



A Novel Hyperparameter Optimization Approach for Supervised Classification: Phase Prediction of Multi-Principal Element Alloys

Syed Hassan Fatimi¹ · Zidong Wang¹ · Isaac T.H. Chang² · Weibo Liu¹ · Xiaohui Liu¹

Received: 10 November 2024 / Accepted: 9 January 2025
© The Author(s) 2025

Abstract

In this paper, a hyperparameter optimization approach is proposed for the phase prediction of multi-principal element alloys (MPEAs) through the introduction of two novel hyperparameters: outlier detection and feature subset selection. To gain a deeper understanding of the connection between alloy phases and their elemental properties, an artificial neural network is employed, with hyperparameter optimization performed using a genetic algorithm to select the optimum hyperparameters. The two novel hyperparameters, outlier detection and feature subset selection, are introduced within the optimization framework, along with new crossover and mutation operators for handling single and multi-valued genes simultaneously. Ablation studies are conducted, illustrating an improvement in prediction accuracy with the inclusion of these new hyperparameters. A comparison with five existing algorithms in multi-class classification is made, demonstrating an improvement in the performance of phase prediction, thereby providing a better perception of the alloy phase space for high-throughput MPEA design.

Keywords Hyperparameter optimization · Outlier detection · Feature selection · Artificial neural networks · Genetic algorithm · Multi-principal element alloys

Introduction

The study of metals and their combinations as alloys has spanned many centuries. Recently, interest has been developed in exploring alloys beyond conventional limits. In 2004, two seminal papers by Cantor [5] and Yeh [62] introduced the study of a new type of alloys, termed multi-principal element alloys (MPEAs). Unlike conventional alloys, MPEAs contain multiple principal elements with (near) equiatomic concentrations. These alloys have been shown to exhibit excellent mechanical and functional properties [16] compared to conventional alloys and are expected to be applied across a wider range of fields. However, the existence of multiple phases

and the large search space make the selection of elements and their compositions for MPEAs with desirable properties challenging. Experimental study of such alloys has traditionally relied on trial-and-error approaches, which have proven insufficient due to the vastness of the search space, making trial-and-error discovery infeasible. This has compelled the adoption of data science approaches for alloy discovery.

With the availability of large datasets and advancements in algorithms, a surge of interest in the field of machine learning has recently been observed. Various machine learning techniques have been applied to different aspects of material science, such as property analysis, the discovery of new materials, and quantum chemistry [56]. Among various applications, phase prediction of MPEAs has received considerable attention due to the fact that alloy phases tend to influence physical properties [25]. Certain phases are more desirable than others when mechanical performance, such as ductility and yield strength, is considered. MPEAs with multiple phases can offer a balance between strength and ductility [17]. These guiding factors are crucial when alloy applications are considered, highlighting the need for a universal indicator for phase prediction of MPEAs. Unfortunately, a

✉ Zidong Wang
Zidong.Wang@brunel.ac.uk

Isaac T.H. Chang
isaac.chang@brunel.ac.uk

¹ Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

² Brunel Centre for Advanced Solidification Technology (BCAST), Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

solid theoretical framework for phase prediction does not exist due to the astronomical number of possible materials. An estimate of the possible alloys that could be formed is approximately $\sim 10^{78}$ [4]. Only a tiny fraction of these alloys has been synthesized and studied. This renders the current trial-and-error method for exploring new alloy configurations obsolete, prompting the need for enhanced search techniques, whether model-driven or data-driven.

When phase prediction of MPEAs is translated into machine learning, it becomes a supervised multi-class classification problem, where each class represents a particular alloy phase, and these phases are treated as mutually exclusive. Numerous studies have been conducted to address this problem using various features and strategies [21, 29, 31, 41, 43, 66]. These studies have involved the application of artificial neural networks, support vector machines, random forests, and deep learning with generative adversarial networks.

It is worth mentioning that Artificial Neural Networks (ANNs) have been widely employed to develop various predictive models. ANNs are computational models inspired by the structure and functioning of biological neurons, such as those in the human brain [19, 33]. These models consist of interconnected nodes, referred to as neurons, which are organized into layers. Each neuron receives input signals, processes them using an activation function, and generates an output signal. ANNs are well suited for handling high-dimensional data, as they can learn complex non-linear relationships between input and output data.

A major challenge in the use of ANNs for predictive modeling is the determination of hyperparameters. Numerous adjustable parameters need to be tuned prior to classification, such as the number of neurons per layer, learning rate, and regularization strength. Identifying the optimal set of hyperparameters is a challenging yet necessary task, as it ultimately influences the performance of the model. Typically, these parameters are determined based on heuristic rules and are manually adjusted, which can be time-consuming. Hyperparameter optimization is the process through which the best set of hyperparameters is identified to maximize the performance of a model [6]. This problem can be framed as the optimization of a loss function over a constrained configuration space. Various approaches have been used in the past for hyperparameter optimization, including grid search, random search, Bayesian optimization, and evolutionary algorithms.

Among various optimization techniques, the genetic algorithm (GA), a subset of evolutionary algorithms, is regarded as a powerful and flexible tool. GA is a nature-inspired algorithm based on the principles of natural selection and Darwinian evolution. Being gradient-free, it is well suited for high-dimensional and complex systems. GA has been utilized for hyperparameter optimization and has demonstrated

significant improvements compared to grid search and Bayesian optimization [1]. Its population-based approach allows for parallelism and diversity in exploration. Due to these technical merits, GA has been widely applied to the hyperparameter optimization of ANNs [8, 40], convolutional neural networks [30], and long short-term memory networks [58].

The approach of ANN predictive modeling for identifying MPEAs has been utilized in the literature, but careful hyperparameter selection has not yet been explored. This may be due to the fact that the application of machine learning to MPEAs is a relatively recent interest within the scientific community, leaving ample room for improvement. Hyperparameter optimization has the potential to significantly enhance the predictive capabilities of ANNs, particularly when dealing with complex and noisy data containing outliers [44]. Due to the nature of experimental errors [11], MPEA data obtained from the literature is often affected by poor-quality data, including imbalanced class distributions, sparsity, and noise in the form of outliers. Any machine learning approach applied to MPEA prediction must be capable of addressing these issues. In our study, one of the key objectives is to incorporate outlier detection into the training process, rather than handling it solely during pre-processing, by integrating it within hyperparameter optimization. This approach will help clean the data during the training process and improve the predictive modeling of MPEAs.

Motivated by the above discussions, the objective of this paper is to develop a framework for supervised classification through hyperparameter optimization, with the inclusion of novel hyperparameters that will aid in handling outliers and feature selection. Specifically, a boxplot-based outlier detection and removal method, along with an optimal feature subset selection, is proposed. This strategy is employed to enhance the classification ability of ANNs when dealing with poor-quality data. The proposed methodology is applied to the phase classification of MPEAs.

The main contributions of this paper can be summarized in the following aspects:

1. A novel hyperparameter optimization algorithm is introduced, which incorporates two innovative parameters: outlier detection and feature subset selection. The outlier detection method is designed to identify and remove noisy or erroneous data points that could otherwise distort the model's performance, while feature subset selection ensures that only the most relevant features are considered during model training, which helps in improving both the accuracy and efficiency of the model. The combination of these two parameters allows the algorithm to better manage complex and noisy datasets, which are frequently encountered in MPEA studies.

2. The proposed hyperparameter optimization algorithm is applied to the phase classification of MPEAs. The experimental results clearly demonstrate the effectiveness of this method. It is shown that, by including the novel hyperparameters, significant improvements in classification performance can be achieved compared to traditional methods. These results confirm that the algorithm enhances the predictive capabilities of ANNs in handling poor-quality data, making it a valuable tool for more precise phase prediction in MPEA research.

The remaining parts of this study are organized as follows: In “**Background**” section, the background of MPEA phase classification is discussed, along with a description of the datasets used. “**Methodology**” section introduces the proposed hyperparameter optimization strategy, which utilizes both static and dynamic length chromosome genetic algorithms. In “**Experimental Results**” section, the experimental results for the phase classification of MPEAs are presented. Finally, the conclusion is provided in “**Conclusion**” section, along with suggestions for suitable future work.

