Influence of Noise and Data Characteristics on Classification Quality of Dispersed Data Using Neural Networks on the Fusion of Predictions

Małgorzata Przybyła-Kasperek
University of Silesia in Katowice
Katowice, Poland *malgorzata.przybyla-kasperek@us.edu.pl*

Kwabena Frimpong Marfo
University of Silesia in Katowice
Katowice, Poland *kwabena.marfo@us.edu.pl*

Abstract

In this paper, the issues of classification based on dispersed data are considered. For this purpose, an approach is used in which prediction vectors are generated locally using the k-nearest neighbors classifier. However, in central server, the final fusion of prediction vectors is made with the use of a neural network. The main aim of the study is to check the influence of various data characteristics (the number of conditional attributes, the number of objects, the number of decision classes) and the degree of dispersion and noise intensity on the quality of classification of the considered approach. For this purpose, 270 data sets were generated that differed by the above factors. Experiments were carried out using these data sets and statistical tests were performed. It was found that each of the examined factors has a statistically significant impact on the quality of classification. However, the number of conditional attributes, degree of dispersion, and noise intensity have the greatest impact. Multidimensionality in dispersed data affects the results positively, but the analyzed method is only resistant to a certain degree of noise intensity and dispersion.

Keywords: Federated Learning, Dispersed Data, Neural Network, Noise, Degree of Dispersion.

1. Introduction

Many different classification methods have been proposed in machine learning so far. These methods have found applications in numerous information systems used in banking, stock exchange, electronic markets, medicine, among others. Most of the traditional classification methods are dedicated to data stored in one decision table, yet, this approach is increasingly seen as insufficient. In today's global society, federated learning is a critical issue [17]. This approach responds to the widespread occurrence of dispersed data provided by various units that wish to keep their data private. In all of the applications mentioned above, examples of dispersed data can be found [10]. Currently, we are dealing with data collected in a dispersed manner by various units, institutions, websites, and mobile devices [1]. When local data is used together, better quality of classification can be obtained than when we rely on one fragment of data. Even so, using dispersed data is not a simple task. First and foremost, there is a great possibility of the presence of inconsistencies in the data – independently collected data may have different set of attributes as well as different set of objects, but the possibility of having common elements among dispersed data is not excluded. In this case, it is not possible to merge such data into one table. Secondly, the difficulty of using dispersed data stems from the fear of freely sharing data. Often, data-owners want to preserve data privacy, thus, we cannot construct a method that accesses all data from various sources.

Scientists wonder if it is necessary to propose many methods for solving real problems [4].

On the other hand, we have a no-free lunch theorem [1] which justifies proposing new methods dedicated to specific problems. In the context of such considerations, it is important to characterize methods in terms of the problem and data characteristics for which the method is dedicated to. To do this, the method should be tested in terms of both varying data and different contexts. The differences in data can be considered in terms of the following characteristics: the number of objects, the number of conditional attributes, the number of decision classes, informativeness and redundancy of attributes. Other important concepts to consider are presence of imbalance decision classes and the presence of noise in the data.

As an instance, suppose we want to build an automated diagnostic system using records of patients across multiple hospitals. Depending on the above mentioned data characteristics, the performance of the learning algorithm could be greatly affected and one would not have a clear approach to address this issue. However, with the use of an information system, we could drill down to the factors that truly cause the deterioration in the learning algorithm. In the paper [12], a method for classification with the use of dispersed data was proposed. This method uses the k-nearest neighbors and the neural networks classifiers. The k-nearest neighbors classifier is used for local data – a prediction vector is generated, which is then transferred to a central server. Then the neural network makes the final decision based on all prediction vectors. In this way, we maintain data privacy as only the prediction vectors are shared. The paper [12] presents that this approach gives unambiguous results, which cannot be said about other fusion methods such as the Majority Voting, the Borda Count method, the Sum Rule, the method based on decision templates and the method based on theory of evidence. Moreover, it was shown that the approach with neural networks achieves better quality of classification than the above-mentioned fusion methods.

However, there are still many questions to be answered. Does the number of conditional attributes/the number of objects/the number of decision classes in local data affect the quality of classification? Does information noise disturb the classification of dispersed data with the use of neural networks, and if so, to what extent? If the informative attributes are dispersed among local data, are we able to make a good classification based on fragmented data? How does high data dispersion affect the quality of classification? The aim of this paper is to answer these questions. In this way, the scope of applications of the classification method for dispersed data using the k-nearest neighbors and the neural network classifiers will be determined.

The article focuses on the extent to which data characteristics mentioned above, as well as noise in data affects the classification accuracy of method proposed in this paper. This issue is very important in the case of dispersed data with noise, for example, data from social media. For this purpose, 18 data sets were generated with varying number of objects, attributes and decision classes. To each of them, noise was added in three different degrees of intensity. The data was divided into five different versions of dispersion (with different number of local data) in such a way as to guarantee different, but not necessarily disjoint sets of attributes in each of the local sets. Thus, 270 dispersed data was obtained. Experiments using these data and different number of neurons in the hidden layer were performed. The obtained results were compared and conclusions were drawn.

