Strategies for Machine Learning on heterogenous datasets

Emirhan Yilmaz

Karlsruhe Institute of Technology

Abstract. Machine learning algorithms are becoming hungry for big datasets. Still, data collectors are often unable to ensure both the quality and quantity of datasets at the same time due to various reasons. This contradiction is even more prominent for some field data collection projects that require the deployment of large sensor networks. In this context, researchers are often confronted with either a union of multiple small datasets from different sources or a large data set littered with mutilations (or even a combination of both in extreme cases). It is exciting that many approaches have been proposed that try to solve this problem from different aspects, for example, by adjusting the data set to make it homogeneous or by modifying the ML algorithm to fit the Heterogeneous datasets. In this paper, we will explore the recent progress on this issue.

Keywords: Machine Learning, Heterogeneous Data sets, Heterogeneous Transfer Learning

1 Introduction

We are living in a time where machine learning has an ever-increasing impact on our daily lives. From speech recognition to production optimization; from self-driving cars to medical diagnosis, we encounter many different machine learning models. Even though these fields of study may look very different and independent from each other, they have one very important thing in common in that their respective machine learning algorithms require training on large datasets of high quality to be accurate with continuous annotations.

With the advancement of sensing and communication technologies, we now have the ability to obtain large amounts of sensor data. However, in order to build an accurate and reliable computational model we still need annotations on these kinds of data. Not only may the data be missing labels, but they also might be heterogeneous. Heterogeneous data is data with high variability of data types and formats. Under real-life conditions, this data may also even be incorrect data. In this paper, one of the points we will talk about is how to effectively train our model on big datasets with a limited amount of annotations and propose solutions to the problems of such heterogeneous data.

Alternatively, we may train our data on a different but related problem. For example, knowledge gained while learning to recognize dogs could apply when trying to recognize cats, where the data set for cats may be heterogeneous

and not alone suitable for training our model. This method is called "Transfer learning". We will also discuss this problem and some similar approaches in this paper.

2 Background

Since the emergence of machine learning and data analysis using machine learning, data heterogeneity has been a significant and unavoidable problem in many fields of study. The research on the impact and the solution space is still actively ongoing in these fields of study. For example, Kumar et al. (2017) try to analyze the heterogeneity of road accident data while Kourou et al. (2014) try to do the same thing for data on cancer prognosis and prediction. In a more obvious use-case of machine learning, big data analysis, Wang (2017) tries to bring more light on the problem of heterogeneity in his article. Evidently, these examples can be multiplied as heterogeneous data naturally arises in every field where machine learning is attempted to be used [1, 2, 3].

3 Types of Heterogeneity

3.1 Missing Values

Missing values are one of the most common phenomena in statistical analysis. Rates of less than 1% missing data are generally expected, a rate of 1-5% is still manageable. However, a rate of 5-15% already requires sophisticated methods to handle, and more than 15% may severely impact any kind of interpretation of the data [4]. Data sets may contain missing values due to various reasons, such as manual data entry procedures, equipment errors and incorrect measurements. Missing values usually appear as "NULL" values in database or as empty cells in spreadsheet table. Some flat-file formats use various symbols for missing values – e.g. ARFF files uses "?" symbol for missing values and in CSV files, NULL values are typically represented by two successive delimiters (e.g. ',,'). These forms of missing values can easily be automatically detected. However missing values can also appears as outliers or wrong data (i.e. out of boundaries). These data must be removed before intended analysis, and are much harder discover [5].

It is possible to classify missing data into three kinds:

- Missing completely at random (MCAR): Missing data are considered MCAR if the probability of an entry missing a value for an attribute is not dependent on the value of the data being observed. Examples of data that is MCAR might include a questionnaire of a person being lost or noise affecting a sensor device reading.
- Missing at random (MAR): Missing data are considered MAR if the probability of an entry missing a value for an attribute is not dependent on the value of the missing attribute itself, but is dependent on other attributes and data. For example, a high-earning household might be less willing to reveal their income in a survey about income and property tax.

– Missing not at random (MNAR): Missing data are considered MNAR if the missingness of the data is related to the value being observed. For example, a person not attending a drug test might be related to the person actually having taken drugs the night before.

