


## Article

# Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset

Jeyabharathy Sadaiyandi <sup>1</sup>, Padmapriya Arumugam <sup>1,\*</sup>, Arun Kumar Sangaiah <sup>2,3</sup> and Chao Zhang <sup>4,\*</sup> 

<sup>1</sup> Department of Computer Science, Alagappa University, Karaikudi 630003, India; jeyabharathys\_phdscholar@alagappauniversity.ac.in

<sup>2</sup> International Graduate School of AI, National Yunlin University of Science and Technology, Douliu 64002, Taiwan; arunks@yuntech.edu.tw

<sup>3</sup> Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon

<sup>4</sup> School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

\* Correspondence: padmapriyaa@alagappauniversity.ac.in (P.A.); czhang@sxu.edu.cn (C.Z.)

**Abstract:** Due to the imbalanced nature of datasets, classifying unbalanced data classes and drawing accurate predictions is still a challenging task. Sampling procedures, along with machine learning and deep learning algorithms, are a boon for solving this kind of challenging task. This study's objective is to use sampling-based machine learning and deep learning approaches to automate the recognition of rotting trees from a forest dataset. Method/Approach: The proposed approach successfully predicted the dead tree in the forest. Seven of the twenty-one features are computed using the wrapper approach. This research work presents a novel method for determining the state of decay of the tree. The process of classifying the tree's state of decay is connected to the issue of unequal class distribution. When classes to be predicted are uneven, this frequently hides poor performance in minority classes. Using stratified sampling procedures, the required samples for precise categorization are prepared. Stratified sampling approaches are employed to generate the necessary samples for accurate prediction, and the precise samples with computed features are input into a deep learning neural network. Finding: The multi-layer feed-forward classifier produces the greatest results in terms of classification accuracy (91%). Novelty/Improvement: Correct samples are necessary for correct classification in machine learning approaches. In the present study, stratified samples were considered while deciding which samples to use as deep neural network input. It suggests that the proposed algorithm could accurately determine whether the tree has decayed or not.

**Keywords:** machine learning; deep learning; imbalanced datasets; stratified sampling; prediction; classification; accuracy; wrapper classes



**Citation:** Sadaiyandi, J.; Arumugam, P.; Sangaiah, A.K.; Zhang, C. Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset. *Electronics* **2023**, *12*, 4423. <https://doi.org/10.3390/electronics12214423>

Academic Editor: Heung-Il Suk

Received: 12 September 2023

Revised: 21 October 2023

Accepted: 25 October 2023

Published: 27 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In data mining and machine learning, classification analysis is a well-researched method. Because of its ability to forecast future outcomes, it is used in a wide range of real-world scenarios. However, classification accuracy is directly proportional to the training data quality utilized. Real-world data frequently has an imbalanced class distribution, with the dominating majority class and ignoring the least ones.

When dealing with an imbalanced class distribution problem, selecting appropriate training data becomes crucial for improving classification accuracy. When all the available data are used for training, the resulting classifier tends to predict most of the incoming data as belonging to the majority class. This leads to the misclassification of minority class instances. Hence, careful selection of training data is essential to address the challenges posed by imbalanced class distributions in classification problems. In the context of forest ecosystems, the need for accurate classification algorithms cannot be overstated. Forests are a critical component of the planet's ecological balance, sequestering and storing massive

amounts of carbon from the atmosphere. The carbon stored in forest biomass is a crucial element of healthy forest ecosystems and the global carbon cycle.

Forests store carbon in various forms that can be challenging to accurately quantify. The estimation of carbon storage in forests depends on several factors, including the density of tree wood, decay class, and density reduction factors. Accurate estimations of carbon storage in forests are essential for effective carbon flux monitoring. Moreover, the classification of forest data is critical in determining the health and productivity of forest ecosystems. Forest classification algorithms can help identify various features of forests, such as tree species, forest density, and biomass, which are essential in monitoring changes in forest structure and function.

Forest-based accurate classification can also help to predict the occurrence and spread of forest disturbances like wildfires, insect infestations, and diseases. Such disturbances can cause significant losses of carbon from forests, negatively impacting the planet's ecological balance. Therefore, the development of accurate and robust classification algorithms for forest datasets is critical for maintaining healthy forest ecosystems and mitigating the impact of natural disasters on the environment. In the realm of predicting tree decay rates in forests, past research has mainly focused on using regression techniques. However, these methods may not be suitable for distinguishing individual dead trees within a forest.

Deep neural network (DNN) architecture is aimed at detecting individual dead trees within the forest more accurately in this study. For that, this research work proposed a novel approach to deal with imbalanced datasets using sampling techniques. The imbalanced nature of forest datasets can make predictions less accurate, particularly when most data points belong to a single class (e.g., living trees). Therefore, by employing sampling techniques, we balanced the dataset, which improved the accuracy of predictions for both dead and living trees. This ultimately improves the accuracy of predictions made with unbalanced forest datasets. The organization of this research work is as follows. The dataset used for this research work is described first. Then, we employ a DNN with sampling techniques to forecast both dead and living trees. This method was then compared to other techniques for its efficacy. Finally, we present our findings and future directions.

Overall, the development of DNN architecture for predicting individual dead trees in forests, coupled with sampling techniques to handle imbalanced datasets, can raise prediction accuracy and contribute to better forest management. It enables forest managers to conserve and protect the forest ecosystem by making informed decisions.

## 2. Literature Review

In general, the process of classifying unbalanced datasets consists of three steps: selecting features, fitting the data distribution, and training a model. The review of the literature is presented below in Table 1.

**Table 1.** Background study.

Reference	Methodology Used	Observations
[1]	CNN	Outlines the open research problems like enhancing the accuracy of tree species classification, applying the approach to various forest types, exploring its potential for estimating forest characteristics, and creating an easy-to-use tool for forest managers and conservationists.
[2]	SMOTE	The approach neglects to consider the computational cost and resource requirements of various algorithms. These resource requirements could be critical in real-world deployment scenarios.
[3]	Stratified with SVM	Limit its scalability to large datasets.
[4]	Classification using SVM and DNN	DNN shows low accuracy.

Table 1. Cont.

Reference	Methodology Used	Observations
[5]	Undersampling	Undersampling may lead to the loss of some useful information by removing significant patterns.
[6]	Oversampling	The performance of this approach may be influenced by the hyperparameters selected for the DCGAN and CNN models. The hyperparameters used in this model were not extensively optimized in this study.
[7]	Synthesizing data using Variational Auto Encoders (VAE) on raw training samples.	Detailed analysis of the computational cost of the proposed method was provided, which may be a concern for large datasets.
[8]	SMOTE	This approach did not consider the impact of SMOTE on real-world data.
[9]	Snag persistence Forest inventory model	This research work did not address the impact of tree species or decay stage on volume estimation accuracy.