The article is organized in the following way. Section 2 provides a literature overview. The next section describes the classification approach of the dispersed data and the method of data generation. In Section 4, the results of the experiments are presented and analyzed. The article ends with conclusion.

2. Literature Review

The issues of data stored in local sets are considered mainly in two contexts; an ensemble of classifiers [2, 3, 11] – where local data is created based on a single data set in order to improve the quality of classification. In this approach we have control over the form of local data created.

Local data meets certain conditions, for example, independence and variety. In this paper, a completely different approach is considered because dispersed data do not have to fulfill any of these constraints.

Dispersed data is also considered in federated learning topics [5, 14]. Similar to the data considered in this paper, the local data are collected independently and we have no influence on their form. Federated learning involves applying machine learning methods locally to each local device separately, without sharing the training objects with a central server [10]. This approach makes it possible to learn a common model while maintaining data privacy. Federated learning issues are widely used in applications such as smart healthcare, smart transportation, Unmanned Aerial Vehicles, smart cities, and smart industry [9].

Neural networks in the context of federated learning are considered in many papers. In [16], neural networks are built locally, and their weights are sent to a central server in order to build the final model. The paper [6] also analyzes the use of neural networks in the context of federated learning, but the main focus is on passing the weights to the central server more efficiently. In [8] a model with a deep graph neural network is proposed to classify the nodes based on their structures and features. In this paper, however, the combination of the k-nearest neighbors classifier, which will be used locally, with a neural network, which will be built on a central server, is considered. Such an approach was investigated in [12] although the influence of various data characteristics, degree of dispersion and noise intensity on the quality of classification of this method was not analyzed there. Such studies are performed in this work.

3. Methods and Data

In this section, we first briefly introduce the dispersed data classification approach that uses the k-nearest neighbors and the neural network classifiers. Next, the method of generating and preprocessing data sets used in the experimental analysis is described.

3.1. Dispersed Classification Method with k-Nearest Neighbors and the Neural Network Classifiers

The approach used in this paper to classify based on dispersed data consists of two steps. We assume that each local data is stored in the form of a local decision table. In the first stage, the calculations are performed independently in local destinations. A modified k-nearest neighbors algorithm is used and predictions from the measurement level are designated. We assume that a set of decision tables $D_{ag} = (U_{ag}, A_{ag}, d)$, $ag \in Ag$ from one discipline is available, where U_{ag} is the universe, a set of objects; A_{ag} is a set of conditional attributes; d is a decision attribute. Based on each of the local tables, a classifier is built. Ag is a set of classifiers, and ag is a single classifier. For each local table and for each test object x, a probability vector over decision classes (denoted by $\mu_{ag}(x)$) is designated. The dimension of vectors $\mu_{ag}(x) = [\mu_{ag,1}(x), \ldots, \mu_{ag,c}(x)]$ is equal to the number of decision classes $c = card\{V^d\}$, where V^d is a set of values of decision attributes from all decision tables and $card\{V^d\}$ is the cardinality of this set. Each coefficient $\mu_{ag,j}(x)$ is determined using the k-nearest neighbors of the test object x belonging to a given decision class j and decision table D_{ag}. The gower similarity measure is used in this approach. For numerical data used in this paper, the gower measure is equivalent to the Manhattan distance.

Only the prediction vectors are made available for centralized computation. A global decision for a test object is generated with the use of a neural network. The structure of the network consists of three layers. The hidden layer has a varying number of neurons that will be studied experimentally. The number of neurons in the input layer is equal to the product of the number of prediction vectors and the dimension of the vector, i.e. $card\{Ag\} \times card\{V^d\}$. The output layer has the number of neurons equal to the number of decision classes. For the hidden layer,

the ReLU (Rectified Linear Unit) activation function is used. For the output layer, the SoftMax activation function is used, which is recommended when we deal with a multi-class problem [7]. The back-propagation method, the Adam optimizer and the categorical cross-entropy loss function are used in the study.

Of course, a neural network must be trained using a certain set of objects. Since the training objects were used to generate the prediction vectors with the k-nearest neighbors classifier, they cannot be reused to train the neural network. Thus, a 10-fold cross-validation method was used for the test set. Each time, the neural network was trained using 9 folds, while the last independent fold was classified using the neural network constructed based on the 9 folds. This procedure was repeated ten times for each of the folds, and the final quality of classification was assigned using the results obtained from these ten performances. For a more detailed description, please refer to [12].

The big advantage of the approach described above is that it generates unambiguous decisions, which cannot be said about many other fusion methods [1]. Moreover, in comparison with fusion methods such as the Majority Voting, the Borda Count method, the Sum Rule, the method based on decision templates and the method based on theory of evidence; the approach described above gives in most cases a better quality of classification. But it is obvious that the proposed approach is not appropriate in every case or for every data set. Therefore, the question remains – with which data does the dispersed classification method with neural network handles best. To answer this question, data sets were generated that differ in many factors. A total of 270 dispersed data sets were tested, which is described below.