ID	Qual.	No. of Bedrooms	Area	YearBuilt	Gar.Area
0	4	3	120	1992	20
1	6	NaN	135	NaN	25
2	8	2	150	1998	32
3	9	NaN	140	2000	35
4	5	4	120	2002	20
5	7	3	15	1993	35
6	8	4	160	1998	30
7	9	3	140	2005	NaN
8	4	2	120	2008	20
9	3	3	130	2000	27

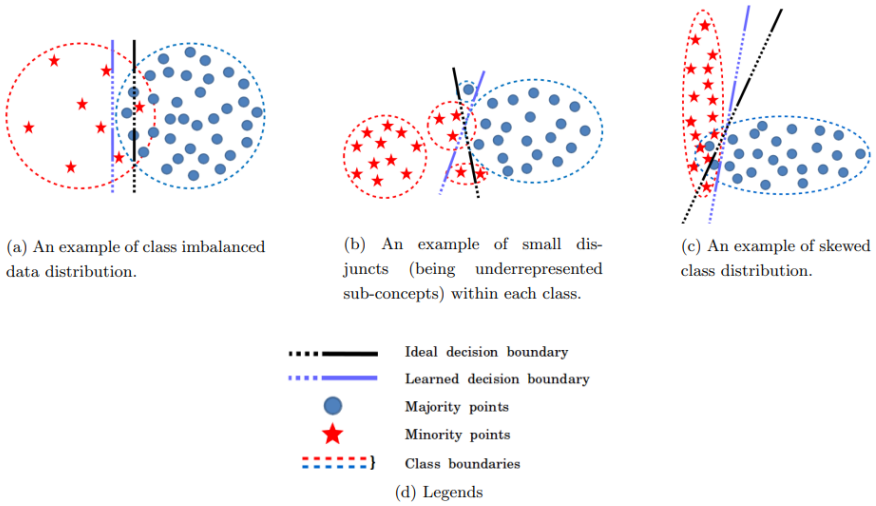
Fig. 1. Example of missing data in a data set [6].

3.2 High Dimensionality

Data sets may contain irrelevant attributes or attributes which have little to no relevance to the training process of the machine learning algorithm. Data sets with irrelevant features are often called high-dimensional data. There are several reasons to reduce the dimensionality of these data sets. First, high-dimensional data impose computational challenges. Moreover, in some situations, high dimensionality might lead to poor generalization abilities of the learning algorithm. Finally, dimensionality reduction can be used for interpretability of the data, for finding the meaningful structure of data, and for illustration purposes Also, as the dimension or the number of features grows, the amount of data needed for training an accurate model grows exponentially. However, if we are not able to provide a sufficient amount of data, sparsity occurs and as a result, finding patterns and grouping properties becomes increasingly difficult. This phenomenon is also called the curse of dimensionality.

3.3 Distribution-based Data Irregularities

Class Imbalance Supervised learning methods require labeled training data, and in classification problems, each data sample belongs to a known class, or category. In a binary classification problem with data samples from two groups, class imbalance occurs when one class, the minority group, contains significantly fewer samples than the other class, the majority group. In many problems, the minority group is the class of interest, i.e., the positive class. A well-known class imbalanced machine learning scenario is the medical diagnosis task of detecting



(a) An example of class imbalanced data distribution.

(b) An example of small disjuncts (being underrepresented sub-concepts) within each class.

(c) An example of skewed class distribution.

(d) Legends

Fig. 2. Examples for distribution-based data irregularities

disease, where the majority of the patients are healthy, and detecting disease is of greater interest. In this example, the majority group of healthy patients is referred to as the negative class. Learning from these imbalanced data sets can be very difficult, especially when working with big data, and non-standard machine learning methods are often required to achieve desirable results.

A thorough understanding of the class imbalance problem and the methods available for addressing it is indispensable, as such skewed data exists in many real-world applications. When a class imbalance exists within training data, learners will typically over-classify the majority group due to its increased prior probability. As a result, the instances belonging to the minority group are misclassified more often than those belonging to the majority group. These side effects make it very difficult to accomplish the typical objective of accurately predicting the positive class of interest. Furthermore, some evaluation metrics, such as accuracy, may mislead the analyst with high scores that incorrectly indicate good performance. Given a binary data set with a positive class distribution of 1a naïve learner that always outputs the negative class label for all inputs will achieve 99% accuracy [7].

