


## Research

# Cognitive pairwise comparison forward feature selection with deep learning for astronomical object classification with sloan digital sky survey

Kevin Kam Fung Yuen<sup>1</sup> 

Received: 11 July 2023 / Accepted: 15 May 2024

Published online: 21 May 2024

© The Author(s) 2024 

## Abstract

This paper proposes a hybrid approach integrating the expert knowledge judgment approach using the Cognitive Pairwise Comparison (CPC) to the Deep Learning, a modern classification approach, for astronomic object classification. The astronomic data with ten thousand samples retrieved from Sloan Digital Sky Survey Sky Server Data Release 15 (SDSS SkyServer DR 15) are used for this study. The CPC is an approach to elicit and encode expert knowledge in the format of a Pairwise Opposite Matrix (POM) to evaluate expert preferences for the features. A forward feature selection algorithm taking the expert choices using CPC for the ordered features is used for the feature selection for the deep learning algorithm to build a heuristic training model based on the astronomic data. Whilst the accuracy of the case of improper feature selection is just 37.1%, the proposed hybrid approach can obtain a very high accuracy of 97.9% for the classification of the astronomic object using the eight scaled features ( $u, g, r, i, z$  redshift,  $ra, dec$ ). To extend this research, the proposed CPC can be used as a human-centered tool to be applied to other areas of data sciences.

**Keywords** Human-centered artificial intelligence · Human-centered feature engineering · Deep learning · Astronomic object classification

## 1 Introduction

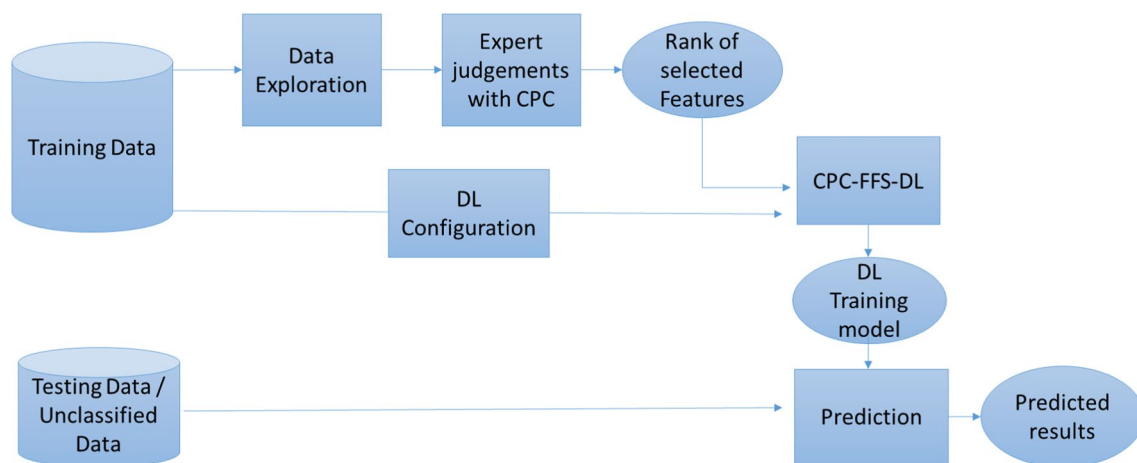
The Sloan Digital Sky Survey (SDSS) provides vast information about the universe regarding the three-dimensional maps, deep multi-colour images of one-third of the sky, spectra for more than three million astronomical objects, and exploration of all phases and surveys in the past, present, and future [9]. Data Release 15 is the third data release of the fourth phase of the Sloan Digital Sky Survey and includes SDSS data taken through June 2017 [10]. Whilst different forms of questions about the universe can be queried from the vast SDSS database, the scope of this paper is to propose the Cognitive Pairwise Comparison Forward Feature Selection (CPC-FFS) as the expert knowledge judgment approach with the FFS algorithm to rank and select features, and the Deep Learning (DL) is applied for model training for astronomical object classification.

Machine learning techniques are increasingly used in SDSS data. Feigelson and Babu [4] demonstrated various statistical and machine learning methods for astronomy study with  $R$  applications. Bazarghan and Gupta [2] described the use of a Probabilistic Neural Network (PNN) for the automatic classification of about 5000 SDSS spectra into 158

---

✉ Kevin Kam Fung Yuen, kevinkf.yuen@gmail.com | <sup>1</sup>Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China.





**Fig. 1** Overview of hybrid CPC-FFS-DL framework

spectral types of a reference library ranging from *O* type to *M* type stars. Buisson et al. [3] described the use of the principal component analysis for feature extraction, and the uses of Random Forest, k-Nearest Neighbours, SkyNet, Support Vector Machine (SVM), Minimum Error Classification (MEC), and Naïve Bayes (NB) to predict whether an object is real on the basis of the dataset of 27,480 objects for each object consisting of three  $51 \times 51$ -pixel difference images (for each image including the *g*, *r* and *i* colour bands). Kim and Brunner [6] presented the use of a convolutional neural network for the classification of star-galaxy based on five photometric bands: *u*, *g*, *r*, *i*, and *z*. It appears that there is a lack of research using human-centered methods that the human decision-making methods are incorporated into the machine learning methods, as the experts of users and designers play significant roles in data modelling during the human-in-the-loop processes for data science projects.

As different combinations of features lead to different performance of machine learning methods, this study utilizes CPC as human centered method to select the features. The cognitive pairwise comparison (CPC) is the core technique of PCNP [12–14] to determine the priorities of the feature choices based on experts' knowledge. The PCNP method is proposed to address the problems of the Analytic Hierarchy Process (AHP)'s paired ratio scale [8], potentially producing misapplications, and the use of the AHP is controversial [7]. The overview of the proposed framework of the Cognitive Pairwise Comparison Forward Feature Selection with Deep Learning (CPC-FFS-DL) approach is presented in Fig. 1. The general steps are described in the rest of this paper and organized below.