The goal of feature selection is to identify subsets of features that are most suited for classifying the unbalanced data while considering the feature class imbalance. This contributes to the development of a more efficient classifier [10–13]. To limit the impact of class imbalances on the classifier, most data preparation procedures, such as various resampling techniques, are used to adjust the data distribution [14–17]. These techniques significantly balance the datasets.

Model training to accommodate unequal data distribution requires primarily adding an enforcement algorithm to an existing classification approach or applying ensemble learning. Standard cost-sensitive learning is an example of the latter [18–20]; it improves minority class classification accuracy by increasing the weights of the class samples. Classification accuracy can be achieved via ensemble learning techniques like boosting and bagging [21–23].

Distribution-level data resampling will resolve the class imbalance. The most significant advantage of this methodology is that the sampling method and the classifier training procedure are independent of one another. Typically, the sample distribution of the training set is changed at the data preprocessing stage to decrease or eliminate class imbalance. The representative methods consist of a few resampling strategies, with the two main categories being oversampling and undersampling.

Oversampling entails adding appropriately created new points to increase the sample points in a minority class to attain sample balance. The synthetic minority oversampling method (SMOTE) and several of its variants, as well as ROS, are examples of prevalent algorithms [24]. SMOTE generates synthetic samples and inserts them between a given sample and its neighbors, whereas datasets are balanced by ROS by adding minority sample points at random.

$$X_{new} = X_j + rand(0, 1) \cdot (X_i - X_j)$$

In Equation (1),  $X_j$  ( $j = 1, 2, 3, 4, 5$ ) represents a minority class point,  $X_{new}$  represents the generated virtual samples based on the nearest neighbors  $X_i$ , and  $rand(0, 1)$  is a random number between 0 and 1 [4].

The earlier study relied heavily on local data to increase sample sizes. Although the number of samples is equalized, since the information on the overall distribution of the data is not taken into consideration, the data distribution of the new dataset following oversampling cannot be guaranteed. Furthermore, utilizing an oversampling approach may result in a big amount of redundant information, increasing the classifier's calculation and training time.

Undersampling decreases the sample size in a majority class by eliminating some of them, and therefore has the apparent benefit of shortening training time. The most basic

undersampling approach is RUS [24], which discards majority class samples at random. To balance the magnitude of primary class samples with the least class samples, another undersampling strategy uses appropriate majority class samples. The training set will be more evenly distributed because of this method, which will also improve the classification accuracy of minority class samples. The disadvantage is that a sizable portion of the majority class sample characteristics could be lost, and the model might not fully acquire the majority class sample properties. As a result, it is crucial to make sure that the learning process is set up so that the bulk of the information covered in class is retained.

### 3. Materials and Methods

This research work is aimed at predicting the decay information of forest trees. Healthy trees absorb the harmful carbon dioxide and emit oxygen. Trees are the carbon sink of our planet. At the same time, decayed and fallen trees emit carbon dioxide. So, the identification of the decay level of a tree is essential to preserve the ecological condition of our planet. In this research work, details about trees in a forest are examined. Several attributes are associated with forest trees. The age of a tree is usually determined by its wood density. During the initial years, the wood density is increasing, and after attaining normal growth, the wood density starts decreasing. Based on the wood density, the trees are classified into five different decay classes ranging from “freshly killed” to “extremely decayed”. The dead trees fall, may cause forest fires, and it may take several years to decompose. Here, the dataset is preprocessed first to compute wood density and identification of decay class (either Not yet decayed or Decayed) using the wrapper method. Due to the imbalance in the dataset after decay class identification, stratified sampling is used to overcome this issue without losing any inputs [25]. The stratified sampled input is fed to the DNN network for drawing predictions about the decay class of a tree.

This section contains a description of the proposed methodology, a description of the forest tree dataset, and the preparation process. The architecture of the proposed stratified sampling-based deep neural networks approach for increasing the prediction accuracy of the unbalanced dataset is shown in Figure 1.

The proposed methodology can be categorized into three phases.

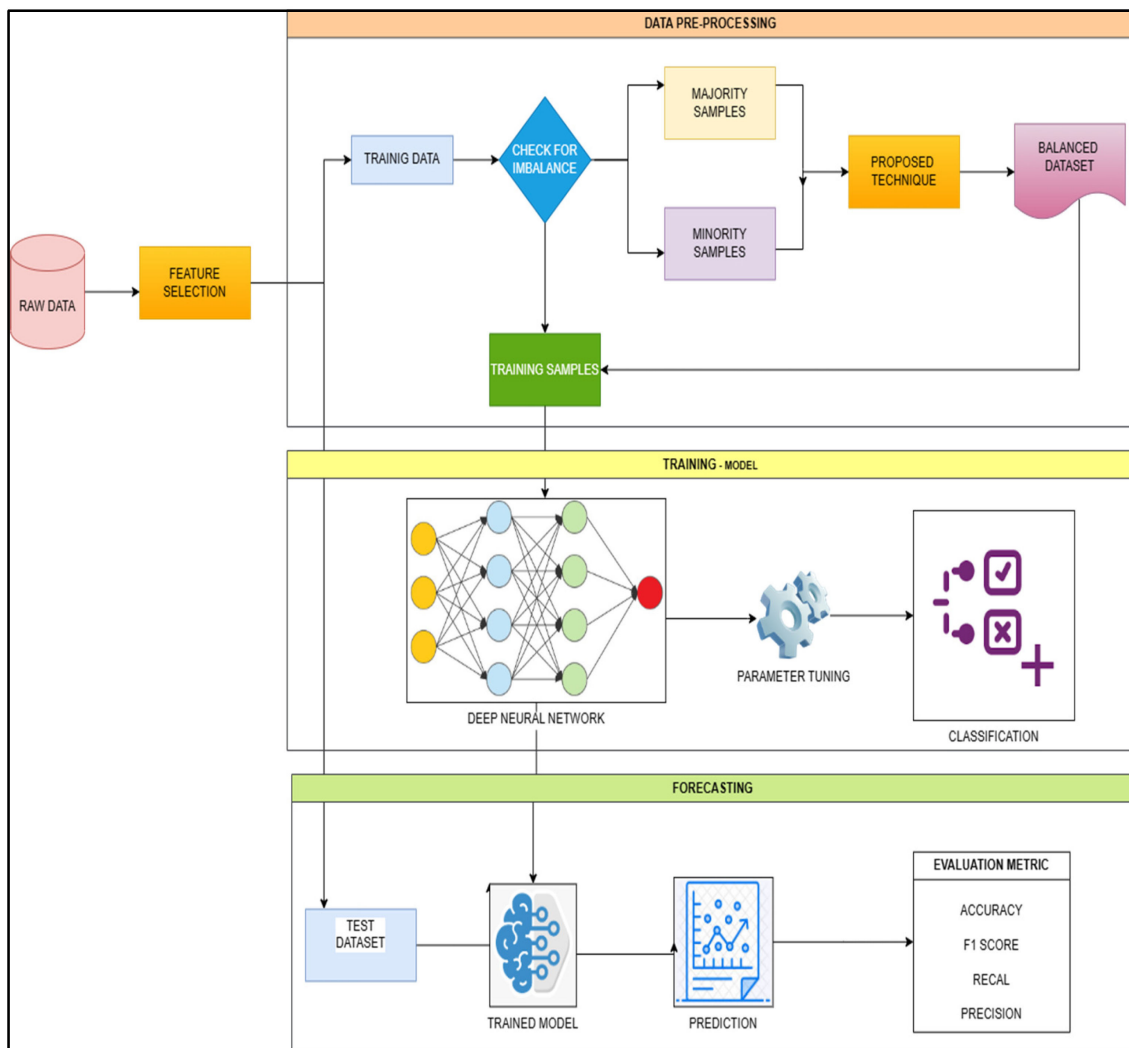
- Data preprocessing phase;
- Training phase;
- Test phase.

