Preprocessing of data from air-gas monitoring sensors during underground mining of gas-bearing coal seams for neural network analysis

Mark Dvoryanchikov[1*], *Larisa* Pavlova[1], and *Inna* Weiss[2]

[1]Siberian State Industrial University, Novokuznetsk, 654007, Russia
[2]EEP Elektro-Elektronik Pranjic GmbH, Gelsenkirchen, 45886, Germany

Abstract. The paper describes the stages of primary study and preprocessing of data obtained from air-gas monitoring sensors for their further analysis using machine learning methods.

1 Introduction

Methane release during the development of gas-bearing coal seams significantly limits possible loads on the mining and tunneling equipment. It also leads to an increase in the cost of coal mining due to the need for additional measures to control gas release [1].

When carrying out underground work, safety comes first. Gas drinaige of coal seams is used to combat gas release. It is important to timely recognize and predict the gas-dynamic phenomenon and take into account many factors in order to make a decision on continuation or stoppage of the work. When the number of properties of the investigated phenomenon increases significantly, it becomes very difficult to establish any connections between them. A special mathematical apparatus which is neural networks was developed to solve such problems. After training, the neural network must be able to reproduce the output data by new input data [2,3].

One of the ways to ensure the safety of mining operations through continuous automatic monitoring of methane level parameters are methane level measurement sensors of the mine air-gas monitoring system. The sensors allow the volume fraction of methane in the atmosphere to be recorded and a decision on further work to be made based on this information. Continuous measurement of these and other variable parameters of production makes it possible to form a database, on the basis of which mathematical methods can be applied for analysis of mutual influence of factors, and to build predictive models.

Thus, when predicting the state of methane in a mine taking into account a large number of measured indicators, it is advisable to use the apparatus of neural networks. An important condition for ensuring effective control of gas release is the correct prediction of methane release at the extraction area and the permissible load on the working face by the gas factor [4, 5].

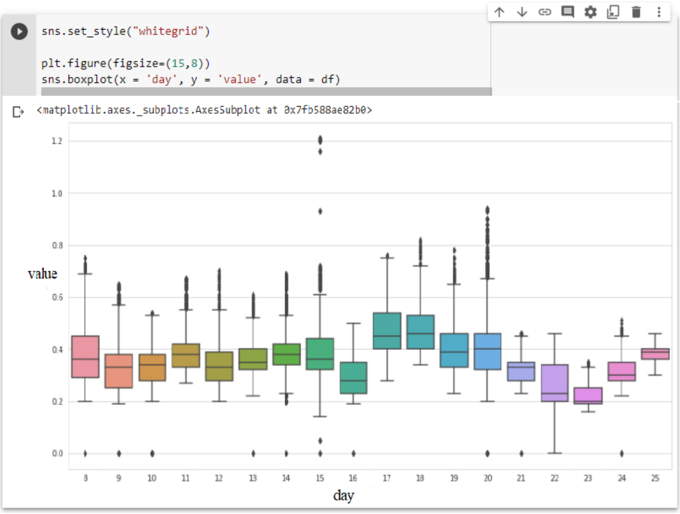
[*] Corresponding author: kicksaflips@gmail.com

2 Initial data

Data preprocessing is an important step which includes removing fluctuations, normalizing, transforming data, extracting useful characteristics, removing rows with empty values, or, if possible, restoring their values. The possibility of using various methods and their effectiveness directly depends on the quality of preprocessing [6].

The data recorded by the sensors of the air-gas dynamic control system have the form of tables with 3 columns: the status of the sensor (working or not), the indicator under investigation (methane concentration in % or rate of methane-air mixture), the third position – date and time accurate to seconds. The sensors are located in the incoming ventilation jet (in_lava_C), the junction jet (kutok_C) and on the outgoing jet (out_lava_C). There is also a sensor on the outgoing jet that measures the air speed (out_lava_speed).

Figure 1 shows the distribution of the methane sensor readings on the outgoing jet by days (in the interval of 25 days) in the form of a box plot.



```
sns.set_style("whitegrid")

plt.figure(figsize=(15,8))
sns.boxplot(x = 'day', y = 'value', data = df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb588ae82b0>
```

Fig. 1. Box plot.

The rectangles in the figure show the interquartile distribution range – 25% (Q1) and 75% (Q3) percentiles. The line inside the rectangle shows the median of the distribution.

The segments represent the entire scatter of points except for outliers, that is, the minimum and maximum values that fall within the interval (Q1 - 1.5 * IQR, Q3 + 1.5 * IQR), where IQR = Q3 - Q1 is the interquartile range. Dots on the graph indicate outliers – values that do not fit into the range of values specified by the segments.

For each of the days, graphs are built showing how the methane readings change during the day (Fig. 2). The background also shows readings for other days.

