpha, \beta, \gamma$ | 90, 90, 120 |
| Space group           | 181              |
In this paper, the notation $S(1)$ refers to $S$ provided by the Definition 1.

**Number of atoms**
The total number of atoms $N$ contained within the system can be derived from $S(1)$ such:

$$N = \sum S(x_i)$$  \hspace{1cm} (5)

where $S(1)$ describes the stoichiometric composition of the material. This feature returns the sum of species of each atom contained in the unit cell multiplied by the instance.

**Atomic weight**
The total atomic weight $T_{Ar}$ of the system with reference to $S(1)$ is given by:

$$T_{Ar} = \sum A_{ri} \times S(x_i)$$ \hspace{1cm} (6)

where $S(1)$ describes the stoichiometric composition of the material and $A_{ri}$ [8] describes the atomic weight of each element. This feature returns the sum of each atomic weight of each species considered individually in the unit cell multiplied by the instance.

**Chemical potential**
The total chemical potential $T_{\mu}$ of the system with reference to $S(1)$ is given by:

$$T_{\mu} = \sum \mu_i \times S(x_i)$$ \hspace{1cm} (7)

where $S(1)$ describes the stoichiometric composition of the material and $\mu_i$ describes the chemical potential of each element. This feature returns the sum of each chemical potential of each species considered individually in the unit cell multiplied by the instance. This feature contains values generated by the software given a stoichiometry value. The chemical potential values are provided from the corresponding VASP POTCAR files.

**$S$, $P$, $D$, $F$ electrons**
The total count of the number of electrons $T_e$ in each type of sub shell within the compound is given by:

$$T_e = \sum e_i$$ \hspace{1cm} (8)

where $e_i$ describes the number of electrons present in the corresponding sub shell. Electron configuration is determined using values from the literature [9].
**S, P, D, F orbitals**
The total count of each type of orbital $T_\sigma$ within the compound is given by:

$$ T_\sigma = \sum \sigma_i $$

(9)

where $\sigma_i$ describes the corresponding number of orbitals present in the element. Orbital configuration is determined using values from the literature [9].

**Symmetry**
The symmetry elements $HS$ [10] associated with the space group of the crystal lattice has been stored in our database. This information is one-hot encoded in a similar fashion to SG (4).

$$ HS(x_{HS1}, x_{HS2}, \ldots, x_{HS63M}) \rightarrow \{ HS_1 = x_{HS1}, HS_2 = x_{HS2}, \ldots, HS_{63M} = x_{HS_{63M}} \} $$

(10)

**Electron affinity**
The total electron affinity $T_{EA}$ with reference to S (1) is given by:

$$ T_{EA} = \sum EA_i \times S(x_i) $$

(11)

where $S (1)$ describes the stoichiometric composition of the material and $EA_i$ describes the electron affinity [11] of each element. This feature returns the sum of each electron affinity of each species considered individually in the unit cell multiplied by the instance.

**Electronegativity**
The total electronegativity $\chi$ is given by the Mulliken electronegativity definition [12, 13]:

$$ \chi = \sum \frac{E_i + E_{ea}}{2} $$

(12)

where $E_i$ describes the first ionisation energy [14] and $E_{ea}$ describes the electron affinity [11].

**Ionisation energy**
The total ionisation energy $T_{IE}$ with reference to S (1) is given by:

$$ T_{IE} = \sum IE_i \times S(x_i) $$

(13)

where $S (1)$ describes the stoichiometric composition of the material and $IE_i$ describes the ionisation energy [14] of each element.

**Mass density**
The total mass density $T_\rho$ with reference to S (1) is given by:

$$ T_\rho = \sum \rho_i \times S(x_i) $$

(14)

where $S (1)$ describes the stoichiometric composition of the material and $\rho_i$ describes the density [15, 9] of each element. This feature returns the sum of each mass density value multiplied by the instance count of the corresponding element.

**Valence electrons**
The total number of valence electrons $T_{Ve}$ with reference to S (1) is given by:

$$ T_{Ve} = \sum Ve_i \times S(x_i) $$

(15)

where $S (1)$ describes the stoichiometric composition of the material and $Ve_i$ describes the number of valences electrons present for each element [15]. Currently, the number of valence electrons is determined primarily from the specification of the chemical elements in the VASP POTCAR file associated with the structure.

**Effective mass**
For a free electron, effective mass [16, 17] is given by

$$ E = \frac{\hbar^2 k^2}{2m_e} $$

(16)

For an electron in a crystal, the effective mass approximation is given by

$$ E' = \frac{\hbar^2 k^2}{2m'_e} $$

(17)

where $m'_e = xm_e$. Thus the dispersion may be rewritten as

$$ E' = \frac{\hbar^2 k^2}{2(x1)} = \frac{k^2}{2x} $$

(18)

Using the second derivative of (18) to calculate $x$

$$ \frac{d^2E'}{dk^2} = \frac{d}{dk} \left( \frac{dE'}{dk} \right) = \frac{d}{dk} \left( \frac{k}{x} \right) = \frac{1}{x} $$

(19)

Fitting a curve to the conduction band minima of an $E$-$k$ diagram using the form $y = ax^2 + bx + c$ yields
\[ E' = ak^2 + bk + c \]  
(20)

Then
\[ \frac{d^2E'}{dk^2} = 2a \]
(21)

And
\[ x = \left( \frac{d^2E'}{dk^2} \right)^{-1} = (2a)^{-1} \]
(22)

Thus our final equation for calculating effective mass (adjusting for atomic units) is given by:
\[ m^* = (2a)^{-1} \]  
(23)

