Automatic steel labeling on certain microstructural constituents with image processing and machine learning tools

Dmitry S. Bulgarevich, Susumu Tsukamoto, Tadashi Kasuya, Masahiko Demura and Makoto Watanabe

Research and Services Division of Materials Data and Integrated System, National Institute for Materials Science, Tsukuba, Japan; School of Engineering, The University of Tokyo, Tokyo, Japan

ABSTRACT
It is demonstrated that optical microscopy images of steel materials could be effectively categorized into classes on preset ferrite/pearlite-, ferrite/pearlite/bainite-, and bainite/martensite-type microstructures with image pre-processing and statistical analysis including the machine learning techniques. Though several popular classifiers were able to get the reasonable class-labeling accuracy, the random forest was virtually the best choice in terms of overall performance and usability. The present categorizing classifier could assist in choosing the appropriate pattern recognition method from our library for various steel microstructures, which we have recently reported. That is, the combination of the categorizing and pattern-recognizing methods provides a total solution for automatic quantification of a wide range of steel microstructures.

ARTICLE HISTORY
Received 26 February 2019
Revised 12 April 2019
Accepted 19 April 2019

KEYWORDS
Metallurgy; machine learning; microstructures; optical microscopy; pattern recognition

CLASSIFICATION
10 Engineering and structural materials; 106 Metallic materials; 404 Materials informatics / Genomics; 505 Optical / Molecular spectroscopy

1. Introduction
Microstructure phase analysis is one of the primary interests in metallurgy because formed microstructures significantly determine the material properties [1–3]. Most of the data for such studies are coming from optical- or electron microscopy imaging techniques. By using modern industrial optical microscopes with advanced automated imaging equipment, scanning stages, and even sample slicing for three-dimensional imaging, the amount of image data...
become overwhelming for manual examinations, and, in some cases, the relevant information could be hidden even for an expert. In this respect, there are big hopes that machine learning (ML) techniques could assist in automatization of routine tasks on big image datasets and that information gains could even unveil the new material paradigms. For metallurgy, the progress in such a direction has the paramount importance. So far, however, the reported image analyses with ML on metallic materials were extremely limited and, in most cases, were not compared/matched to the standard analytical methods in the metallurgical industry [4]. This situation is in strong contrast with an explosive number of scientific publications on ML applications to material science problems by using the physicochemical and structural property datasets/databases [5–9] as well as other knowledge sources [10,11].

Consequently, we have recently reported the successful application of random forest (RF) ML classifiers to segment the optical microscopy images of typical steel materials on ferrite (F), pearlite (P), bainite (B), and martensite (M) microstructures, with possibilities of further segmentation/analysis of their corresponding sub-phases [4]. The key point of this technique was the possibility to get the excellent quality of quantitative results, which was comparable or even better than manual estimations by experts, not to mention the benefits of microstructure phase area visualizations, analysis speed boosts, and automatizations with computers.

As with any single techniques, there were also some limitations of our previously established RF classifiers. One of them was that number and type of microstructure phases/classes \(j = A, B, C, ..., n\) present in training image dataset must correspond to the same set of \(j\) in dataset being analyzed in order to get the correct results with the specific RF classifier. This poses less of a problem for an expert who recognizes well the type of \(j\) in images for segmentation, but for any use in a large scale of industrial analytical laboratory or for the use by a non-expert in steel/metal microstructures, this limitation should be properly addressed. The prime aim of the current work is to develop additional image processing/analysis protocols for assisting in choosing the appropriate RF classifier to the image of interest.

Figure 1 shows the envisioned general workflow of image analysis for metallic materials with ML and image processing tools. The current analysis module and its contribution to the workflow are shown by highlighted/spotlighted orange color. In the case of the human expert in metallurgy field, it is often enough to have just a brief look at the microscopy image of steel material for judging what types of microstructures, e.g. F/P-, F/P/B-, or B/M-type ones, the image shows. To do it on a computer in an automated way could bring the great benefits, for example, in analysis of a welded part of steels that should contain all sorts of the microstructure types at different areas. The high-resolution image data of the whole welded part could contain thousands of images, which hold a lot of useful information, but which are difficult/impractical to sort and analyze manually. This is our first attempt to address this problem.