3.2. Data

The data was generated artificially as the aim was to systematically compare the results obtained from data with specific characteristics. The aim was to compare results obtained with the use of dispersed classification method with neural network in relation to the following issues with respect to the impact on the performance of the proposed algorithm:

- the impact that the number of conditional attributes in data has on the quality of classification (multidimensionality of data),
- the impact that the number of objects in data has on the quality of classification,
- the impact that the number of decision classes in data has on the quality of classification,
- the impact that the degree of dispersion has on the quality of classification,
- the impact that the noise intensity in data has on the quality of classification.

The generation of artificial data sets was carried out in several stages:

1. In the first stage, data sets were generated using the Weka [13] software. For this purpose, the RandomRBF was used. This function, at first, randomly generates centers for each decision class. Then to each center, a weight is randomly assigned as well as a central point per attribute, and a standard deviation. A new object is generated as follows; a center is selected according to the weights. Then attribute values are randomly generated and offset from the center. After, the vector is scaled so that its length is equal to a value sampled randomly from the Gaussian distribution of the center. In this way, 18 data sets were generated with different number of conditional attributes, decision classes, objects and centroids. The number of objects in decision classes is imbalanced. The characteristics are presented in Table 1.

There are two main reasons why the training and testing methods used in this paper are the best methods to evaluate the quality of classification for dispersed data. To begin, when

Table 1. Data set characteristics.

Data Set	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
No. of objects	650	650	650	1300	1300	1300	650	650	650	1300	1300	1300	650	650	650	1300	1300	1300
No. of conditional attributes	30	50	70	30	50	70	30	50	70	30	50	70	30	50	70	30	50	70
No. of decision classes	10	10	10	10	10	10	5	5	5	5	5	5	5	5	5	5	5	5
No. of centroids	100	100	100	100	100	100	50	50	50	50	50	50	100	100	100	100	100	100

we deal with dispersed data, it significantly increases computational complexity. Also, there are different sets of conditional attributes in different local tables, and the test object must have specific values on all of these attributes. Thus, at first, each of the data sets was randomly but in a stratified way divided into training set (70% of the data) and testing set (30% of the data).

One of the study goals was to check the influence of noise intensity on the quality of classification. For this purpose, three different levels of noise intensity were applied to the training set (70% of the data), thus, the density of the noise was 100% of the training set. After dividing each data into training and testing set, 3 training data sets with Gaussian noise intensity were further constructed. For each set, mean value equal to 0 and different values of standard deviation (std) were used: $\{mean = 0, std = 0.01\}, \{mean = 0, std = 0.1\}$ and $\{mean = 0, std = 0.2\}$ respectively. In this way, based on 18 training sets, 54 data sets with different noise intensities were created.

Another research goal was to check the influence of the degree of dispersion on the quality of classification. Each of the 54 data sets prepared in the previous step was divided into five versions of dispersion – $3, 5, 7, 9, 11$ local tables were constructed from each training data set with different number of conditional attributes. Local decision tables were constructed in a way such that each local table has a unique set of conditional attributes and also some conditional attributes that are present in other local tables. The number of attributes in local tables varied from 3 to 35. For a finer dispersion, a greater number of local tables contain a smaller number of conditional attributes.

The above described approach is used to generate 270 dispersed data. Thus, for each of the original training set (70% of the data), we have dispersed data with 3, 5, 7, 9, 11 local tables for $mean = 0, std \in \{0.01, 0.1, 0.2\}$ Gaussian noise intensities.

The code of the function that defines the Gaussian noise intensities for each training data set is given in Listing 1.

Listing 1. Gaussian Noise Intensity for Training Data Sets

```
# df: Original training data set read into a Dataframe
# col: A list of conditional attributes
# mu: mean value for Gaussian noise
# sigma: standard deviation value for Gaussian noise

def get_noise(df, col:list, mu:float,sigma:float):
noise = np.random.normal(mu, sigma, size(df))
return pandas.DataFrame(noise, columns=col)
```

The quality of classification was evaluated based on the test set using the estimator of classification error e. It is defined as a fraction of the total number of objects in the test set that were classified incorrectly. With the use of the analyzed approach, the decisions generated are always unambiguous – thus, one decision class is always generated by the system.

Table 2. Results of classification error e for the dispersed system with neural network