Background

Multi-Principal Element Alloys

Multi-principal element alloys (MPEAs), also referred to as multi-component alloys or high-entropy alloys, are materials composed of three or more principal elements in significant proportions, typically with each element contributing between 5 and 35% by atomic composition. These alloys are distinguished by their complex and diverse compositions, in contrast to conventional binary or ternary alloys. Unique properties arise from their high configurational entropy, which results from the multi-element composition. MPEAs have been shown to exist in stable, disordered solid solution phases [15]. Due to their nature, MPEAs are capable of exhibiting excellent physical properties, such as high strength and hardness, exceptional wear resistance, remarkable high-temperature strength, good structural stability, as well as resistance to corrosion and oxidation [42, 50].

Determining the stable phases present in MPEAs has been an important focus of their study, as the resulting phases influence the mechanical properties of the alloys, thereby affecting their potential applications. One fundamental question regarding MPEAs is which atomic phase will form when a large number of different elements are mixed together. Surprisingly, the resulting phases tend to be simple structures [5]. Classic Hume-Rothery rules define the factors that influence the formation of solid solutions in binary systems, where only two element types are present. These factors include atomic size mismatch, valence electron

concentration, electronegativity, enthalpy of mixing, and configurational entropy, among others [3]. Studies on the phase formation of MPEAs have drawn conclusions similar to those of Hume-Rothery rules for binary systems [45, 65].

Unlike parametric approaches used in the past [65], machine learning-based phase prediction is regarded as a robust tool that can extract insights from given data with relatively low bias and high efficiency. Accurate and fast algorithms have been developed in recent years for the purpose of material discovery [55]. These approaches are employed to guide experimental analysis of complex multi-dimensional systems, such as MPEAs, and have become vital tools for material discovery in recent years.

Description of Data Set

Sources of MPEA Phase Data

The data used in this study is obtained from various sources in the literature [28, 36, 49, 67]. After the removal of duplicates and alloys containing radioactive elements, the merged dataset consists of 2218 compositions, ranging from binary to multi-component alloys. In the compiled dataset, each data point represents an individual alloy composition, along with its respective chemical features and atomic phase.

Characteristics of Features

Ten features are considered in this study, and their numerical values are obtained using the following equations [9, 65, 68]: 1) $\bar{Z} = \sum_{i=1}^n c_i Z_i$, 2) $\bar{r} = \sum_{i=1}^n c_i r_i$, 3) $\bar{m}_a = \sum_{i=1}^n c_i m_{a,i}$, 4) $\text{VEC} = \sum_{i=1}^n c_i \text{VEC}_i$, 5) $\Delta\chi = \sqrt{\sum_{i=1}^n c_i (\chi_i - \bar{\chi})^2}$, 6) $\delta = 100 \sqrt{\sum_{i=1}^n c_i (1 - r_i/\bar{r})^2}$, 7) $T_{\text{melt}} = \sum_{i=1}^n c_i T_i$, 8) $\Delta S = -R \sum_{i=1}^n c_i \ln c_i$, 9) $\Delta H_{\text{mix}} = \sum_{i=1, i < j}^n 4H_{ij} c_i c_j$.

Here, n is number of elements in an alloy, c_i is atomic concentration of i^{th} element, \bar{Z} is mean atomic number, Z_i is atomic number of i^{th} component, \bar{r} is mean atomic radius, r_i is atomic radius of i^{th} element, \bar{m}_a is mean atomic mass, $m_{a,i}$ is atomic mass of i^{th} element, VEC is mean valence electron concentration, VEC_i is valence electron concentration of i^{th} element, $\Delta\chi$ is Pauling electronegativity difference, χ_i is electronegativity of i^{th} element, δ is atomic size mismatch, T_{melt} is mean melting temperature, T_i is melting temperature of i^{th} element, ΔS is configurational entropy, R is ideal gas constant, ΔH_{mix} is enthalpy of mixing, and H_{ij} is enthalpy of binary alloy i, j . Values of binary mixing enthalpy are obtained from the Miedema model [47], and required elemental data for calculation of features is obtained from [45]. Number of elements in an alloy is considered as a separate feature.

Characteristics of Classes

The phases are categorized as follows: face-centered cubic (FCC), body-centered cubic (BCC), hexagonal closed pack (HCP), mixed phases (MP), amorphous (AM), intermetallic (IM), and BCC+FCC. A distinct imbalance is observed in the class distribution, which is attributed to the fact that certain alloy phases have been studied more extensively than others due to their potential applications and relevance [36]. For MPEAs, single-phase solid solutions (FCC, BCC, HCP) are of primary interest because their presence is associated with excellent mechanical properties, while the presence of intermetallics is considered undesirable due to their negative impact on mechanical behavior [4, 16]. The details of the data pre-processing are presented in “Experimental Results” section.

Problems of Dataset

As previously discussed in “Introduction” section, the compiled alloy dataset contains class imbalance and outliers. The class imbalance is depicted in Fig. 1, which illustrates the distribution of phases. The MP class contains the majority of alloy samples, while the IM class contains the fewest. It is also worth noting that the BCC and FCC classes, which are single-phase solid solutions, include a significant number of alloys. Due to the nature of metallurgic experiments and the presence of systematic and random errors in experimental analysis, MPEA data tends to be noisy and unsuitable for predictive modeling. Rather than modifying the noise, this study focuses on identifying and removing outliers. The physical nature of alloys makes it difficult to modify experimental data using statistical methods, as altering values of physical parameters could result in an unphysical representation of alloy systems. For this reason, the approach in this study

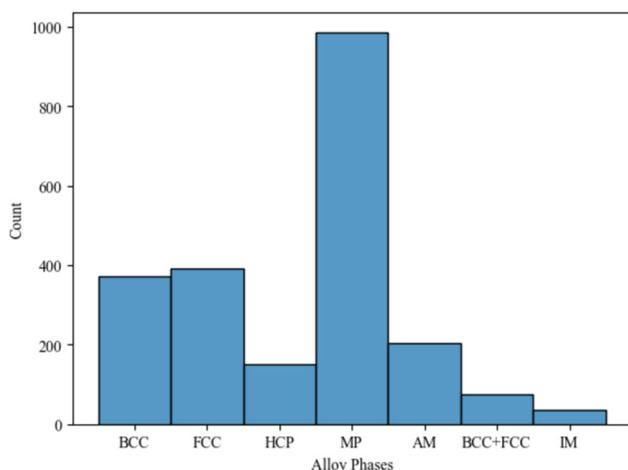


Fig. 1 Distribution of classes in alloy dataset depicting imbalance

is centered on outlier detection, which involves the removal of data points rather than their modification. This strategy will help ensure that the phase prediction model is reliable and can be effectively used by metallurgists in the search for MPEAs with desirable properties.

Methodology

ANN-Based Classification

To establish a framework for phase prediction, a supervised classification approach is formulated, where each alloy composition is characterized by ten distinct features as inputs and the phase as the output class. The ANN architecture can be described with consecutive input, hidden, and output layers. An initial network is proposed, consisting of two hidden layers and dropout regularization. Dropout is a regularization method used to prevent overfitting and can be easily implemented without significant computational overhead [18]. Each hidden layer employs the ReLU activation function [39]. Cross-entropy is chosen as the loss function for the network, and it is defined as

$$\text{loss} = -\frac{1}{N} \sum_{n=1}^N \sum_{i=1}^K T_{ni} \ln Y_{ni} \quad (1)$$

where N is number of observations, K is number of classes, T_{ni} is target value, and Y_{ni} is predicted value of the network.

The training process of an ANN is an iterative procedure in which a series of linear operations between consecutive layers is used to enhance the performance of the network by tuning its parameters, specifically the weights and biases. The operation process can be defined as

$$x_j^{(l)} = f \left(\sum_i w_{i,j}^{(l)} x_i^{(l-1)} + b_j^{(l)} \right) \quad (2)$$

where $x_j^{(l)}$ is the output of j^{th} neuron in l^{th} layer, $x_i^{(l-1)}$ is input of i^{th} neuron in previous layer, whereas $w_{i,j}^{(l)}$ and $b_j^{(l)}$ are weights and biases of j^{th} node in l^{th} layer. f represents a non-linear activation function. In this study, the architecture of ANN consists of an input layer for alloy features, two hidden layers with equal number of nodes in each layer, and an output layer for alloy phase.