The VASP software package can produce EIGENVAL files which contain the Kohn-Sham eigenvalues for all \( k \)-points. We have developed software to parse these files and produce the appropriate band structure diagrams, to which a parabola may be fitted. The EIGENVALS output usually appears in the following format:

```
44 44 1 1
0.1722398E+02 0.7561903E-09 0.7561903E-09 0.1382445E-08
0.0000000E+00 0.0000000E+00 0.3571429E-02
CAR
Ag1Cr4O14Tl3_ICSD_421926
364 280 294
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.3571429E-02
1 -43.9433
2 -43.9433
...
293 15.1138
294 15.2782
```

The following line in this file contains important information required during processing:

```
364 280 294
0.2553226E-01 0.2553226E-01 0.3244774E-02 0.3571429E-02
1 -43.9433
2 -43.9433
...
```

The values on this line are the number of electrons, number of \( k \)-points, and number of bands respectively. Lines that contain 4 double-values contain information regarding the 3-dimensional position in \( k \)-space \((x, y, z)\), as well as a weighting factor (not used by our software):

```
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.3571429E-02
```

These values are parsed into a vector and stored in memory. Immediately following the coordinate lines are lines containing energies associated at that coordinate:

```
1 -43.9433
2 -43.9433
...
293 15.1138
294 15.2782
```

Coordinate vectors represent direct coordinate values \((x, y, z)\) and are require further processing to be useful for \( m^* \) calculation.

\[ f(\alpha) = (2 \times \pi \times \alpha) \]
(24)

\[ x' = f(x) \]
\[ y' = f(y) \]
\[ z' = f(z) \]

The reciprocal lattice is a \( 3 \times 3 \) matrix defined as

\[ G_m = m_1 b_1 + m_2 b_2 + m_3 b_3 \]  
(25)

where the reciprocal primitive vectors are defined as

\[ b_1 = \alpha \hat{i}_1 + \beta \hat{j}_1 + \gamma \hat{k}_1 \]
\[ b_2 = \alpha \hat{i}_2 + \beta \hat{j}_2 + \gamma \hat{k}_2 \]
\[ b_3 = \alpha \hat{i}_3 + \beta \hat{j}_3 + \gamma \hat{k}_3 \]  
(26)

The reciprocal lattice (values sourced from the OUTCAR file, another VASP output file) is used to transform coordinate vectors \( \mathbf{v}_i = [x'_i \, y'_i \, z'_i]^T \) by the reciprocal lattice such:

\[ \mathbf{v}_i = [x'_i \, y'_i \, z'_i]^T \times G_m \]  
(27)

Finally, the distance between two 3-dimensional \( k \)-coordinate vectors \( \mathbf{v}_i \) and \( \mathbf{v}_j \) is described by:

\[ d = \sqrt{(\mathbf{v}_{ix} - \mathbf{v}_{ix})^2 + (\mathbf{v}_{iy} - \mathbf{v}_{iy})^2 + (\mathbf{v}_{iz} - \mathbf{v}_{iz})^2} \]  
(28)

This value is used as the \( k \)-value (converted from units of \( \text{Å}^{-1} \text{→} \mu_B^{-1} \)) along the \( x \)-axis in the following \( E-k \) diagram, with \( E \) (converted from units of \( eV \text{→} Ha \)) comprising the \( y \)-axis values. This process ostensibly provides...
energy values that correspond to the associated position in the Brillouin zone.

Figure 1 shows the band structure ($E$-$k$) diagram of $Si_2$ generated by the Hadoken software. Conduction bands are shaded green, with the lowest unoccupied molecular orbital (LUMO) shown as bold green. Valence bands are shaded blue, with the highest occupied molecular orbital (HOMO) shown as bold blue. An orange parabola has been fitted to the LUMO minimum in the $\Gamma$-$X$ segment, and it is this curve that is used to calculate effective mass. Also shown are red parabolae fitted to the HOMO maxima.

Fitting a parabola in the quadratic form \( y = ax^2 + bx + c \) yields the coefficient \( a \) which can then be used by (23) to obtain the final $m^*$ value.

Should more than one fit per $k$-space segment be possible, then the resultant values are averaged to yield the final effective mass value. Currently, only $m^*$ values calculated in the $\Gamma$-$X$ segment via this method are persisted.

The following Table 4 displays the entire feature set, including sourced and derived values, corresponding example values and units.

**Deep learning model training process**

All models were trained using the same process:

1. Features in the entire 0.477 GB dataset were normalised.
2. Data was split into two streams: training and validation at a ratio of 0.7/0.3.
3. An artificial neural network based on a sequential DL model from the Keras framework on a TensorFlow backend with dense layers (100, 50) was used and trained over 300 iterations.
4. Verification that over-fitting was not occurring. Over-fitting is characterised by an increase in loss which will be reflected in the training history. Even after 300 iterations, the loss recorded continues to converge, indicating that the algorithm is still learning (i.e., not over-fitting). Had the training process resulted in an increase in loss, we could be sure over-fitting was occurring.
5. The neural network was optimised by training it with all the data over 1000 iterations. The following Table 5 demonstrates that the optimisation process may yield extra accuracy when training a model for production deployment.
6. Information about the neural network was serialised for production use (layers, weights, biases, activation functions etc.).
7. Optimised models are persisted for future use via the https://www.hadokenmaterials.io/ website and associated API

The models described in this document have been made available for use at https://www.hadokenmaterials.io/ with the API documentation available at https://www.
hadok enmat erials. io/ Home/ Api. These models are also made available via GitHub at https://github.com/carlyman77/MaterialsDiscoveryML.

### Determination of model accuracy

We include three different loss functions used to determine accuracy for the predictive models, and a single loss function for the classification model. All metrics should be considered when evaluating the accuracy of a model, as each method has advantages in certain applications. For example, if the average errors are evenly distributed then both Mean Absolute Error and Root Mean Squared Error outputs should converge. However, Root Mean Squared Error will penalise large outlier errors as the errors are squared before an average is taken.

