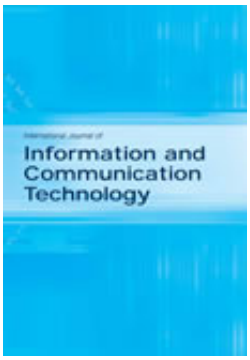
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Unbalanced data identification based on Bayesian optimisation convolutional neural network

Yanzhen Wang

Article History:

Received:	08 December 2024
Last revised:	18 December 2024
Accepted:	18 December 2024
Published online:	22 January 2025

Unbalanced data identification based on Bayesian optimisation convolutional neural network

Yanzhen Wang

School of Electronic and Computer Engineering,
Zhengzhou College of Finance and Economics,
Zhengzhou 450044, China
Email: yanzhenwang@126.com

Abstract: The difficulty of unbalanced datasets in classification issues has become more noticeable with the fast expansion of data science and machine learning approaches. When confronted with uneven data, conventional machine learning methods often produce poor prediction of a few classes. Based on Bayesian optimisation (BO), we propose in this work an enhanced convolutional neural network (CNN) framework (BO-CNN) meant to optimise the hyperparameter configuration of CNNs while resolving the class bias problem in unbalanced data. Experimental results reveal that BO-CNN shows benefits on challenging datasets, lowers miss-detection and false alarms, and efficiently enhances the capacity of the model to manage unbalanced data. These results offer a fresh approach for unbalanced data categorisation and a useful guide for the future optimisation and implementation of deep learning models.

Keywords: BO; Bayesian optimisation; convolutional neural network; CNN; unbalanced data recognition.

Reference to this paper should be made as follows: Wang, Y. (2025) 'Unbalanced data identification based on Bayesian optimisation convolutional neural network', *Int. J. Information and Communication Technology*, Vol. 26, No. 2, pp.96–111.

Biographical notes: Yanzhen Wang received her Master's degree from Nanjing University of Science and Technology in 2015. From 2019 to 2024, she studied in Dhurakij Pundit University and received her PhD in 2024. She is currently working in Zhengzhou College of Finance and Economics. Her research focuses on data analysis and machine learning.

1 Introduction

Data science and machine learning are developing quickly, hence the range and volume of data are expanding (Najafabadi et al., 2015; Ali et al., 2016). But many datasets in practice – especially those related to classification problems – often show extreme class imbalance. Conventional machine learning methods often struggle on unbalanced datasets (Thabtah et al., 2020), which produces bad predictions for a few classes. Although several techniques – such as oversampling, undersampling, and cost-sensitive learning – have been suggested to solve this issue – these approaches are typically

challenging to efficiently raise the performance of classification models on unbalanced data (Song et al., 2018).

In the realm of unbalanced data recognition, conventional classification systems often perform badly in the prediction of a few classes of samples, therefore generating large misclassification and omission rates (Susan and Kumar, 2021). Scholars have put up some ideas to solve this issue. Among the most often used methodologies are integrated learning approaches, cost-sensitive learning, and resampling methods.

The traditional approaches for addressing unbalanced data include oversampling and undersampling (Luengo et al., 2011). While undersampling balances the dataset by cutting the amount of majority class samples, oversampling replicates minority class samples. These techniques may cause over-generation or discarding of samples, therefore reducing the generalisation ability of the model even if they can somewhat increase the recognition rate of minority class samples.

By modifying the cost of classification error for various classes, cost-sensitive learning guides the model to give more focus on the appropriate classification of minority groups. Although this method can solve the issue of unbalanced datasets, it depends on the creation of a suitable cost function, which is usually challenging in reality (Zhou and Liu, 2015).

Furthermore, as deep learning develops, convolutional neural network (CNN) and other neural network models are progressively turning into useful instruments for unbalanced data recognition solution (Jiao et al., 2020). Many researchers have been motivated to employ CNN to cope with unbalanced datasets, particularly in multi-classification activities, by their success in the field of image recognition. Some methods increase the capacity of the model to learn from a few classes of samples by means of adaptive weighting loss function and attention mechanism, therefore improving the performance of CNN on unbalanced data. CNN can automatically learn the feature information of various classes in the imbalanced data identification job; but, due to the imbalance of classes, conventional CNN are prone to favour the classes with larger sample numbers, therefore reducing the recognition capability of minority classes (Sampath et al., 2021).

Bayesian optimisation (BO) has been progressively used in recent years as an effective hyperparameter optimisation tool for deep learning models (Wu et al., 2019). By means of BO, CNN's hyperparameters may be efficiently changed within a constrained amount of iterations, hence enhancing the model's performance on unbalanced data.

All things considered, studies on unbalanced data recognition have put forth a range of approaches with pros and drawbacks. This work presents an enhanced CNN framework based on BO (BO-CNN), which attempts to optimise the hyperparameter configuration of the CNN while tackling the problem of category bias in imbalanced data.

This work mostly produces advancements in the following spheres:

1 One proposes the Bayesian optimisation-convolutional neural network (BO-CNN) framework. Especially in unbalanced data, the framework uses the effective search strategy of BO to automatically control the hyperparameters of the CNN, so greatly enhancing the model's capacity to identify samples of a few classes by precisely optimising the hyperparameters of the model.

2 We investigate the combined BO with CNN approach and suggest its usage in unbalanced data recognition. This work generates a new framework combining the adaptive property of BO with the depth property of CNN to efficiently address the unbalanced data problem, therefore offering fresh approaches for unbalanced data recognition.