	Noise $std = 0.01$					Noise $std = 0.1$					Noise $std = 0.2$				
Data	No. of local tables														
Set	3	5	7	9	11	3	5	7	9	11	3	5	7	9	11
1	0.039	0.049	0.059	0.09	0.138	0.049	0.065	0.085	0.133	0.247	0.179	0.241	0.324	0.483	0.636
2	0.029	0.034	0.026	0.032	0.031	0.026	0.029	0.013	0.036	0.031	0.041	0.056	0.076	0.092	0.142
3	0.003	0	0	0.001	0	0.005	0	0	0.001	0.003	0.006	0.013	0.022	0.016	0.016
4	0.017	0.035	0.04	0.058	0.094	0.028	0.065	0.058	0.088	0.203	0.069	0.138	0.383	0.584	0.699
5	0.001	0.002	0.003	0.005	0.006	0.001	0.003	0.005	0.006	0.012	0.002	0.018	0.036	0.084	0.152
6	0	0	0	0	0	0	0	0	0	0	0	0.002	0.006	0.005	0.035
7	0.017	0.022	0.029	0.029	0.041	0.01	0.025	0.03	0.035	0.071	0.018	0.041	0.063	0.102	0.193
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0.003	0.001	0	0	0.004	0.002	0.008	0.007	0.005	0.01	0.007	0.007
10	0.01	0.017	0.019	0.028	0.031	0.01	0.021	0.017	0.029	0.042	0.017	0.024	0.035	0.072	0.123
11	0.002	0.002	0.001	0.002	0.005	0.002	0.002	0.002	0.002	0.005	0.002	0.004	0.003	0.003	0.005
12	0.001	0.003	0.002	0.003	0.003	0.003	0	0.002	0.003	0.003	0.002	0.003	0.003	0.003	0.003
13	0.048	0.058	0.068	0.077	0.115	0.056	0.058	0.061	0.09	0.145	0.073	0.131	0.211	0.366	0.591
14	0.005	0.006	0.011	0.008	0.013	0.01	0.008	0.015	0.01	0.016	0.008	0.025	0.027	0.041	0.051
15	0	0	0.005	0	0.003	0	0	0	0	0	0.003	0.008	0.011	0.005	0.023
16	0.007	0.017	0.026	0.047	0.064	0.016	0.023	0.031	0.055	0.078	0.024	0.047	0.11	0.347	0.636
17	0	0	0	0.001	0.001	0	0	0	0.001	0.003	0	0.001	0.01	0.021	0.028
18	0	0	0	0	0.005	0.001	0.001	0	0.001	0.003	0.002	0	0.002	0.003	0.01

4. Results

For each dispersed set (one of the 270 analyzed) the experiments were carried out according to the following scheme:

- Generating vectors of predictions based on local tables using the k-nearest neighbors classifier. For each data set, three different values of the k parameter were tested, namely $k \in \{1, 5, 10\}$. One parameter value was selected for each dispersed data set that produced the best overall results. For the majority of data – $k = 1$ were selected. Only for data set 4 and the dispersion with 9 and 11 local tables, $k = 5$ was selected.

- Generating a global decision using a neural network with one hidden layer and different number of neurons in the hidden layer. For each data set, the following number of neurons in the hidden layer were tested: $\{1, 3, 4, 4.25, 4.5, 4.75, 5\}\times$the number of neurons in the input layer. Different number of neurons in the hidden layer was also checked. However, it was noticed that the accuracy of the respective models improves as the number of neurons in the hidden layer increases, but significant improvement declines around $5\times$ the number of neurons in the input layer. The number of neurons in the input layer depends on the number of local tables. Thus, the more dispersed data we have, the more complex the structure of the neural network is.

It should be noted once again that to use the neural network, a 10-fold cross-validation was used on the test set, i.e., the neural network was trained 10 times with 9 folds and tested on one remaining fold. In addition, each test was performed three times to ensure that the results were reliable and not distorted by the influence of randomness. The results for the neural network approach that are given below are the average of the obtained results.

The results obtained for the optimal number of neurons in the hidden layer are presented in Table 2. We do not present the individual results obtained for a different number of neurons in the hidden layer ($\{1, 3, 4, 4.25, 4.5, 4.75, 5\}\times$ the number of neurons in the input layer) due to the limited space. However, for many data sets, the optimal number was around $4\times$ the number of neurons in the input layer. In Table 2, 18 data sets are listed in the rows, the columns distinguish between three different noise levels and five different versions of dispersion.

As can be seen, some data sets were trivial for the analyzed approach. These are data sets 6, 8, 9, 15, 17 and 18 for which the classification error was almost always equal to 0 regardless of the version of dispersion and noise intensity. As can be seen from data characteristics – Table 1, the main factor affecting data simplicity is the number of conditional attributes occurring in the data. The data sets 6, 9, 15 and 18 have 70 attributes (which were split into local tables). However, the approach of using dispersed data perfectly copes with the information stored in

local tables and makes correct decisions. For the two remaining data sets 8 and 17, the number of conditional attributes was equal to 50 - so it is also a large number, while here the factor influencing the simplicity of the data was a small number of centroids (for set 8) and a large number of objects (for set 17).

The general hypotheses that can be made based on the results in Table 2 in relation to the effect each data set had on the performance of the algorithm proposed in this paper are as follows:

- The more conditional attributes occurred in the data, the better the quality of classification. Multidimensionality is beneficial for dispersed data.
- The greater the number of training objects, the better the quality of classification.
- The greater the number of decision classes, the worse the quality of classification.
- For greater dispersion (number of local tables 3, 5, 7, 9, 11) the quality of classification deteriorates.
- The analyzed method is not immune to noise; high noise and significant dispersion (a large number of local tables) gives poorer quality of classification.

The tests of statistical significance for all of the above hypotheses are presented below.