Small Disjuncts Rare or exceptional cases correspond to small numbers of training examples in particular areas of the feature space. When learning a concept, the presence of rare cases in the domain is an important consideration. The reason why rare cases are of interest is that they cause small disjuncts to occur which are known to be more error prone than large disjuncts. In more detail learning systems usually create concept definitions that consist of several dis-

juncts. Each disjunct, in turn, is a conjunctive definition of a subconcept of the original concept. The coverage of a disjunct corresponds to the number of training examples in correctly classifies, and a disjunct is considered to be a "small disjunct" if that coverage is low. In fact, small disjuncts are not inherently more error prone than large disjuncts. What makes them more error prone are the bias of the classifiers as well as the effect of attribute noise, missing attributes, class noise and training set size on the rare cases which cause them [8].

Skewed Class Distribution the problem of the dominance of one class in the region of overlap can arise in learning problems irrespective of the presence of global class imbalance. In the absence of class imbalance, such a situation may arise when the class distributions are disparate so that one class is sparse in the region of overlap while the other is abundant. Let us revisit the previous illustration to gain a better understanding of the class distribution skew problem. The disparity in the orientation of the two elliptical classes is evident from the figure. As a result of the disparate orientations, the star class (despite being the minority class) has a greater number of representatives around the region of overlap, resulting in the borderline instances from the circle class being misclassified. Such situations can occur irrespective of the presence of global class imbalance. This results in poor learning in the vicinity of the region of overlap for one or more of the classes [9].

4 Solutions

4.1 Missing Values

Reducing the Data Set: The simplest solution for the missing values imputation problem is the reduction of the data set and elimination of all missing values. This can be done by elimination of samples (rows) with missing values or elimination of attributes (columns) with missing values. Both approaches can be combined. Elimination of all samples is also known as complete case analysis. Elimination of all samples is possible only when large data sets are available, and missing values occur only in a small percentage of samples and when analysis of the complete examples will not lead to serious bias during the inference. Elimination of attributes with missing values during analysis is not possible solution if we are interested in making inferences about these attributes. Both approaches are wasteful procedures since they usually decrease the information content of the data [5].

Acquire Missing Values: In practice, a missing value may be obtainable by incurring a cost, such as the cost of performing a diagnostic test or the cost of acquiring consumer data from a third party. To maximize expected utility one must estimate the expected added utility from buying the value, as well as that of the most effective missing-value treatment. Buying a missing value is only appropriate when the expected net utility from acquisition exceeds that

of the alternative. However, this decision requires a clear understanding of the alternatives and their relative performances [10].

Imputation

- **Mean Imputation:** This method replaces the missing values for an attribute with the mean of all already known values of that attribute in the class where instance was missing attribute belongs to. Let us consider that the value x_{ij} of the k-th class, C_k, is missing. Then, the missing value will be replaced by

$$\hat{x}_{ij} = \sum_{i=x_{ij} \in C_k} \frac{x_{ij}}{n_k},$$

  where n_k represents the number of non-missing values in the j-th feature of the k-th class [4]. However, the drawback of this method of that all the missing values are now equal to the mean, which means the data is negatively biased and that the variance is underestimated [11].

- **Median Imputation:** Since the mean is affected by the presence of outliers it seems natural to use the median instead just to assure robustness. In this case the missing data for a given attribute is replaced by the median of all known values of that attribute in the class where the instance with the missing value belongs [4]. This method is also a recommended choice when the distribution of the values of a given feature is skewed. Let us consider that the value x_{ij} of the k-th class, C_k, is missing. It will be replaced by

$$\hat{x}_{ij} = median_{\{i=x_{ij} \in C_k\}}\{x_{ij}\}.$$

- **Most Common Value Imputation:** This method simply uses most common attribute value for missing value imputation [12]. The most common value of all values of the attribute is used. This method is usable only for symbolic attributes and is usually combined with replacing missing values with missing values imputation using mean for numeric attributes.