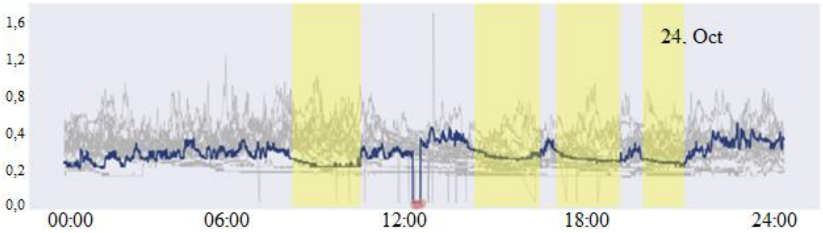


Fig. 2. Sensor readings on outgoing jet, October 24.

The figure shows the periods when the readings practically do not change (in the figures they are indicated by rectangles). Probably, the production does not work during these periods. At some moments, the readings go to zero – the sensor is faulty at that time.

3 Bringing data to a single scale

One of the datasets (out_lava_C) covers a shorter interval of time – 410 hours than the rest – 456 hours. This requires shortening the length of all sets to its length for shared use. In addition, datasets contain a different number of values (Fig. 3). The sensors are triggered at different periodicity. To combine the data, you need to align the values across the rows.

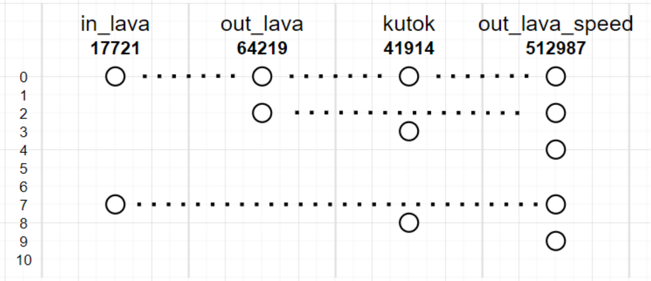


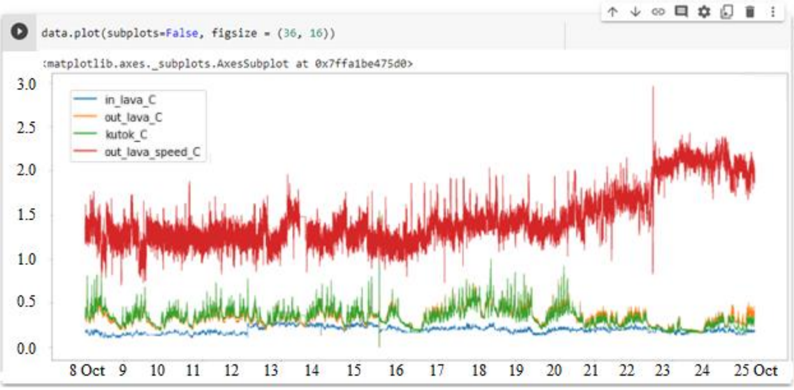
Fig. 3. Illustration of the information receipt from the sensors.

Taking into account the fact that the parameters change insignificantly, if the sensor does not work (readings are not recorded at all), the following algorithm can be used to combine data arrays:

1. Write in one set the information of two sets.
2. Sort values by date.
3. Fill in the missing values with the previous sensor readings

In addition, the values should be excluded from the data in case of sensor disconnect (sensor status is not 0):

The resulting dataset is presented on a single scale in Fig. 4.

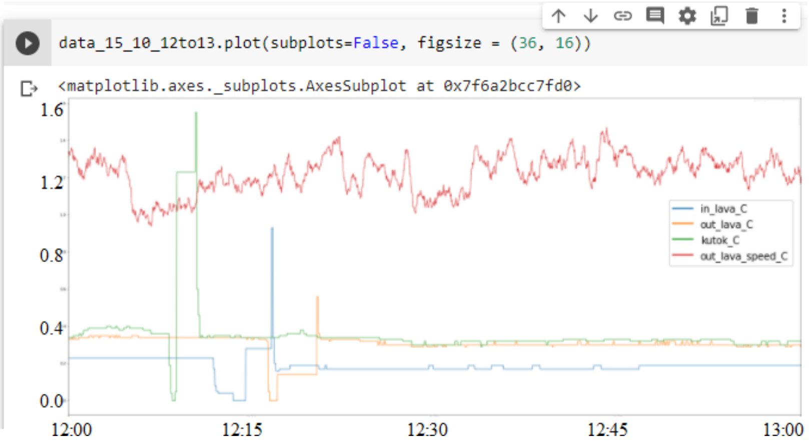


```
data.plot(subplots=False, figsize = (36, 16))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ffa1be475d0>
```

Fig. 4. Combined data on one scale.