#### Mean absolute error (MAE)

This value is derived from the mean_absolute_error [19] function which produces a risk metric corresponding to the expected value of the absolute error loss or $l_1$-norm loss.

| Name                   | Example                | Units          | Data type | Aggregated | Calculated |
|------------------------|------------------------|----------------|-----------|------------|------------|
| Species                | CaCuGeO                | String         | Yes       | No         |            |
| Compound               | Ca2Cu2Ge4O12           | String         | Yes       | No         |            |
| Band gap               | 1.2007                 | eV             | Double    | Yes        | No         |
| Density                | 4.60489                | eV             | Double    | Yes        | No         |
| DFT type               | 1                      | Int32          | Yes       | No         |            |
| Energy                 | 121.07                 | eV             | Double    | Yes        | No         |
| Energy per Atom        | 6.05349                | eV             | Double    | Yes        | No         |
| Fermi energy           | 3.4726                 | eV             | Double    | Yes        | No         |
| Gap type               | InsulatorIndirect      | String         | Yes       | No         |            |
| Geometry A             | 6.949605               | Å              | Double    | Yes        | No         |
| Geometry B             | 6.949605               | Å              | Double    | Yes        | No         |
| Geometry C             | 5.44499                | Å              | Double    | Yes        | No         |
| Geometry alpha         | 76.82593               | Degrees        | Double    | Yes        | No         |
| Geometry beta          | 76.82593               | Degrees        | Double    | Yes        | No         |
| Geometry gamma         | 83.10932               | Degrees        | Double    | Yes        | No         |
| Kْ-Space               | Γ-Y-F-L-Z...-N-Z-F₁    | String         | Yes       | No         |            |
| Number of atoms        | 20                     | Int32          | Yes       | No         |            |
| Space group            | 15                     | Int32          | Yes       | No         |            |
| Volume                 | 248.674                | Å$^3$ or Bohr$^3$[18] | Double | Yes | No |
| Effective mass         | 0 or NULL              | Double         | No        | Yes        |            |
| Total atomic weight    | 689.756                | Double         | No        | Yes        |            |
| Total chemical potential| 8390.4896              | Double         | No        | Yes        |            |
| Total electron affinity| 24.9832632             | Double         | No        | Yes        |            |
| Total electron negativity| 6.19191658            | kJ/mol         | Double    | No        | Yes        |
| Total ionisation energy| 222.6934               | eV             | Double    | No        | Yes        |
| Total density          | 42.309148              | eV             | Double    | No        | Yes        |
| Total number of S Orbitals| 56                  | Int32          | No        | Yes        |            |
| Total number of P Orbitals| 96                  | Int32          | No        | Yes        |            |
| Total number of D Orbitals| 30                  | Int32          | No        | Yes        |            |
| Total number of F Orbitals| 0                   | Int32          | No        | Yes        |            |
| Total number of electrons| 322                 | Int32          | No        | Yes        |            |
| Total number of S electrons| 110                | Int32          | No        | Yes        |            |
| Total number of P electrons| 152                | Int32          | No        | Yes        |            |
| Total number of D electrons| 60                  | Int32          | No        | Yes        |            |
| Total number of F electrons| 0                   | Int32          | No        | Yes        |            |
| Valence electrons      | 94                     | Int32          | No        | Yes        |            |

| Name                   | Example                | Units          | Data type | Aggregated | Calculated |
|------------------------|------------------------|----------------|-----------|------------|------------|
| Species                | CaCuGeO                | String         | Yes       | No         |            |
| Compound               | Ca2Cu2Ge4O12           | String         | Yes       | No         |            |
| Band gap               | 1.2007                 | eV             | Double    | Yes        | No         |
| Density                | 4.60489                | eV             | Double    | Yes        | No         |
| DFT type               | 1                      | Int32          | Yes       | No         |            |
| Energy                 | 121.07                 | eV             | Double    | Yes        | No         |
| Energy per Atom        | 6.05349                | eV             | Double    | Yes        | No         |
| Fermi energy           | 3.4726                 | eV             | Double    | Yes        | No         |
| Gap type               | InsulatorIndirect      | String         | Yes       | No         |            |
| Geometry A             | 6.949605               | Å              | Double    | Yes        | No         |
| Geometry B             | 6.949605               | Å              | Double    | Yes        | No         |
| Geometry C             | 5.44499                | Å              | Double    | Yes        | No         |
| Geometry alpha         | 76.82593               | Degrees        | Double    | Yes        | No         |
| Geometry beta          | 76.82593               | Degrees        | Double    | Yes        | No         |
| Geometry gamma         | 83.10932               | Degrees        | Double    | Yes        | No         |
| Kْ-Space               | Γ-Y-F-L-Z...-N-Z-F₁    | String         | Yes       | No         |            |
| Number of atoms        | 20                     | Int32          | Yes       | No         |            |
| Space group            | 15                     | Int32          | Yes       | No         |            |
| Volume                 | 248.674                | Å$^3$ or Bohr$^3$[18] | Double | Yes | No |
| Effective mass         | 0 or NULL              | Double         | No        | Yes        |            |
| Total atomic weight    | 689.756                | Double         | No        | Yes        |            |
| Total chemical potential| 8390.4896              | Double         | No        | Yes        |            |
| Total electron affinity| 24.9832632             | Double         | No        | Yes        |            |
| Total electron negativity| 6.19191658            | kJ/mol         | Double    | No        | Yes        |
| Total ionisation energy| 222.6934               | eV             | Double    | No        | Yes        |
| Total density          | 42.309148              | eV             | Double    | No        | Yes        |
| Total number of S Orbitals| 56                  | Int32          | No        | Yes        |            |
| Total number of P Orbitals| 96                  | Int32          | No        | Yes        |            |
| Total number of D Orbitals| 30                  | Int32          | No        | Yes        |            |
| Total number of F Orbitals| 0                   | Int32          | No        | Yes        |            |
| Total number of electrons| 322                 | Int32          | No        | Yes        |            |
| Total number of S electrons| 110                | Int32          | No        | Yes        |            |
| Total number of P electrons| 152                | Int32          | No        | Yes        |            |
| Total number of D electrons| 60                  | Int32          | No        | Yes        |            |
| Total number of F electrons| 0                   | Int32          | No        | Yes        |            |
| Valence electrons      | 94                     | Int32          | No        | Yes        |            |
Table 5 Comparison of unoptimised ML and optimised Models