The neural network is chosen for classification in this research work over the SVM, Random Forest, and Naïve Bayes because of its ability to handle imbalanced data, feature learning capabilities, model nonlinear relationships, and the ability to fine-tune hyperparameters for optimal performance.

#### 3.1. Description of the Dataset

The dataset was obtained from the USDA repository [26]. Data collection began in 1985 and is expected to last until 2050. The Douglas fir, red cedar, Pacific silver fir, and Western hemlock tree are the four species used for investigation. The data gathered for this study compare the breakdown of tiny logs (20–30 cm in diameter and 2 m in length) in a stream channel at the H.J. Andrews Experimental Forest to that of logs on an adjacent upland site. Above the intersection of Mack Creek and Lookout Creek, the stream is of the third order. A portion of the logs are periodically resampled to assess changes in volume, bark cover, density, and nutrient reserves. Dry mass and volume, as determined by dimensional measures, are used to calculate wood density. Table 2 shows the attributes in the dataset.

For training and testing, different proportions of the dataset were employed. The decay class and wood density of the relevant species are in the training dataset. Also, the wood density threshold value is present in the training dataset. The test data includes information on four species, including circumference, tree’s age, volume, dry weight, and moisture. A total of 54,000 instances with 21 attributes are available in the test dataset.



**Figure 1.** Stratified sampling-based deep neural network (SSDNN) approach for predicting decay class of forest trees.

### 3.2. Preprocessing the Dataset

The dataset is preprocessed before the technique is applied. In this forest dataset, the data distribution is uneven among the live and decayed trees. A tree may belong to a not-yet-rotted or a decayed tree group. Out of the 11,387 trees in the dataset, 9132 belong to the not-yet-rotted group, whereas only 2255 trees belong to the decayed trees group. The data can be either overfit or underfit. This kind of uneven data distribution will have a critical impact on the problem of prediction and categorization, so the data need to be preprocessed.

The preprocessing stage consists of feature selection and checking the skewness of the data. This process will help to reduce the time consumption in handling the unbalanced forest dataset.

#### 3.2.1. Feature Selection Method

The dataset is preprocessed with the feature selection approach back elimination for identifying the optimal subset attributes for forecasting the tree’s wood density (Kusy and Zajdel, 2021). Six of the twenty-one features that are essential for prediction were chosen via the wrapper method–back elimination.

**Table 2.** Dataset specifications.

Attributes	Description
Log num	Log number
Species	Four categories of trees in this region
Time	The tree's age in years
Year	Year of the tree
Subtype	Hard, soft, and other tree types
Rad pos	The location of the measurement
D1	Tree circumference
D2	Tree's circumference in various positions
D3	Tree's circumference in various positions
D4	Tree's circumference in various positions
VOL1	Tree's volume
VOL2	Tree's volume
Wet Wt	Weight of the water content in the tree
DRYWT	The dried weight of the tree
MOIST	Wood's moisture content
Decay	The tree's level of decay
WDENSITY	The tree's wood density with respect to vol1
Den2	The tree's wood density with respect to Vol2
Knot Vol	The wood's volume at a knot
Sample Date	Sample collected date
Comments	Other features of the tree

The model is iteratively trained on several subsets of features using the wrapping technique, and the best subset of features is chosen. The choice of the feature subset selection is based on the inferences from the model. A feature selection strategy called backward elimination starts with a model that incorporates all the available features and gradually eliminates the least significant ones until a stopping requirement is met. This strategy, also known as a wrapper, is typically combined with statistical models to choose a subset of important features. By repeatedly removing elements that are the least significant based on the selected significance level, backward elimination assists in identifying the most pertinent characteristics. Table 3 shows the extracted features using feature selection methods for further processing. Before assessing the feature subsets, these strategies train and test the model using a variety of feature combinations. The strategy reduces overfitting and eliminates pointless or unnecessary features to enhance the model's performance and interpretability.

**Table 3.** Reduced attributes after preprocessing the dataset.

Attributes	Description
Species	Four categories of trees in this region
Year	Tree's age
D1	Tree's circumference
VOL1	Tree's volume
DRYWT	The dried weight of the tree
WDENSITY	Tree's wood density based on vol1

In the experimental dataset, the explanatory variables Species, Diameter, Volume, Wet Weight, Dry Weight, and Decay are considered for multiple linear regression, and the target variable is Wood density  $W_i$  of the tree.

The prediction equation is given below.

$$W_i = \beta_0 + \beta_1 \text{Species} + \beta_2 \text{Diameter} + \beta_3 \text{Volume} + \beta_4 \text{Wetwt} + \beta_5 \text{Drywt} + \beta_6 \text{Decay} + \epsilon$$

where, for  $n$  observations

$W_i$  is the dependent variable, and Species, Diameter, Volume, Wetwt, DryWt, and Decay are the explanatory variables,

$\beta_0$  is the y-intercept (constant term)

$\beta_j$  are the slope coefficients for each explanatory variable ( $j$  indicates attribute index)

$\epsilon$  is the model's error term (also known as the residuals)

### 3.2.2. Checking the Skewness of the Data

Classifiers are built up in machine learning to eliminate misclassification errors and, as a result, optimize predictive accuracy. The class imbalance problem, which refers to an uneven distribution of response variable values, is one of the most prevalent issues that influence raw data.

An unbalanced dataset is one in which the number of samples in different classes is highly uneven, making classification difficult. With uneven data, modern machine learning techniques struggle because they focus on reducing error rates serving the dominant class while disregarding the underrepresented group. Classification becomes extremely difficult because the results may be skewed by dominant class values.