3 We employ two standard and unbalanced datasets. Different approaches including comparison tests, fusion studies, and ablation experiments confirm the great adaptability of the BO-CNN architecture in several application situations. By means of these tests, this work not only confirms the efficiency of the suggested approach but also offers a reference for deep learning applications in other domains, so highlighting the vast possibilities of BO along with CNN.

2 Relevant technologies

2.1 Bayesian optimisation

Particularly suited for optimisation problems when the objective function is computationally costly or not derivable is BO, a global optimisation technique (Locatelli and Schoen, 2021). The fundamental idea is to build an agent model to approximate the objective function and choose the sampling sites most likely to increase the value of the objective function by guiding the search process via an acquisition function. The Acquisition Function to choose the next sampling point and the Gaussian process (GP) as the agent model define the main components of BO (Obrezanova et al., 2007).

Assume that the function we wish to maximise is the objective function $f(x)$; BO models the objective function by constructing an agent model $p(f(x) \mid D)$, where D is an already seen data set. Suppose that at every point x in the input space the values of the goal function have a Gaussian distribution. Typically initialised to 0 for the mean function and 0 for the covariance function, the GP is defined by a mean function $\mu(x)$ and a covariance function $k(x, x')$.

$$f(x) \sim GP\left(0, k(x, x')\right) \tag{1}$$

Often used covariance functions are the radial basis function (RBF) expressed as:

$$k\left(x, x'\right) = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) \tag{2}$$

where l is the length scale; σ^2 is the signal's variance; the control function's smoothness is determined here.

The objective function $f(x)$ is unknown in the BO process; hence, the GP model and current data points help to deduce the goal function. Bayes' theorem allows one to derive the posterior distribution of the objective function at the new point x^* assuming n data points $X = [x_1, x_2,\ldots,x_n]$ and the matching objective function values $y = [f(x_1), f(x_2),\ldots,f(x_n)]$. The nature of the GP indicates that the following equation determines the expected value and objective function variance:

$$\mu^*\left(x^*\right) = k\left(x^*, X\right)^T K(X, X)^{-1} y \tag{3}$$

$$\sigma^2(x^*) = k(x^*, x^*) - k(x^*, X)^T K(X, X)^{-1} k(x^*, X) \tag{4}$$

where $K(X, X)$ is the covariance matrix between the data points; y is the known objective function value; $k(x^*, X)$ is the vector of covariances between the new point x^* and the known data points.

The aim function is to choose a new input point x^* at every BO step. BO uses maximising the acquisition function to choose the next evaluation point. A fundamental part of BO, the acquisition function decides how to balance exploration with exploitation. Formulated as expected degree to which the objective function is better than the current ideal value f^* at a given location x^*, expected improvement (EI) is a widely used acquisition function (Xu et al., 2021).

$$\alpha_{EI}(x^*) = (\mu^*(x^*) - f^*)\Phi\left(\frac{\mu^*(x^*) - f^*}{\sigma^*(x^*)}\right) + \sigma^*(x^*)\phi\left(\frac{\mu^*(x^*) - f^*}{\sigma^*(x^*)}\right) \tag{5}$$

where $\phi(\cdot)$ is the probability density function of the standard normal distribution and $\Phi(\cdot)$ is its cumulative distribution function. BO chooses the next evaluation point x^* by optimising the acquisition function, hence optimising the search efficiency.

Actually, especially in the realm of deep learning, BO is frequently employed for hyperparameter optimisation (Karl et al., 2023). The performance of the model in neural network training depends much on the choice of hyperparameters like learning rate, regularisation factor, batch size, etc.BO can provide better results with less experiments by effectively searching the hyperparameter space, therefore avoiding the inefficiencies of the conventional grid search or random search strategies. The optimisation problem can be stated assuming that the hyperparameter θ has to be improved and the aim is to minimise the loss $L(\theta)$ of the validation set:

$$\theta^* = \arg\min_{\theta} L(\theta) \tag{6}$$

BO aims to maximise the acquisition function by means of hyperparameter θ^*, therefore optimising the model performance.

In unbalanced data categorisation issues, BO might also be really crucial. Often times, the loss function must be changed for unbalanced data to enhance the classification performance for a small number of classes. One often used weighted cross-entropy loss function has the form:

$$L_{\text{weighted}} = -\sum_{i=1}^{N} w_{y_i} \log p(y_i | x_i) \tag{7}$$

where w_{yi} are the weights of category y_i, BO can automatically change these weights to balance the effect of several categories on the loss, hence enhancing the capacity to classify a limited number of categories.

Usually, BO is implemented by means of GP and acquisition function computations (Surianarayanan et al., 2023). Libraries including scikit-optimise, GPyOpt, etc. let BO be applied. Here's a simple Python code example that shows how to use scikit-optimise to Grid search:

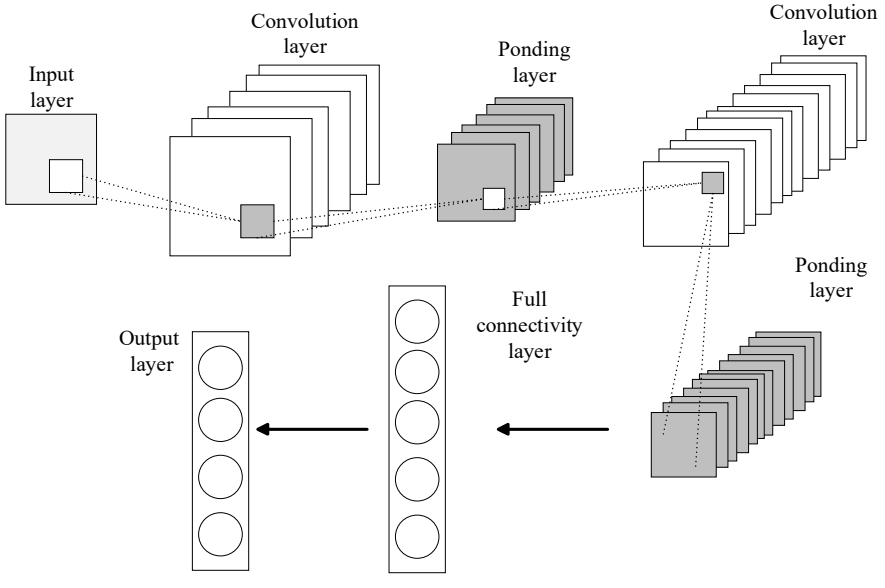
```
from skopt import gp_minimise
from skopt.space import Real, Integer

def objective(params):
    learning_rate, batch_size = params
    return train_model(learning_rate, batch_size)

space = [Real(1e-5, 1e-1, name='learning_rate'),
         Integer(16, 128, name='batch_size')]

result = gp_minimize(objective, space, n_calls=50, random_state=42)

print("Best parameters:", result.x)
print("Best validation loss:", result.fun)
```

Iteratively updating the agent model and maximising the acquisition function helps the next evaluation point to be chosen effectively in BO implementation. This procedure increases the efficiency of BO application in high-dimensional space over conventional random or grid search.

2.2 Convolutional neural network

Especially in cases of unbalanced data, CNN can enhance performance by automatically changing its hyperparameters with the help of BO. The fundamental ideas and CNN structure will next take the stage in this part.

Figure 1 Structure of CNN



Particularly suited for processing data with a grid structure – that of photos, videos, and audio signals – CNN is a deep learning model. As demonstrated in Figure 1, CNN's central concept is to automatically extract characteristics from the data by means of several convolutional, pooling, and fully connected layers (Akhtar and Ragavendran, 2020). CNN is able to automatically discover the ideal representation of the features during the training phase, unlike conventional human feature engineering techniques, thereby considerably simplifying processing difficult tasks.

CNNs' architecture makes the convolutional layer the most important element. By sliding a filter – or known convolution kernel – over the input data and doing a weighted summation over a particular region, the convolution operation extracts local features. Divining the convolution operation into two parts assuming the input data is a two-dimensional matrix X and the convolution kernel is a matrix W The point-by-point product of the convolution kernel and the input data comes first; then, the summation of these products yields the convolution result at last:

$$Z(i,j) = X(i,j) \cdot W(i,j) \tag{8}$$

$$Y(i,j) = \sum_{m} \sum_{n} Z(i+m, j+n) \tag{9}$$

where $Z(i, j)$ is the indicated result of the point-by- point product of the input data and the convolution kernel elements following convolution. $Y(i, j)$ denotes the result of the convolution operation; m and n indicate the convolution kernel's dimensions. This allows the convolutional layer to efficiently extract low-level features such image edges and textures, therefore providing the basis for later high-level feature learning.

Usually, pooling layers lower the output spatial dimensionality of a convolutional layer (He et al., 2015), therefore lowering computational cost and improving model resilience. Max Pooling is the most often used pooling technique since it chooses the maximum value in a limited area therefore lowering the dimensionality of the data. The pooling procedure has this formula below:

$$Y(i,j) = \max_{m,n} X(i+m, j+n) \tag{10}$$

By reducing the number of parameters in the model while preserving the important information, the pooling operation increases the computational efficiency.

Usually following the convolutional and pooling layers, CNNs feature one or more fully connected layers. By linking all the input nodes to the output nodes, the fully connected layer conducts worldwide feature learning. Based on the characteristics extracted from the convolutional and pooling layers, the network further processes in this layer to produce the final classification result. The calculation of the completely connected layer can be stated assuming that the input of the fully connected layer is x, the weight is W, the bias is b and the output is y as follows:

$$y = f(Wx + b) \tag{11}$$

where f is a nonlinear connection introducing activation function such ReLU or Sigmoid.

The usual training approach of CNNs may be influenced by the minority class samples when handling unbalanced datasets, therefore producing a model with great prediction accuracy for the majority class but poor recognition of the minority class. Usually, the training process is tuned – that is, using weighted loss functions,

oversampling or undersampling methods, etc. – to solve this challenge. In order to better handle the difficulties presented by unbalanced datasets, BO is essential in this process and helps to automatically modify the hyperparameters of the CNN, including the learning rate, batch size, convolutional kernel size, etc., in order.

Together with BO, the CNN training process gets more effective. By helping to choose the ideal hyperparameter setup, BO can enable CNN to reach great classification performance in less training cycles. When confronted with complicated and unbalanced datasets, this capacity to automatically modify the hyperparameters helps the CNN to greatly increase the generalisation capacity and accuracy of the model.