2. Experimental details

All samples of A-type steel (see composition in Figure 2) obtained by cooling of this alloy from 1400°C at 0.3, 1, 3, and 10°C/s rates were prepared by a conventional polishing and subsequent etching with 0.5% picric and 0.5% nitric acids in ethanol [12]. The microstructures in our samples correspond to the typical ones appearing in a welded part of steels. They were imaged with same spatial resolution on BX53M optical microscope (Olympus, Japan) equipped with MPLN50x objective (Olympus, Japan) and DP22 CCD camera (Olympus, Japan), which satisfied the pixel size requirement for maximum optical resolution (see grayscale image in Figure 2). The samples at 0.3, 1, 3, and 10°C/s rates were identified as \((F_{gb}+F_{gb}+F_{sp})/P\), \((F_{gb}+F_{gb}+F_{sp})/P\), \((F_{gb}+F_{gb}+F_{sp})/P\), and \(B/M\)-type microstructures, respectively, with different contributions of ferrite sub-phases: allotriomorphic \((F_{gb})\), grain boundary polygonal \((F_{gb})\), and side plate \((F_{sp})\) ferrites.

The image pre-processing, analysis, and segmentation were conducted by using the open-source FIJI software package with Trainable Weka Segmentation plugin on two-CPU 6128 Optiron Workstation with 128 GB RAM [13–17]. The ML on training/test datasets with statistical image data for corresponding image labeling on present microstructures was performed by using the open-source WEKA software package, which is a collection of ML algorithms for data mining tasks [18].

3. Results and discussion

Before discussing image labeling, Figure 2 shows the example of the application of the RF ML algorithm to train it on four microscopy optical images of A-type steel and segment them on F and P microstructure phases, which are formed by cooling from 1400°C at 1°C/s rate. The successful application of automatic pattern recognition with such well-trained RF classifier was confirmed by an expert with manual visual inspection and line-intercept analysis (see percentage numbers in Figure 2). In principle, such RF classifier could be applied to any number of other images containing F and P microstructures if such images have similar quality and same spatial resolution.
Figure 1. General outline of the workflow for the development of new metallurgical materials with image analysis protocols by using machine learning and image processing tools. The orange highlighted arrows and spotlighted text boxes indicate the main theme of the current work.

| Steel A | C    | Si  | Mn  | P   | S   |
|---------|------|-----|-----|-----|-----|
| mass%   | 0.152| 0.015 | 1.51 | 0.007 | 0.0016 |

A-Type Steel (cooling rate at 1 °C/s)

Overlay of original and auto segmented image

Figure 2. Example of image segmentation protocol on F and P microstructure phases by using the image pre-processing and random forest classifier tools. The color coding on F and P microstructure areas of A-type steel is given on insert together with results of quantitative manual and automated analyses.
However, as discussed in Section 1, the required prior knowledge on steel microstructures (i.e. the type of microstructures in the image of interest) could be problematic for choosing the RF classifier correctly. As one possibility to solve this problem, we extracted the statistical attributes from image datasets for various cooling rates and tried to classify/label each image into $F/P$, $F/P/B$, and $B/M$-type microstructures with ML tools. Figure 3 shows the protocol of image processing and subsequent attribute extraction: (1) automatic conversion of color image stacks to 8-bit gray level ones for each A-steel cooling rate; (2) consistent automatic optimization of brightness and contrast of resulted image stacks based on analysis of the image histograms; (3) automatic conversion of resulted image stacks to binary ones (black and white) by analyzing image histograms and subsequent thresholding; and (4) automatically count and measure the black objects/particles in stacks of resulted binary images.