| Model          | State       | MAE      | RMSE     | R²        | 99%       |
|----------------|-------------|----------|----------|-----------|-----------|
| Band Gap       | Unoptimised | 0.079572 | 0.297179 | 0.914471  | 3.044744  |
| Band Gap       | Optimised   | 0.057742 | 0.214150 | 0.955388  | 2.315024  |
| Band Gap       | Unoptimised | 0.067828 | 0.300045 | 0.912595  | 2.321237  |
| Band Gap       | Optimised   | 0.049818 | 0.162154 | 0.974421  | 1.590898  |
| Band Gap       | Unoptimised | 0.045086 | 0.162154 | 0.974421  | 1.590898  |
| Band Gap       | Optimised   | 0.046388 | 0.175463 | 0.970050  | 2.175463  |
| Fermi energy   | Unoptimised | 0.249163 | 0.379878 | 0.975808  | 2.765868  |
| Fermi energy   | Optimised   | 0.308781 | 0.392329 | 0.974224  | 2.141287  |

\[ R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \] (32)

where \( \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \) and \( \sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} \epsilon_i^2 \).

Modelling and results

Overview

Models are produced by the ML training process, and contain the refined weights, biases and activation functions required to operate independently of the original dataset. Models are software assets that can be used to perform complex algorithmic tasks such as prediction or classification.

Band gap

Band gap \( E_g \) is an energy range between the uppermost valence band (valence band maximum) and the lowest conduction band (conduction band minimum) of a crystal. Electrons in the valence bands can transition into the conduction bands upon excitation. This size of the band gap is a critical feature that many of the material's possible applications.

Photovoltaic (PV) materials are semiconductors, and so it follows that \( E_g \) is a key metric when considering a material's suitability for PV applications.

Deep learning to predict band gap (Single feature)

This model attempts to predict \( E_g \) from stoichiometry only. This model uses a single feature, stoichiometry \( S \), such:

\( E_g(S) = M(S) \)

where \( E_g(S) \) describes the predicted result computed by \( M \) from \( S \).

Results

Figure 2 displays the predicted \( E_g \) values generated by the model with the original \( E_g \) values. A clear linear trend is evident.

Figure 3 displays errors in 0.1 eV buckets. The majority of predicted results appear in the first negative bucket, indicating that for most predictions, the resultant value is no more than 0.1 eV different from the original value.

Figure 4 displays the loss values generated by during the model training process.

Table 6 details the overall predictive accuracy metrics for the model.

Deep learning to predict band gap (minimal features)

This model attempts to predict \( E_g \) from the fewest features considered logical that are also easily sourced, i.e.,
they can be found in literature and/or do not require complex computation to derive. This model uses the feature geometry $G$ which is decomposed into cell parameters (the unit cell's lengths and angles).

This model uses 3 main features, stoichiometry $S$ (1), geometry $G$ (3), and space group $SG$ (4), such:

$$E_g(S, G, SG) = M(S, G, SG)$$

where $E_g(S)$ describes the predicted result computed by $M$ from $S$ (1), $G$ (3), and $SG$ (4).
Results

Figure 5 displays the predicted $E_g$ values generated by the model with the original $E_g$ values. A clear linear trend is evident, and the spread of data points from this trend is much less than the previous model.

Figure 6 displays errors in 0.1 eV buckets. The majority of predicted results appear in the first negative bucket,

Table 6 Single feature model performance metrics

| Name                    | Value    |
|-------------------------|----------|
| Mean absolute error     | 0.079572 |
| Root mean squared error | 0.297179 |
| $R^2$                   | 0.914471 |
| 99% Quantile error      | 3.044744 |

Fig. 4 $E_g$ Model Training History - Single Feature

Fig. 5 Simulated vs. predicted $E_g$ - minimal features
indicating that for most predictions, the resultant value is no more than 0.1 eV different from the original value.

Figure 7 displays the loss values generated by during the model training process.

Table 7 details the overall predictive accuracy metrics for the model.

| Name                           | Value    |
|--------------------------------|----------|
| Mean absolute error            | 0.072204 |
| Root mean squared error        | 0.300485 |
| $R^2$                          | 0.912558 |
| 99% Quantile error             | 3.212371 |
Deep learning to predict band gap (maximal features)

This model attempts to predict $E_g$ from the maximum number of features available from the collated dataset. This model is described as such:

$$E_g(F_{ALL}) = M(F_{ALL})$$

where $E_g(F_{ALL})$ describes the predicted result computed by $M$ from $F_{ALL}$, and $F_{ALL}$ describes all features in the dataset.

Results

Figure 8 displays the predicted $E_g$ values generated by the model with the original $E_g$ values. A clear linear trend is

![Fig. 8 Simulated vs. predicted $E_g$ - maximal features](image1)

![Fig. 9 $E_g$ Model residuals-maximal features](image2)
evident, with the spread of data points from this trend much similar to the previous model.

Figure 9 displays errors in 0.1 eV buckets. As with the previous model, the majority of predicted results appear in the first negative bucket, indicating that for most predictions, the resultant value is no more than 0.1 eV different from the original value.