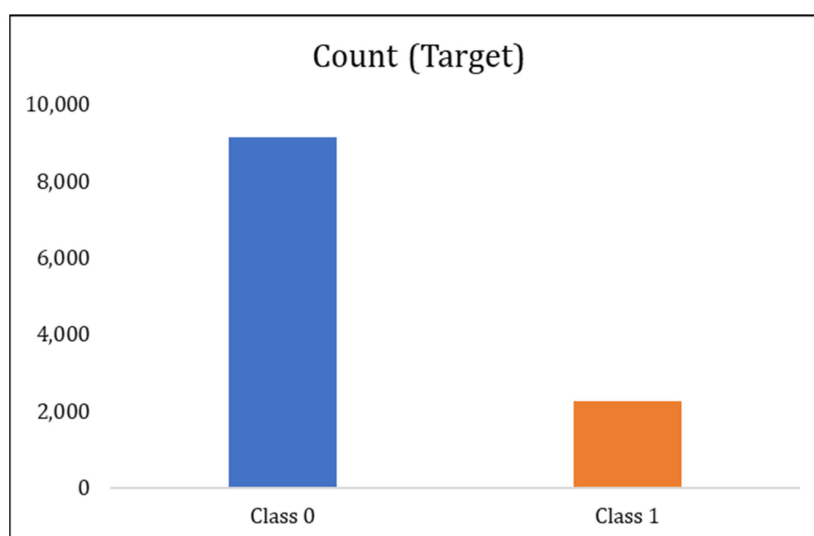
As per the experimental dataset, a tree may belong to any one of the five different decay levels ranging from 1 to 5. If a tree belongs to class 1, it means it is not yet decayed; otherwise, it has a decaying component. Since our aim is to classify trees, we considered only two classes, namely "Not yet Decayed" trees and "Decayed" trees. The dataset is considered for the experimental study of the class imbalance problem. As mentioned earlier, there are possibilities of overfit or underfit.

The class details are given below.

Class 0: 9132 (Not yet Decayed)

Class 1: 2255 (Decayed Trees)

The class imbalance problem in the experimental dataset is depicted in Figure 2.



**Figure 2.** Depiction of class imbalance problem in the experimental dataset.

### 3.3. Stratified Sampling-Based Deep Neural Network (SSDNN) Approach

The process of classifying unbalanced datasets involves three main steps: feature selection, fitting the data distribution, and model training. Feature selection helps to identify the most suitable subsets of features for classifying unbalanced data while considering the class imbalance among the features. Various resampling approaches that minimize the impact of class inequality on the classifier can be used to fit the data distribution.

The most common resampling strategies are oversampling and undersampling. These strategies aim to balance the datasets by increasing or decreasing the sample points in the minority and majority classes, respectively. However, oversampling algorithms may generate duplicate information and increase the training time of the classifier, while undersampling may result in the loss of the majority of class information.

Both random oversampling (ROS) and random undersampling (RUS) violate the law governing data distribution. The generated samples might not be helpful in illustrating the distribution. SMOTE has drawbacks like supersampling the noisy samples and uninformative data. It is highly challenging to determine the closest neighbors of anonymous synthesized samples. Also, the SMOTE samples are always contained within the samples, and pruning them will lead to an increase in misclassification rate.

We proposed stratified random sampling method to resolve said issue, which will perform the task of test input selection for DNNs. According to the sampling theory, stratified random sampling involves dividing a population into smaller groups without any duplication and avoiding records. The proposed method increases the computation efficiency in the reliability evaluation of the model.

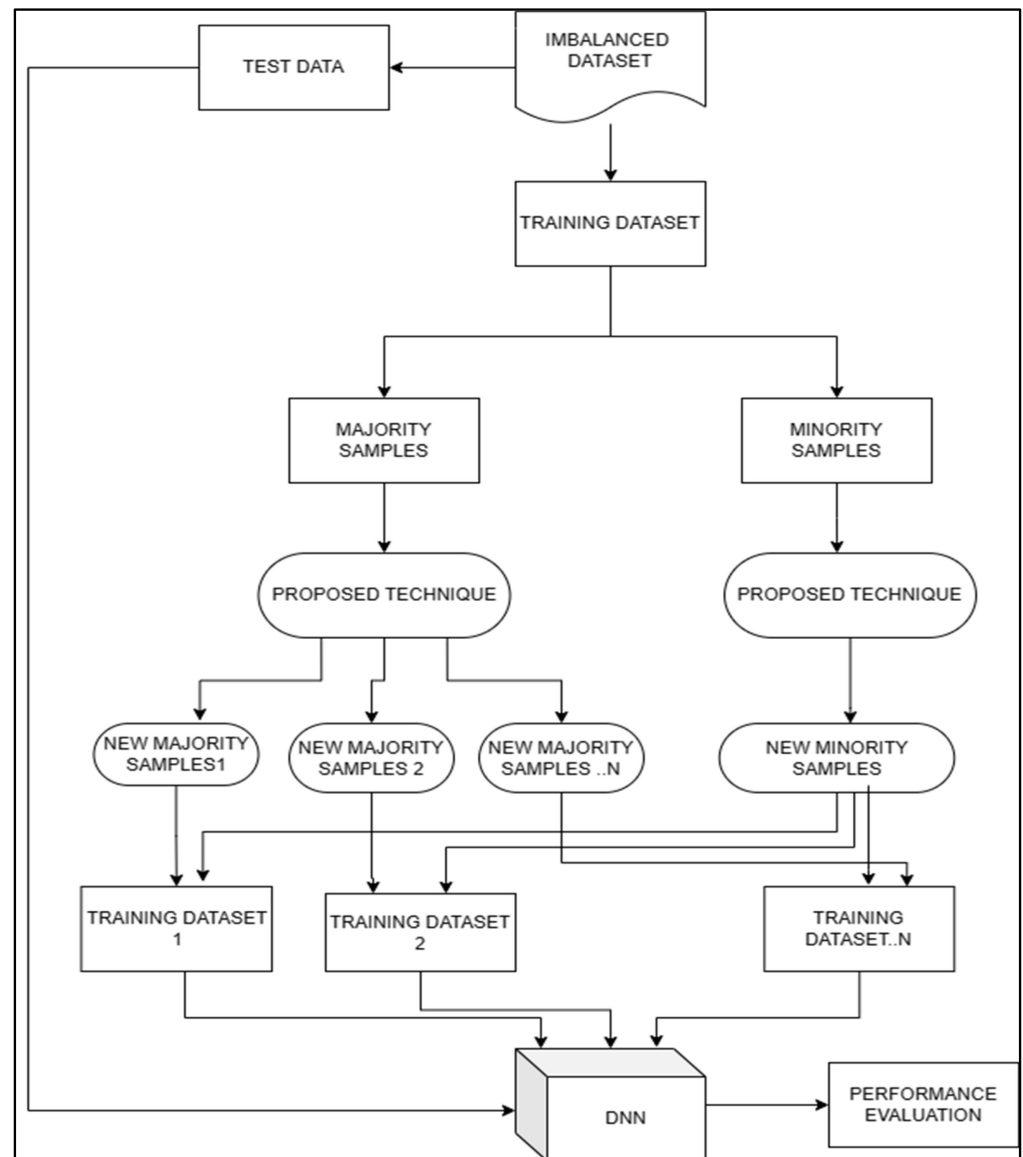
The stratified sampling approach divides the data into blocks based on specified values to extract the structural facts of the data and then draws samples at random from these distinct data blocks. Stratification makes it simple to find representative samples. In the case of forest datasets, stratified sampling can be applied to guarantee that the number of samples for each class is balanced and that the variance of the data within each class is considered when choosing the optimum number of samples. This helps to preserve the original data structure feature information while also ensuring a balance in the number of samples for the majority and minority classes. The specific procedure is to randomly select some examples from both positive and negative occurrences and then combine the training samples for classification. Stratified sampling is best suited for the uneven distribution of data, and it is applied to different domains [25,27–29].