Table 1 lists the names and descriptions of absolute or mean values of estimated attributes for each image (one data point) in resulted training and test datasets. In total, there were 117 and 73 of training and test data points/images in 21-dimensional attribute space to work with, respectively. They were used to train and test the different ML classifiers in order to classify into the four types of microstructures. Note that some attributes in Table 1 were completely irrelevant to our classification problem, redundant, and correlated to others. They were left on purpose and used to check the attribute selection/reduction tools and robustness of different classifiers to such datasets. This behavior is important to know and consider if adding new attributes in the future will be necessary or they will become available.

Some differences between training and test datasets were also added to model the possible real-world applications: (1) different samples for test dataset were prepared time apart from training dataset, but with the same composition and cooling rates; (2) the data in test dataset for $3^\circ C/s$ cooling rate were extracted from images with bad quality of sample etching for B-phase visualization; and thus, F and P phases were mainly visible in those images. Such data were used on purpose in order to check the sensitivity of attributes and classifiers to data quality variations, which could be typically encountered in actual experiments.

Figure 4 shows the examples of 3D and 2D slices of such 21-dimensional attribute space. In spite of...
considerable scattering, there is a visible tendency for data points in training and test datasets to group depending on types of microstructure combinations. It is also seen in the 2D slice of surface density vs. aspect ratio (plotted on gray plane) that largest deviations between training (dark green) and test data (light green) are for 3°C/s cooling rate ones, which is reasonable due to bad etching quality mentioned above. This can also be observed on 3D/2D slices for different attributes (see Supplemental Material). Then, we have applied several ML techniques to train, classify, and validate the resulted classification accuracy with them. For comparison, Table 2 lists the used methods and results of 10-fold cross-validation on the training dataset, which was utilized to build classifiers. On

![Figure 4](image-url)  
Figure 4. The 3D/2D slices of the multidimensional attribute space for training and test datasets of image statistics. Each attribute data point corresponds to the absolute or mean value for a single image (see Figure 3 and Table 1). The mapping example of 3D points to the 2D plane is shown for clarity with blue color lines. Additional 2D slices in the form of partial scatter matrix are available as Supplemental Material.

### Table 1. Attributes in training and test datasets corresponding to single image characteristics.

| No. | Attribute | Definition | Comment |
|-----|-----------|------------|---------|
| 1   | Count     | Black objects | Absolute value, correlated |
| 2   | Total area| Black objects | Absolute value, correlated |
| 3   | Size      | Black objects | Mean value |
| 4   | %Area     | Black objects | Absolute value, correlated |
| 5   | Mean      | Average value of all pixels in black objects | Constant value, irrelevant |
| 6   | Mode      | Most frequently occurring value of pixels in black objects | Constant value, irrelevant |
| 7   | Perimeter | Black objects | Mean value |
| 8   | Major     | Axis of fitted ellipse into black particle | Mean value |
| 9   | Minor     | Axis of fitted ellipse into black particle | Mean value |
| 10  | Angle     | Angle between the fitted ellipse axis of the black particle and a line parallel to the X-axis of the image | Mean value, irrelevant |
| 11  | Circularity | Black objects, $4\pi \times |Size| / |Perimeter|$ | Mean value, robust to image scale |
| 12  | Solidity  | Black objects, $4\pi \times |Size| / |Convex area|$ | Mean value, dimensionless, robust to image scale |
| 13  | Feret     | Maximum caliper of black objects | Mean value |
| 14  | FeretX    | Starting X coordinates of the Feret | Mean value, irrelevant |
| 15  | FeretY    | Starting Y coordinates of the Feret | Mean value, irrelevant |
| 16  | FeretAngle| The angle of the Feret | Mean value, irrelevant |
| 17  | MinFeret  | Minimum caliper of black objects | Mean value |
| 18  | IntDen    | The sum of the values of the pixels in black objects | Mean value, correlated |
| 19  | Aspect ratio | $|Major| / |Minor|$ | Mean value, dimensionless, robust to image scale |
| 20  | Density   | $|Count| / |Total Area|$ | Mean value, correlated |
| 21  | Surface density | $|Count| / |\mu m^2|$ | Mean value, robust to image scale, correlated |
Table 2. Performance comparisons between random forest, neural network, and Auto-WEKA classifiers. The a, b, c, and d columns/row labels correspond to the instances of the classification on 0.3, 1, 3, and 10°C/s or (F_gb+F_dp+F_sp)/P, (F_gb+F_all+F_sp)/P, and B/M classes, respectively. The numbers in brackets are calculated by omitting the data at 0.3°C/s. All attributes from Table 1 were used.