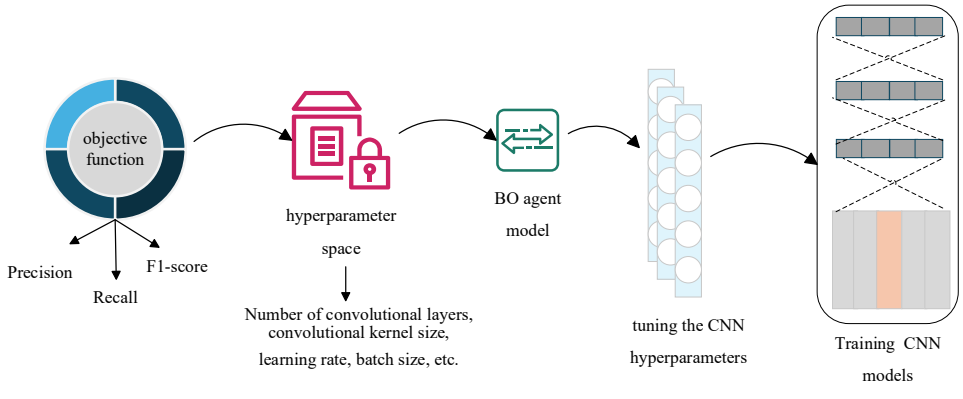
We acknowledge the role of integrated learning in handling imbalanced data. While our primary focus is on the BO-CNN framework, we incorporate elements of integrated learning through the BO process, which implicitly considers various model configurations to optimise hyperparameters. This approach allows us to leverage the benefits of multiple learning strategies without explicitly training an ensemble, streamlining our method and enhancing its adaptability to class imbalance.

3 BO-CNN: a BO-CNN based framework for unbalanced data recognition

3.1 BO-CNN framework recognition process

Two basic modules make up the BO-CNN architecture: BO and CNN, the former is mostly in charge of optimising the hyper-parameter configuration of the CNN while the CNN module is utilised to extract features from the data and carry categorisation. BO-CNN automatically changes CNN's hyperparameters using BO to enhance the model's performance on a small number of classes of samples on unbalanced datasets. BO-CNN consists of the following five components. Figure 2 shows the BO-CNN framework's workflow:

Figure 2 Structure of BO-CNN (see online version for colours)



1 Define the objective function

CNN on unbalanced datasets is evaluated using a criterion known as the objective function. Usually, our optimisation goal is F1-score, Recall or Precision. The equation looks like this:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

where TP is the true case count; FP is the false positive case count; FN is the false negative case count.

2 Selection of hyperparameter space

The hyperparameter space of the CNN must be precisely defined in the BO-CNN framework if BO is to conduct efficient search (Song et al., 2014). Common hyperparameters consist in the number of convolutional layers, convolutional kernel size, learning rate, batch size, and so on. Assume the hyperparameter space is:

$$\Theta = \{\text{Kernel Size}, \text{Number of Layers}, \text{Learning Rate}, \text{Batch Size}\} \tag{15}$$

Through exploration of this area BO will discover the ideal mix of hyperparameters.

3 Perform BO

BO searches for the best hyperparameter combination step by step and models the hyperparameter space using an agent model – e.g., a GP. Assuming θ_t as the present hyperparameter combination, BO chooses a new hyperparameter θ_{t+1} by maximising the objective function using agent model evaluation.

BO aims to identify the optimal hyperparameters by raising the expected value $\hat{f}(\theta)$ of the agent model – for instance, a GP – as highest as feasible. The recipe calls for:

$$\theta_{t+1} = \arg\max_{\theta} \hat{f}(\theta) \tag{16}$$

4 Optimisation of hyperparameters

Particularly the recognition ability on a few classes, the BO automatically changes the hyperparameter configurations of the CNN to enhance the performance of the model. By evaluating the performance of the present set of hyperparameters on the validation set, the optimisation process maintains the best hyperparameters current state.

One can find the BO iterative updating procedure as follows:

$$\theta_{t+1} = \theta_t + {}_{\Delta}\theta \tag{17}$$

where depending on the prediction outcomes, $\Delta\theta$ is the hyperparameter variance modified by BO in every iteration.

5 Training the CNN model

Training the last CNN model will be done using BO-tuned hyperparameters. Using the modified hyper-parameter settings, we will train the optimal CNN, hence improving the performance of the model on the unbalanced data.

3.2 Imbalanced data processing strategy in BO-CNN

Using the hybrid loss method helps the BO-CNN framework recognise minority class samples in the processing of unbalanced datasets. This approach combines cross-entropy loss with contrast loss, therefore enabling the model to pay greater attention to the learning of minority classes and preserve a high degree of classification accuracy appropriate for many application situations of unbalanced datasets.

To maximise model performance, the hybrid loss function aggregates conventional cross-entropy loss with contrast loss (Goceri, 2024). Minority class samples are less important in most unbalanced datasets, hence using cross-entropy loss by itself could lead to the weak identification of minority classes. Through the blending of the two losses, the framework helps the model to maintain the general classification accuracy while increasing the attention on the minority class, so boosting the recognition of minority class data.

The hybrid loss function has the particular formula:

$$L_{hybrid} = \lambda L_{ce} + (1-\lambda) L_{contrastive} \tag{18}$$

where equation represents the cross-entropy loss and L_{ce} marks:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{19}$$

where p_i is the expected probability of sample i and y_i is the actual label of the sample. Mostly, the cross-entropy loss helps to maximise the model's general classification accuracy.

Conversely, contrast loss enhances the discriminative power of the model by increasing the boundary variations between classes (Huang et al., 2021), particularly in regard to minority class data. The contrast loss formula is:

$$L_{contrastive} = \frac{1}{2N}\sum_{i=1}^{N}\left[y_i d(x_i, y_i)^2 + (1-y_i)\max\left(0, m - d(x_i, y_i)\right)^2 \right] \tag{20}$$

where m is a constant denoting the minimum distance between classes and $d(x_i, x_j)$ is the Euclidean distance between samples x_i and x_j.