The diversified dataset  $N$  is split into similar groups,  $S_0$ ,  $S_1$ , and so forth. For data selection,  $S_n$ , also known as strata, utilizes uniform random or systematic sampling in each stratum. The reduction in estimation error is the primary advantage of stratified sampling over other sampling techniques. Within strata, a sample for data analysis is taken via random sampling after relative homogenous data objects are grouped together based on the necessary parameters.

The Stratified sampling-based deep neural networks approach is shown below in Figure 3.

Deep learning is a feed-forward neural network with one or more hidden layers. Deep learning is a subfield of machine learning that emphasizes the use of numerous linked layers to transform input into features and predict associated outputs. Artificial neural networks are the core of it. Input, output, and numerous hidden layers are all present in deep neural networks (DNNs). The hidden layer is in the middle, after the input layer and preceding the output layer. In the training of a deep neural network, the following steps are taken: first, initialization is performed according to requirements, and the structure of the DNN is set; second, the layer is then communicated between layers to obtain an error using forward; and finally, the layer is transferred between layers to obtain an error using forward.





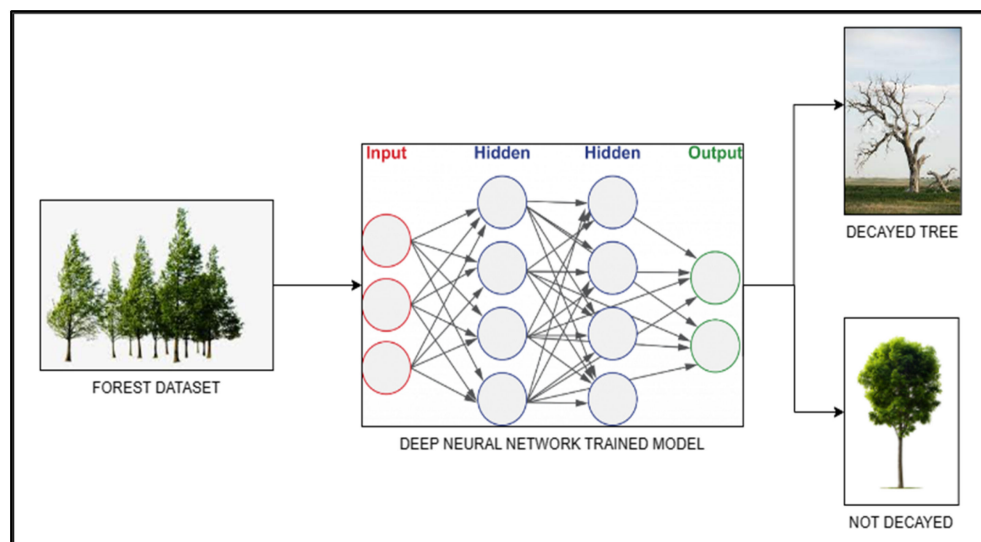
**Figure 3.** SSDNN model.

DNNs can handle both linear and nonlinear issues by monitoring the probability of each output layer by layer with an appropriate activation function. In essence, DNNs are fully linked neural networks. A deep neural network is sometimes known as a multi-layer perceptron (MLP). The hidden layer alters the input feature vectors, which eventually arrive at the output layer, where the binary classification result is obtained.

Environments have been interested in determining functional links between carbon storage and plant uncertainty of wood density; an appropriate technique is required. Developing empirical models to forecast the DECAY CLASS of the tree is the focus of this research. A deep neural network, a subset of expert systems, predicts the DECAY CLASS of the tree more accurately than standard models. Because there was no constraint for constructing models in DNN, the outcomes are more accurate predictions than the ensemble model. The data loss was achieved using the training data, as shown in the topology of the model, implying that there was no overfitting.

The suggested work's learning model has four layers: one input layer, two hidden layers, and one output layer is shown in Figure 4. At the last three layers, the ReLu activation function was utilized, and the sigmoid function was used at the output layer. The binary cross-entropy loss between the input was used to establish the objective function,

which should be minimized in the NN. Adam's optimization was chosen above other existing optimization techniques because it was more efficient. To create a model, each dataset was first randomly divided into two parts: a 75% training set and a 25% test set.



**Figure 4.** Prediction of tree decay class using DNN model.

The training set is examined for skewness and, if necessary, balanced using a stratified sampling procedure. The balanced training set is then used to develop DNN models and train them, while the test sets are utilized to evaluate the performance of the predictive models. We used the following easy method to choose the best threshold. The curve of balanced accuracy as a function of prediction is first plotted. The best threshold was finally determined to be where the DNN achieved the most balanced accuracy. The unbalanced learn library from Python was then used to apply each data-balancing technique to each training batch. The model has been tried with different numbers of mini-batches as 10, 50, 25, and 100 and determined 100 as the best choice with epoch sizes as 10, 25, and 50.

### 3.3.1. Algorithm for SSDNN Model

The algorithm for the proposed SSDNN model is given in Algorithm 1. This proposed SSDNN model will first extract the features required for the job and verify whether the ratio of the dataset is unbalanced or not. Next, it will choose the right samples for prediction. Below is a representation of the suggested model algorithm.

---

#### Algorithm 1. Proposed Algorithm for SSDNN Model

---

1. Import the dataset
  2. Perform the Wrapper method (Back Elimination Method)
  3. Check the Skewness of the dataset
  4. Apply Stratified
  5.       Update the imbalanced dataset
  6. Load the training dataset
  7. Train the DNN
  8.       Shuffle and Split as 75% and 25%
  9.       Use SVM-Kernel for classification
  10. Tune the Parameters
  11. Apply to the test dataset
  12. End
-

### 3.3.2. DNN with Hyperparameter Tuning

Deep neural network hyperparameter tuning employs a random search to identify the ideal hyperparameter combination from a set of hyper parameter values. Random search resulted in a set of 20 hyperparameter combinations. The following are the best hyperparameters found via random search.

Finally, the model is hyper-tuned using a random search approach, where the optimum parameters are 2 hidden layers, 400 neurons, ReLu, 50 epochs, and a batch size of 100 as listed in Table 4.