| Classifier                  | Main details                                               | Ten-fold cross-validation on training dataset | Validation on test dataset |
|-----------------------------|------------------------------------------------------------|---------------------------------------------|---------------------------|
|                             | Confusion matrix, correct classification                  | Confusion Matrix, correct classification    |                           |
| Random forest 0.1 s to build | Number of trees: 100                                      | a b c d                                    | a b c d                    |
|                             | Number of randomly chosen attributes: int[log2(Attributes) + 1] | 30 0 0 0 a                                | 23 1 0 0 a                |
|                             | Multilayer perceptron (neural networks) 1.0 s to build     | 0 38 0 0 b                               | 0 16 0 0 b                |
|                             | Learning rate: 0.3                                        | 0 1 18 0 c                              | 0 10 5 2 c                |
|                             | One single hidden layer, number of neurons: (attributes + classes)/2 | 0 1 1 29 d                              | 0 0 0 0 29 d              |
| Auto-WEKA 15 min to build   | Best classifier:                                          | a b c d                                    | a b c d                    |
|                             | Attribute Selected Classifier                             | 30 0 0 0 a                                | 22 2 0 0 a                |
|                             | Arguments:                                                | 0 38 0 0 b                               | 0 16 0 0 b                |
| Auto-WEKA 12 h to build     | Greedy Stepwise, Correlation-based                        | 0 0 19 0 c                              | 0 17 0 0 c                |
|                             | Feature Subset Selection, Random Forest                   | 0 0 0 30 d                               | 0 0 0 29 d                |
| Auto-WEKA 24 h to build     | Best classifier:                                          | a b c d                                    | a b c d                    |
|                             | Logistic Model Trees                                       | 30 0 0 0 a                                | 23 1 0 0 a                |
|                             | Arguments:                                                | 0 38 0 0 b                               | 0 16 0 0 b                |
|                             | Greedy Stepwise, Correlation-based                        | 0 0 19 0 c                              | 0 16 0 1 c                |

The next step, such classifiers were applied on test dataset for additional validations. The corresponding results are listed in Table 2.

In addition to RF, multilayer perceptron (MP), and other individual classifiers, we also used the state-of-the-art Auto-WEKA, which is the RF-based Bayesian optimization module for tuning hyperparameters of the classification algorithms built into the WEKA software package [19,20]. It works with 27 base ML algorithms, 10 meta- and 2 ensemble-methods, combining 3 search and 8 evaluator techniques. In total, 789 hyperparameters from all classification algorithms and feature selectors/evaluators could be used in optimization depending on accuracy, time, and PC resources. Table 2 lists several Auto-WEKA runs with names of classifiers and main arguments, which is selected for the given computation time. Note that Auto-WEKA generates a new classifier with optimized hyperparameters, which could be conveniently applied in ordinary WEKA way to the test datasets. The Auto-WEKA deals with hyper-dimensional surface, which has many local minima; therefore, more classifiers could be created with different or same parameters depending on the number and setting of the Auto-WEKA runs. To our knowledge, this is the first application of Auto-WEKA technique to the material science problems.