BO is a useful hyperparameter optimisation tool that may automatically change the hyperparameter λ in the hybrid loss function throughout the training phase. λ is a hyperparameter here that decides the weight distribution between the cross-entropy loss and the contrast loss.BO aims to maximise the performance of the model on the validation set and to automatically choose the ideal value of λ, therefore enabling the BO-CNN framework to use varied unbalanced datasets to obtain optimal learning outcomes.

BO has as its objective function:

$$\hat{\theta} = \arg\max_{\theta} f(\theta) \tag{21}$$

where $f(\theta)$ is the performance metric of the model on the validation set, say accuracy or F1 score; θ is the collection of hyperparameters to be optimised.

By including hybrid loss – which combines contrast loss and cross-entropy loss – the category imbalance issue in unbalanced datasets can be essentially resolved in the BO-CNN architecture. This method guarantees increased recognition of minority class samples while preserving the general classification performance. Furthermore, the use of BO in the framework enables the intelligent and efficient choice of hyperparameter λ, hence optimising the performance of the model on unbalanced data.

4 Experimental results and analyses

4.1 Data sets

In order to assess the effectiveness of the BO-CNN framework in unbalanced data recognition, this work chooses two common unbalanced datasets: the UCI Adult dataset and the KDD Cup 99 dataset. These two sets of data obviously exhibit category imbalance. They thus provide application cases in the domains of tabular data and cybersecurity, respectively, so enabling complete testing of the model's performance with various kinds of data.

We obtained the UCI Adult dataset from the UCI Machine Learning Repository in order to forecast whether a person's annual salary surpasses $50,000. With 32,561 samples total, more than half of which fell into the minority category – that is, earning less than $50K. The dataset has a somewhat large category imbalance of roughly 1:3. There are 14 features in the dataset – age, education, job type, etc.; the feature types are numerical and categorical.

Table 1 Overview of datasets

Item	UCI Adult Dataset	KDD Cup 99 Dataset
Dataset name	UCI Adult Dataset	KDD Cup 99 Dataset
Sample size	32,561	4,898,434
Feature count	14 (e.g., age, workclass, education, marital-status, etc.)	41 (e.g., protocol_type, service, src_bytes, dst_bytes, etc.)
Class distribution	Imbalanced; Income > 50K (24%) vs. Income <= 50K (76%)	Highly Imbalanced; Normal (97%) vs. Anomalous (3%)
Class labels	Income > 50K, Income <= 50K	Normal, DoS (Denial of Service), Probe, R2L, U2R
Missing values	Some missing values in 'workclass', 'occupation', and 'native-country' fields	Some features have missing values, mostly categorical fields
Feature types	Mixed (Numerical: age, hours-per-week; Categorical: workclass, education)	Mixed (Numerical: src_bytes, dst_bytes; Categorical: protocol_type, service)
Imbalance degree	High (Minority class makes up only 24% of total samples)	Extreme (Normal traffic overwhelmingly outnumbers attacks)

Considered as a tool for network intrusion detection, the KDD Cup 99 dataset distinguishes several network threats. There are 49 attack types in the dataset; some attack categories have a somewhat low sample count while normal traffic and some assault categories have most of the samples. With a great degree of category imbalance, the dataset has 4,898,434 total samples, most of which fall into regular traffic or common attack categories. Mostly numerical, the features comprise a range of network communication statistics including traffic amount, connection length, protocol type, etc.

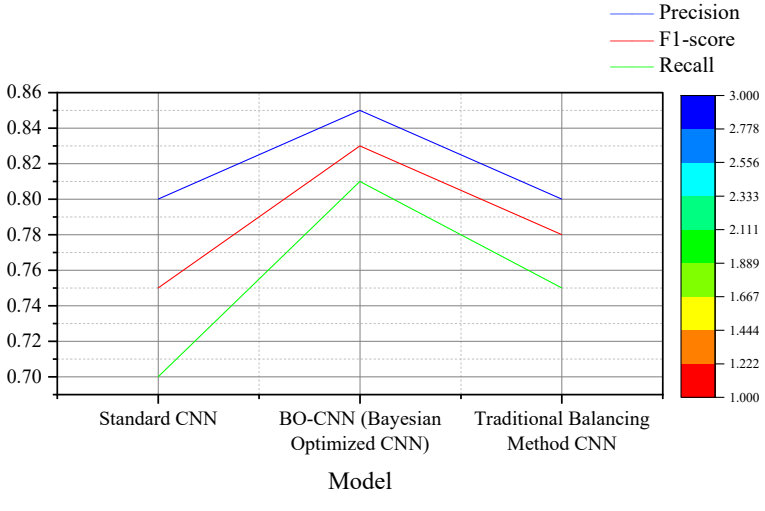
The foundation for the studies in this work will be Table 1, which lists the main characteristics, sample sizes, and category distributions of these two datasets.

4.2 Comparative experiments

We have developed a model comparison experiment with the intention of validating the advantages of the BO-CNN framework in unbalanced data recognition by means of other standard models handling unbalanced data and evaluating BO-CNN on several criteria. Two imbalanced datasets will be used in the experiment under which we apply the following models for training and testing: normal CNN (without BO), BO-CNN, and CNN models employing classical data balancing techniques including SMOTE and weighted loss function. Every model will be tested and trained using the same dataset and hyperparameter setup.