**Table 4.** DNN hyperparameters.

Hyperparameter	Value/Type
Hidden Layers	2
Neurons	400
Optimizer	Adam
Hidden Layer	ReLu
Output Layer	Softmax
Epochs	10, 25, 50
Batch size	100

When the number of epochs increases, the accuracy of the proposed method also increases, and we obtain maximum accuracy when the epoch is closer to 100. The built model is compared with the existing models, and the performance is analyzed in the results and discussion section.

## 4. Experiment Results and Discussion

To recognize dead and live trees, we used the forest tree dataset to perform our classification. The dataset was preprocessed to determine the relevance of the variables for categorization. The dataset was split into two parts: training and testing. We used a training dataset to train the DNN and a test set to evaluate classifier performance. We conducted a huge number of trials to discover the ideal DNN design and parameters, using various combinations of batch sizes, number of hidden units, and learning rate.

Because of the imbalanced dataset, DNN accuracy is good, but other performance metrics like F1Score, Precision, and Recall value are low. As a result, the dataset is balanced via stratified sampling, and the resulting strata are supplied to DNN as a training set. The result of the proposed model is compared with the previous model SVM, Naïve Bayes, and Random Forest. Earlier, we tried to perform the classification using these three models with different datasets. Each model has its own credits and pitfalls. For the smaller datasets, SVM produces better results but is not promising for larger ones. At the same time, Random Forest is one of the best choices for larger datasets but is time consuming. Naïve Bayes is simple, and it is not preferred for large datasets. It assumes that the variables are independent.

It is evident from the results that the proposed model gives high accuracy in addition to performing well in the case of large datasets. The proposed approach is written in Python using Jupyter Notebook and uses the Keras package on a 64-bit OS with an X64 CPU, and the model worked well on the Google Lab platform. Thus, by combining DNN with a stratified sampling-based deep learning model, the prediction and classification of dead trees in the forest are successfully completed. Forest managers will be able to predict the early stages of decaying trees with this information. The proposed method can also be applied to similar datasets belonging to different domains.

#### 4.1. Performance Metrics

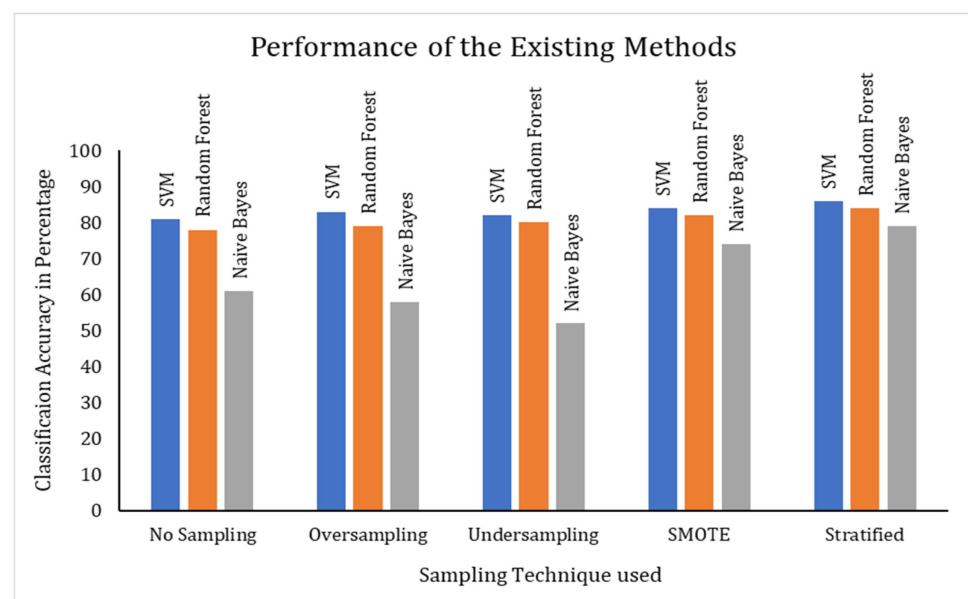
The efficiency of the proposed method is analyzed using classification accuracy, Precision, Recall, and F1 Score. The performance of the three approaches, namely SVM, Random Forest, and Naïve Bayes, with different sampling techniques, are depicted in the following figure.

#### 4.2. Results and Discussion

Table 5 shows the comparison of test accuracy among the proposed DNN models with sampling methods. The performance in terms of accuracy of the existing and proposed algorithms along with different sampling techniques are shown in Figure 5.

**Table 5.** Comparison between proposed and existing methods.

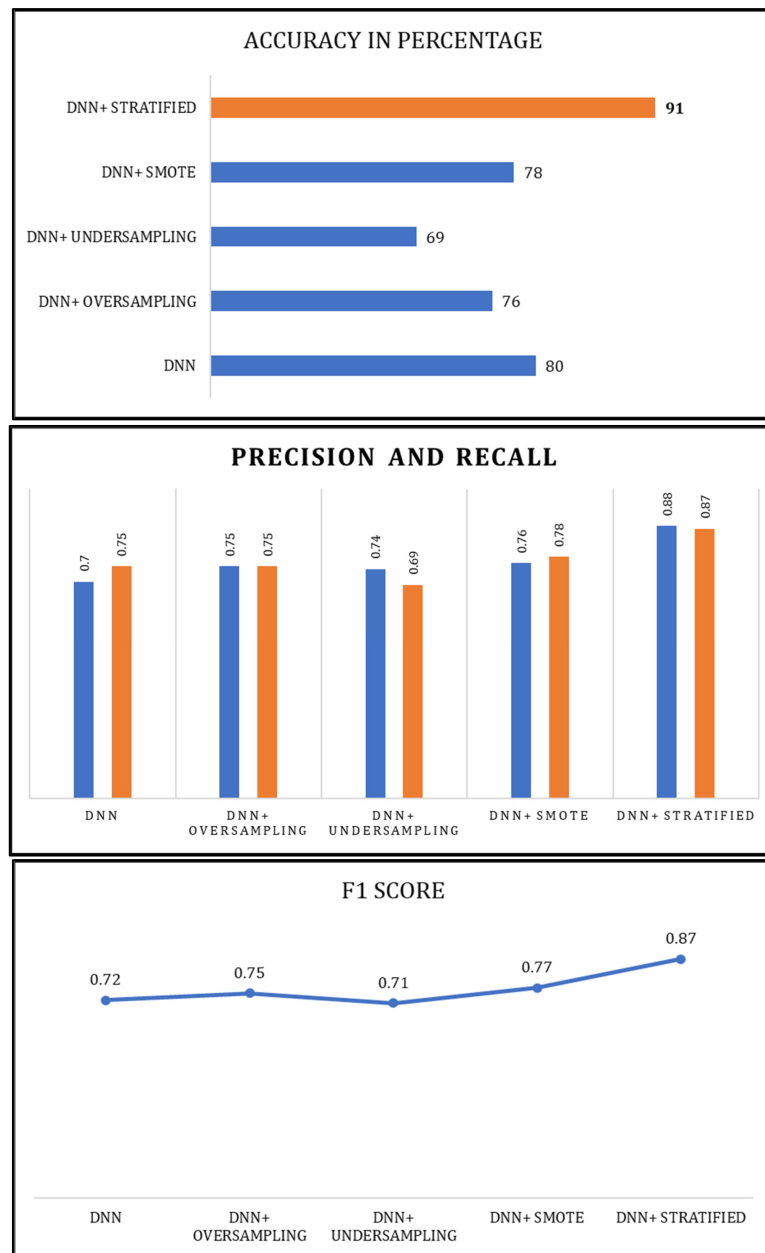
Methods	Accuracy	Precision	Recall	F1 Score
DNN	80	0.7	0.75	0.72
DNN+OVERSAMPLING	76	0.75	0.75	0.75
DNN+UNDERSAMPLING	69	0.74	0.69	0.71
DNN+SMOTE	78	0.76	0.78	0.77
DNN+STRATIFIED	91	0.88	0.87	0.87