From Table 2, it can be seen that RF, MP, and Auto-WEKA can easily achieve the 96.6–100% accuracy with 10-fold cross-validation on our training dataset. Note that diagonal and off-diagonal elements in confusion matrix (CM) correspond to the number of correct and incorrect classifications, respectively (see Equation (1)). All other errors and statistical values can be easily calculated from such matrices:

\[
CM = \begin{pmatrix}
    a & b & c & d \\
    a & b & c & d \\
    a & b & c & d \\
    a & b & c & d \\
\end{pmatrix}
\]

(1)

For example, Equations (2)–(5) are the true-positive rate ($TPR_a$) or recall, i.e. the proportion of actual positives that are correctly identified; the false-positive rate ($FPR_a$), i.e. the proportion of all negatives that still yield positive test outcomes; the positive predictive or precision value ($PPV_a$), i.e. the proportions of positive results that are true positive; and the effectiveness measurement ($F - Measure_d$), i.e. the weighted
average of the \( PPV \) and \( TPR \), for classes \( a, b, c, \) and \( d \), respectively:

\[
TPR_a = \frac{TP_a}{P_a} = \frac{TP_a/(TP_a + FN_a)}{1 - FNR_a} = \frac{TP_a}{P_a + TP_a + FN_a} = \frac{TP_a}{P_a + FN_a} = 1 - FNR_a
\]

(2),

\[
FPR_b = \frac{FP_b}{N_b} = \frac{FP_b/(FP_b + TN_b)}{1 - TNR_b} = \frac{FP_b}{P_b + FN_b} = 1 - TNR_b
\]

(3),

\[
PPV_c = \frac{TP_c}{(TP_c + FP_c)}
\]

(4),

\[
F-\text{Measure}_d = 2 \times \frac{PPV_d \times TPR_d}{(PPV_d + TPR_d)}
\]

(5).

In Equation (2), the \( TP_a = aa, FN_a = ba + ca + da, P_a = TP_a + FN_a, \) and \( FNR_a \) are the true-positive (hit), false-negative (miss), condition-positive, and false-negative rate values for class \( a \), respectively, with \( FN_a \) to be the proportion of positives which yield negative test outcomes. In Equation 3, the \( FP_b = ba + bc + bd, N_b = \sum_{row=a}^d CM, TN_b = N_b - FP_b, \) and \( TNR_b \) are the false-positive (false alarm), number of real-negative cases, true-negative (correct prediction of not belonging to class), and true-negative rate values for class \( b \), respectively, with \( TN_b \) to be the proportion of actual negatives that are correctly identified.

Though Auto-WEKA classifiers slightly outperformed the RF and MP ones on the training dataset, but by applying these trained classifiers on the test dataset, the accuracy of correct prediction dropped significantly and differently for all of them with the lowest decrease for RF (see the last column in Table 2). Below, we will demonstrate in more details the overall superiority of RF classifier for our microstructure-labeling problem. Note that increased misclassification on test dataset was mainly due to the instances at \( 3^\circ C/s \) cooling rate. Actually, this was the reasonable and good result since these data were obtained from images with insufficient etching to visualize B microstructures. Then, the classifiers mainly assigned these test data to the F/P-type microstructure (\( b \)-column in \( CM \)), which indicates that our method spotted this problem automatically. The small misclassification between data at 0.3 and \( 1^\circ C/s \) cooling rates was also understandable since both samples are F/P-type microstructures and differentiated in F sub-microstructures, i.e. in relative contributions of \( F_{abl}, F_{gb}, \) and \( F_{sp}. \) However, in total, our method can distinguish reasonably well such F/P microstructures at 0.3 and \( 1^\circ C/s \) cooling rates. In principle, more precise labeling on the relative contribution of particular F sub-microstructure could be feasible with some attributes from Euclidean distance conversion technique (see, for example, [4]). If to remove \( 3^\circ C/s \) data from test dataset, then classification accuracies had improved significantly with all classifiers (see values in parentheses in Table 2). By comparing these values, it can be noted that classifiers with the RF algorithm in their core produced the most accurate predictions.