1. Data preparation: the data obtained from different tables from SDSS are divided into two parts: training data and test data, which are used to build training models and test the training models previously constructed. The details are presented in Sect. 2.1.
2. Data exploration: whilst insufficient features can produce an under-fit model, irrelevant features can create bias for the model leading to low accuracy. The relevant features must be selected to construct a learning model after the data is explored and visualized. Some outliers may be identified and removed. The details are discussed in Sect. 2.2.
3. CPC process: whilst the motivation for using CPC is presented in Sect. 3.1, the details of CPC are offered in Sect. 3.2. To elicit expert knowledge, the experts conduct a CPC evaluation, in which the CPC scores are stored in the form of a Pairwise Opposite Matrix (POM) to produce the ranks of the features.
4. DL Configuration process: parameters setting is a complex process in DL. After several trials, the good enough setting is proposed to simplify the illustration of the use of CPC-FFS-DL. The details of the setting for DL are illustrated in Sect. 3.3.
5. CPC-FFS-DL process: after the POM and DL configurations are established, The CPC-FFS-DL algorithm, applying the deep learning algorithm to proceed with the data by evaluating additional features based on the CPC ranks, is used to produce a promising DL training model. The details are discussed in Sect. 3.4.
6. Prediction process: the testing data, i.e., data without target or class labels, are taken for the DL training model to produce the predicted results. If the testing data are with ground truth values (or target labels), the prediction accuracy can be calculated. The details are discussed in Sect. 4.
7. The conclusions and prospects of the proposed approach are presented in Sect. 5.

---

```
sqlQuerySTAR = "SELECT TOP 3500 p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z, p.run, p.rerun, p.camcol, p.field, s.specobjid,
s.class, s.z as redshift, s.plate, s.mjd, s.fiberid FROM PhotoObj AS p JOIN SpecObj AS s ON s.bestobjid = p.objid WHERE
s.class = 'STAR'"

dr15="http://skyserver.sdss.org/dr15/SkyServerWS/SearchTools/SqlSearch?"

csvQuerySTAR = getForm(dr15, cmd = sqlQuerySTAR, format = "csv")
```

---

**Fig. 2** SQL query in R for 'STAR' class

The following sections demonstrate the details in the CPC-FFS-DL framework, which is used for the astronomical object classification problem.

## 2 SDSS astronomical data

### 2.1 SDSS source data

The online platform of Sloan Digital Sky Survey (SDSS) SkyServer DR15 offers a substantial amount of open data for the universe. SDSS provides SQL search on the web<sup>1</sup> to query the dataset. The SQL database structures are documented on the website of the schema browser.<sup>2</sup> Users can refer to the information about API and Tools for Programmer's Reference<sup>3</sup> to use SDSS web services to access the data.

Figure 2 presents the R code for SQL query with SDSS API web services to obtain the data used in this paper. Whilst there are three classes for the target variable, the value of the parameter, *s.class* = 'STAR', is changed to 'GALAXY' and 'QSO' respectively in Fig. 2. The 'STAR' and 'GALAXY' classes have 3500 records respectively, whilst the 'QSO' class has 3000 records. Therefore, the sample data comprises 10,000 records.<sup>4</sup> According to some hints on searching SkyServer,<sup>5</sup> the "TOP <n>" SQL construct is not a deterministic ordering. That means the "TOP <n>" objects may differ if the same query is performed again.

According to the SQL query in Fig. 2, 18 features are selected. The feature definitions<sup>6</sup> are shown in Table 1. The variable, *class*, is the target label class comprising GALAXY, QSO, and STAR. The rest are the features potentially related to the class prediction. Two identity variables, *objid* and *specobjid*, are not used for building a learning model. Therefore, 15 features are used for creating a learning model.

### 2.2 Data exploration and visualization

To understand the 15 features related to the astronomic object classification problem, the statistics summary and data visualization are presented in Table 2 and Figs. 3–4 respectively. Regarding the outliers or missing values, eight observations with five filter bands equal to – 9999 are removed. The descriptive statistics for the selected features are presented in Table 2. The *rerun* feature is not related to the classification problem as it is just a single value, 301.

To understand the GALAXY, QSO, and STAR classes, the coverage of two features *ra* and *dec* of the sample data produced by R *ggplot2* [11] is presented in Fig. 3. The features *ra* and *dec* are related to the label class of astronomic object classification as objects of the same class are likely grouped. The image instances for these three classes are presented in Fig. 4. The images are obtained from the *getjpeg* web method in SkyServer web services.<sup>7</sup> By browsing the images, five filter band features are highly related to the astronomic object classification label class.

<sup>1</sup> SQL Search, <http://skyserver.sdss.org/dr15/en/tools/search/sql.aspx>

<sup>2</sup> Schema browser, <http://skyserver.sdss.org/dr15/en/help/browser/browser.aspx>

<sup>3</sup> API and Tools for Programmer's Reference, <http://skyserver.sdss.org/dr15/en/help/docs/api.aspx>

<sup>4</sup> The data extracted for this paper are available at <https://github.com/kkfuyen/SDSS>

<sup>5</sup> Hints, <http://skyserver.sdss.org/dr15/en/help/docs/realquery.aspx#hints>

<sup>6</sup> Glossary, <https://www.sdss.org/dr15/help/glossary>

<sup>7</sup> An example to find an QSO object by passing *ra* and *dec* parameters to the skyserver 's web method using the link as below: <http://skyserver.sdss.org/dr15/SkyServerWS/ImgCutout/getjpeg?ra=182.47399&dec=-0.49460302&width=50&height=50>

**Table 1** Definitions of data features

<i>ra, dec</i>	J2000 right ascension and declination
<i>u,g,r,i,z</i>	Five filter bands: ultraviolet, green, red, near infrared, infrared
<i>run</i>	A <i>run</i> is the length of a strip observed in a single continuous image observing scan, bounded by lines of $\mu$ and $\nu$ . $\mu$ corresponds to <i>ra</i> , or longitude whilst $\nu$ corresponds to <i>dec</i> , or latitude
<i>rerun</i>	A reprocessing of an imaging run. The underlying imaging data are the same, but the software version and/or calibration are different
<i>camcol</i>	Camera column. A <i>camcol</i> is the output of one camera column of CCDs (each with a different filter) as part of a Run. Therefore, 1 <i>camcol</i> = 1/6 of a Run. It is also a portion of a scanline
<i>field</i>	A field is part of a <i>camcol</i> that is processed by the Photo pipeline at one time
<i>redshift</i>	Final Redshift
<i>plate</i>	It is plate ID. Each spectroscopic exposure employs a large, thin, circular metal plate that positions optical fibers via holes drilled at the locations of the images in the telescope focal plane. These fibers then feed into the spectrographs. Each plate has a unique serial number, called plate in views such as SpecObj in the CAS
<i>mjd</i>	Modified julian date, used to indicate the date that a given piece of SDSS data (image or spectrum) was taken
<i>fiberid</i>	The SDSS spectrograph uses optical fibers to direct the light at the focal plane from individual objects to the <i>slithead</i> . Each object is assigned a corresponding <i>fiberID</i>