A key point to note is that our network is not fully connected due to the presence of dropout regularization, which randomly removes connections between nodes based on a fixed probability p_{dropout} . This process is essential in overcoming overfitting, a phenomenon that occurs when a model fits too closely to its training data, thereby reducing its ability

to generalize and predict testing data. An added benefit of using dropout is that it makes the ANN more lightweight and significantly reduces computational complexity by removing certain node connections.

Genetic Algorithm

The GA is a type of optimization algorithm inspired by the processes of natural selection and evolution. It is used to solve optimization and search problems by mimicking the process of natural selection, allowing a population of solutions to evolve over successive generations. GA is particularly useful for problems involving large and complex search spaces, making it an ideal candidate for use in this study.

Before the implementation of GA, the data must be encoded so that each individual solution is represented as a chromosome, with each gene within a chromosome signifying decision variables.

The improvement of solutions in GA depends on two key operators: crossover and mutation. Crossover generates new solutions by combining solutions from the previous generation, while mutation randomly alters a decision variable within a particular solution. The process of GA can be depicted as follows:

- 1) Initialization of population to represent search space;
- 2) Evaluation of fitness through an objective function;
- 3) Selection of suitable individuals for reproduction;
- 4) Crossover through combination of individuals;
- 5) Mutation by randomly altering an offspring's genetic code.

This above process is iterative, with each iteration referred to as a generation. In every generation, evaluation, selection, crossover, and mutation take place. The process continues until a termination criterion is met, which could be based on the number of generations or a minimum tolerance in the fitness value. The GA process is illustrated in Fig. 2 as a flowchart. The performance of GA can be sensitive to the choice of parameters such as population size, crossover rate, mutation rate, and selection criteria. In this study, GA will be employed for hyperparameter optimization, incorporating a novel strategy for handling variable-sized chromosomes.

Encoding data within the framework of genes is crucial for the operation of GA. The vast majority of GA studies employ binary encoding, where data is represented as 0s and 1s. However, in this case, due to the presence of both real-numbered and categorical data, it is more appropriate to utilize real encoding. Each chromosome is composed of N genes, where N represents the number of hyperparameters in the problem.

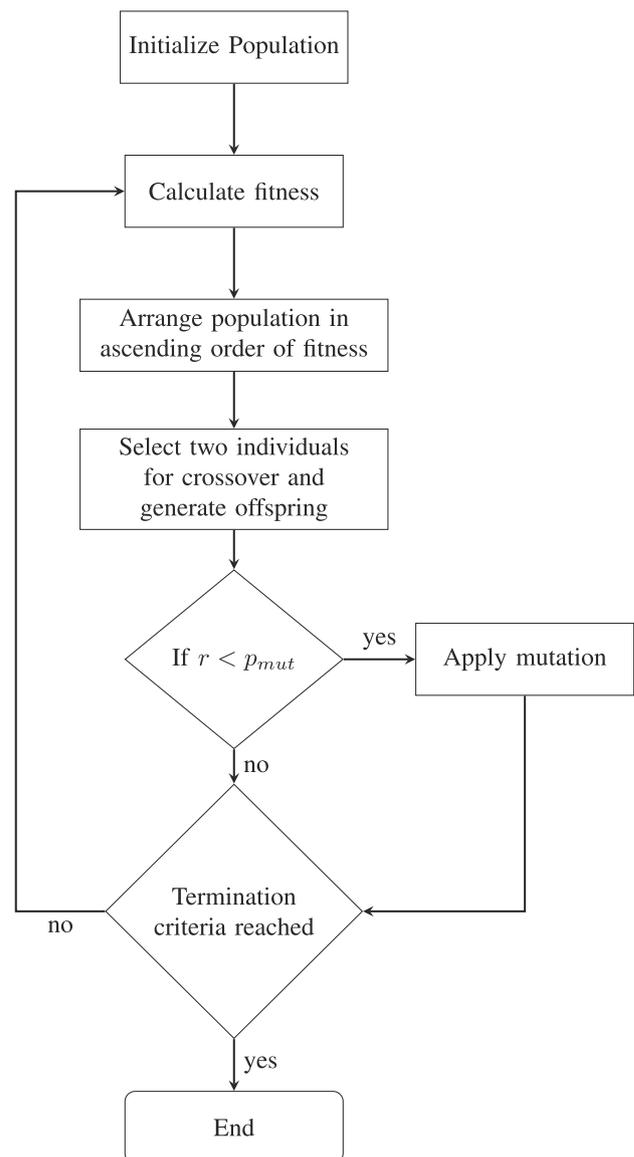


Fig. 2 Flowchart of genetic algorithm. p_{mut} is mutation probability, and r is a uniformly distributed random number between 0 and 1

In this study, single-point crossover and single-point mutation are employed, with a fixed mutation probability of $p_{mut} = 0.1$. Selection is based on the fitness of each individual, and instead of replacing all members of the previous generation, a few of the best individuals from the current generation are retained and passed on to the next. This method, known as elitism, ensures that the best solutions found so far are preserved across generations, preventing the loss of valuable genetic information due to random variation. Elitism helps maintain stability and ensures that promising solutions are not lost prematurely. It is particularly useful when the fitness landscape is dynamic or when the algorithm is prone

to premature convergence, a situation where the population converges to suboptimal solutions too early. By preserving the best individuals, elitism helps guide GA toward better solutions over time. A comprehensive depiction of the GA employed in the current study is presented as pseudocode in Algorithm 1.

Algorithm 1 The procedure of genetic algorithm.

- 1: Initialize population of hyperparameters with population size P .
 - 2: Calculate fitness of each individual.
 - 3: Arrange population in increasing order of fitness
 - 4: Select two individuals. Select a random point of intersection and apply crossover at that point to generate new offspring.
 - 5: If $r < p_{mut}$, select a random decision variable and initialize it with a different value within its respective range.
 - 6: Select the best 10% of current population and keep them for the next generation
 - 7: Terminate the algorithm if maximum number of generations is reached. If not, repeat Steps 2-6.
-

Dynamic Length Chromosome Operators

Canonical GA and its respective operators are designed based on a fixed number of decision variables, with each variable representing a gene in a fixed-length chromosome and multiple chromosomes forming a population. A significant limitation of this approach is that GA is not inherently suited to handle variable-length data, whether it involves a variable number of genes or a variable length of a particular gene. To address this deficiency, several studies have been conducted to develop approaches for variable-length chromosome GA by modifying the crossover and mutation operators to suit the specific problem. These approaches have been applied to various domains, such as topology design [23, 24] and hyperparameter optimization [58].

GA strategies exist for both fixed and variable-length chromosomes. However, a gap in the current research lies in the ability to handle variable-length genes, meaning the capability to manage chromosomes containing both single and multiple values within individual genes. An illustration of this phenomenon is shown in Fig. 3, where a standard chromosome is contrasted with a chromosome containing variable-length genes.

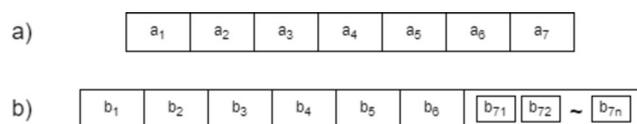


Fig. 3 Depiction of chromosomes where **a** shows normal chromosome with fixed-length genes; **b** shows chromosome having a single variable-length gene which consists of values $b_{71}, b_{72} \dots b_{7n}$, where n is an integer which can vary between a lower and upper bound

By examining Fig. 3, it can be discerned that there is a need for modified crossover and mutation operators capable of handling chromosomes containing both single-valued and multi-valued genes, where the length of multi-valued genes can vary. To address this, a robust framework of genetic operators has been developed, enabling effective crossover and mutation operations for the specific problem at hand.