- **Interpolation:** When dealing with temporal or spatial continuous data, interpolation can be used. Examples of temporal data might be the hourly annotations of weather quality in Karlsruhe or data about the yearly average screen time for a young adult. The simplest interpolation technique is linear interpolation. Here missing data is interpolated using the nearest neighbours of the missing data [13].

4.2 High Dimensionality

The simplest way to remove irrelevant features is to apply domain knowledge. For example, if we are interested in clustering text documents, it is obvious that articles, such as “a,” “an” and “the” are irrelevant variables. However, this

approach is feasible only when a domain scientist can easily identify irrelevant attributes, which is rarely the case [14].

Feature subset selection is another well-known task of data mining and machine learning. Genetic algorithms, Hill Climbing, and Simulated Annealing, etc. are commonly used algorithms for feature subset selection tasks. The dimensionality reduction made by an Feature Subset Selection process can provide several advantages: 1) a faster induction of the final classification model, 2) an improvement of the final classification model's comprehensibility, and 3) an improvement in classification accuracy.

Techniques for feature selection can be divided in two approaches: feature ranking and subset selection. In the first approach, features are ranked by some criteria and then features above a defined threshold are selected. In the second approach, one searches a space of feature subsets for the optimal subset. Moreover, the second approach can be split in three parts: 1) filter approaches — people select the features first, then they use this subset to execute a classification algorithm; 2) embedded approaches — the feature selection occurs as part a classification algorithm; and 3) wrapper approaches — an algorithm for classification is used over the data set to identify the best features [3].

Other modern approaches to dimensionality reduction include Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA is a statistical procedure which uses an orthogonal transformation. PCA converts a group of correlated variables to a group of uncorrelated variables. PCA is used for exploratory data analysis. Also PCA can be used for examination of the relationships among a group of variables. Hence it can be used for dimensionality reduction.

LDA is another popular dimensionality reduction approach for pre-processing step in data mining and machine learning applications [39]. The main aim of LDA is to project a dataset with high number of features onto a less-dimensional space with good class-separability. This will reduce computational costs. The approach followed by LDA is very much analogous to that of PCA. Apart from maximizing the variance of data (PCA), LDA also maximizes separation of multiple classes. The goal of Linear Discriminant Analysis is to project a dimension space onto a lesser subspace without disturbing the class information [15].

4.3 Class Imbalance

Under-sampling The most naive under-sampling method is random under-sampling, a non-heuristic method trying to balance class distributions through the random elimination of majority class examples. This leads to discarding potentially useful data that could be important for classifiers. There have been several heuristic under-sampling methods proposed or introduced from data cleaning in recent years. They are based on either of two different noise model hypotheses. One thinks examples that are near to the classification boundary of the two classes are noise, while the other considers examples with more neighbors of different labels are noise.

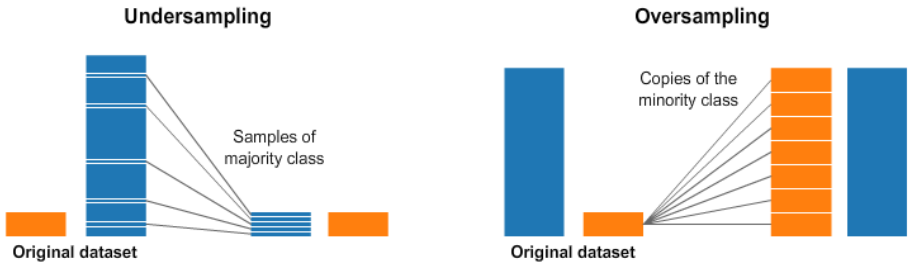


Fig. 3. An Illustration of under-sampling and over-sampling (Rafael Alencar, 2017)

Over-sampling Random over-sampling is a non-heuristic method that aims to balance class distributions through the random replication of minority class examples. Random over-sampling has two shortcomings. First, it will increase the likelihood of occurring over-fitting, since it makes exact copies of the minority class examples. Second, oversampling makes the learning process more time-consuming if the original data set is already fairly large but imbalanced [16].