**Table 2** Descriptive statistics of sample data

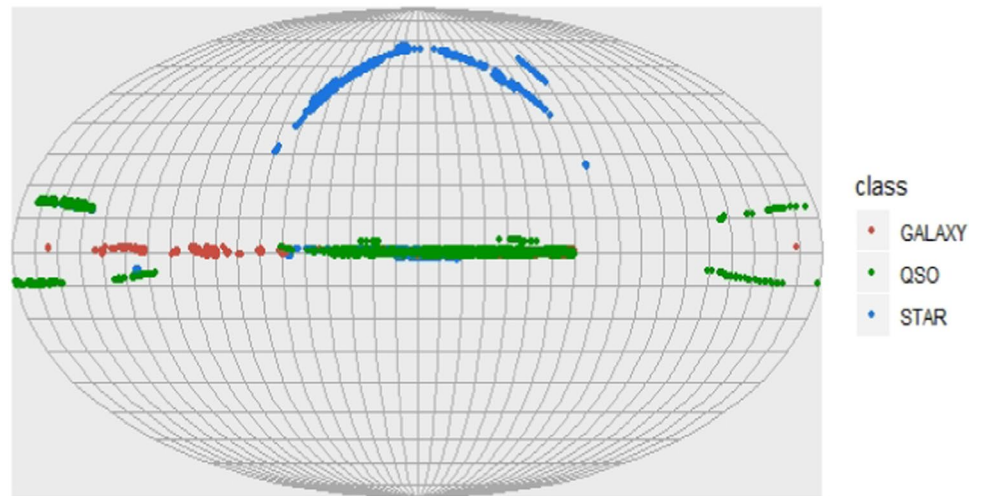
	Min	Median	Mean	Max
<i>ra</i>	0.0724	184.9148	164.8349	359.9189
<i>dec</i>	− 9.97555	0.08735	6.25947	68.01452
<i>u</i>	13.99	20.65	20.94	27.76
<i>g</i>	12.79	19.5	19.55	26.25
<i>r</i>	12.15	18.93	18.81	29.73
<i>i</i>	11.77	18.58	18.42	24.59
<i>z</i>	11.65	18.29	18.18	26.47
<i>run</i>	94	752	805.2	1754
<i>rerun</i>	301	301	301	301
<i>camcol</i>	1	3	3.256	6
<i>field</i>	11	315	329.6	811
<i>redshift</i>	− 0.010875	0.158429	0.613957	7.008873
<i>plate</i>	266	2193	2194	8198
<i>mjd</i>	51608	53905	53666	57401
<i>fiberid</i>	1	365	380.1	1000
Class	GALAXY:3495; QSO:2997; STAR:3500			

### 3 Cognitive pairwise comparison foreword features selection with deep learning

#### 3.1 Feature selection challenges

Whilst there are several alternative ways for heuristic feature combinations, domain experts may have their preferences to introduce the features one by one (or batch by batch) to build the machine learning models and observe how these results are accordingly changed. The motivation may include their preferences to see how changes are made by adding and evaluating new feature(s), and the confidence to build a suitable model with fewer features but higher prediction accuracy. The Pairwise Opposite Matrix (POM) is a promising method to encode his implicit knowledge explicitly, rather than just getting higher ranking results.

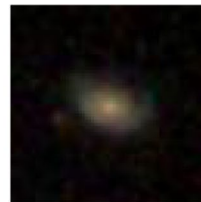
**Fig. 3** SDSS coverage for sample data



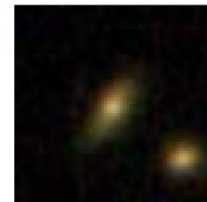
GALAXY



ra:238.07057  
dec:0.37041427



ra:195.64151  
dec:0.29202235



ra:187.49950  
dec:-1.20770979

QSO



ra:182.47399  
dec:-0.49460302

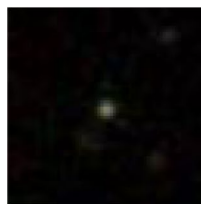


ra:21.55023  
dec:14.03830388



ra:186.39759  
dec:-0.34425949

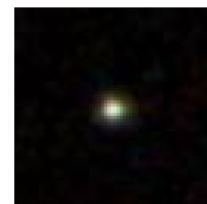
STAR



ra: 220.9688  
dec:-0.772322208



ra: 162.6211  
dec:-0.003768721



ra: 162.7797  
dec:1.095266744

**Fig. 4** Image instances of GALAXY, QSO, and STAR

Whilst there are 15 features, the number of combinations of the features,  $N$ , is calculated below.

$$N = \sum_{i=1}^{15} C_i^{15} = 32,767 \quad (1)$$

**Table 3** POM, priority, and rank for astronomic features for astronomic object classification

POM	( <i>ra,dec</i> )	<i>ugriz</i>	<i>run</i>	<i>rerun</i>	<i>camcol</i>	<i>field</i>	<i>redshift</i>	<i>plate</i>	<i>mjd</i>	<i>fiberid</i>	Priority	Rank
( <i>ra,dec</i> )	0	−6	1	5	3	2	−3	2	3	3	0.113	3
( <i>u,g,r,i,z</i> )	6	0	8	8	8	8	5	8	8	8	0.184	1
<i>run</i>	−1	−8	0	4	2	1	−4	1	2	2	0.099	4
<i>rerun</i>	−5	−8	−4	0	−2	−2	−8	−2	−2	−2	0.056	10
<i>camcol</i>	−3	−8	−2	2	0	−1	−6	−1	0	0	0.076	8
<i>field</i>	−2	−8	−1	2	1	0	−5	0	1	0	0.085	5
<i>redshift</i>	3	−5	4	8	6	5	0	5	6	6	0.148	2
<i>plate</i>	−2	−8	−1	2	1	0	−5	0	1	0	0.085	5
<i>mjd</i>	−3	−8	−2	2	0	−1	−6	−1	0	0	0.076	8
<i>fiberid</i>	−3	−8	−2	2	0	0	−6	0	0	0	0.079	7

The domain expert may consider *u*, *g*, *r*, *i*, and *z* to be used together as a batch of filters, and *ra* and *dec* to be used together as a pair of positions. Therefore, the number of features to be ranked is reduced to 10, and the number of combinations is calculated below.

$$N = \sum_{i=1}^1 C_i^{10} = 1,023 \quad (2)$$

Although the number of combinations is significantly reduced, 1023 combinations are still a large number. The proposed CPC-FSS for DL can address the problem of selecting features for a machine-learning model such as DL.