To simplify the problem, two different crossover and mutation operators are defined: one for single-length genes and another for variable-length genes. The genetic operators for single-length genes function as standard crossover and mutation, where parents are segmented and joined together, with random mutations occurring at a single point. For the variable-length operators, only the gene with variable array length is handled. Consequently, both single-length and variable-length operators function simultaneously for each individual in the population.

The variable-length crossover operates by selecting the variable-length gene from two parents, ensuring that the length of parent 1 is less than the length of parent 2, i.e., $len(P_1) < len(P_2)$. Next, the length of the offspring is defined within the range $[len(P_1), len(P_2)]$, ensuring that the offspring's length always lies between the lengths of both parents. The crossover point c is then selected within the range $[1, len(P_1)]$ such that the point lies within bounds of P_1 . The first half of offspring O_1 is initialized by selecting values within P_1 until point of crossover c , such that $O_1 = P_1(1, \dots, c)$. The second half of O_1 is defined by checking each member of P_2 . If the member is not present in O_1 , it is added to O_1 ; otherwise, it is rejected. This process avoids repetitions and ensures that the characteristics of both parents are represented in the resulting offspring. The pseudocode for this process is shown in Algorithm 2.

Algorithm 2 The procedure of variable length crossover.

- 1: Select parents P_1 and P_2 such that $len(P_1) < len(P_2)$.
 - 2: Define length of offspring O_1 within $[len(P_1), len(P_2)]$ range.
 - 3: Select point of crossover c within range $[1, len(P_1)]$.
 - 4: Initialize first half of O_1 with $P_1(1, \dots, c)$.
 - 5: Initialize second half of O_1 by looping over P_2 and selecting members not present in O_1 .
-

For the variable-length mutation, the same strategy is applied as used for the traveling salesman problem. A random permutation of length l is generated, where l is a randomly selected length within the bounds of the decision variable. Thus, the mutation process involves sampling from a distribution of random permutations of the variable-length decision variable.

Hyperparameter Optimization

In an ANN, numerous tunable parameters, referred to as hyperparameters, influence its performance and control various aspects of the learning process. These parameters are often selected on an ad hoc basis, which is an ineffective practice that limits the capabilities of the ANN. To address this, hyperparameter optimization is employed to select the optimal parameters using an optimization technique that aims to maximize the network's performance, specifically by improving testing accuracy.

In this study, hyperparameter optimization is formulated as a minimization problem, where the goal is to identify the set of hyperparameters for a given model that returns the best performance when measured on a test set. This process can be mathematically represented as

$$x^* = \arg \min_{x \in \Theta} f(x) \quad (3)$$

where $f(x)$ is the objective function to be minimized; x^* is the set of hyperparameters that yield the lowest value of the objective function. Here, x can take any value within the domain Θ , which consists of all possible combinations of hyperparameters. In this study, the objective function is the root mean squared error (RMSE), evaluated on the test set.

Canonical hyperparameters for ANN considered in this study are

- Learning rate
- Batch size
- Number of epochs
- Number of nodes in each hidden layer
- Rate of dropout

The above parameters have been widely utilized in the literature across various types of optimization techniques, datasets, and problems involving both classification and regression [2, 12, 35].

In this study, two *new* concepts are incorporated as hyperparameters within the optimization framework: outlier detection and feature subset selection. These topics are discussed in more detail in the following subsection.

Outlier Detection

Outlier detection is the process of identifying data points or observations that deviate significantly from the majority of the data in a dataset. These outliers are typically observations that are unusual compared to the rest of the data and may indicate anomalies or errors. A well-known statistical approach for identifying outliers in a distribution is the boxplot method [13]. A boxplot provides a visual summary of key statistics

for a sample dataset, including the 25th and 75th percentiles, maximum and minimum values, as well as the median of the distribution. Outliers, in the context of boxplots, are defined as data points that fall outside the interquartile range (IQR) multiplied by a constant, k_{iqr} . The interquartile range is the difference between the 25th and 75th percentiles, with outliers identified as those data points beyond the range of 25th percentile \geq outlier \geq 75th percentile. The constant value essentially determines the range beyond which any data point is considered an outlier. A larger value for the constant will result in fewer data points being classified as outliers, and vice versa. Therefore, the choice of k_{iqr} is crucial, as it allows for the elimination of the maximum number of outliers without negatively affecting the information contained within each distribution.

The trade-off associated with k_{iqr} and number of observations considered by the model is illustrated in Fig. 4, where the value of k_{iqr} influences the number of outliers identified by the boxplot method. It is evident that a low k_{iqr} value is undesirable, as it identifies large portions of the data as outliers and eliminates them from the dataset. A reasonably moderate to high k_{iqr} value is needed for efficient identification and removal of outliers. Therefore, k_{iqr} is treated as a hyperparameter for our model, and its value will be determined through hyperparameter optimization.

Feature Subset Selection

Feature subset selection is the process of choosing a subset of relevant features (variables, predictors) from a larger set to build a model. The objective is to enhance the model's performance by reducing overfitting, simplifying the model, and potentially improving interpretability. In many real-world datasets, numerous features may be present, but not all

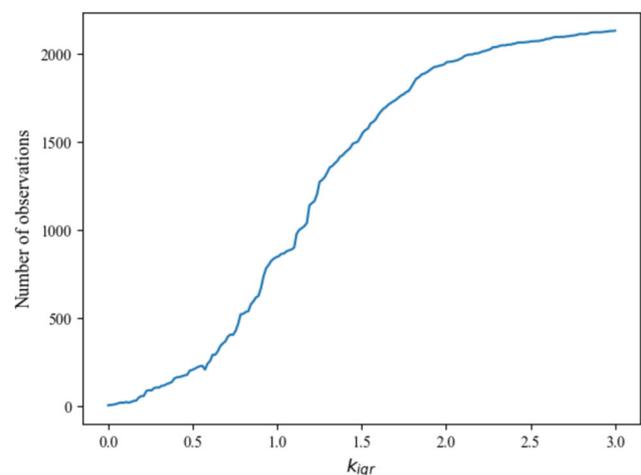


Fig. 4 Variation of number of outliers with k_{iqr}

Table 1 List of hyperparameters and their respective ranges

Hyperparameter	Range
Learning rate	$[10^{-4}, 10^{-1}]$
Batch size	[5, 200]
Number of epochs	[10, 1000]
Number of nodes	[5, 50]
Dropout rate	[0, 1]
k_{iqr}	[1.16, 3]
Feature subset	[5, 10]

contribute equally to the predictive capability of the model. Some features may be redundant or irrelevant, adding noise to the model or leading to overfitting. Feature subset selection methods aim to identify and select the most informative and relevant features, while discarding the less useful ones. This process is crucial for building efficient and effective machine learning models, particularly when dealing with high-dimensional data.

Many feature subset selection methods have been utilized in the past, such as correlation-based filtration [63], wrapper methods [48], and embedded methods like LASSO [38]. Each method has its own merits and demerits. However, feature subset selection has not yet been incorporated within the framework of hyperparameter optimization. Integrating feature selection with hyperparameter optimization allows for the simultaneous selection of optimum features and suitable network hyperparameters, removing extra computational overhead and optimizing the process to maximize testing accuracy.

A comprehensive list of hyperparameters considered in this study, along with their respective ranges, is shown in Table 1. The range of the feature subset indicates the lowest and highest number of possible features that can be considered by the algorithm, which in this case is 5 and 10.

Experimental Results

Data Statistics

Descriptive statistics are a valuable tool for summarizing and describing the main features of a dataset, providing a concise overview that facilitates easy interpretation and understanding. The descriptive statistics for the features considered in this study are listed in Table 2, where the statistical parameters include mean, median, mode, range, variance, standard deviation, skewness, and kurtosis. Mean, median, and mode are used to describe central tendency, providing information about the center and average of a feature. Range, variance, and standard deviation describe the variability of features, helping to understand how much the data points deviate from the center. Skewness and kurtosis describe the distribution of data, offering insights into the symmetry and shape of the distribution.