Actually, this was not a surprise. It was reported that by applying 179 classifiers on 121 datasets from the UCI database, the RF classifier versions produced the best results in most of the cases [21]. A similar conclusion was also derived for smaller scale investigation with 65 WEKA classifiers on 3 datasets [22]. In our case, the basic RF classifiers even slightly outperformed the Auto-WEKA classifier with RF in its core (98.6 vs. 97.1%). Probably, it was due to only three trees in RF with Auto-WEKA and better internal attribute selection in base RF algorithm compared to greedy stepwise and feature subset selection techniques applied to the training dataset prior to RF classifier creation in Auto-WEKA. Figure 5(a) demonstrates such internal attribute selection in the process of base RF classifier build-up by plotting of normalized attribute counts from all RF trees. Figure 5(a) also shows that there was little difference between 100 and 1000 trees in the forests, which related to the well-known leveling-off the classification accuracy for forests with more than \( \sim 100-200 \) trees [23,24] and better RF classifier robustness with respect to noise [25]. The attributes, which are irrelevant to our classification task, such as Mode, Mean, FeretX, and FeretY got zero or very low counts in tree nodes due to insignificant statistical effect on information gains in a process of tree growth. Figure 5(a) also demonstrates that RF classifier is not a ‘black box’ since attribute importance could be extracted and each tree has very clear meaning (see Figure 5(b)). The base RF classifier can be also created much faster compared to Auto-WEKA one. Therefore, the automated hyperparameter optimizations still require the careful comparisons. Nevertheless and apart from required PC time and resources, all classifiers listed in Table 2 can produce quite good results with \( \sim 86-99\% \) accuracy (see values in parentheses).

From a practical viewpoint, it is better to do deal with attributes in datasets, which are robust to image scale/area. Otherwise, the images for testing should be collected with the same spatial resolution or additional normalization is necessary. Such robust attributes are the Surface Density, Aspect Ratio, Circularity, and Solidity ones. However, this manual attribute reduction could lead to the decrease in classification accuracy. Table 3 shows the performance of several well-known ML techniques with such reduced training/test datasets. Again, the best classification accuracy was obtained with RF classifier. In this case, the drop in accuracy was only \( \sim 1.5\% \) compared to the use of all 21 attributes.

Figure 6 demonstrates the differences between classifiers in terms of their decision boundaries, which
were estimated with smooth interpolations. As indicated by labels, each class maximum probability is assigned with single-color code. Then, RGB pixel color in discretized attribute space is defined by a linear combination of estimated class probabilities (weighted averages based on kernel density estimators), which are calculated by sampling of the points in corresponding attribute space with classification models [26]. Except k-means and k-nearest neighbors classifiers with over-fragmented boundaries, other

Figure 5. Extracted information from random forest classifier: (a) the attribute importance and (b) the example structure of one tree.
tested ones have more similarities in this respect. Nevertheless, the RF classifier decision boundaries automatically partitioned the relevant attribute space in the simplest way but in accordance with our intuitively expected distribution of such boundaries for different steel types.

Here, it should be mentioned that Tables 2 and 3 list results with default WEKA settings for all classifiers since they still produced over 90% labeling accuracy on the training dataset, except with k-Means one. Note that these settings are based on known behaviors of these algorithms, i.e. they are not arbitrary for a good start. In principle, the multi-parameter optimizations could be further applied on single classifier from Table 3 with MultiSearch meta-method in WEKA, which could optimize arbitrary number of user-defined parameters and their ranges after attribute selection/filtering with other tools. We did not proceed rigorously with such single classifier tuning due to plausible gains with unjustifiable efforts compared to already achieved high accuracy with default settings for RF algorithm: ~98/99% accuracy on training/test datasets without any external attribute selection/filtering (see Table 2). Nevertheless, these tools could be useful for other training datasets and they are available.

4. Conclusions

We have developed the protocol with image processing and ML tools to label the images of steel materials on typical F/P-, F/P/B-, and B/M-type steels. The RF algorithm or its modifications performed overall better compared to other ML tools due to its ensemble, unbiased, and stable nature. Our technique of image attribute extraction and consequent ML application on image datasets could find potential applications in metallurgical research/analytical laboratories.