Three primary phases comprise the comparison experiment: data preprocessing, model training, and model evaluation. We first normalise the data for every dataset so that it satisfies the models' input needs. Second, using the same training set and test set, all models – standard CNN, BO optimised CNN, conventional data balancing method CNN) – will be trained. Except for the BO portion, which will be hyperparameter tuned in the BO-optimised CNN model, the first values of hyperparameters, the number of training rounds, and the optimiser settings remain the same for all models, thereby guaranteeing the fairness of the trials.

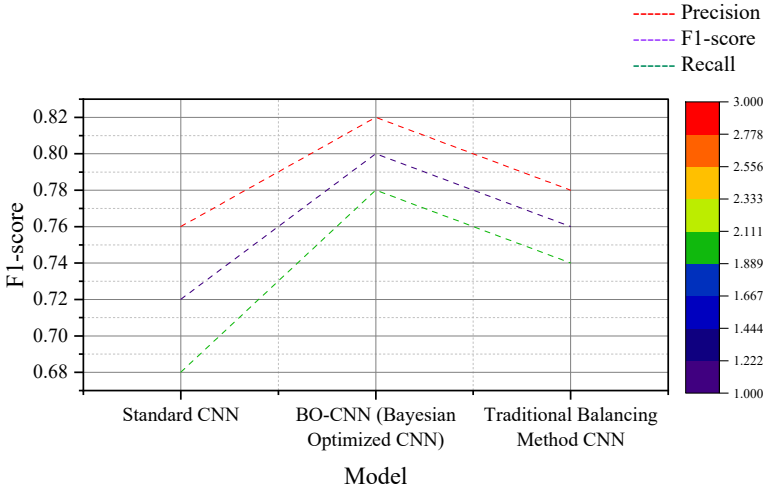
Figure 3 Results of the comparison experiment on the UCI Adult dataset (see online version for colours)



Every model will undergo 10-fold cross validation to validate their training process; each fold's F1-score, recall, and precision will then be assessed. Each model's performance will be thoroughly assessed using these criteria in handling the imbalanced data, particularly with regard to the model's capacity to recognise minority classes – that is, less classes in the unbalanced data. Figure 3 and Figure 4 exhibit the experimental outcomes.

With an F1-score of 0.75, recall of 0.70, and precision of 0.80, the standard CNN performs somewhat poorly in the experiments on the UCI Adult dataset, particularly in regard to lower in Recall. By modifying the hyperparameters via BO, BO-CNN increases the F1-score to 0.83, Recall to 0.81, and Precision to 0.85, so greatly improving the model's capacity to manage unbalanced data.

Figure 4 Results of the comparison experiment on the KDD Cup 99 dataset (see online version for colours)



The F1-score of the conventional CNN in the KDD Cup 99 dataset is 0.72, recall is 0.68, and precision is 0.76, so suggesting low recall in intrusion detection. On complicated datasets, BO-CNN greatly increases the recall and precision; after BO, the F1-score of BO-CNN is enhanced to 0.80 and recall and precision are 0.78 and 0.82, respectively.

Especially in the indices of F1-score, recall and precision, BO-CNN clearly beats both conventional balanced method CNN and standard CNN taken combined. Particularly in imbalanced data, BO is able to efficiently control the hyperparameters and enhance the performance of the model; it also helps to successfully increase the recall rate and precision and so lower the leakage and false alarms.

4.3 Hyperparametric sensitivity analysis experiments

First we choose three important hyperparameters for examination in the hyperparameter sensitivity study: learning rate, batch size, and convolutional kernel size. The fundamental elements of CNN performance, these hyperparameters significantly affect the training process and model output. We developed two experimental setups: one for

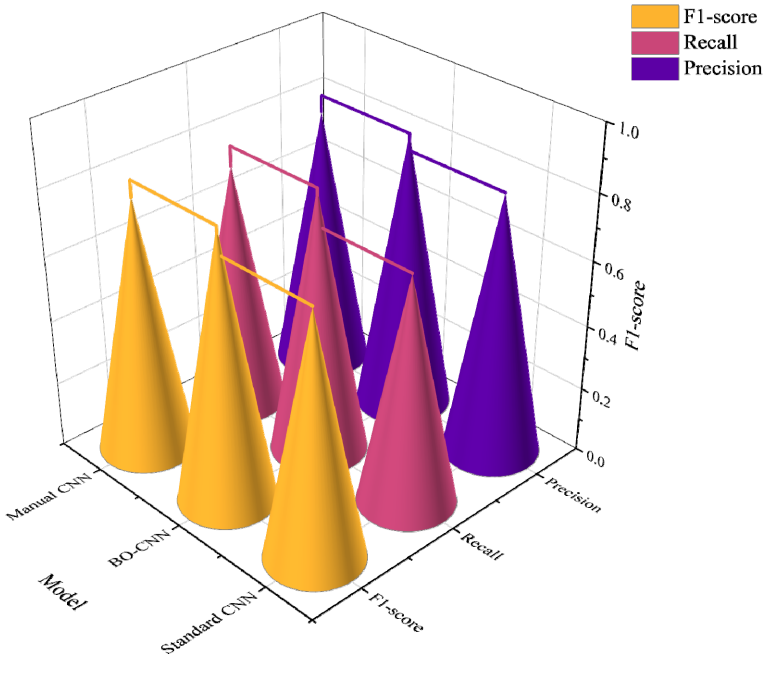
manually choosing hyperparameters and the other for automatically altering hyperparameters by BO to further assess the benefits of BO for hyperparameter tuning.