4.1. Comparison of Experimental Results for Different Numbers of Conditional Attributes

In order to investigate how the number of conditional attributes affects the quality of classification for dispersed data and the approach using neural network, all results from Table 2 were used – each number of conditional attributes as a separate group. Thus, we have three independent samples for data with 30 conditional attributes, 50 conditional attributes, 70 conditional attributes. Each sample containing 90 observations – results obtained for different versions of dispersion and noise intensity with a constant number of conditional attributes. The Kruskal-Wallis test confirmed that differences among the classification error in these three groups are significant, with a level of $p = 0.000001$, $\chi^2(2) = 151.257$. Then, to determine the pairs of groups between which statistically significant differences occur, the Mann-Whitney test were performed. The test showed that there is a significant difference with $p < 0.0005$ between each pair.

Additionally, comparative box-plot chart for the values of the classification error was created (Figure 1) – the classification error values obtained for the data with different numbers of conditional attributes are presented. As can be observed, distributions of the classification error values in groups are very different. For dispersed data, multidimensionality is very good. The total number of conditional attributes occurring in local tables above 30 already gives a very good quality of classification when using neural network as a fusion method. Anyway, it also reflects the real situation when various units participate in making joint decision. The variety of attributes occurring in local tables has a positive effect on the quality of decisions made.

4.2. Comparison of Experimental Results for Different Numbers of Training Objects

In order to investigate how the numbers of training objects affects the quality of classification for dispersed data and the approach using neural network, all results from Table 2 were used – each number of training objects in data as a separate group. Thus, we have two independent samples for data with 455 training objects and 910 training objects. Each sample containing 135 observations – results obtained for different versions of dispersion and noise intensity with a constant number of training objects. The Mann-Whitney test for independent groups were performed. The test showed that there is significant difference with $p < 0.03$ between groups.

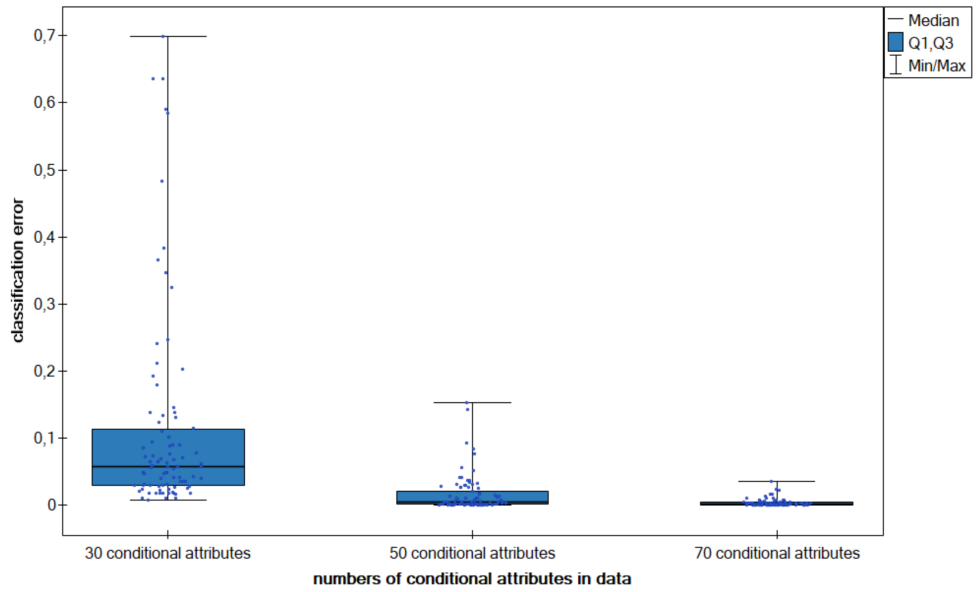

Fig. 1. Box-plot chart with (Median, the first quartile – Q1, the third quartile – Q3) the value of classification error e for the neural network with different numbers of conditional attributes in data.

Additionally, comparative box-plot chart for the values of the classification error was created (Figure 2) – the classification error values obtained for the data with different numbers of training object are presented. As can be observed, distributions of the classification error values in groups are not so different as at the previous graph. This means that the number of objects in dispersed data does not affect the quality of classification as much as the number of conditional attributes. Of course, the difference in results is statistically significant, so the more training objects, the better, but the total number of conditional attributes in dispersed data is much more important.