### 3.2 Cognitive pairwise comparison

The Cognitive Pairwise Comparison is a human-centered method as an interface for humans to interact with machine learning algorithms. The CPC is the core technique of PCNP [12–14] to determine the ranks of features. The Pairwise Opposite Matrix (POM) is used to interpret the priority of each feature. Let an ideal utility set be  $V = \{v_1, \dots, v_n\}$ , and a comparison score to represent the difference between two feature priorities be  $b_{ij} \cong v_i - v_j$ . The ideal pairwise opposite matrix is  $\tilde{B} = [v_i - v_j]$ . A subjective judgmental pairwise opposite matrix using paired interval scales is  $B = [b_{ij}]$ .  $\tilde{B}$  is determined by  $B$  as follows:

$$\tilde{B} = [\tilde{b}_{ij}] = [v_i - v_j] \cong [b_{ij}] = B \quad (3)$$

The  $b_{ij}$  is chosen from the paired rating scale  $\left\{-\frac{8}{\kappa}, \dots, -\frac{1}{\kappa}, 0, \frac{1}{\kappa}, \dots, \frac{8}{\kappa}\right\}$  representing {“extremely less important than”, ..., “weekly less important than”, “equal to”, “weekly more important than”, ..., “extremely more important than”}. The normal utility  $\kappa$  represents the mean of the priorities of the features. By default,  $\kappa$  is set to 8.

In this case, after the domain expert explores the data mentioned in Sect. 2.2, a POM for comparing astronomic features is conducted on the basis of the expert’s CPC evaluation. The POM results are presented in Table 3. For example, the score 5 of (*ra, dec*) vs. *rerun* means that (*ra, dec*) is 5 units more important than the *rerun*. Mathematically, the form is  $v_{(ra,dec)} - v_{rerun} = 5$ . A cognitive pairwise matrix  $B$  is verified by the Accordance Index ( $AI$ ) as below.

$$AI = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \sqrt{\frac{1}{n} \sum_{p=1}^n \left( \frac{(b_{ip} + b_{pj} - b_{ij})}{\kappa} \right)^2} \quad (4)$$

$AI \geq 0$ , and the normal utility  $\kappa$  is equal to 8 by default. If  $AI = 0$ , then  $B$  is perfectly accordant; If  $0 < AI \leq 0.1$ ,  $B$  is acceptable; If  $AI > 0.1$ ,  $B$  may be required to be revised. The accordance index for POM in Table 3 is 0.095, which is within the acceptable range.

To derive the priority, the Row Average plus the normal Utility (RAU) below is used as the prioritization operator.

$$v_i = \left( \frac{1}{n} \sum_{j=1}^n b_{ij} \right) + \kappa, \forall i \in \{1, \dots, n\} \quad (5)$$

The individual priority  $v_i$  from POM is rescaled as a normalized priority vector by the rescale function (or normalization function) as below.

$$w_i = \frac{v_i}{n\kappa}, \forall i \in \{1, \dots, n\} \quad (6)$$

The priorities and ranks for all features are presented in Table 3. To illustrate the calculation, the priority of (*ra*, *dec*) is calculated as below.

$$v_1 = \frac{(0 + (-6) + 1 + 5 + 3 + 2 + (-3) + 2 + 3 + 3)}{10} + 8 = 9$$

Thus, the relative priority of (*ra*, *dec*) is calculated as below.

$$w_1 = \frac{9}{10 \times 8} = 0.1125$$

The priorities of the rest of the features are calculated by the same methods as demonstrated above.

### 3.3 Deep learning configuration

The CPC-FFS-DL algorithm (Algorithm 1) includes two major components: the feature ranks evaluated by an expert using CPC, which is presented in the previous sub-section, and the deep learning model configured, built, and tested according to the feature ranks. Deep learning is an Artificial Neural Network (ANN) with deep layers and each layer consists of neurons. The word "deep" is not related to deeper understanding about mind in psychology or in our common sense, but is just mentioned to the depth of an ANN mathematical model.

*Keras* is a Python deep learning library that runs on top of TensorFlow, CNTK, or Theano, and enables fast experimentation [5]. This research applies the *Keras* package (Version 2.15.0, 2024-04-19) in R language (Version 4.4.0, 2024-04-24) [1] as an interface to use the *Keras* (Version 3.3.3) on top of TensorFlow (Version 2.16.1) in Python language (Version 3.12.3, 2024-4-9). Difference versions may have slight difference for the performance. Whilst there are many configurations of deep learning models, a multi-layer perceptron is a good enough initiative for CPC-FFS-DL to address the proposed astronomical object classification problem in this study.

Figure 5 presents the R code to implement the fully connected multi-layer perceptron with the configuration information after several trials of the parameter tuning manually. Figure 6 demonstrates the depth of the DL model of eight inputs: *u*, *g*, *r*, *i*, *z*, *redshit*, *ra*, and *dec*. Whilst the number of features increases, the number of neurons in the first layer in the network increase by times accordingly. There are 11587 parameters to be tuned in the model. Comparing with the classical neural network with a few parameters, this network is called "deep". The definitions of functions and configurations are found in *Keras* Documentation [5].

### 3.4 CPC-FFS-DL algorithm

Forward Features Selection (FFS) is a greedy approach that iteratively evaluates a new feature on a set of selected features, and adds the new feature if there is an improvement. The problem with basic FFS is that the order of features has significant impacts on the model construction and prediction performance. Incorporating human knowledge with CPC can be a promising solution to achieve the optimal model performance result. The CPC approach can be applied to rank the features for the use of FFS. The domain expert is supposed to have sufficient knowledge to rank. The CPC is used to encode the preference in POM data structure denoted by *B* and understand how such features are compared and ranked.