Table 3. Performance comparisons between random forest and other popular classifiers. The a, b, c, and d columns/row labels correspond to the instances of the classification on 0.3, 1, 3, and 10°C/s or \((F_{gb} + F_{all} + F_{sp})/P, (F_{gb} + F_{all} + F_{sp})/P, (F_{all} + F_{sp})/P/P/B,\) and B/M classes, respectively. The numbers in brackets are calculated by omitting the data at 0.3°C/s. Attributes, which are robust to image scale, were used from Table 1.

| Classifier with default settings in WEKA | Ten-fold validation on training dataset | Validation on test dataset |
|----------------------------------------|-----------------------------------------|---------------------------|
|                                        | Confusion matrix, correct classification | Confusion matrix, correct classification |
| Random forest                          | a 30 0 0 0 a 22 2 0 0 a                  | a 30 0 0 0 a 30 30 0 0 a          |
|                                        | b 0 0 0 0 b 16 0 0 0 b                  | b 0 0 0 0 b 30 30 0 0 b           |
|                                        | c 0 3 16 0 c 0 10 0 0 c                 | c 0 3 16 0 c 30 30 0 0 c          |
|                                        | d 0 0 0 30 d 0 0 0 29 d                 | d 0 0 0 30 d 30 30 0 0 d          |
|                                        | ~97.4%                                  | ~77.9% (97.1%)                  |
| Logistic regression                    | a 29 1 0 0 a                            | a 29 1 0 0 a                     |
|                                        | b 30 0 0 0 a                            | b 30 0 0 0 a                     |
|                                        | c 0 3 16 0 c                            | c 0 3 16 0 c                     |
|                                        | d 0 0 0 30 d                            | d 0 0 0 30 d                     |
|                                        | ~99.1%                                  | ~44.2% (55.1%)                  |
| Multilayer perceptron (neural networks)| a 30 2 0 0 a                            | a 30 2 0 0 a                     |
|                                        | b 0 0 0 0 b                            | b 0 0 0 0 b                      |
|                                        | c 0 3 16 0 c                            | c 0 3 16 0 c                     |
|                                        | d 0 0 0 30 d                            | d 0 0 0 30 d                     |
|                                        | ~98.3%                                  | ~74.4% (92.8%)                  |
| Naive Bayes                            | a 28 2 0 0 a                            | a 28 2 0 0 a                     |
|                                        | b 30 0 0 0 a                            | b 30 0 0 0 a                     |
|                                        | c 0 3 16 0 c                            | c 0 3 16 0 c                     |
|                                        | d 0 0 0 30 d                            | d 0 0 0 30 d                     |
|                                        | ~93.2%                                  | ~87.2% (95.7%)                  |
| k-Means                                | a 22 8 0 0 a                            | a 22 8 0 0 a                     |
|                                        | b 16 16 6 0 b                           | b 16 16 6 0 b                    |
|                                        | c 1 0 18 0 c                            | c 1 0 18 0 c                     |
|                                        | d 0 0 3 27 d                            | d 0 0 3 27 d                     |
|                                        | ~70.9%                                  | ~67.4% (63.8%)                  |
| k-Nearest neighbors                    | a 27 3 0 0 a                            | a 27 3 0 0 a                     |
|                                        | b 0 3 17 1 c                            | b 0 3 17 1 c                     |
|                                        | c 0 0 1 29 d                            | c 0 0 1 29 d                     |
|                                        | ~94.9%                                  | ~69.4% (87.0%)                  |
Acknowledgments

Financial support from Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), ’Structural Materials for Innovation’, and ’Materials Integration for Revolutionary Design System of Structural Materials’ (Funding agency: JST) is gratefully acknowledged.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), ’Structural Materials for Innovation’, and ’Materials Integration for Revolutionary Design System of Structural Materials’ (Funding agency: JST).

References

[1] Bramfitt BL. Structure/property relationships in irons and steels. In: Davis JR, editor. Metals handbook desk edition. 2nd ed. Novelty (OH): ASM International; 1998. p. 153–173.

[2] Kumar A, Singh SB, Ray KK. Influence of bainite/martensite-content on the tensile properties of low carbon dual-phase steels. Mater Sci Eng. 2008; A474:270–282.