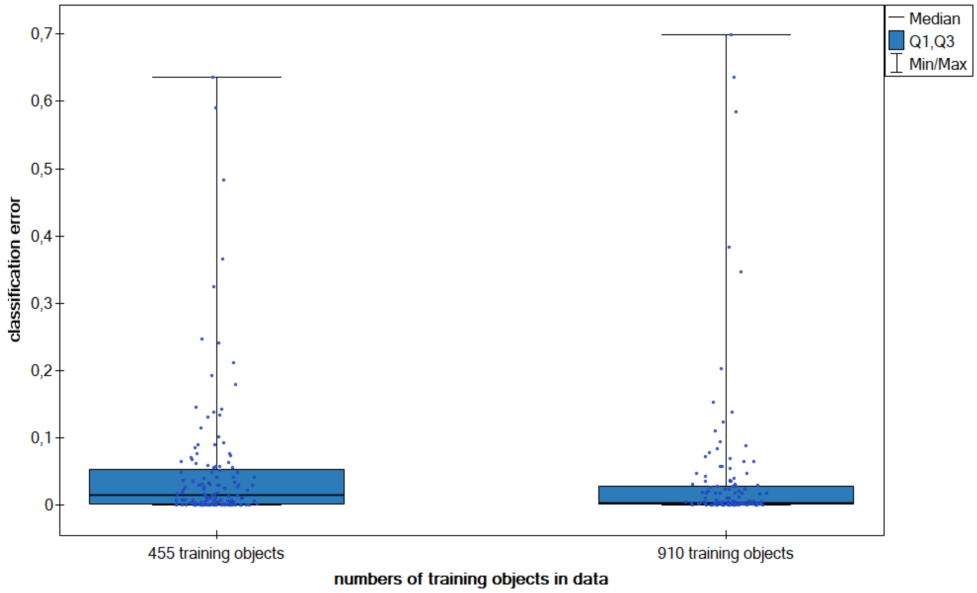

Fig. 2. Box-plot chart with (Median, the first quartile – Q1, the third quartile – Q3) the value of classification error e for the neural network with different numbers of training objects in data.

4.3. Comparison of Experimental Results for Different Numbers of Decision Classes

In order to investigate how the numbers of decision classes affects the quality of classification for dispersed data and the approach using neural network, all results from Table 2 were used – each number of decision classes in data as a separate group. Thus, we have two independent samples for data with 5 decision classes and 10 decision classes. The first sample contains 180 observations and the second sample contains 90 observations – results obtained for different versions of dispersion and noise intensity with a constant number of decision classes. The Mann-Whitney test for independent groups confirmed that there is significant difference with $p < 0.0005$ between groups.

Additionally, comparative box-plot chart for the values of the classification error was created (Figure 3) – the classification error values obtained for the data with different numbers of decision classes are presented. As can be observed, the difference between distributions of the classification error values in groups is less noticeable than for the number of conditional attributes, but more visible than for the number of training objects. It is more or less obvious that the more decision classes we have, the more difficult it is to make a correct decision. However, it can be concluded that the number of decision classes in the dispersed data set is less significant in terms of data difficulty than the number of conditional attributes.

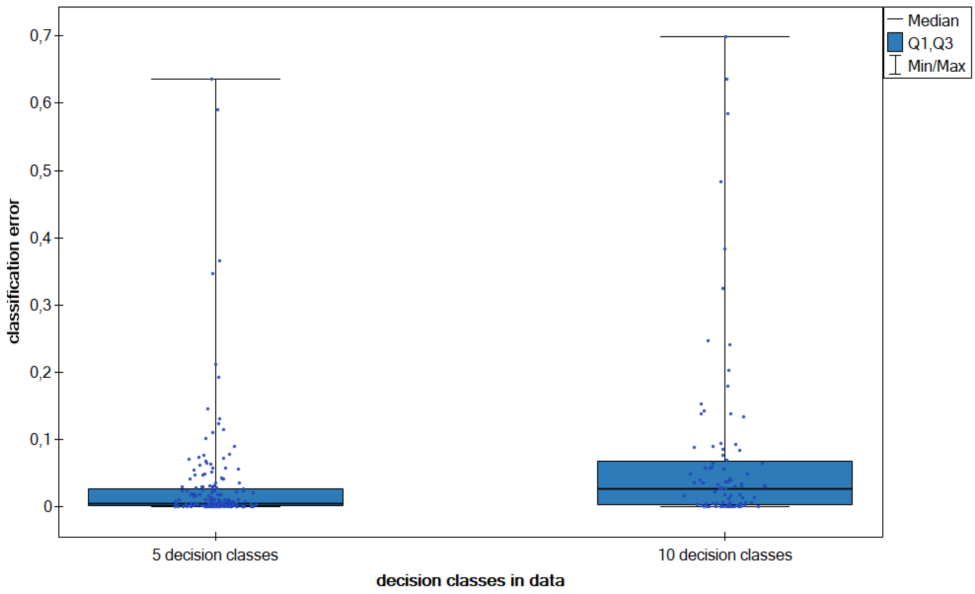

Fig. 3. Box-plot chart with (Median, the first quartile – Q1, the third quartile – Q3) the value of classification error e for the neural network with different number of decision classes in data.

4.4. Comparison of Experimental Results for Different Degree of Dispersion

As the degree of dispersion, we understand the number of local tables occurring in the dispersed data. In order to investigate how the numbers of local tables affects the quality of classification for dispersed data and the approach using neural network, all results from Table 2 were used – each number of local tables in data as a separate group. Thus, we have five dependent samples (as one data set was divided into a different number of local tables) for data with 3, 5, 7, 9 and 11 local tables. Each sample contain 55 observations – results obtained for different data sets and noise intensity with a constant number of local tables. The Friedman's test confirmed that differences among the classification error in these five groups are significant, with a level of $p = 0.000001$. Then, to determine the pairs of groups between which statistically significant

differences occur, the Wilcoxon pair test for dependent groups were performed. The test showed that there is significant difference with $p < 0.00004$ between each pair.