**Fig. 5** Configuration of multi-layer perceptron model using Keras in R code

```

model <- keras_model_sequential()
model %>%
  layer_dense(units = 128,activation = "relu",input_shape = ncol(X)) %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 64, activation = "relu") %>%
  layer_dropout(rate = 0.1) %>%
  layer_dense(units = 32, activation = "relu") %>%
  layer_dense(units = classSize, activation = "softmax")
model %>% compile(loss = "categorical_crossentropy",
  optimizer = optimizer_rmsprop(), #optimizer_adagrad(),
  metrics = c('accuracy'))
## X is the input matrix, Y is the Class labels
model %>% fit(X, Y, validation_split = 0.10,
  epochs=500, batch_size = 50, shuffle = TRUE)
model

```

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 128)	1152
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dense (Dense)	(None, 3)	99
Total params: 11587 (45.26 KB)		
Trainable params: 11587 (45.26 KB)		
Non-trainable params: 0 (0.00 Byte)		

**Fig. 6** Summary of a DL instance of eight inputs (*u, g, r, i, z, redshit, ra* and *dec*)



**Algorithm 1** CPC-FFS-DL

---

$CFD(X, B, \kappa, DL, \psi, \rho)$
1. Calculate the importance of the features $\{w_i\} = RAU(B, \kappa)$
2. Get feature indices by ranking $\{w_i\}$ in descending order $\theta = \{j\} = rank(\{w_i\})$
3. Initialize the selected feature index set $\beta^+$ including the most referred variable located in the $\theta_1$ column of $X$ . $\beta^+ = \{\theta_1\}$
4. Split $X$ into training and test data, i.e., $X_{trn}$ and $X_{tst}$ .
5. Evaluate the additional feature. If there is significant improvement $\rho$ , the new feature index is added. For each $j$ in $\theta$ do $\beta = \beta^+ \cup j$
$\text{If } \psi(DL(X_{trn}[\beta^+]), X_{tst}) + \rho < \psi(DL(X_{trn}[\beta]), X_{tst})$ $\beta^+ = \beta$
Return: the selected feature index set $\beta^+$ .

---

Algorithm 1 presents the calculation steps of the cognitive pairwise comparison forward features selection with DL, denoted by  $CFD(X, B, \kappa, DL, \psi, \rho)$ . Whilst CPC component is shown in Steps 1–2, FFS with DL is shown in Steps 4–5.  $X$  is the data table with  $m$  rows by  $n + 1$  columns including one column of the target variable.  $X$  is split into training data  $X_{trn}$  and testing data  $X_{tst}$ .  $B$  is a POM and  $\kappa$  is the normal utility.  $DL$  is the configured deep learning model presented in Sect. 3.3.  $DL(X_{trn}[\beta^+])$  is the  $DL$  training model based on the training data  $X_{trn}$  and selected feature index set  $\beta^+$ .  $\psi$  is a measure function to evaluate the performance of predicted results, e.g., accuracy.  $\rho$  is the baseline for the significant improvement.  $\psi(DL(X_{trn}[\beta^+]), X_{tst})$  yields the performance score for the classification results from testing data  $X_{tst}$  based on the  $DL$  training model, i.e.,  $DL(X_{trn}[\beta^+])$ , with the selected feature index set  $\beta^+$ . The demonstration of the use of Algorithm 1 is presented in Sect. 4.

## 4 Simulations and discussions

### 4.1 Background and random selections

As the details are described in Sect. 2, 10,000 records from SDSS have been obtained in the sample data. After 8 outliers or missing values are removed, 9992 records are used for this simulation. A random number function, *runif*, with a seed number of 999 is used to randomize the order of the sample data retrieved from the SQL query. The first 9000 records are chosen as the training data ( $X_{trn}$ ), whilst the rest 992 records are chosen as the testing data ( $X_{tst}$ ). The data consists of 15 features. The feature indices are defined in Table 4 according to the rank in Table 3. The scaling method is based on standardization method. As all instances of the “rerun” feature is of the same value, the rescaled value is NA due to division by zero. However, the packages of *R Keras* on top of *TensorFlow* still can handle the NA values using default setting, which may lead to lower testing accuracy.

To illustrate the challenges of feature selection, Table 4 presents seven cases with random feature selections for raw or scaled training data to train the Deep Learning model, and the accuracy results are predicted by testing data. The accuracy is calculated by the number of successful classification cases over the total number of classification cases. It is observed that some features can contribute to significant accuracy increment, whilst some have a negative impact. The combinations influence the overall accuracy. As indicated by Eq. (1), it is challenging to find the best or optimal feature combinations among 32,767 possibilities. Although limitations exist, the forward feature selection is one of the possible ways to address this issue.

**Table 4** Simulation results using random feature selection

Feature index	Feature	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
1–5	(u,g,r,i,z)							Y
6	redshift						Y	
7,8	(ra,dec)					Y	Y	Y
9	run	Y		Y	Y			
10	field	Y		Y	Y			
11	plate	Y		Y	Y			
12	fiberid		Y	Y	Y			
13	camcol		Y	Y	Y			
14	mjd		Y	Y	Y			
15	rerun				Y			
Accuracy (%) (Raw data)		81.8	37.1	37.1	37.1	78.5	96.5	88.3
Accuracy (%) (Scaled data)		86.1	82.4	93.1	93.5	86.2	98.5	95.7

**Table 5** Simulation results using forward features selection without CPC order

Case	Feature sequence	Selected features (raw data)	Selected features (scaled data)	Accuracy (%) (raw)	Accuracy (%) (scaled)
1	15,14,13,12,11,10,9,8,7,6,5,4,3,2,1	15,12,11,10,9,8,7,6,1	15	97.1	37.1
2	12,11,5,6,10,3,1,13,7,8,14,15,9,2,4	12,5,6,3	12,11,5,6,10,3	94	96.7
3	2,11,14,6,3,1,10,5,7,12,9,4,15,13,8	2,11,10,5,7,13	2,11,14,6,7	92.5	96.9
4	4,14,5,6,12,10,15,2,9,3,1,7,8,13,11	4,14	4,14,5,6,2,7	37.1	97.4
5	10,15,8,2,9,7,14,4,1,12,13,3,11,5,6	10,15,8,2,9,4,6	10,8,2,9,14,1,5	95.8	96.7
6	8,9,4,5,1,15,13,12,3,11,14,2,7,6,10	8,9,4,5,1,13,11,2,6	8,9,4,1,3,11	97	96.5
7	12,8,2,14,3,6,10,15,9,4,5,13,11,7,1	12,8,2,3,6	12,8,2,14,3,6,9	95.6	97.7