**Figure 5.** Performance of the existing approaches with different sampling methods.

The performance of the proposed SSDNN method with different existing sampling techniques is shown in Figure 6.

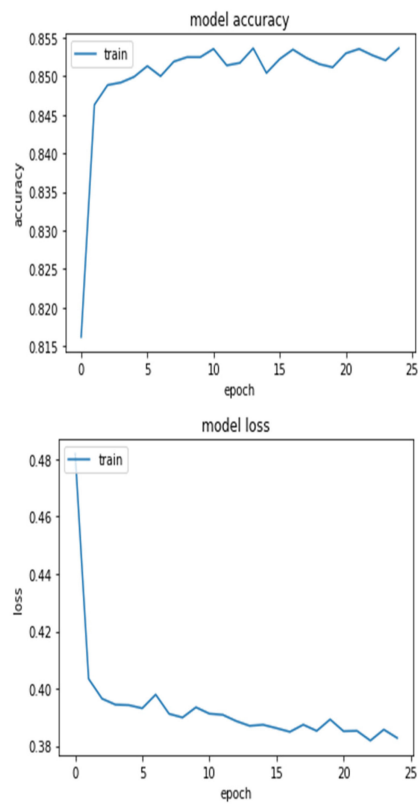
The DNN, DNN+ oversampling, DNN+ undersampling, DNN+ SMOTE, and DNN+ stratified sampling yields test accuracy of 80%, 76%, 69%, 78%, and 91%, respectively. First, the DNN model was created and tested on the prepared dataset, yielding low accuracy. The DNN model was analyzed for the reason of yielding low accuracy, and it was found that the dataset was unbalanced. The imbalanced dataset was subsequently handled using a stratified sampling technique, which divided the training dataset into groups of distinct strata for each class. The data from each stratum was distributed uniformly to the deep neural network, resulting in good accuracy, precision, recall, and F1 score. Several tests using the tree dataset were carried out to determine the optimal deep neural network.



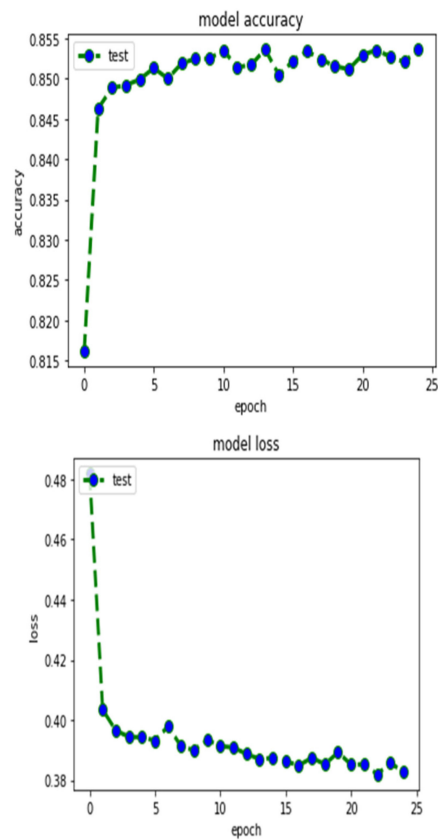
**Figure 6.** Comparison between existing methods with proposed SSDNN.

The training, as well as testing accuracy and loss of the proposed SSDNN, is visualized in Figure 6. From the figure, during the initial epochs, accuracy is not appreciable, and at the same time, the loss is highly noticeable. But in the subsequent epochs, the results are more promising. Similarly, the same parameters are analyzed for the testing phase. The testing phase also has the same impact on model accuracy and model loss. To observe the variations more clearly, the chart is prepared up to 25 epochs.

Also, the training/testing accuracy and loss of the proposed method is shown in Figure 7. The proposed DNN + stratified sampling results in an accuracy of 91% with higher efficiency. The proposed model was compared to the ensemble SVM kernel algorithm used in prior work, and the results show that the proposed DNN + stratified model is more efficient. The proposed method is robust compared to the traditional methods due to hyper-tuning, low false positive, and high recall.



(a) Training accuracy and training loss of SSDNN



(b) Testing accuracy and testing loss of SSDNN

**Figure 7.** Performance in terms of training/testing accuracy, as well as loss of the proposed SSDNN.

## 5. Conclusions

In this research, we experimented to find the best model to classify the forest tree as a dead or live tree. For predicting the decay class of a tree, the classification models DNN, DNN+ oversampling, DNN+ undersampling, DNN+ SMOTE, and DNN+ stratified sampling were applied to the dataset. The results show that DNN+ stratified sampling offers better performance with high accuracy.

The proposed method correctly classifies a tree as either dead or alive compared to other models. The proposed model will be suitable to handle any imbalanced dataset for classification. In deep learning, classification accuracy often increases when the amount of data used for training increases; thus, using a larger dataset for training can be a good research direction to continue improving our classification accuracy of forest tree classification. This paper suggests that identifying decaying trees earlier will help forest managers in removing them before they begin to emit carbon back into the atmosphere.

This research promotes reforestation by planting a new tree after removing a dead tree to reduce pollution and forest fires. In the case of stratified sampling, the research gap discovered is that the number of records in both classes is not equal; hence, deficit records occur when training the model. To address this issue, the deficit class is oversampled, strata are shuffled, and the model is trained to increase model efficiency. In future work, the proposed method can be applied to smart forest management. Since there may be uneven data or irrelevant data during data collection, we can use IOT-based RFID for each tree to automate data collection for the tree and also to indicate its level of decay and carbon absorption.

**Author Contributions:** Conceptualization, P.A.; methodology, J.S. and P.A.; validation, A.K.S. and C.Z.; writing—original draft preparation, P.A. and J.S.; writing—review and editing, A.K.S. and C.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Rashtriya Uchchatar Shiksha Abhiyan (RUSA) Phase 2.0 [grant sanctioned vide Letter No.F.24-51/2014-U, Policy (TNMulti-Gen), Department of Education, Government of India, Date 9 October 2018].