Table 8 details the overall predictive accuracy metrics for the model.

| Comparison among deep learning models |
|---------------------------------------|
| Table 9 summarises the predictive accuracy metrics for each model. All 3 models are extremely accurate, and of note is the diminishing returns realised by the addition of many extra features: the model using a single feature is almost as accurate as the model that uses 20 features. |

**Fermi energy**

Fermi energy is also an attribute useful for the design and discovery of materials, however some online data sources do not store this value. We provide a model for the prediction of this property.

**Deep learning to predict fermi energy**

This model attempts to predict $E_F$ from the fewest features. This model uses 2 main features, stoichiometry (one-hot encoded) $S$ (1), and geometry $G$ (3). This model is described as:

$$E_F(S, G) = M(S, G)$$

**Table 8** Maximal feature model performance metrics

| Name               | Value     |
|--------------------|-----------|
| Mean absolute error| 0.082444  |
| Root mean squared error | 0.311686 |
| $R^2$              | 0.905917  |
| 99% Quantile error | 3.232445  |

**Table 9** Comparison metrics with feature and encoded feature count

| Name  | $F$ | $F_E$ | MAE    | RMSE  | $R^2$  | 99%   |
|-------|-----|-------|--------|-------|--------|-------|
| Single| 1   | 100   | 0.079572| 0.297179| 0.914471| 3.044744|
| Minimal| 8   | 311   | 0.072204| 0.300485| 0.912558| 3.212371|
| Maximal| 20  | 348   | 0.082444| 0.311686| 0.905917| 3.232445|
Results

Figure 11 displays the predicted $E_F$ values generated by the model with the original $E_F$ values. A clear linear trend is evident, with most data points clustered on or around this trend.

Figure 12 displays errors in 0.1 eV buckets. This model is accurate to within 0.5 eV for the majority of predicted values.

Figure 13 displays the loss values generated by during the model training process.
Table 10 details the overall predictive accuracy metrics for the model.

**Gap type**
Gap type is an important attribute used to classify the type of band gap present in a material. Typically the gap type relates directly to the usefulness of a material for a specific application. For example, metals have no band gap and such make excellent conductors, whilst semiconductors may have a direct or indirect band gap (an indirect band gap is characterised by the phonon-assisted transmission). Insulators typically have a very large band gap.

**Deep learning to classify gap type**
This model uses 2 main features, stoichiometry (one-hot encoded) $S (1)$, and space group $SG (4)$, that are encoded (or decomposed) into values of varying size. This model is described as:

$$\text{GapType}(S, SG) = M(S, SG)$$

**Results**
Figure 14 displays the accuracy of gap type predictions per gap type. This model is most useful at predicting whether a gap type is an direct insulator or a metal.

Figure 15 displays the loss values generated during the model training process.

Table 11 details the overall predictive accuracy metrics for the model.

Table 12 details the metrics for each class of the model.

**Production deployment of machine learning models**
In addition to development of the preceding models, we have developed a lightweight and efficient method for deploying models to a production environment.

Multiple files are produced by Keras when persisting a model, namely the weights and structure of the network. The weights are stored in the HDF5 format, neither of which are suitable for a number of reasons: JSON offers no schema support, or mature query language, comments, or meta-data. JSON is also a terse format designed to be used when the contract is pre-agreed upon, and therefore does not make a good candidate to support rich, searchable data models. The HDF5 format is not human readable and is not easily parsed.
Unifying these two files in a more appropriate format is a welcome improvement.

In addition to this, no information is saved with the model about how it is intended to be used. For example,
inputs are not labelled, and no normalising parameters are included, which renders the model not portable and useless for production consumption. To address this, we have developed a simple, portable XML format that is searchable and can be validated against a schema. Only a single file is required to instantiate a usable model in a production environment that is guaranteed to produce reliable results from minimal code.

Artificial neural network function
Provisioning of ML models from the XML definition is provided via the Hadoken.ML.NeuralNetwork type located in the Hadoken.ML assembly. This custom-built Artificial Neural Network (ANN) functions as a series of completely connected layers using the following method:

1. Inputs are multiplied by weights and forwarded to the nodes in each layer
2. Each node introduces a bias and another weight and sends the value to the next layer via the activation function

Figure 16 displays the map of a typical neural network. Inputs are fully connected with the first hidden layer, which is in turn fully connected to each following layer. This process is completed for each hidden layer, with results forwarded to the output layer.

Figure 17 displays the map of a neural network node. Inputs are multiplied by a weight and then added to a bias value. The sum of these operations is forwarded to an activation function which determines the final output value.

| Class                  | Accuracy       | Recall         | Precision       | F-Score        |
|------------------------|----------------|----------------|-----------------|----------------|
| Half metal             | 0.981922       | 0.588644       | 0.336201        | 0.427970       |
| Insulator direct       | 0.967027       | 0.659524       | 0.567789        | 0.610228       |
| Insulator direct, Spin Polarised | 0.9967527 | 0.500000       | 0.459016        | 0.478632       |
| Insulator indirect     | 0.957695       | 0.717451       | 0.678095        | 0.697218       |
| Insulator Indirect, Spin Polarised | 0.992452 | 0.498920       | 0.408127        | 0.448980       |
| Metal                  | 0.960757       | 0.965741       | 0.989014        | 0.977239       |
Supported activation functions

Table 13 details the activation functions provided by the software.

| Name               | Form          |
|--------------------|---------------|
| Hyperbolic tangent | $f(x) = \tanh x$ |
| Rectified linear unit | $f(x) = \max(0, x)$ |
| Sigmoid            | $f(x) = \frac{1}{1 + e^{-x}}$ |
| Softmax            | $f(x) = \frac{x^e}{\sum e^x}$ |

Supported normalisers

Table 14 details the normalisers provided by the software.

| Name     | Form                      |
|----------|---------------------------|
| Mean     | $y = \frac{x - \text{mean}(x)}{\text{max}(x) - \text{min}(x)}$ |
| Min/Max  | $y = \frac{x - \text{min}(x)}{\text{max}(x) - \text{min}(x)}$ |

Software platform

Architecture

A bespoke software platform (codename: Hadoken) was created for the express purpose of aggregating materials data from disparate representational state transfer (REST) APIs such as Materials Project [7] and AFLOW [6]. Data from these sources is collected via an aggregator and stored in a relational database. Additional supporting files that may be of use (such as associated VASP [23] files) are also downloaded and stored for later use. Useful attributes such as Fermi energy $E_F$ that are not present in REST API data are sourced from the VASP files and added to the database. Curated data is then used for the purposes of training ML models for predictive tasks.