## 4.2 Basic forward feature selection with DL

Forward feature selection is one of the popular feature selection methods in feature engineering. The major steps of the basic FFS without CPC are shown in Steps 4–5 in Algorithm 1. The training data with the selected feature is used to train the DL model whilst testing data with the selected feature is used for benchmarking. Table 5 exhibits sequences of different feature orders in the dataset that can lead to different combinations of selected features resulting in different accuracy values, in addition to another factor that the data is either scaled or not.

To understand the reasons that features are selected from the sequences of different feature orders, Table 6 exhibits how accuracy (%) changes with FFS based on raw data, whilst Table 7 exhibits the accuracy changes based on scaled data. The significant improvement  $\rho$  is set to 1%, which means that a new feature contributing 1% to the accuracy is included. For example, Feature Sequence Index 1 for Case 1 in Table 6 means Feature 15 for Case 1 in Table 5, which refers to *rerun* in Table 4. If the FSI 2, i.e., *mjd* (Feature 14), is added for the DL training model, the testing accuracy does not improvement, and thus *mjd* is not included. Similarly, when the FSI 3 is added into DL training model, the testing accuracy has no improvement, and FSI 3 is also not included. When the FSI 4 (*fiberid*) is added and has more than 1% improvement, the new feature is included.

The orders of feature sequence and data transformation have significant impacts on the performance of feature selection results. The problem of how to order the features must be addressed. By utilizing expert knowledge, the Human Centered Forward Feature Selection with CPC is a promising solution introduced in the next subsection.

## 4.3 Human-centered forward feature selection with CPC and DL

For the CPC-FFS-DL algorithm shown in Algorithm 1, the objective of the simulations is to observe changes in the accuracy by adding one feature or a batch of features according to the feature rank evaluated by an expert using CPC. Table 8

**Table 6** Accuracy (%) changes using forward feature selection with raw data

Feature Sequence Index	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
1	37.1	37.1	48.7	35.5	42.5	69.9	39.8
2	37.1	37.1	54.6	37.1	54.5	72.7	63.5
3	37.1	38.7	37.1	37.1	80.6	80.9	68.9
4	40.9	90.5	37.1	37.1	84.6	82.6	37.1
5	70.4	89.9	37.1	37.1	86.7	84.4	73.1
6	76.9	94	37.1	37.1	87.5	85.3	95.6
7	80.2	94.5	65.4	37.1	37.1	87.5	95.2
8	89.8	91.2	73.8	37.1	90.6	83.3	93.4
9	91	92.6	91.5	37.1	90.9	87.4	91
10	95.7	89.9	90.8	37.1	90	90.6	95.4
11	93.9	37.1	89.6	37.1	90.9	37.1	94.6
12	95.7	88.9	91.6	37.1	88.7	92.5	95.7
13	94.4	94.1	88.8	37.1	86.9	88.7	94.8
14	96.2	94.1	92.5	37.1	90.1	97	95.5
15	97.1	93.5	89.8	37.1	95.8	96.8	95.4

**Table 7** Accuracy (%) changes using forward feature selection with scaled data

Feature sequence index	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
1	37.1	41.7	55.1	54.7	55.7	69.1	41.6
2	37.1	73.7	78.2	78.9	37.1	85	70
3	37.1	80.8	79.3	85	83.9	90.1	77.5
4	37.1	94	95.7	93.9	86.8	89.7	87.2
5	37.1	95.3	96.3	94.3	91.5	93.8	91.5
6	37.1	96.7	96.3	94.4	91.5	37.1	96.4
7	37.1	96.8	96.3	37.1	94.1	94.6	96.8
8	37.1	97.2	96.1	95.6	94.1	94.3	37.1
9	37.1	97.5	96.9	96.5	95.5	95.2	97.7
10	37.1	96.7	96.4	95.8	95.6	96.5	97.5
11	37.1	96.3	97.7	96.2	96.4	96.4	97.8
12	37.1	37.1	96.8	97.4	95.8	95.9	98.5
13	37.1	96.8	37.1	97.7	96.3	96.3	97.8
14	37.1	96.7	97.8	98	96.7	97	97.4
15	37.1	96.3	97.4	97.4	97.3	96.5	98