In the hand-selected hyperparameter experiment, we selected a set of typical hyperparameter values grounded on knowledge and literature. For example, we choose the batch size to be 32, the learning rate to be 0.01, and the convolutional kernel size to be 5. Using typical CNNs, we trained on the UCI Adult dataset and the KDD Cup 99 dataset in this arrangement; the evaluation metrics – F1-score, recall, and precision – were noted on each dataset. This phase aims to offer a baseline performance for next BO.

We next automatically tweak these hyperparameters using BO-CNN under the BO framework.BO explores the hyperparameter space and chooses the optimal configuration step by step, hence optimising the model. BO-CNN trains again on the same two datasets and automatically optimises the hyperparameters including learning rate, batch size, and convolutional kernel size during the training process. BO's method not only lowers the bias of human-selected hyperparameters but also investigates a larger hyperparameter range, so enhancing the model's performance.

We evaluate and analyse the performance of the standard CNN and BO-CNN correspondingly under every hyperparameter setting. In this sense, we can see how automatically changing the hyperparameters BO enhances the recognition performance of the CNN on unbalanced datasets. Figure 5 and Figure 6 present the experimental results for two datasets: KDD Cup 99 and UCI Adult dataset respectively.

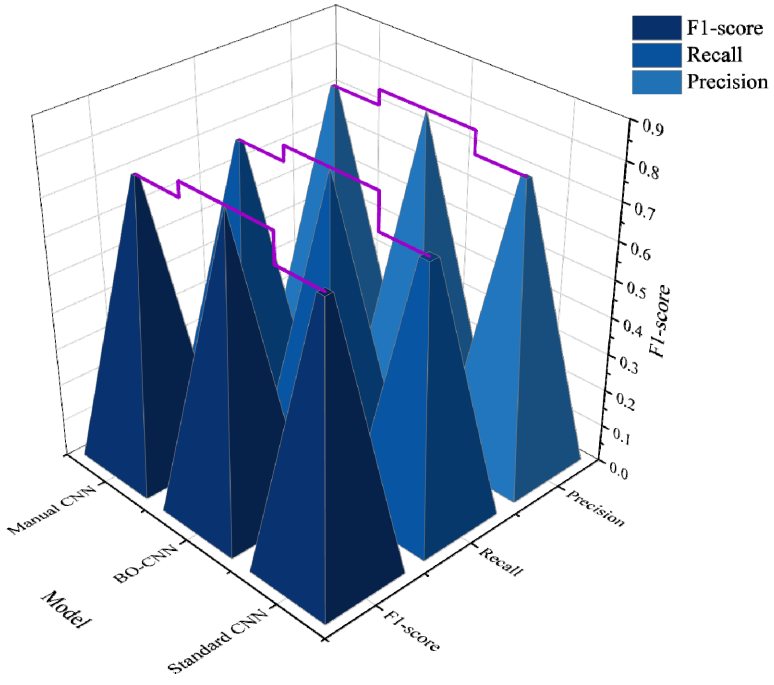
Figure 5 UCI Adult dataset experiment results (see online version for colours)



With low recall – shown by missing some positive class samples – the F1-score of the standard CNN on the UCI Adult dataset is 0.75. Following BO's BO-CNN, the F1-score rises to 0.83 and the recall and precision are much enhanced, therefore demonstrating the

value of BO in managing unbalanced data. Though still not as good as BO-CNN, the manually tweaked CNN performs somewhat better.

Figure 6 KDD Cup 99 dataset experiment results (see online version for colours)



With limited recall and prone to miss detection of intrusion events, the F1-score of the standard CNN is 0.72 on the KDD Cup 99 dataset. By BO, the BO-CNN increases the F1-score to 0.80 by means of notable recall and precision enhancement as well as a better identification of samples of few classes. Though still performs poorly relative to BO-CNN, the manually tweaked CNN also improves.

Especially in recall and precision, BO-CNN exhibits notable benefits on both the UCI adult dataset and the KDD Cup 99 dataset by means of comparison of their experimental results. Dealing with an unbalanced dataset, BO is able to automatically modify the hyper-parameters to maximise the performance of the model, hence improving F1-score, recall and precision. On the other hand, although the manual CNN improves the hyperparameters to some degree, it cannot equal the performance of BO-CNN in terms of handling unbalanced data; the conventional CNN performs really poorly. BO-CNN thus has a broad spectrum of uses and more adaptability and performance in unbalanced data recognition tasks, particularly for complicated and high-dimensional datasets.

5 Conclusions

This paper suggests a BO-based CNN architecture aiming at unbalanced data recognition problem solution. The work is conducted with two usual imbalanced datasets for experimental validation. Especially in the identification and recall of minority class

samples, the experimental results reveal that BO-CNN shows more notable improvement in the evaluation metrics such F1-score, Recall and Precision than the conventional standard CNN and manually optimised CNN. BO-CNN shows benefits on challenging datasets since it can more effectively handle unbalanced datasets and lower false alarms and missed detections than conventional balancing techniques.

The BO-CNN model suggested in this work has certain restrictions even if it shows improved experimental results in unbalanced data recognition. First of all, the computing cost of the optimisation process is significant and the BO process itself could be more time-consuming, particularly in the case of big-scale datasets and intricate models. Although this work uses a quite small dataset for validation, how to effectively execute hyperparameter tweaking in useful applications is still a topic of interest that calls more investigation. Second, this work mostly ignores the possibilities of other deep learning models (e.g., recurrent neural networks, graph neural networks, etc.) in unbalanced data processing by concentrating just on the mix of CNN and BO.