4.4 Small disjuncts

Cluster-Based Oversampling This strategy tackles the problem of small disjuncts by separately clustering the training data of each class and performing random oversampling on all clusters. Before oversampling, the training data in the classes are clustered using a clustering algorithm. Once the process of clustering is complete, oversampling on the classes is applied so that no between-class and no within-class imbalance remains [8].

5 Other related solutions

5.1 Missing Labels

Sensor-based human activity recognition (HAR) has been playing a significant role in many applications, which enables the provision of customized services to suit people's current context. Various machine learning including recent deep learning techniques have been applied to HAR in feature extraction and sensor fusion and achieved promising accuracy in recognizing daily activities. However, these techniques heavily rely on labeled training data to build a robust computational model. Labeling sensor data with activities is a time-consuming and cost-sensitive task. Reducing the reliance on training data has long been a challenging research topic, and different approaches have been attempted, including unsupervised learning, active learning, co-training, and transfer learning. The majority of these approaches have reduced the annotation to a certain degree, but the annotation burden on individual users is still heavy and does not scale to a large number of users. To directly tackle this challenge we propose the following approaches:

Sharing Classifiers To share classifiers, Ye, J. et al. describe a uncertainty-driven algorithm; that is, when the classifier from the current data set cannot confidently infer an activity label to a given example, then we acquire labels from the classifiers from the other datasets. This is inspired by active learning; that is, identify uncertain examples and query human operators for annotation. The difference here is that we do not query human operators, but classifiers in the other datasets. This process is described in the Algorithm in Figure4 below.

Sharing Labels A classic approach to dealing with a small amount of training data is leveraging unlabelled data. That is, for each data set, we train a classifier on its labeled data and then use it to iteratively infer the labels on its unlabelled examples for T rounds or until the algorithm converges. For each iteration, we select the top k most confident examples to enlarge the labeled data pool and iteratively update the classifier. However, given our assumption that the labeled data might be too little and have not covered the whole set of activities of interest, this basic approach can only assign the labels that have been observed in the training data. We will need to leverage labels from the other datasets. This gives rise to the Algorithm in Figure 5 below. With feature remapping, for each data set, we will not only train a classifier on its train data but also on the transferred training data from the other datasets [17].