From Table 2, it is evident that ΔH_{mix} and \bar{r} have high positive skewness, which can be observed by the fact that the majority of binary H_{mix} values lie within range $[-50, 50]$, but the long positive tail of the distribution indicates the presence of alloys with large positive ΔH_{mix} . The long positive tail of \bar{r} can be explained by the presence of alloys containing elements with large metallic radii, such as potassium, rubidium, and cesium. Features with low skewness, like ΔS , indicate a more centralized distribution with low asymmetry. This can be attributed to the presence of both low-entropy conventional alloys and high-entropy MPEAs, which balance the distribution on either side of the median.

Features with high positive kurtosis, such as ΔH_{mix} , are likely to generate a large number of outliers, whereas platykurtic features with low negative kurtosis are less likely to contain outliers, as their distribution is closer to a thin-tailed normal distribution, as seen with n and ΔS .

Table 2 Descriptive statistics values of features in our dataset

Feature	Mean	Median	Mode	Range	Variance	Standard deviation	Skewness	Kurtosis
n	3.71	4	2	8	3.13	1.77	0.37	-1.19
\bar{Z}	97.31	32.48	27	78.25	236.55	15.38	0.92	0.06
\bar{m}_a	8.7×10^1	7.39×10^1	3.22×10^1	2.17×10^2	1.65×10^3	4.06×10^1	9.88×10^{-1}	1.58×10^{-1}
\bar{r}	1.41	1.36	1.29	1.45	2.86×10^{-2}	1.69×10^{-1}	2.18	6.79
VEC	6.65	7.0	8	13	4.31	2.08	-1.67×10^{-1}	-2.35×10^{-1}
$\Delta\chi$	1.74	1.76	2.2	1.7	7.83×10^{-2}	2.8×10^{-1}	-1.81×10^{-1}	4.91×10^{-1}
δ	6.27×10^{-2}	5.16×10^{-2}	4.34×10^{-3}	3.36×10^{-1}	2.70×10^{-3}	5.19×10^{-2}	1.62	3.32
T_{melt}	1.88×10^3	1.82×10^3	1.12×10^3	3.37×10^3	3.54×10^5	5.95×10^2	1.57×10^{-1}	4.63×10^{-1}
ΔS	9.05	9.13	5.76	1.83×10^1	1.83×10^1	4.27	5.8×10^{-2}	-1.49
ΔH_{mix}	-6.33	-5.89	0	2.07×10^2	3.33×10^2	1.82×10^1	2.51	1.6×10^1

As depicted in Fig. 1, the imbalance present in the dataset could severely affect the distribution and sampling of data in the training and testing sets. To address this and prevent skewed sampling, stratified sampling is utilized when separating the data into training and testing sets. This ensures that the proportions of each class are preserved in both sets and that every minor class is represented in the testing set—something that might not occur with normal splitting, as the minority class is less likely to appear in both sets. A sample distribution illustrating the stratified sampling of the training and testing sets is shown in Fig. 5. It can be observed that the class distribution is identical in both sets, demonstrating the effectiveness of stratified sampling for an imbalanced dataset. In this case, the training–testing split is 70:30.

Data Pre-Processing

As mentioned previously in “Background” section, there are ten features associated with each alloy in the dataset. To avoid unnecessary influence among different features, it is necessary to normalize each instance across all features so that they can be treated equally during the training process. For this reason, Min-Max normalization is applied as part of the pre-processing step. Min-Max normalization is defined as

$$X_{N_i} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \tag{4}$$

where X_{N_i} denotes i^{th} normalized data point of variable X ; X_{\min} and X_{\max} represent minimum and maximum value of X , respectively.

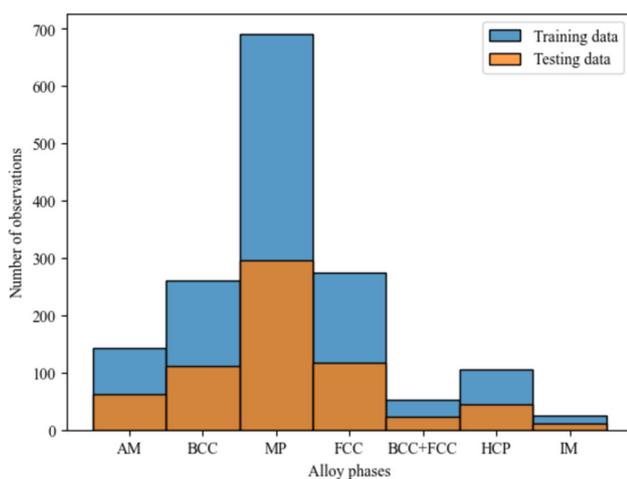


Fig. 5 Distribution of classes within training and testing sets when using stratified sampling

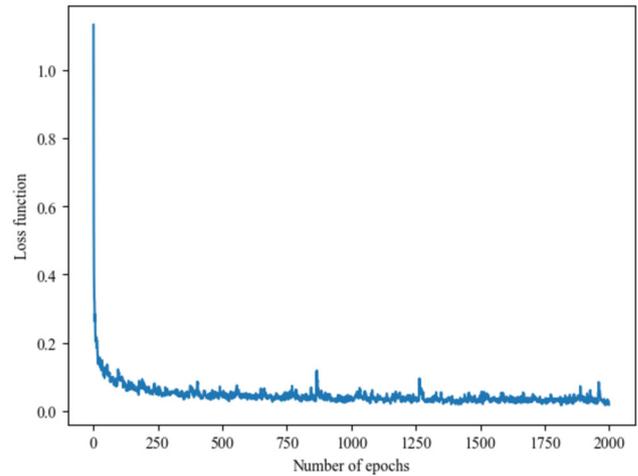


Fig. 6 Training process of ANN, depicting evolution of loss function with number of epochs

Hyperparameter Optimization

The CPU used in these experiments is Intel Core i7 - 11700 with 16GB RAM. The programming platform used here is Python 3.12.2 with Jupyter Notebook as IDE, for easy execution of algorithm and visualization of results. Pytorch framework is utilized for generating ANNs.

To ensure that the ANN framework is functioning as expected, a network is trained with random hyperparameters, and its training behavior is visualized in Fig. 6. As expected, the training loss decreases rapidly during the initial phase of training and eventually reaches saturation, with oscillations becoming evident. At this point, further epochs are unlikely to improve the ANN’s performance and may even contribute to overfitting. This is why the number of epochs is included as a hyperparameter, with its selection aimed at maximizing testing accuracy.

For the implementation of hyperparameter optimization via GA, the population size is initially set to 20, and the number of generations to 5, with elitism set to 2. The results of fitness evolution across generations are depicted in Fig. 7. Fitness initially decreases rapidly and then begins to oscillate towards the end, suggesting the presence of local minima. Since the population size and number of generations are relatively small, the algorithm does not sample a sufficiently large search space and is more likely to become stuck in a local minimum.

To enhance the capabilities of hyperparameter optimization, the experiment is rerun with a population size of 40, number of generations set to 20, and elitism set to 3. The results of fitness evaluation across each successive generation of GA with the updated parameters are illustrated in

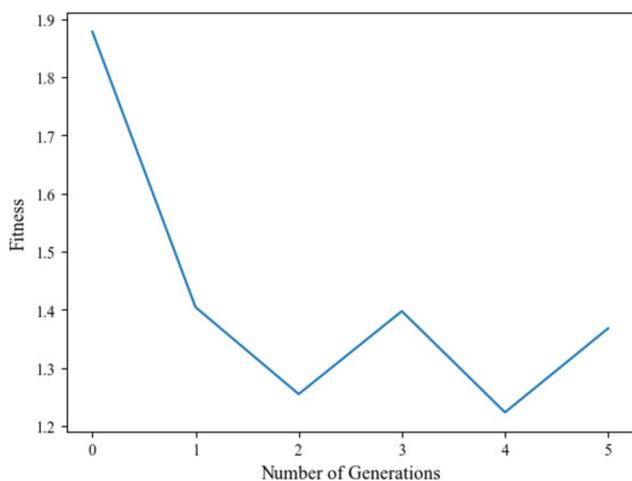


Fig. 7 Fitness plot of GA with respect to number of generations. Here, population size = 20, number of generations = 5, and elitism = 2

Fig. 8. While the fitness appears to stabilize beyond the second generation, this is not the case, as demonstrated by the zoomed-in view enclosed within the plot. This view shows that fitness continues to improve, though at a slower rate as it approaches saturation. It can also be observed that a larger population size results in a higher initial fitness value, reflecting the more diverse sampling of the search space. This diversity is crucial for obtaining high-quality hyperparameters and will contribute to a phase prediction model with high accuracy, which is the goal of this study.