Future investigations might look into the following:

1 Improving BO efficiency: Future research could investigate more effective optimisation algorithms, such as sampling strategies to accelerate BO, or combining other optimisation methods (e.g., genetic algorithms, particle swarm optimisation, etc.), so improving the optimisation efficiency. The computational overhead of the BO process is a possible bottleneck in this study.

2 Extension to other deep learning models: Future studies can widen the mix of BO with other deep learning architectures. BO can be used, for instance, on sequence models like RNN, LSTM, etc. to investigate the possible applications in time series analysis and imbalanced data processing.

3 Integration of multiple unbalanced data processing strategies: Although BO can improve CNN's performance in unbalanced data processing greatly, real applications depend much on other elements including data pretreatment and sampling techniques. Combining oversampling, undersampling, SMOTE, and other methods with BO will help to maximise the CNN model and thereby increase the accuracy and robustness of unbalanced data identification.

Acknowledgements

This work is supported by the Key Project on Research and Practice of Higher Education Teaching Reform in Henan Province (No. 2024SJGLX0223), Soft Science Project of Henan Science and Technology Department (No. ZZFXJ2023044), and Henan Province Teacher Education Curriculum Reform Research Project (No. 2025-JSJYZD-065).

References

Akhtar, N. and Ragavendran, U. (2020) 'Interpretation of intelligence in CNN-pooling processes: a methodological survey', *Neural Computing and Applications*, Vol. 32, No. 3, pp.879–898.

Ali, A., Qadir, J., Rasool, R.U. et al. (2016) 'Big data for development: applications and techniques', *Big Data Analytics*, Vol. 1, pp.1–24.

Goceri, E. (2024) 'Polyp segmentation using a hybrid vision transformer and a hybrid loss function', *Journal of Imaging Informatics in Medicine*, Vol. 37, No. 2, pp.851–863.

He, K., Zhang, X., Ren, S. et al. (2015) 'Spatial pyramid pooling in deep convolutional networks for visual recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 9, pp.1904–1916.

Huang, D., Wang, M., Zhang, L. et al. (2021) 'Learning rich features with hybrid loss for brain tumor segmentation', *BMC Medical Informatics and Decision Making*, Vol. 21, pp.1–13.

Jiao, J., Zhao, M., Lin, J. et al. (2020) 'A comprehensive review on convolutional neural network in machine fault diagnosis', *Neurocomputing*, Vol. 417, pp.36–63.

Karl, F., Pielok, T., Moosbauer, J. et al. (2023) 'Multi-objective hyperparameter optimization in machine learning – an overview', *ACM Transactions on Evolutionary Learning and Optimization*, Vol. 3, No. 4, pp.1–50.

Locatelli, M. and Schoen, F. (2021) '(Global) optimization: historical notes and recent developments', *EURO Journal on Computational Optimization*, Vol. 9, p.100012.

Luengo, J., Fernández, A., García, S. et al. (2011) 'Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling', *Soft Computing*, Vol. 15, pp.1909–1936.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M. et al. (2015) 'Deep learning applications and challenges in big data analytics', *Journal of Big Data*, Vol. 2, pp.1–21.

Obrezanova, O., Csányi, G., Gola, J.M. et al. (2007) 'Gaussian processes: a method for automatic QSAR modeling of ADME properties', *Journal of Chemical Information and Modeling*, Vol. 47, No. 5, pp.1847–1857.

Sampath, V., Maurtua, I., Aguilar Martin, J.J. et al. (2021) 'A survey on generative adversarial networks for imbalance problems in computer vision tasks', *Journal of Big Data*, Vol. 8, pp.1–59.

Song, C., Cao, J., Zhao, Q. et al. (2024) 'A high-precision crown control strategy for hot-rolled electric steel using theoretical model-guided BO-CNN-BiLSTM framework', *Applied Soft Computing*, Vol. 152, p.111203.

Song, Q., Guo, Y. and Shepperd, M. (2018) 'A comprehensive investigation of the role of imbalanced learning for software defect prediction', *IEEE Transactions on Software Engineering*, Vol. 45, No. 12, pp.1253–1269.

Surianarayanan, C., Lawrence, J.J., Chelliah, P.R. et al. (2023) 'A survey on optimization techniques for edge artificial intelligence (AI)', *Sensors*, Vol. 23, No. 3, p.1279.

Susan, S. and Kumar, A. (2021) 'The balancing trick: optimized sampling of imbalanced datasets – a brief survey of the recent State of the Art', *Engineering Reports*, Vol. 3, No. 4, p.e12298.

Thabtah, F., Hammoud, S., Kamalov, F. et al. (2020) 'Data imbalance in classification: experimental evaluation', *Information Sciences*, Vol. 513, pp.429–441.

Wu, J., Chen, X-Y., Zhang, H. et al. (2019) 'Hyperparameter optimization for machine learning models based on Bayesian optimization', *Journal of Electronic Science and Technology*, Vol. 17, No. 1, pp.26–40.

Xu, Z., Guo, Y. and Saleh, J.H. (2021) 'Efficient hybrid Bayesian optimization algorithm with adaptive expected improvement acquisition function', *Engineering Optimization*, Vol. 53, No. 10, pp.1786–1804.

Zhou, Z-H. and Liu, X-Y. (2005) 'Training cost-sensitive neural networks with methods addressing the class imbalance problem', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 1, pp.63–77.