Additionally, comparative box-plot chart for the values of the classification error was created (Figure 4) – the classification error values obtained for the dispersed data with different numbers of local tables are presented. As can be observed, the difference between distributions of the classification error values in groups is most noticeable between the extreme degrees of dispersion (3 local tables and 11 local tables). The results can be summarized that the greater the dispersion, the worse the quality of classification.

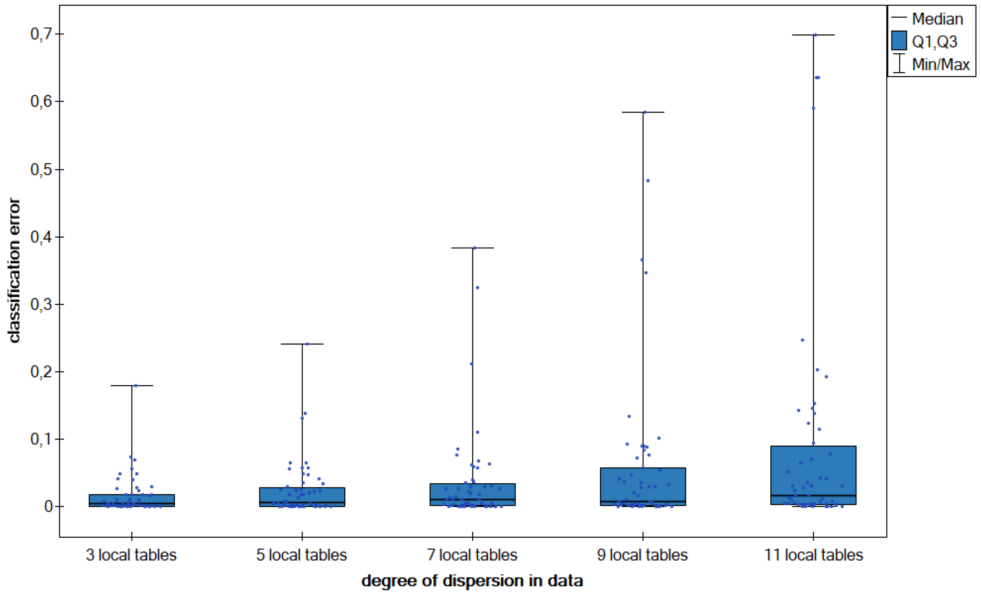

Fig. 4. Box-plot chart with (Median, the first quartile – Q1, the third quartile – Q3) the value of classification error e for the neural network with different degree of dispersion in data.

4.5. Comparison of Experimental Results for Different Noise Intensity in Data

In order to investigate how the intensity of noise affects the quality of classification for dispersed data and the approach using neural network all results from Table 2 were used – each noise intensity (Gaussian noise level with $std \in \{0.01, 0.1, 0.2\}$) in data as a separate group. Thus, we have three dependent samples (as three different noise levels were generated based on one data set) for data with noise $std = 0.01$, $std = 0.1$ and $std = 0.02$. Each sample contain 90 observations – results obtained for different data sets and number of local tables with a constant noise intensity. The Friedman's test confirmed that differences among the classification error in these three groups are significant, with a level of $p = 0.000001$. Then, to determine the pairs of groups between which statistically significant differences occur, the Wilcoxon pair test for dependent groups were performed. The test showed that there is significant difference with $p < 0.00002$ between each pair.

Additionally, comparative box-plot chart for the values of the classification error was created (Figure 5) – the classification error values obtained for the data with different noise intensity $std \in \{0.01, 0.1, 0.2\}$ are presented. For noise intensities equal to $std = 0.01$ and $std = 0.1$, the difference in the distributions is not so noticeable. Only when the noise intensity significantly increased to the level of $std = 0.2$, we can observe a significant increase in the classification error in comparison with the lower noise level. This means that although the classification method for dispersed data with neural network is immune to noise to some extent, it does not cope well with information noise at the $std = 0.2$ level.

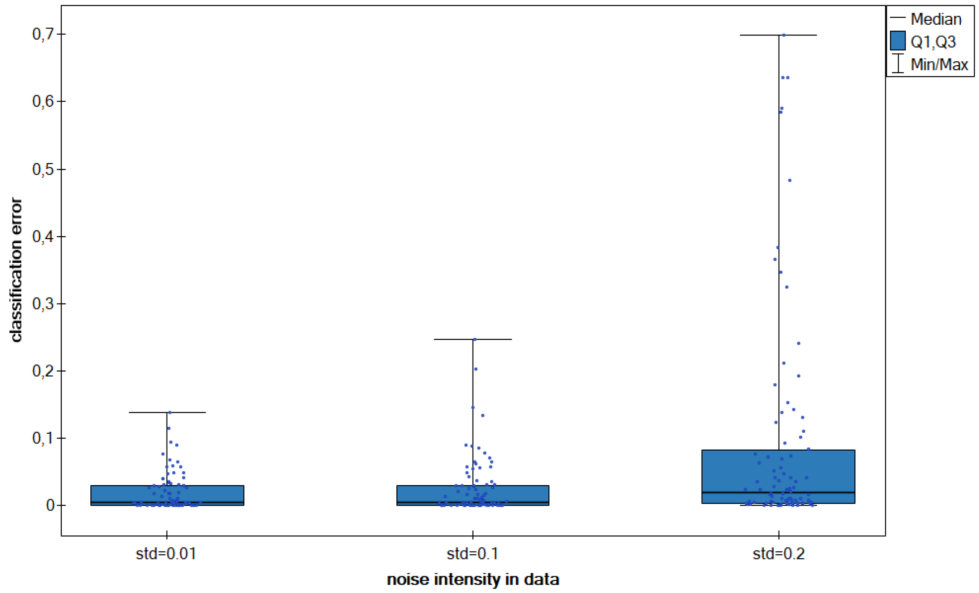

Fig. 5. Box-plot chart with (Median, the first quartile – Q1, the third quartile – Q3) the value of classification error e for the neural network with different noise intensity in data.