To avoid contingency and ensure fair results, 10 experiments are run with population size = 40, number of generations = 20, and elitism = 4, where each run has a different random seed. The best accuracy obtained from these experiments is

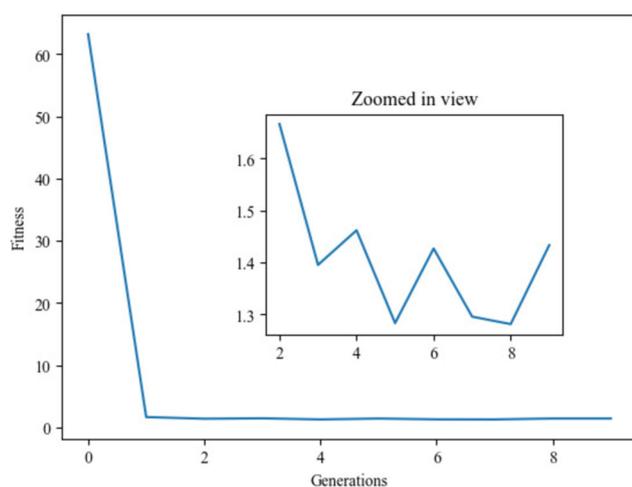


Fig. 8 Fitness plot of GA with respect to number of generations, with a zoomed-in view enclosed within the plot depicting change in fitness close to saturation. Here, population size = 40, number of generations = 20, and elitism = 3

Table 3 List of hyperparameters with the best testing accuracy

Hyperparameter	Range
Learning rate	1.3036×10^{-2}
Batch size	1.89×10^2
Number of epochs	6.09×10^2
Number of nodes	44
Dropout rate	9.5782×10^{-2}
k_{iqr}	2.5989
Feature subset	$[\bar{Z}, \bar{r}, \text{VEC}, \Delta\chi, \delta, T_{melt}, \Delta S, \Delta H_{mix}]$

a fitness value of 1.1320 which equates to testing accuracy of 88.34% via $\text{fit} = 1/T_{acc}$. The resulting set of hyperparameters obtained from experiment with the best fitness is described in Table 3. Upon examining the selected feature subset, it becomes evident that out of the ten features, only eight are chosen by the algorithm. This indicates that the remaining two features are either considered redundant or their contribution to testing accuracy is minimal compared to the other features in the universal set. Incorporating feature subset selection within the framework of hyperparameter optimization ensures that the most relevant features are selected, maximizing the performance of phase prediction and accurately representing the search space of the MPEA phase dataset.

Analysis of Results

To analyze the prediction quality of the network, a confusion matrix is utilized, as illustrated in Fig. 9. This matrix shows the correct and incorrect predictions for individuals in the testing set. The AM and IM classes exhibit the best prediction quality, with low false positives and false negatives. This

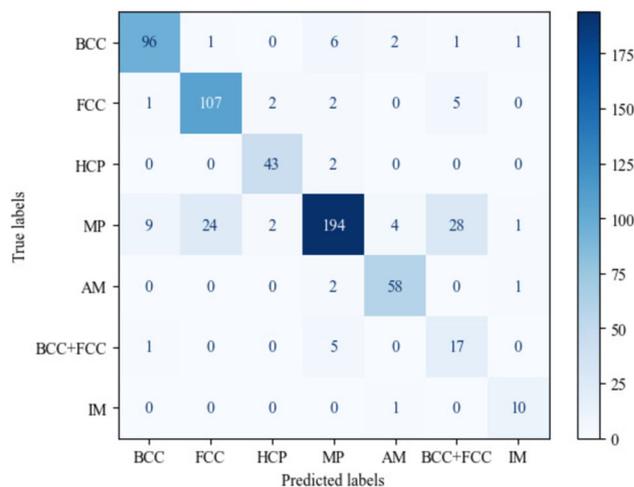


Fig. 9 Confusion matrix of testing data

Table 4 Prediction results of new alloys for validation of model

Alloy	Actual	Predicted
Fe _{4.5} Co ₃ CrVMn _{0.5} [61]	SS	SS
CoCu ₂ Mn ₃ Ni ₄ [22]	SS	SS
Al _{0.75} Cu _{3.0833} Fe _{3.0833} Mn _{3.0833} [37]	SS	SS
Cu _{3.33} Fe _{3.33} Mn _{3.33} [37]	SS	SS
Co _{1.75} Cr _{1.25} Fe _{5.5} NiMo _{0.3} C _{0.2} [27]	IM	IM

is surprising, as IM is the smallest minority class, indicating that the current network, with hyperparameters from Table 3, is well suited for minority classes. However, the BCC+FCC class has the worst prediction quality, with 13% false positives and 50% false negatives, where the network incorrectly labels MP as BCC+FCC 35% of the time. This may be due to the fact that BCC+FCC can itself be considered a mixed phase, especially when phases other than BCC and FCC are present in the microstructure. However, caution must be exercised when considering the removal of a class, as the aim is to generate an accurate depiction of the phase search space, and oversimplification could lead to an unphysical representation of the physical systems.

Validation of Prediction Model

To validate the phase prediction framework, the resulting ANN is used to predict the phases of additional MPEA alloys reported in recent studies [22, 27, 37, 61], which were not included in our dataset. The prediction results are listed in Table 4. It is observed that the model successfully estimates the phases for all five validation MPEAs, demonstrating that the approach developed in the current study is a viable strategy for phase prediction of MPEAs.

To further validate the approach chosen in this study, the best results of our algorithm are compared with several commonly utilized classification algorithms. Table 5 lists the outcomes of these comparisons, including the name of the technique and the corresponding best testing accuracy for 10 independent runs. It is evident that our model,

Table 5 Comparison among various classification algorithms

Technique	Best accuracy
ANN with HPO and additional hyperparameters	0.84
Support vector machines	0.68
Decision tree	0.75
Random forest	0.71
XGBoost classifier	0.79
Naive Bayes	0.59

which employs an ANN with hyperparameter optimization incorporating outlier detection and feature subset selection as additional hyperparameters, achieves the best prediction accuracy when compared to five widely used multi-class classification techniques applied to MPEA phase prediction. This comparison highlights that conventional classification techniques are not well suited to handle the complex and high-dimensional MPEA data, which contains outliers and class imbalances.

Ablation Studies

In the context of machine learning, ablation studies refer to the scientific examination of a machine learning system by removing its components to gain insight into its performance. These studies are useful for illustrating the importance of key building blocks in an ANN model. For this purpose, ablation studies are conducted to highlight the significance of the novel additional parameters introduced in our study, namely outlier detection and feature subset selection.

To perform ablation studies, the performance of the model is compared with and without these key additional hyperparameters, as well as between the two parameters. Each comparison consists of 20 runs, and the results of the mean testing accuracy are presented in Table 6. Here, *O* denotes outlier detection hyperparameter, and *F* denotes feature subset selection hyperparameter.

From Table 6, the performance of the model can be assessed based on the comparison between the presence and absence of *O* (outlier detection) and *F* (feature subset selection). Here, the absence of *O* and *F* indicates that all other five hyperparameters are still present. In the first column, it is evident that the model performs significantly better when the *O* hyperparameter is added to the set of hyperparameters, supporting the initial hypothesis that including outlier detection enhances the quality of data in the MPEA dataset.