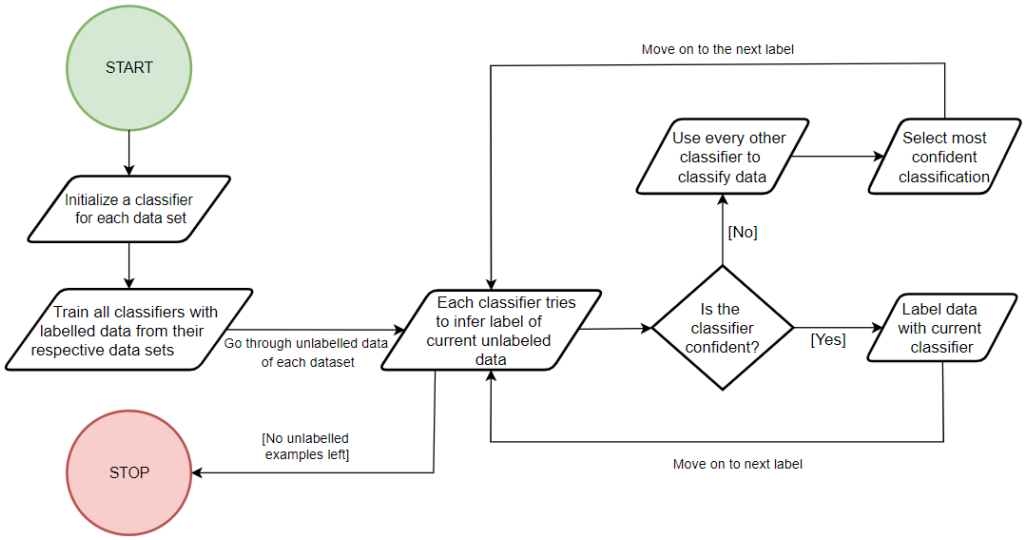


Fig. 4. An algorithm for sharing classifiers

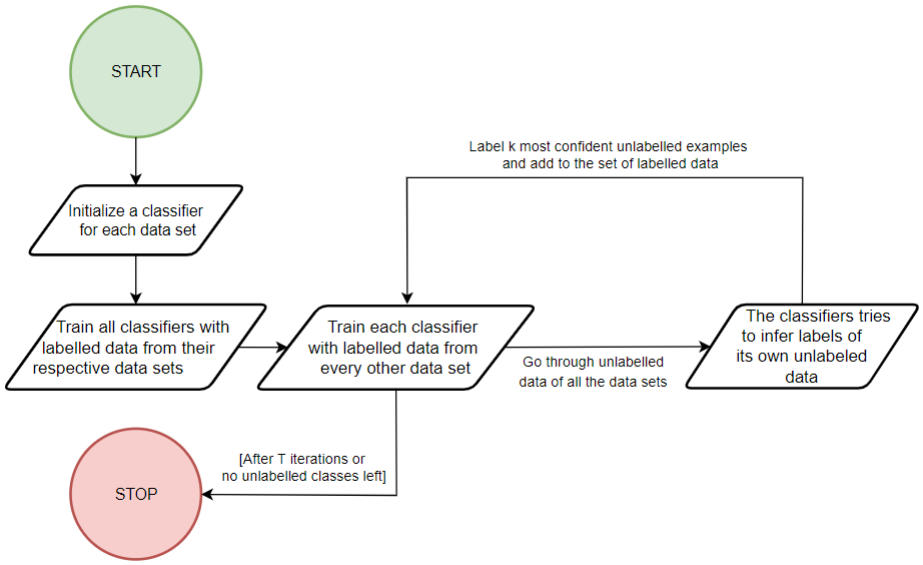


Fig. 5. An algorithm for sharing data between classifiers

5.2 Transfer Learning

Human learners appear to have inherent ways to transfer knowledge between tasks. That is, we recognize and apply relevant knowledge from previous learning experiences when we encounter new tasks. The more related a new task is to our previous experience, the more easily we can master it. Common machine learning algorithms, in contrast, traditionally address isolated tasks. Transfer learning attempts to change this by developing methods to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task (see Figure 6). Techniques that enable knowledge transfer represent progress towards making machine learning as efficient as human learning [18].

Ideally, we would like to be able to use labeled data from a different domain to improve learning in the target domain. One example would be to use labeled data from one or more smart apartments to recognize activities in a new smart apartment which may have a different layout, different residents, or different lifestyles or behavioral patterns. Another example would be using the labeled data from a smart apartment to perform activity recognition in a smart office. Traditional supervised machine learning techniques rely on the assumptions that the training data and test data have similar probability distributions and that the classification task is the same for both datasets. However, in the previous examples, the source and target data are drawn from different probability distributions. In these cases, traditional machine learning techniques often fail to correctly classify the test data. However, with transfer learning, we may train

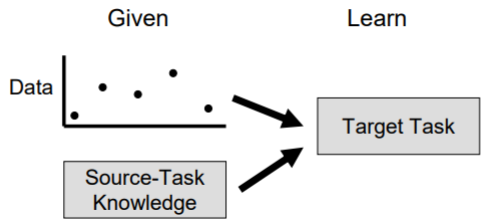


Fig. 6. Machine learning incorporating expertise from one or more related tasks as an additional information source in addition to the usual training data is known as transfer learning.

our machine learning model on one data set and reuse this model on structurally different data sets. Other important benefits of transfer learning In the field of machine learning are for example less time spent learning new tasks, less information required of experts (usually humans), and more situations being able to be handled effectively, making the learned model more robust. These potential benefits have led researchers to apply transfer learning techniques to many domains with varying degrees of success [19].

Feature-Space Remapping Traditional supervised machine learning techniques rely on the assumptions that the training data and test data are drawn from the same probability distributions and that the classification task is the same for both datasets. However, in practice it is often convenient to relax these assumptions and allow the test data to be drawn from a different probability distribution or to allow the classification task to change. In these cases, traditional machine learning techniques often fail to correctly classify the test data. Feuz and Cook propose a transfer learning technique between heterogeneous data sets called "Feature Space-Remapping (FSR)". This algorithm essentially works like the following: First, the FSR computes the meta-features as means to relate source and and target features. These are used to select features in the source space which are most similar to features in the target space. Next, FSR computes a similarity matrix based upon the feature-feature pairs and the meta-features. Then, the FSR computes mappings from target features to source features by selecting source feature with maximal similarity to target feature as given by the similarity matrix. Finally, FSR applies the computed mapping to the target data to be classified using the hypothesis learned on the source data [20].