**Data Availability Statement:** <https://andrewsforest.oregonstate.edu/data> (accessed on 11 September 2023).

**Conflicts of Interest:** The authors declare that they have no conflict of interest.

## References

1. Briechle, S.; Krzystek, P.; Vosselman, G. Silvi-Net—A dual-CNN approach for combined classification of tree species and standing dead trees from remote sensing data. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *98*, 102292. [CrossRef]
2. Karatas, G.; Demir, O.; Sahingoz, O.K. Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. *IEEE Access* **2020**, *8*, 32150–32162. [CrossRef]
3. Cao, L.; Shen, H. CSS: Handling imbalanced data by improved clustering with stratified sampling. *Concurr. Comput. Pr. Exp.* **2020**, *34*, e6071. [CrossRef]
4. Li, K.; Chen, X.; Zhang, R.; Pickwell-MacPherson, E. Classification for Glucose and Lactose Terahertz Spectrums Based on SVM and DNN Methods. *IEEE Trans. Terahertz Sci. Technol.* **2020**, *10*, 617–623. [CrossRef]
5. Mînaştireanu, E.-A.; Meşniţă, G. Methods of Handling Unbalanced Datasets in Credit Card Fraud Detection. *BRAIN. Broad Res. Artif. Intell. Neurosci.* **2020**, *11*, 131–143. [CrossRef]
6. Shoohi, L.M.; Saud, J.H. DCGAN for Handling Imbalanced Malaria Dataset based on Over-Sampling Technique and using CNN. *Medico-Legal Update* **2020**, *20*, 1079–1085.
7. Sheikh, T.S.; Khan, A.; Fahim, M.; Ahmad, M. Synthesizing data using variational autoencoders for handling class imbalanced deep learning. In Proceedings of the International Conference on Analysis of Images, Social Networks and Texts, Kazan, Russia, 17–19 July 2019; pp. 270–281.
8. Elreedy, D.; Atiya, A.F. A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. *Inf. Sci.* **2019**, *505*, 32–64. [CrossRef]
9. Oberle, B.; Ogle, K.; Zanne, A.E.; Woodall, C.W. When a tree falls: Controls on wood decay predict standing dead tree fall and new risks in changing forests. *PLoS ONE* **2018**, *13*, e0196712. [CrossRef]

10. Tallo, T.E.; Musdholifah, A. The Implementation of Genetic Algorithm in Smote (Synthetic Minority Oversampling Technique) for Handling Imbalanced Dataset Problem. In Proceedings of the 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 7–8 August 2018; pp. 1–4. [[CrossRef](#)]
11. Moayedikia, A.; Ong, K.-L.; Boo, Y.L.; Yeoh, W.G.; Jensen, R. Feature selection for high dimensional imbalanced class data using harmony search. *Eng. Appl. Artif. Intell.* **2017**, *57*, 38–49. [[CrossRef](#)]
12. Maldonado, S.; López, J. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. *Appl. Soft Comput.* **2018**, *67*, 94–105. [[CrossRef](#)]
13. Maldonado, S.; Weber, R.; Famili, F. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Inf. Sci.* **2014**, *286*, 228–246. [[CrossRef](#)]
14. Ng, W.W.; Hu, J.; Yeung, D.S.; Yin, S.; Roli, F. Diversified sensitivity-based under-sampling for imbalance classification problems. *IEEE Trans. Cybern.* **2014**, *45*, 2402–2412. [[CrossRef](#)] [[PubMed](#)]
15. Sáez, J.A.; Krawczyk, B.; Woźniak, M. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recogn.* **2016**, *57*, 164–178. [[CrossRef](#)]
16. González, S.; García, S.; Lázaro, M.; Figueiras-Vidal, A.R.; Herrera, F. Class Switching according to Nearest Enemy Distance for learning from highly imbalanced data-sets. *Pattern Recognit.* **2017**, *70*, 12–24. [[CrossRef](#)]
17. Cao, L.; Shen, H. Imbalanced data classification using improved clustering algorithm and under-sampling method. In Proceedings of the 20th International Conference on Parallel and Distributed Computing, Applications and Technologies, Gold Coast, Australia, 5–7 December 2019.
18. Cheng, F.; Zhang, J.; Wen, C.; Liu, Z.; Li, Z. Large cost-sensitive margin distribution machine for imbalanced data classification. *Neurocomputing* **2016**, *224*, 45–57. [[CrossRef](#)]
19. Cao, C.; Wang, Z. IMCStacking: Cost-sensitive stacking learning with feature inverse mapping for imbalanced problems. *Knowl.-Based Syst.* **2018**, *150*, 27–37. [[CrossRef](#)]
20. Ohsaki, M.; Wang, P.; Matsuda, K.; Katagiri, S.; Watanabe, H.; Ralescu, A. Confusion-Matrix-Based Kernel Logistic Regression for Imbalanced Data Classification. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1806–1819. [[CrossRef](#)]
21. Sun, Z.; Song, Q.; Zhu, X.; Sun, H.; Xu, B.; Zhou, Y. A novel ensemble method for classifying imbalanced data. *Pattern Recognit.* **2015**, *48*, 1623–1637. [[CrossRef](#)]
22. Feng, W.; Huang, W.; Ren, J. Class Imbalance Ensemble Learning Based on the Margin Theory. *Appl. Sci.* **2018**, *8*, 815. [[CrossRef](#)]
23. Chen, Z.; Lin, T.; Xia, X.; Xu, H.; Ding, S. A synthetic neighborhood generation based ensemble learning for the imbalanced data classification. *Appl. Intell.* **2018**, *48*, 2441–2457. [[CrossRef](#)]
24. Japkowicz, N. The class imbalance problem: Significance and strategies. In Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000), Las Vegas, NV, USA, 26–29 June 2000.
25. Zhao, X.; Liang, J.; Dang, C. A stratified sampling based clustering algorithm for large-scale data. *Knowl.-Based Syst.* **2019**, *163*, 416–428. [[CrossRef](#)]
26. Available online: <https://www.nal.usda.gov/data/find-data-repository> (accessed on 10 October 2023).
27. Wang, W.; Zhao, Y.; Zhang, T.; Wang, R.; Wei, Z.; Sun, Q.; Wu, J. Regional soil thickness mapping based on stratified sampling of optimally selected covariates. *Geoderma* **2021**, *400*, 115092. [[CrossRef](#)]
28. Alogogianni, E.; Virvou, M. Handling Class Imbalance and Class Overlap in Machine Learning Applications for Undeclared Work Prediction. *Electronics* **2023**, *12*, 913. [[CrossRef](#)]
29. Wu, Z.; Wang, Z.; Chen, J.; You, H.; Yan, M.; Wang, L. Stratified random sampling for neural network test input selection. *Inf. Softw. Technol.* **2023**, *165*, 107331. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.