Technology stack

The technology stack mirrors current popular industry standard for rapid application development (RAD), and is based on Microsoft’s .NET Core Framework [24] and Microsoft SQL Server 2017 [25]. ML technologies include Python 3.5 [26] and TensorFlow [27] as well as Azure ML Studio [28].

Data collection

Data are initially sourced from two streams, on-line and off-line. On-line data sources are actively maintained network resources which release edits in real (or near to
real) time. These often take the form of REST web API offerings including Materials Project and AFLOW. Some of these web services include information gathered from other sources, such as the Inorganic Crystal Structure Database (ICSD) [29].

These RESTful web services provide an industry standard method for querying and retrieving data. Data is provided in JSON format, which is then parsed into a common object model and stored locally. On-line data sources are much easier to work with than off-line, as they provide instant access to data stores that are pro-actively curated.

Off-line data sources may not be actively maintained, or may only release edits periodically (such as with a new publication) and typically include information contained within texts, files, or databases which may have been produced by a lab during the research process. It is most likely that each off-line source differs in its storage format or layout, especially in the case of textural publications, and thus must have a bespoke parser written for it. This process is very time consuming and so these sources are currently avoided.

Data curation and post processing

Data is collected currently on an ad-hoc basis, however when a new model is to be trained a snapshot of the database is taken so that continual data collection may occur. These snapshots are completely disconnected from the original data source, thus any updates to the database are not reflected in any dataset used by the model training process.

Post processing is the first step in data curation, and involves processing values and schema structure to assist with preparing data for curation. As an example, the AFLOW schema is mostly flat, however the Materials Project schema is nested. The Hadoken software prepares nested data by moving it to a simple normalised schema ready for the curation and matching process.

Data curation is a process that involves the careful selection and combination of data sources. Data sources may have differing, non-identical schemas applied to them, which will affect the storage and representation of underlying data. During the process, attributes from disparate data sources are matched where possible. Decisions must also be made about the treatment of nullable attributes. For example, it may be possible to replace null values with a default initialisation value, such as 0 for a null integer or an empty string for a null string. These decisions are realised in code, and applied in the software and underlying database schema.

All data collected must be curated, and this process involved dividing the data into two streams: high-quality and low-quality. Data must attain a completeness factor of 100% in order to be useful, so efforts are made to achieve this.

The completeness factor $F_C$ is the ratio of features that contain non-null values $F_{NN}$ to the total number of features $F_{Tot}$ in the dataset:

$$F_C = \frac{F_{NN}}{F_{Tot}}$$

where $F_{NN}$ defines the number of features with non-null values and $F_{Tot}$ defines the total number of features present.

Data is considered high-quality if its $F_C > 0.9$, with the additional constraint that any missing attributes can be retro-fitted by reading them from associated files or calculating them directly.

Data is considered low-quality if its $F_C \leq 0.9$. Records that contain missing attributes cannot be used by model training as they may mislead the model. Low-quality data is stored, but shelved for use later, as it may be possible to reconstruct missing attributes via ML, or, the data may be updated when matched with a future high-quality data source.

API access

We present a lightweight REST API for accessing the machine learning models built from this curated data. The API is built on current industry standards supporting both JSON and XML data exchange formats. The full API definition is located on the Hadoken Materials website here: https://hadokenmaterials.io/Home/Api. Registration is required to use the API (https://hadokenmaterials.io/Account/SignUp) and is fast (and free), however registration is not required to use the web UI interfaces provided for each model.

Whilst use of the website is free, any use of the website or API for research purposes, commercial or otherwise, are governed by terms defined in the citing document available on the website. More information is available here https://hadokenmaterials.io/Home/Citing.

Upon completion of registration, an API key in the form of an 128-bit GUID is allocated and API access is granted to the entire platform. This API key must be presented during each request.

By way of example, a typical API request for a band gap prediction for the compound $Ca_2Cu_2Ge_4O_{12}$ follows:
The response from this request (some headers omitted for brevity):

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8
X-Powered-By: ASP.NET

{
  "bandGap": 1.3985653904114555472784324321,
  "stoichiometry": "Ca2Cu2Ge4O12"
}
```

### API reference

Currently, the API supports a single version: 1. In the future, different versions will become available; to use those versions replace the current version number. Table 15 details the entire API URI reference. Table 16 details all optional query string parameters used by the API.

#### Table 16 Optional query string parameters

| Parameter | Default | Minimum | Maximum |
|-----------|---------|---------|---------|
| Size      | 100     | 1       | 200     |
| Start     | 1       | 1       | 2147483647 |

#### Machine learning API URI reference

##### Band gap-single feature

URL format: /api/v{Version}/MachineLearning/BandGap/Single

JSON fragment template:

```
{
  "Stoichiometry": "{Stoichiometry}"
}
```

JSON fragment example:

```
{
  "bandGap": 1.3985653904114555472784324321,
  "stoichiometry": "Ca2Cu2Ge4O12"
}
```

##### Band gap-minimal features

URL format: /api/v{Version}/MachineLearning/BandGap/SpaceGroup Geometry

JSON fragment template:

```
{
  "bandGap": 1.2049858165280045952682033686,
  "stoichiometry": "Ca2Cu2Ge4O12"
}
```

### Table 15 Full URI reference-https://www.hadokenmaterials.io/Home/Api

| Format | Method | Description |
|--------|--------|-------------|
| /api/v{Version}/Species | GET | Retrieve a list of resources |
| /api/v{Version}/Species/{Name} | GET | Retrieve a list of resources by name |
| /api/v{Version}/Species?Start={Start}&Size={Size} | GET | Retrieve a list of resources constrained by arguments |
| /api/v{Version}/Species/{GUID} | GET | Retrieve a single resource by unique identifier |
| /api/v{Version}/Compounds | GET | Retrieve a list of resources |
| /api/v{Version}/Compounds/{Name} | GET | Retrieve a list of resources by stoichiometry (Fe2In1P1) |
| /api/v{Version}/Compounds?Start={Start}&Size={Size} | GET | Retrieve a list of resources constrained by arguments |
| /api/v{Version}/Compounds/{GUID} | GET | Retrieve a single resource by unique identifier |
| /api/v{Version}/Simulations | GET | Retrieve a list of resources |
| /api/v{Version}/Simulations/{Name} | GET | Retrieve a list of resources by stoichiometry (Cu1In1Se2) |
| /api/v{Version}/Simulations?Start={Start}&Size={Size} | GET | Retrieve a list of resources constrained by arguments |
| /api/v{Version}/Simulations/Simulation/{GUID} | GET | Retrieve a single resource by unique identifier |
| /api/v{Version}/MachineLearning/BandGap/Single | POST | Compute a prediction from the posted data |
| /api/v{Version}/MachineLearning/BandGap/SpaceGroup Geometry | POST | Compute a prediction from the posted data |
| /api/v{Version}/MachineLearning/BandGap/SpaceGroupHighSymmetryDerived | POST | Compute a prediction from the posted data |
| /api/v{Version}/MachineLearning/FermiEnergy/Geometry | POST | Compute a prediction from the posted data |
| /api/v{Version}/MachineLearning/GapType/SpaceGroup | POST | Compute a prediction from the posted data |
{  
  "Stoichiometry": "Ca2Cu2Ge4O12"
}

**JSON fragment example:**

```
{
  "BandGap": 1.072278625397415602174867483,
  "GeometryA": 6.955802,
  "GeometryAlpha": 76.73364,
  "GeometryB": 6.955802,
  "GeometryBeta": 76.73364,
  "GeometryC": 5.44479,
  "GeometryGamma": 83.12188,
  "SpaceGroup": 15,
  "Stoichiometry": "Ca2Cu2Ge4O12"
}
```

**Band gap-maximal features**

URL format: /api/vVersion/MachineLearning/BandGap/

**SpaceGroup HighSymmetryDerived**

JSON fragment template:

```
{
  "SpaceGroup": "SpaceGroup",
  "Stoichiometry": "Stoichiometry"
}
```

**JSON fragment example:**

```
{
  "FermiEnergy": 3.54172624835756478074564780714516288,
  "GeometryA": 7.642811,
  "GeometryAlpha": 59.99344,
  "GeometryB": 7.643063,
  "GeometryBeta": 59.99962,
  "GeometryC": 7.643013,
  "GeometryGamma": 60.00009,
  "Stoichiometry": "Cr4Cu1In1Se8"
}
```

**Fermi energy**

URL format: /api/vVersion/MachineLearning/FermiEnergy/

**Geometry**

JSON fragment template:

```
{
  "FermiEnergy": FermiEnergy,
  "GeometryA": GeometryA,
  "GeometryAlpha": GeometryAlpha,
  "GeometryB": GeometryB,
  "GeometryBeta": GeometryBeta,
  "GeometryC": GeometryC,
  "GeometryGamma": GeometryGamma,
  "Stoichiometry": Stoichiometry
}
```

**JSON fragment example:**

```
{
  "HalfMetal": 0.0000000120987131097922044,
  "InsulatorDirectSpinPolarised": 0.000000000000000033320132,
  "InsulatorDirect": 0.027628844767050804013304393,
  "InsulatorIndirect": 0.97054870642805664504263889,
  "InsulatorIndirectSpinPolarised": 0.000000000000000011510869471,
  "Metal": 0.0003162447594771147566699205,
  "SpaceGroup": 129,
  "Stoichiometry": "Ag10O8P2Te2"
}
```

**Gap type**

URL format: /api/vVersion/MachineLearning/GapType/

**SpaceGroup**

JSON fragment template:

```
{
  "SpaceGroup": SpaceGroup,
  "Stoichiometry": Stoichiometry
}
```

**JSON fragment example:**