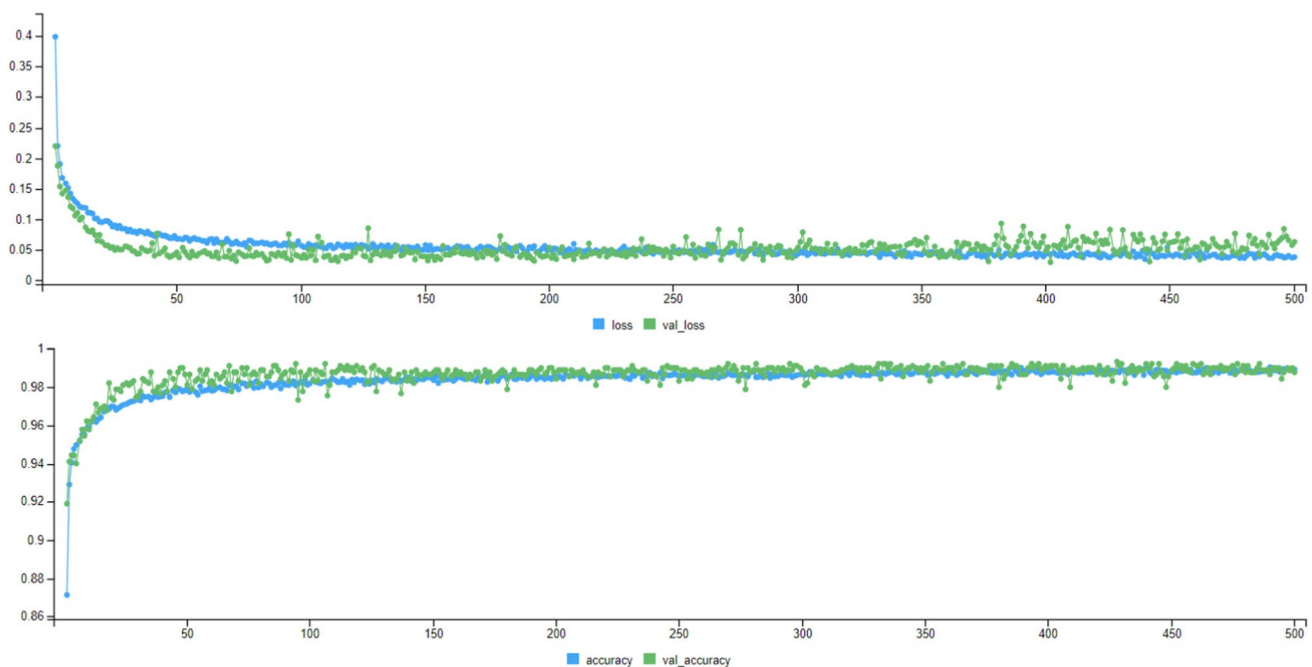
presents the simulation results for the forward features selection based on the rank evaluated by the expert using CPC shown in Table 3. The features of higher preferences are selected first.

For the first six simulation cases in Table 8, the data for the same setting of the Deep Learning model has two types: raw data and scaled data. Regarding the data transformation, standardization is applied for scaled data. Different configurations very likely result in different training times and accuracy. After several trial, the tuned configuration of the DL model is presented in Fig. 5. When accuracy and loss are converged, the DL training model is good enough to be used. Figure 7 presents the convergence of an instance of DL training model. Whilst the maximum of iterations is set to 500 in this experiment, the loss and accuracy start to converged after 300 iterations.

Table 9 presents the selection results with  $\rho$  of 0.2% and a pure element-by-element approach, instead of combining some relevant features as batches such as  $(u, g, r, i, z)$  and  $(ra, dec)$ . The results of Table 9 are quite similar to Table 8. If we run the DL model code again using Keras, the results may be slightly different even with the same seeds, as DL is a heuristic approach.

**Table 8** Simulation results using CPC batch forward features selection

Index	Feature	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
1–5	<i>(u,g,r,i,z)</i>	Y	Y	Y	Y	Y	Y
6	<i>redshift</i>		Y	Y	Y	Y	Y
7,8	<i>(ra,dec)</i>			Y	Y	Y	Y
9	<i>run</i>				Y	Y	Y
10	<i>field</i>				Y	Y	Y
11	<i>plate</i>				Y	Y	Y
12	<i>fiberid</i>					Y	Y
13	<i>camcol</i>					Y	Y
14	<i>mjd</i>					Y	Y
15	<i>rerun</i>						Y
Accuracy (%) (raw data)		86	95.5	97	95.6	37.1	37.1
Accuracy (%) (scaled data)		90.3	95.7	97.9	98.6	98.9	98.7

**Fig. 7** Convergence of training iterations

To conclude the simulation results in Table 8, when the features of raw data are added one by one according to the rank order of the CPC selection, the accuracy increases whilst *redshift* and *(ra, dec)* are added into *(u,g,r,i,z)*, but decreasing whilst the rest of features are continuously added. This situation reflects that the ranking results from the CPC make reasonable sense. If a significant improvement  $\rho$  is set to 1% for the CPC-FFS-DL algorithm with batch selection, eight features are chosen to build a robust DL model when raw data are either scaled or not, although there may be a slight improvement when an additional feature is added, whilst some additional features may lead to lower accuracy.

On the other hand, if the data is scaled by standardization, accuracy increases further when more scaled features are included until a *rerun* is added. The main reason is that *rerun* is just a single value of 301 (or a constant) in the sample data (Table 2) and becomes a *null value* after standardizing the feature value. When a feature of *NA* value is used for the proposed DL model of the package, the performance of the DL model may be distorted. In summary, all features except for *rerun* can be used to build the DL model when raw data are standardized.

Concerning random feature selections in Table 4 and the forward feature selection with initial random feature sequence in Table 5, it appears that none of the combinations with scaled data have better accuracy than the CPC methods shown in Table 8 (98.9%). It is concluded that the order of feature sequences influences the combinations, which further results

**Table 9** Accuracy (%) changes using CPC element-by-element forward feature selection

Index		Accuracy(raw)	Selection (raw)	Accuracy(scaled)	Selection (scaled)
1	u	35.8	1	49	1
2	g	69.1	1	72	1
3	r	79.5	1	83.1	1
4	i	85.2	1	88.4	1
5	z	84.9	0	90.4	1
6	redshift	95.2	1	95.9	1
7	ra	96.3	1	97.7	1
8	dec	98.0	1	97.9	1
9	run	97.0	0	98.2	1
10	field	98.2	1	98.6	1
11	plate	97.4	0	98.4	0
12	fiberid	97.8	0	98.5	0
13	camcol	97.7	0	99.0	1
14	mjd	37.1	0	98.9	0
15	rerun	97.6	0	99.0	0

in the performance of the deep learning model, and expert evaluation for the order of selection using CPC can benefit the model prediction performance. If we blindly choose all features, inappropriate features are included, and the accuracy could be only 37.1% (subject to a DL design as well), as indicated in Table 8. To address this issue, the proposed CPC can be a promising method to support domain experts to make better decisions for feature selections.