5. Conclusion

In the paper, the approach to classification based on dispersed data was analyzed, in which the k-nearest neighbors algorithm was used as a local classifier and the neural network as a fusion method. The analyzed approach takes into account the assumptions of data protection and privacy like in federated learning domain and does not interfere with the form of local data. The main aim of the study was to examine the impact of different data characteristics, the degree of dispersion and noise intensity on the classification quality of the above-mentioned approach. For this purpose, 270 different data sets were generated and experiments were performed. Analysis of obtained results and statistical tests were made.

The main conclusions are as follows. Regarding the data sets characteristics, in the case of dispersed data, the number of conditional attributes has the greatest impact on the quality of classification. Multidimensional dispersed data guarantees better quality of classification. The second most important factor is the number of decision classes. The fewer decision classes, the easier the set for classification. The number of training objects has the least influence on the quality of classification. The more objects in the data set, the better the quality of classification we get. The degree of dispersion has a significant impact on the quality of classification. With significant dispersion (11 local tables), the quality of classification significantly decreases in comparison with the results obtained for data with low dispersion (3 local tables). In the case of small dispersion (3, 5, 7 local tables) the impact on the results is not that significant. Noise occurrence also negatively affects the quality of classification. To some extent it can be said that the method is immune to noise. For Gaussian noise level with an average value 0 and standard deviation not exceeding 0.1, the method generates results in which the difference is not that drastic. Nevertheless, for the Gaussian noise level with the standard deviation equal to 0.2, large decrease in the quality of classification has been noted. The method does not cope well with a high noise level.

In future study, it is planned to propose a classification method for dispersed data, which enables conflict analysis and coalitions creation in order to eliminate the negative impact of high degree of dispersion on the quality of classification. In addition, it is planned to apply the proposed approach to dispersed business data, more specifically stock exchange data.

References

1. Adam, S. P., Alexandropoulos, S. A. N., Pardalos, P. M., Vrahatis, M. N.: No free lunch theorem: A review. Approximation and optimization, 57–82, (2019)
2. Blachnik, M.: Ensembles of instance selection methods: A comparative study. International Journal of Applied Mathematics and Computer Science, 29(1), (2019)
3. Bolon-Canedo, V., Alonso-Betanzos, A.: Ensembles for feature selection: A review and future trends. Information Fusion, 52, 1–12, (2019)
4. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? The journal of machine learning research, 15(1), 3133–3181, (2014)
5. Kołodziej, T., Rościszewski, P.: Towards Scalable Simulation of Federated Learning. In International Conference on Neural Information Processing, 248–256. Springer, Cham, (2021)
6. Konecny, J. H., McMahan, B., Yu, X., Richtarik, P., Suresh, A.T., Bacon, D.: Federated Learning: Strategies for Improving Communication Efficiency, NIPS Workshop on Private Multi-Party Machine Learning (2016)
7. Li, X.; Li, X.; Pan, D.; Zhu, D. On the learning property of logistic and softmax losses for deep neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 34, 4739–4746, (2020)
8. Mei, G., Guo, Z., Liu, S., Pan, L.: SGNN: A Graph Neural Network Based Federated Learning Approach by Hiding Structure, 2019 IEEE International Conference on Big Data (Big Data), 2560–2568, (2019)
9. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Poor, H. V.: Federated learning for internet of things: A comprehensive survey. IEEE Communications Surveys and Tutorials, (2021)
10. Pfitzner, B., Steckhan, N., Arnrich, B.: Federated learning in a medical context: A systematic literature review. ACM Transactions on Internet Technology (TOIT), 21(2), 1–31, (2021)
11. Pławiak, P.: Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals. Swarm and evolutionary computation, 39, 192–208, (2018)
12. Przybyła-Kasperek M, Marfo KF. Neural Network Used for the Fusion of Predictions Obtained by the K-Nearest Neighbors Algorithm Based on Independent Data Sources. Entropy 23(12):1568, https://doi.org/10.3390/e23121568 (2021)
13. Russell, I.; Markov, Z. An introduction to the Weka data mining system. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March, 742–742, (2017)
14. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 13(3), 1–207, (2019)
15. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1–19, (2019)
16. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In International Conference on Machine Learning, 7252–7261, PMLR, (2019)
17. Zimmermann, A., Schmidt, R., Sandkuhl, K.: Multiple perspectives of digital enterprise architecture. In ENASE, 547–554, (2019)