```
{
  "HalfMetal": 0.0000000000120987131097922044,
  "InsulatorDirectSpinPolarised": 0.000000000000000033320132,
  "InsulatorDirect": 0.027628844767050804013304393,
  "InsulatorIndirect": 0.97054870642805664504263889,
  "InsulatorIndirectSpinPolarised": 0.000000000000000011510869471,
  "Metal": 0.0003162447594771147566699205,
  "SpaceGroup": 129,
  "Stoichiometry": "Ag10O8P2Te2"
}
```

**Machine learning web URI reference**

Visit the URLs listed below to use the corresponding ML model via a web UI.
Conclusions
In this paper we show that it is possible to develop a number of highly accurate ML models to inexpensively predict the properties of materials using information previously generated from computationally expensive simulations.

The ML models demonstrate that a stoichiometry definition alone is a high value feature, containing (in most cases) all the information required to accurately compute the band gap associated with that material. Initial experimenting has demonstrated that the addition of other features (such as Density or Total Atomic Weight) has yielded little, if any additional accuracy. This suggests that DFT computation may not be required to perform this type of calculation.

The prospect of fast, efficient DFT-free computation of materials properties using only consumer hardware is tantalising and implies that further investigation into properties implied by stoichiometry related to $E_g$ is required. This development could in turn greatly reduce the amount of time spent on simulations, managing simulation software, and budgets spent on supercomputing.

This project also lays the foundation for expansion to the prediction of other materials properties in the future using a similar process, and the development of an industry standard platform for the production development of said models should facilitate the exhaustive profiling of compounds to develop novel materials by the wider research community in general.

Abbreviations
ANN: Artificial Neural Network; API: Application Programming Interface; ARC: Australian Research Council; DFT: Density Functional Theory; DL: Deep Learning; HOMO: Highest Occupied Molecular Orbital; HT: High Throughput; JSON: JavaScript Object Notation; LUMO: Lowest Unoccupied Molecular Orbital; MAE: Mean Absolute Error; ML: Machine Learning; PBE: Perdew-Burke-Ernzerhof; RAD: Rapid Application Development; REST: Representational State Transfer; RMSE: Root Mean Squared Error; SQL: Structured Query Language; UI: User Interface; URL: Uniform Resource Locator; VASP: Vienna Ab initio Simulation Package; XML: Extensible Markup Language.

Acknowledgements
The authors are grateful to the Editor and the anonymous reviewers for their invaluable comments and suggestions that greatly improved the contribution and overall quality of this manuscript. The authors would also like to thank Dr. Sherif Abbas at RMIT for several helpful discussions.

Authors’ contributions
All authors read and approved the final manuscript.

Funding
This work was supported by the Australian Government through the Australian Research Council (ARC) under the Centre of Excellence scheme (project number CE170100026). It was also supported by computational resources provided by the Australian Government through the National Computational Infrastructure National Facility and the Pawsey Supercomputer Centre.

Availability of data and materials
Hadoken API: https://www.hadokenmaterials.io/. Additional resources: https://github.com/carlyman77/MaterialsDiscoveryML

Declarations
Ethics approval and consent to participate
Not applicable.

Consent for publication
Consent is granted for publication.

Competing interests
Not applicable.

Author details
1ARC Centre of Excellence in Exciton Science, RMIT University, Melbourne 3000, Australia. 2School of Science, RMIT University Australia, 124 La Trobe Street, 3000 Melbourne, Australia.

Received: 1 December 2020 Accepted: 10 May 2021
Published online: 27 May 2021

References
1. Perdew JP, Burke K, Ernzerhof M (1996) Phys. Rev. Lett. 77:3865
2. Perdew JP, Burke K, Ernzerhof M (1997) Phys. Rev. Lett. 78:1396
3. Blöchl PE (1994) Phys. Rev. B 50:17953
4. Kresse G, Joubert D (1999) Phys. Rev. B 59:1758
5. Taylor Richard H et al (2014) A RESTful API for exchanging materials data in the AFLLOWLIB.org consortium. Comput Mater Sci 93:185
6. Taylor Richard H et al (2014) A RESTful API for exchanging materials data in the AFLLOWLIB.org consortium. Comput Mater Sci 93:178–192
7. Jain A, Ong* SP, Hautier G, Chen W, Richards WD, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G, Persson KA (*=equal contributions), (2013). The Materials Project: A materials genome approach to accelerating materials innovation. APL Materials, 1(1):011002
8. Lide David R (2004). CRC Handbook of Chemistry and Physics 86th Edition, 1-7, 1-8
9. Lide David R (2004) CRC Handbook of Chemistry and Physics 86th Edition, 13-14
10. Bradley CJ, Cracknell Arthur P (1972) The mathematical theory of symmetry in solids; representation theory for point groups and space groups
11. Lide David R (2004) CRC Handbook of Chemistry and Physics 86th Edition, 10-156, 10-157
12. Mulliken RS (1934) A new electroaffinity scale; together with data on valence states and on valence ionization potentials and electron affinities. J Chem Phys 2(11):782–793
13. Mulliken RS (1935) Electronic structures of molecules XI. electroaffinity, molecular orbitals and dipole moments. J Chem Phys 3(9):573–585
14. Lide David R (2004) CRC Handbook of Chemistry and Physics 86th Edition, 1-13, 1-14
15. Lide David R (2004) CRC Handbook of Chemistry and Physics 86th Edition, 4-1, 4-43
16. Simon Steven H (2017) The Oxford Solid State Basics 103–104
17. John Singleton, Band Theory and Electronic Properties of Solids, 42-44, (2014)
18. Taylor Richard H et al (2014) A RESTful API for exchanging materials data in the AFLOWLDB.org consortium. Comput Mater Sci 93:188
19. 3.3. Metrics and scoring: quantifying the quality of predictions - scikit-learn 0.23.1 documentation, https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error, viewed 2020
20. 3.3. Metrics and scoring: quantifying the quality of predictions - scikit-learn 0.23.1 documentation, https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error, viewed 2020
21. 3.3. Metrics and scoring: quantifying the quality of predictions - scikit-learn 0.23.1 documentation, https://scikit-learn.org/stable/modules/model_evaluation.html#2-score-the-coefficient-of-determination, viewed 2020
22. The HDF5 Library & File Format - The HDF Group. https://www.hdfgroup.org/solutions/hdf5, viewed 2020
23. VASP - Vienna Ab initio Simulation Package, https://www.vasp.at/, viewed 2020
24. NET | Free. Cross-platform. Open Source., https://dotnet.microsoft.com/, viewed 2020
25. SQL Server 2017 on Windows and Linux | Microsoft, https://www.microsoft.com/en-us/sql-server/sql-server-2017, viewed 2020
26. Welcome to Python.org, https://www.python.org/, viewed 2020
27. Tensorflow, https://www.tensorflow.org/, viewed 2020
28. Microsoft Azure Machine Learning Studio (classic), https://studio.azureml.net/, viewed 2020
29. Inorganic Crystal Structure Database (ICSD) | Physical Sciences Data science Service, https://www.psisds.ac.uk/icsd, viewed 2020

**Publisher’s Note**
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.