#### 4.4 Comparisons

To compare with the existing approaches, the Recursive Feature Elimination (RFE) function of the *Caret* R package [15] is selected. RFE is a feature selection method based on existing ML algorithms. Whilst complete and exhaustive comparison

**Table 10** Simulation results using recursive feature elimination with five machine learning models

features	k-nearest neighbors		Naive bayes		CART		Regularized logistic regression		Support vector machine	
	Raw	Scaled	Raw	Scaled	Raw	Scaled	Raw	Scaled	Raw	Scaled
u	Y	Y	Y	Y			Y	Y	Y	Y
g	Y	Y	Y	Y			Y	Y	Y	Y
r	Y	Y	Y	Y			Y	Y	Y	Y
i	Y	Y	Y	Y			Y	Y	Y	Y
z	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
redshift	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ra		Y					Y	Y		Y
dec	Y	Y	Y	Y			Y	Y	Y	Y
run	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
field		Y					Y	Y		Y
plate		Y			Y	Y	Y	Y		Y
fiberid		Y					Y	Y		Y
camcol		Y					Y	Y		Y
mjd		Y					Y	Y		Y
rerun		Y					Y	Y		Y
Accuracy (%)	96.1	95.7	93.1	93.1	91.5	91.5	96.7	96.7	96.3	97.3

for discussion is not the focus of this study, five popular machine learning algorithms are selected for RFE: C5.0 using *C5.0* function, k-Nearest Neighbors using *knn* function, Naive Bayes using *nb* function, Classification and Regression Trees (CART) using *rpart* function, Regularized Logistic Regression using *glmnet* function, and Support Vector Machine with linear kernel using *svmLinear* function. The *Caret* depends on the other packages in which readers may refer to the related documents such as in [15]. As some ML function cannot handle NA value, the scaled value of *rerun* feature is set to 1.

The train and RFE method in *Caret* R Package automatically tune the parameters for each selection machine learning methods. Comparing to the proposed CPC-FFS due to proper expert involvement, the RFE method requires more computational resources for more exhaustive search with more subsets of features and more partitions of data to achieve the higher accuracy. Five-fold cross validation is used to train the model. Table 10 presents simulation results of feature selection and accuracy using recursive feature elimination with the five machine learning models. Generally, the accuracy is slightly lower than the proposed method shown in Case 3 of Table 8.

## 5 Conclusions and future study

Whilst the selection of the best combination of features among many possible combinations is challenging, this paper proposes a Cognitive Pairwise Comparison Forward Feature Selection with Deep Learning (CPC-FFS-DL) for astronomical object classification. The data is obtained from the SDSS SkyServer SQL web service. The CPC approach is used to rank the choices of features according to the domain expert's preference. The DL models are implemented according to choices of features ranked by experts using the CPC. The DL model can achieve very high accuracy (97% with raw data and 98.9% with scaled data) for the astronomical object classification using the eight features (*u*, *g*, *r*, *i*, *z*, *redshift*, *ra*, *dec*) with raw data and additional six features (*run*, *field*, *plate*, *fiberid*, *camcol*, *mjd*) with scaled data. Without proper expert involvement, the accuracy potentially can only be 37.1%. The comparisons demonstrated that the proposed method is promising.

As the rating scores from the domain experts significantly influence the results, the CPC is demonstrated as a promising tool to explicitly capture the implicit knowledge from the domain experts, but the limitation is that the model result depends on how knowledgeable an engaged domain expert is. If the data is standardized, more features added may increase the accuracy further, but the limitation is that the computational workloads will increase.

The study provides the initiatives for the use of CPC for feature selection applied to deep learning for the insight of astronomical physics study. The Forward Features Selection with CPC can be applied to not only the Deep Learning model but also the other supervised classification approaches, such as C5.0, CART, Random Forest, support vector machine, and k-Nearest Neighbors, which will be conducted in future studies. Based on the proposed CPC-FFS-DL applying to the astronomical object classification, the proposed approach can be applied to other astronomical classification problems with the other features from the SDSS or the other open platforms. To further extend, the CPC-FFS-DL can be applied to the other areas of human-centered data sciences.

**Acknowledgements** The author is very grateful for the Editor-in-Chief, Handling Editor, Assistant Editor and the anonymous referees for their time and efforts to improve and recommend this work.

**Author Contribution** This research is solely conducted by K.K.F. Yuen.

**Funding** This research has not been funded by any company or organization.

**Data availability** The datasets generated during and analyzed during the current study are available in the GitHub repository, <https://github.com/kkfyuen/SDSS>.

## Declarations

**Competing interests** The author states that there is no conflict of interest.

**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

1. Allaire JJ and Chollet F, "keras: R interface to 'Keras'", 2024. <https://CRAN.R-project.org/package=keras>.
2. Bazarghan M, Gupta R. Automated classification of sloan digital sky survey (SDSS) stellar spectra using artificial neural networks. *Astrophys Sp Sci*. 2008;315:201–10.
3. du Buisson L, Sivanandam N, Bassett BA, Smith M. Machine learning classification of SDSS transient survey images. *Mon Not R Astron Soc*. 2015;454:2026–38.
4. Feigelson ED, Babu GJ. *Modern statistical methods for astronomy with R Applications*. Cambridge: Cambridge University Press; 2012.
5. Keras team. "keras documentation,". 2021. <https://keras.io>.
6. Kim EJ, Brunner RJ. Star–galaxy classification using deep convolutional neural networks. *Mon Not R Astron Soc*. 2017;464:4463–75.
7. Koczkodaj WW, Mikhailov L, Redlarski G, Soltys M, Szybowski J, Tamazian G, Wajch E, Yuen KKF. Important facts and observations about pairwise comparisons. *Fund Inform*. 2016;144:1–17.
8. Saaty TL. *Analytic hierarchy process: planning, priority, setting, resource allocation*. New York: McGraw-Hill; 1980.
9. Sloan Digital Sky Survey. "SDSS website,". 2021. <https://www.sdss.org>.
10. SDSS DR15. "SkyServer DR15". 2021. <http://skyserver.sdss.org/dr15/en/home.aspx>.
11. Wickham H. "ggplot2: elegant graphics for data analysis." New York: Springer-Verlag; 2016.
12. Yuen KKF. Cognitive network process with fuzzy soft computing technique in collective decision aiding, The Hong Kong Polytechnic University, Ph.D. thesis. 2009.
13. Yuen KKF. The primitive cognitive network process in medical decision making: comparisons with the analytic hierarchy process. *Appl Soft Comput*. 2014;14:109–19.
14. Yuen KKF. Fuzzy cognitive network process: comparisons with fuzzy analytic hierarchy process in new product development strategy. *IEEE Trans Fuzzy Syst*. 2014;22:597–610.
15. Kuhn. M. Recursive feature elimination. 2024 <https://topepo.github.io/caret/recursive-feature-elimination.html>.

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