The second comparison is somewhat surprising, as it shows only a slight improvement when adding *F* to the list of hyperparameters. The third comparison, between the presence of only *O* and only *F*, indicates that the inclusion of outlier detection is far superior in terms of testing accuracy compared to feature subset selection. The underwhelming performance of feature subset selection suggests the need

Table 6 Comparison results of ablation studies

Setting 1	Accuracy	Setting 2	Accuracy
Without <i>O</i> and <i>F</i>	0.73	With only <i>O</i>	0.84
Without <i>O</i> and <i>F</i>	0.73	With only <i>F</i>	0.76
With only <i>O</i>	0.84	With only <i>F</i>	0.76
Without <i>O</i> and <i>F</i>	0.73	With both <i>O</i> and <i>F</i>	0.85

for further analysis of the physical representation of all features in the dataset. The addition of other chemical properties as features in future studies may provide more insight into the importance of feature selection in phase prediction of MPEAs.

The fourth and final comparison, with and without both O and F , further solidifies the success of our approach in using additional hyperparameters, which lead to a significant improvement in the mean testing accuracy. Combining the results of the ablation studies and the validation using new alloys in Table 4, it can be concluded that the model is well suited for predicting the phases of alloys based on their respective compositions and will be valuable for discovering new MPEAs with desirable properties.

Conclusion

In this paper, a hyperparameter optimization approach for ANNs has been proposed for phase prediction of MPEAs. Specifically, outlier detection and feature subset selection have been incorporated as novel additions to the hyperparameter optimization process, with GA employed as the optimization algorithm. To make GA suitable for feature subset selection, a new strategy has been developed to handle variable-length chromosomes, where a single gene can have a variable length. Experimental results have demonstrated the viability of this approach as a phase prediction framework. The analysis of GA results has illustrated the successful optimization of both real-valued variables and dynamically sized categorical arrays. To validate the current approach, the best obtained model has been used to successfully predict the phases of new MPEAs not included in the dataset. Ablation studies conducted on the model have indicated that outlier detection outperforms feature subset selection, and the combination of both has significantly enhanced prediction accuracy, which will contribute to the high-throughput development of MPEAs by assisting experimental design through data science and providing a deeper understanding of their physical characteristics.

Future work can be summarized as follows: (1) considering additional chemical and mechanical features beyond Hume-Rothery rules for phase prediction of MPEAs [32, 59]; (2) employing different selection, crossover, and mutation strategies and evolutionary computation methods for hyperparameter optimization [7, 46, 60]; (3) designing a similar strategy for supervised regression to predict mechanical properties of MPEAs [20, 34, 64]; (4) creating a framework for multi-objective optimization, where ANN phase prediction is utilized as a search space to identify alloys with desirable properties [10, 14, 51, 54]; (5) applying the proposed hyperparameter optimization approach to other material science and data analysis applications [52, 53, 57].

Acknowledgements Syed Hassan Fatimi, Zidong Wang, Weibo Liu, and Xiaohui Liu are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. Isaac T. H. Chang is with Brunel Centre for Advanced Solidification Technology (BCAST), Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom.

Author Contributions Syed Hassan Fatimi: conceptualization, formal analysis, investigation, methodology, software, visualization, writing—original draft. Zidong Wang: conceptualization, formal analysis, methodology, resources, supervision, writing, review and editing. Isaac T. H. Chang: conceptualization, resources, supervision, validation, review and editing. Weibo Liu: formal analysis, methodology, writing, original draft, writing, review and editing. Xiaohui Liu: conceptualization, methodology, supervision, writing, review and editing.

Data Availability The data that support the findings of this study are not openly available due to data privacy and are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alibrahim H Ludwig SA. Hyperparameter optimization: comparing genetic algorithm against grid search and Bayesian optimization, In: Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 2021. pp. 1551–1559.
- Bergstra J, Yamins D, Cox D. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures, In: Proceedings of the International Conference on Machine Learning, Atlanta, USA, 2013. pp. 115–123.
- Cahn RW, Haasen P. Physical metallurgy, Elsevier, 1996.
- Cantor B. Multicomponent and high entropy alloys. *Entropy*. 2014;16(9):4749–68.
- Cantor B, Chang ITH, Knight P, Vincent AJB. Microstructural development in equiatomic multicomponent alloys. *Mater Sci Eng A*. 2004;375:213–8.
- Chen H, Chen Q, Shen B, Liu Y. Parameter learning of probabilistic Boolean control networks with input-output data. *Int J Netw Dyn Intell*. 2024;3(1):100005.
- Dai D, Li J, Song Y, Yang F. Event-based recursive filtering for nonlinear bias-corrupted systems with amplify-and-forward relays. *Syst Sci Control Eng*. 2024;12(1):2332419.