[3] Zare A, Ekrami A. Influence of martensite volume fraction on tensile properties of triple phase ferrite–bainite–martensite steels. Mater Sci Eng. 2011; A530:440–445.

[4] Bulgarevich DS, Tsukamoto S, Kasuya T, et al. Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures. Sci Rep. 2018; 8:2078–1–8.

[5] Liu Y, Zhao T, Ju W, et al. Materials discovery and design using machine learning. J Materiomics. 2017; 3:159–177.

[6] Butler KT, Davies DW, Cartwright H, et al. Machine learning for molecular and materials science. Nature. 2018; 559:547–555.

[7] Jalem R, Nakayama M, Noda Y, et al. A general representation scheme for crystalline solids based on Voronoi-tessellation real feature values and atomic property data. Sci Technol Adv Mater. 2018; 19:231–242.

[8] Pham TL, Kino H, Terakura K, et al. Machine learning reveals orbital interaction in materials. Sci Technol Adv Mater. 2017; 18:756–765.

[9] Shin D, Lee S, Shyam A, et al. Peta-scale supercomputing to accelerate the design of high-temperature alloys. Sci Technol Adv Mater. 2017; 18:828–838.

[10] Onishi T, Kadohira T, Watanabe I. Relation extraction with weakly supervised learning based on process-structure-property-performance reciprocity. Sci Technol Adv Mater. 2018; 19:649–659.

[11] Ito S, Nagao H, Kasuya T, et al. Grain growth prediction based on data assimilation by implementing 4DVar on multiphase-field model. Sci Technol Adv Mater. 2017; 18:857–868.

[12] Ikawa H, Oshige H, Tanoue T. Study on the martensite-austenite constituent in weld-heat affected zone of high strength steel. J Jap Weld Soc. 1980; 49:467–472.

[13] Schneider CA, Rasband WS, Eliceiri KW. NIH image to imageJ: 25 years of image analysis. Nat Methods. 2012; 9:671–675.

[14] Collins TJ. ImageJ for microscopy. BioTechniques. 2007; 43:S25–S30.

[15] Schindelin J, Arganda-Carreras I, Frise E, et al. Fiji: an open-source platform for biological-image analysis. Nat Methods. 2012; 9:676–682.

[16] Arganda-Carreras I, Kaynig V, Rueden C, et al. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics. 2017; 33:2424–2426.

[17] Ferreira T, Rasband W ImageJ user guide IJ 1.46r. 2012 [cited Feb 26 2016]. p. 198. Available from: http://imagej.nih.gov/ij/docs/guide

[18] Frank E, Hall MA, Witten IH. The WEKA workbench. In: Online appendix for data mining: practical machine learning tools and techniques. 4th ed. San Francisco (CA): Morgan Kaufmann; 2016. p. 128.

[19] Thornton C, Hutter F, Hoos HH, et al. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. KDD ’13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data
mining; 2013 August 11–14; Chicago, Illinois: ACM New York, NY; p. 847–855.
[20] Kotthoff L, Thornton C, Hoos HH, et al. Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J Mach Learn Res. 2017;18:1–5.
[21] Fernández-Delgado M, Cernadas E, Senén B, et al. Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res. 2014;15:3133–3181.
[22] Tatsis VA, Tjortjis C, Tzirakis P. Evaluating data mining algorithms using molecular dynamics trajectories. Int J Data Min Bioin. 2013;8:169–187.
[23] Ko BC, Kim SH, Nam J-Y. X-ray image classification using random forests with local wavelet-based CS-local binary patterns. J Digit Imaging. 2011;24:1141–1151.
[24] Wright MN, Ziegler A. A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw. 2017;77:1–17.
[25] Breiman L. Random forests. Mach Learn. 2001;45:5–32.
[26] Frank E, Hall M. Visualizing class probability estimators. In: editors, Lavrač N, Gamberger D, Todorovski L, et al. Knowledge discovery in databases: PKDD 2003. PKDD 2003. Lecture notes in computer science. Vol. 2838. Berlin, Heidelberg: Springer; 2003. p. 168–179.