6 Conclusion

In the current state of machine learning and neural networks, the causes and effects of heterogeneous data have been extensively studied and are still ongoing.

In this paper, we have tried to categorize the heterogeneity of the data and briefly explained the nature of these heterogeneities. We've also explored literature that encounters the problem of heterogeneous data and how they address these kinds of problems in various kinds of ways. However, it is important to note that heterogeneity comes in all shapes and sizes so not every solution presented in this paper is one-size-fits-all. The domain of the problem needs to be carefully analyzed to determine which solution is best applicable and what compromises we can settle on.

References

[1] Sachin Kumar, Durga Toshniwal, and Manoranjan Parida. "A comparative analysis of heterogeneity in road accident data using data mining techniques". In: *Evolving systems* 8.2 (2017), pp. 147–155.

[2] Konstantina Kourou et al. "Machine learning applications in cancer prognosis and prediction". In: *Computational and structural biotechnology journal* 13 (2015), pp. 8–17.

[3] Lidong Wang. "Heterogeneous data and big data analytics". In: *Automatic Control and Information Sciences* 3.1 (2017), pp. 8–15.

[4] Edgar Acuna and Caroline Rodriguez. "The treatment of missing values and its effect on classifier accuracy". In: *Classification, clustering, and data mining applications*. Springer, 2004, pp. 639–647.

[5] Jiři Kaiser. "Dealing with Missing Values in Data." In: *Journal of Systems Integration (1804-2724)* 5.1 (2014).

[6] Karshiev Sanjar et al. "Missing data imputation for geolocation-based price prediction using KNN–mcf method". In: *ISPRS International Journal of Geo-Information* 9.4 (2020), p. 227.

[7] Justin M Johnson and Taghi M Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (2019), pp. 1–54.

[8] Taeho Jo and Nathalie Japkowicz. "Class imbalances versus small disjuncts". In: *ACM Sigkdd Explorations Newsletter* 6.1 (2004), pp. 40–49.

[9] Swagatam Das, Shounak Datta, and Bidyut B Chaudhuri. "Handling data irregularities in classification: Foundations, trends, and future challenges". In: *Pattern Recognition* 81 (2018), pp. 674–693.

[10] Maytal Saar-Tsechansky and Foster Provost. "Handling missing values when applying classification models". In: (2007).

[11] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.

[12] Jerzy W Grzymala-Busse and Ming Hu. "A comparison of several approaches to missing attribute values in data mining". In: *International Conference on Rough Sets and Current Trends in Computing*. Springer. 2000, pp. 378–385.

[13] Norazian Mohamed Noor et al. "Comparison of linear interpolation method and mean method to replace the missing values in environmental data set".

In: *Materials Science Forum*. Vol. 803. Trans Tech Publ. 2015, pp. 278–281.

[14] Hussein A Abbass, Ruhul Sarker, and Charles S Newton. *Data Mining: A Heuristic Approach: A Heuristic Approach*. IGI global, 2001.

[15] G Thippa Reddy et al. "Analysis of dimensionality reduction techniques on big data". In: *IEEE Access* 8 (2020), pp. 54776–54788.

[16] Xinjian Guo et al. "On the class imbalance problem". In: *2008 Fourth international conference on natural computation*. Vol. 4. IEEE. 2008, pp. 192–201.

[17] Juan Ye. "Shared learning activity labels across heterogeneous datasets". In: *Journal of Ambient Intelligence and Smart Environments* 13.2 (2021), pp. 77–94.

[18] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[19] Kyle Dillon Feuz and Diane J Cook. "Heterogeneous transfer learning for activity recognition using heuristic search techniques". In: *International Journal of Pervasive Computing and Communications* (2014).

[20] Kyle D Feuz and Diane J Cook. "Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR)". In: *ACM transactions on intelligent systems and technology (TIST)* 6.1 (2015), pp. 1–27.