8. Erden C. Genetic algorithm-based hyperparameter optimization of deep learning models for PM2.5 time-series prediction. *Int J Environ Sci Technol.* 2023;20(3):2959–82.
9. Fang S, Xiao X, Xia L, Li W, Dong Y. Relationship between the widths of supercooled liquid regions and bond parameters of mg-based bulk metallic glasses. *J Non-Cryst Solids.* 2003;321(1–2):120–5.
10. Fang W, Shen B, Pan A, Zou L, Song B. A cooperative stochastic configuration network based on differential evolutionary sparrow search algorithm for prediction. *Syst Sci Control Eng.* 2024;12(1):2314481.
11. Ferro R, Cacciamani G, Borzone G. Remarks about data reliability in experimental and computational alloy thermochemistry. *Intermetallics.* 2003;11(11–12):1081–94.
12. Feurer M, Hutter F. *Automated machine learning: methods, systems, challenges*, Springer Nature, 2019.
13. Frigge M, Hoaglin DC, Iglewicz B. Some implementations of the boxplot. *Am Stat.* 1989;43(1):50–4.
14. Gao P, Jia C, Zhou A. Encryption-decryption-based state estimation for nonlinear complex networks subject to coupled perturbation. *Syst Sci Control Eng.* 2024;12(1):2357796.
15. Gao MC, Yeh J-W, Liaw PK, Zhang Y. *High-entropy alloys: fundamentals and applications*, Springer, 2016.
16. Gludovatz B, Hohenwarter A, Thurston KV, Bei H, Wu Z, George EP, Ritchie RO. Exceptional damage-tolerance of a medium-entropy alloy CrCoNi at cryogenic temperatures. *Nat Commun.* 2016;7(1):10602.
17. Guo S. Phase selection rules for cast high entropy alloys: an overview. *Mater Sci Technol.* 2015;31(10):1223–30.
18. Hinton GE, Srivastava N., Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. [arXiv:1207.0580](https://arxiv.org/abs/1207.0580). 2012.
19. Hu S, Lu J, Zhou S. Learning regression distribution: information diffusion from template to search for visual object tracking. *Int J Netw Dyn Intell.* 2024;3(1):100006.
20. Jin F, Ma L, Zhao C, Liu Q. State estimation in networked control systems with a real-time transport protocol. *Syst Sci Control Eng.* 2024;12(1):2347885.
21. Kaufmann K, Vecchio KS. Searching for high entropy alloys: a machine learning approach. *Acta Mater.* 2020;198:178–222.
22. Kim DG, Jo YH, Park JM, Choi W-M, Kim HS, Lee B-J, Sohn SS, Lee S. Effects of annealing temperature on microstructures and tensile properties of a single FCC phase CoCuMnNi high-entropy alloy. *J Alloys Compounds.* 2020;812:152111.
23. Kim IY, de Weck O. Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Struct Multidisc Optim.* 2005;29:445–56.
24. Kim IY, de Weck O. Variable chromosome length genetic algorithm for structural topology design optimization. In: *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, California, USA, 2004.
25. King DJM, Middleburgh SC, McGregor AG, Cortie MB. Predicting the formation and stability of single phase high-entropy alloys. *Acta Mater.* 2016;104:172–9.
26. Kingma DP, Adam JB. A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980). 2014.
27. Kwon H, Moon J, Bae JW, Park JM, Son S, Do H-S, Lee B-J, Kim HS. Precipitation-driven metastability engineering of carbon-doped CoCrFeNiMo medium-entropy alloys at cryogenic temperature. *Scripta Mater.* 2020;188:140–5.
28. Lee K, Ayyasamy MV, Delsa P, Hartnett TQ, Balachandran PV. Phase classification of multi-principal element alloys via interpretable machine learning. *NPJ Comput Mater.* 2022;8(1):25.
29. Lee SY, Byeon S, Kim HS, Jin H, Lee S. Deep learning-based phase prediction of high-entropy alloys: optimization, generation, and explanation. *Mater Des.* 2021;197:109260.
30. Lee S, Kim J, Kang H, Kang D-Y, Park J. Genetic algorithm based deep learning neural network structure and hyperparameter optimization. *Appl Sci.* 2021;11(2):744.
31. Li B, Li W. Distillation-based user selection for heterogeneous federated learning. *Int J Netw Dyn Intell.* 2024;3(2):100007.
32. Li J, Suo Y, Chai S, Xu Y, Xia Y. Resilient and event-triggered control of singular Markov jump systems against cyber attacks. *Int J Syst Sci.* 2024;55(2):222–36.
33. Ma C, Cheng P, Cai C. Localization and mapping method based on multimodal information fusion and deep learning for dynamic object removal. *Int J Netw Dyn Intell.* 2024;3(2):100008.
34. Ma G, Wang Z, Liu W, Fang J, Zhang Y, Ding H, Yuan Y. Estimating the state of health for lithium-ion batteries: a particle swarm optimization-assisted deep domain adaptation approach. *IEEE/CAA J Autom Sinica.* 2023;10(7):1530–43.
35. Melis G, Dyer C, Blunsom P. On the state of the art of evaluation in neural language models. [arXiv:1707.05589](https://arxiv.org/abs/1707.05589). 2017.
36. Miracle DB, Senkov ON. A critical review of high entropy alloys and related concepts. *Acta Mater.* 2017;122:448–511.
37. Moon J, Park JM, Bae JW, Do H-S, Lee B-J, Kim HS. A new strategy for designing immiscible medium-entropy alloys with excellent tensile properties. *Acta Mater.* 2020;193:71–82.
38. Muthukrishnan R, Rohini R. Lasso: a feature selection technique in predictive modeling for machine learning. In: *Proceedings of the 2016 IEEE international conference on advances in computer applications (ICACA)*, Coimbatore, India, 2016. pp. 18–20.
39. Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, Haifa, Israel, 2010. pp. 807–814.
40. Nikbakht S, Anitescu C, Rabczuk T. Optimizing the neural network hyperparameters utilizing genetic algorithm. *J Zhejiang Univ Sci A.* 2021;22(6):407–26.
41. Oñate A, Sanhueza JP, Zegpi D, Tuninetti V, Ramirez J, Medina C, Melendrez M, Rajas D. Supervised machine learning-based multi-class phase prediction in high-entropy alloys using robust databases. *J Alloys Compounds.* 2023;962:171224.
42. Ouyang G, Singh P, Johnson DD, Kramer MJ, Perepezko JH, Senkov ON, Miracle D, Cui J. Design of refractory multi-principal-element alloys for high-temperature applications. *Comput Mater.* 2023;9(1):141.
43. Pei Z, Yin J, Hawk JA, Alman DE, Gao MC. Machine-learning informed prediction of high-entropy solid solution formation: beyond the Hume-Rothery rules. *NPJ Comput Mater.* 2020;6:50.
44. Reunanen N, Rätty T, Lintonen T. Automatic optimization of outlier detection ensembles using a limited number of outlier examples. *Int J Data Sci Anal.* 2020;10(4):377–94.
45. Sheng G, Liu CT. Phase stability in high entropy alloys: formation of solid-solution phase or amorphous phase. *Prog Nat Sci Mater Int.* 2011;21(6):433–46.
46. Sheng M, Ding W, Sheng W. Differential evolution with adaptive niching and reinitialisation for nonlinear equation systems. *Int J Syst Sci.* 2024;55(10):2172–86.
47. Takeuchi A, Inoue A. Calculations of mixing enthalpy and mismatch enthalpy for ternary amorphous alloys. *Mater Trans, JIM.* 2000;41(11):1372–8.
48. Talavera L. An evaluation of filter and wrapper methods for feature selection in categorical clustering. In: *Proceedings of the International Symposium on Intelligent Data Analysis*, Madrid, Spain, 2005. pp. 440–451.
49. Tsai M-H, Tsai R-C, Chang T, Huang W-F. Intermetallic phases in high-entropy alloys: statistical analysis of their prevalence and structural inheritance. *Metals.* 2019;9(2):247.
50. Tsai M-H, Yeh J-W. High-entropy alloys: a critical review. *Mater Res Lett.* 2014;2(3):107–23.

51. Wang D, Wen C, Feng X. Deep variational Luenberger-type observer with dynamic objects channel-attention for stochastic video prediction. *Int J Syst Sci.* 2024;55(4):728–40.
52. Wang L, Sun W, Pei H. Nonlinear disturbance observer-based geometric attitude fault-tolerant control of quadrotors. *Int J Syst Sci.* 2024;55(11):2337–48.
53. Wang Y, Wen C, Wu X. Fault detection and isolation of floating wind turbine pitch system based on Kalman filter and multi-attention 1DCNN. *Syst Sci Control Eng.* 2024;12(1):2362169.
54. Wang W, Ma L, Rui Q, Gao C. A survey on privacy-preserving control and filtering of networked control systems. *Int J Syst Sci.* 2024;55(11):2269–88.
55. Ward L, Agrawal A, Choudhary A, Wolverson C. A general-purpose machine learning framework for predicting properties of inorganic materials. *NPJ Comput Mater.* 2016;2(1):1–7.
56. Wei J, Chu X, Sun X-Y, Xu K, Deng H-X, Chen J, Wei Z, Lei M. Machine learning in materials science. *InfoMat.* 2019;1(3):338–58.
57. Wu Y, Huang X, Tian Z, Yan X, Yu H. Emotion contagion model for dynamical crowd path planning. *Int J Netw Dyn Intell.* 2024;3(3):100014.
58. Xiao X, Yan M, Basodi S, Ji C, Pan Y. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. [arXiv:2006.12703](https://arxiv.org/abs/2006.12703). 2020.
59. Xue J, Shen B. A survey on sparrow search algorithms and their applications. *Int J Syst Sci.* 2024;55(4):814–32.
60. Xue Y, Li M, Arabnejad H, Suleimenova D, Jahani A, Geiger BC, Boesjes F, Anagnostou A, Taylor SJE, Liu X, Groen D. Many-objective simulation optimization for camp location problems in humanitarian logistics. *Int J Netw Dyn Intell.* 2024;3(3):100017.
61. Yang J, Jo YH, Kim DW, Choi W-M, Kim HS, Lee B-J, Sohn SS, Lee S. Effects of transformation-induced plasticity (TRIP) on tensile property improvement of Fe₄₅Co₃₀Cr₁₀V₁₀Ni₅-xMn_x high-entropy alloys. *Mater Sci Eng A.* 2020;772:138809.
62. Yeh J-W, Chen S-K, Lin S-J, Gan J-Y, Chin T-S, Shun T-T, Tsau C-H, Chang S-Y. Nanostructured high-entropy alloys with multiple principal elements: novel alloy design concepts and outcomes. *Adv Eng Mater.* 2004;6(5):299–303.
63. Yu L, Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, Washington DC, USA, 2003. pp. 856–863.
64. Zhang R, Liu H, Liu Y, Tan H. Dynamic event-triggered state estimation for discrete-time delayed switched neural networks with constrained bit rate. *Syst Sci Control Eng.* 2024;12(1):2334304.
65. Zhang Y, Zhou YJ, Lin JP, Chen GL, Liaw PK. Solid-solution phase formation rules for multi-component alloys. *Adv Eng Mater.* 2008;10(6):534–8.
66. Zhang W, Fan Y, Song Y, Tang K, Li B. A generalized two-stage tensor denoising method based on the prior of the noise location and rank. *Expert Syst Appl.* 2024;255:124809.
67. Zhou Z, Zhou Y, He Q, Ding Z, Li F, Yang Y. Machine learning guided appraisal and exploration of phase design for high entropy alloys. *NPJ Comput Mater.* 2019;5(1):128.
68. Zhu J, Liaw P, Liu C. Effect of electron concentration on the phase stability of NbCr₂-based Laves phase alloys. *Mater Sci Eng A.* 1997;239:260–4.

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