Let us consider in more detail the release by the parameter out_lava_C -15 October (Fig. 5). The release corresponds to the methane concentration on the conjanction jet during the period indicated in the figure below. The values greater than expected are located before and after processed periods when the status was not equal to 0 (sensor was faulty). Also, before the event occurred, at some point in time, the sensor readings on the conjugate jet were equal to zero. Later, a similar situation is recorded by sensors located in the incoming ventilation jet and on the outgoing one, but the emission values were processed (previously excluded). Probably, the sensors were sequentially disconnected and then reconnected. And they give overestimated values at some setting moment.



```
data_15_10_12to13.plot(subplots=False, figsize = (36, 16))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6a2bcc7fd0>
```

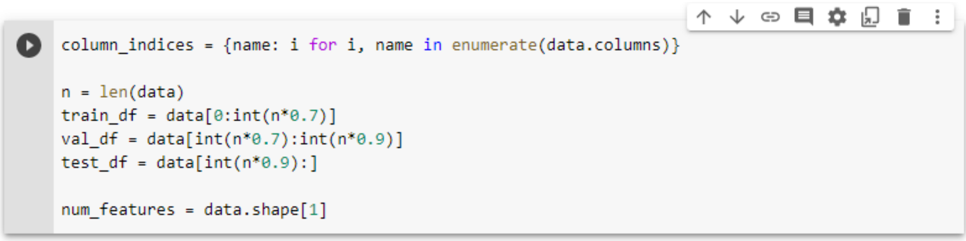
Fig. 5. Fluctuations on October 15. The time period from 12:00 to 13:00.

During these periods of time, the sensors were reconnected, and due to the anomalies that arose after this (jumps in the values during the tuning period), these values must be excluded from further analysis.

The question remains with the uneven intervals between the readings of the sensors. If a sequence is taken with different shifts between elements, then it cannot be argued that data vectors with the same parameters are fed to the input of the model. In this regard, the data must be transferred to a uniform scale.

4 Data separation for training

Data separation into 70% - 20% - 10% for training, validation and testing (Fig. 6).



```
column_indices = {name: i for i, name in enumerate(data.columns)}

n = len(data)
train_df = data[0:int(n*0.7)]
val_df = data[int(n*0.7):int(n*0.9)]
test_df = data[int(n*0.9):]

num_features = data.shape[1]
```

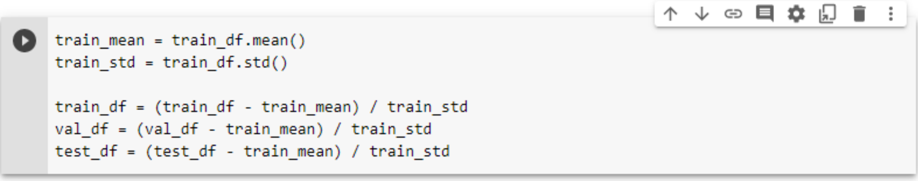
Fig. 6. Separation of the set into training, validation and test samples.

5 Standardization

Scaling (scaling) the data is an important step before training a neural network. different values, that vary in different ranges, can be obtained by the inputs and outputs of neural network after information coding. It is desirable to bring all input variables to a single range and normalize (the maximum absolute value of the input variables should not exceed one). Otherwise, errors caused by variables varying over a wide range will have a stronger effect on network learning than errors of variables varying over a narrow range. One common way of performing scaling is standardization, which is done by subtracting the mean u and dividing by the standard deviation s for each feature x.

$$x\_scaled = (x - u) / s \qquad (1)$$

The mean and standard deviation should be calculated only using the training data so that the models do not have access to the values in the validation and test sets (Fig.).

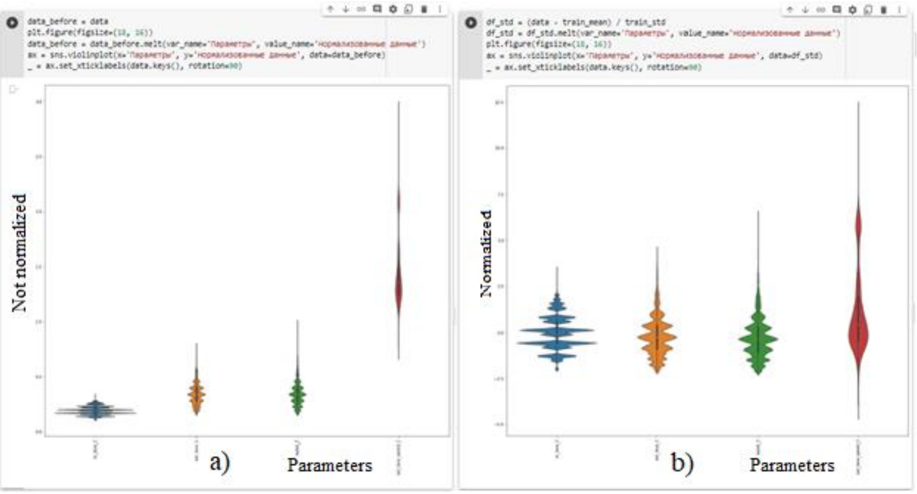


```
train_mean = train_df.mean()
train_std = train_df.std()

train_df = (train_df - train_mean) / train_std
val_df = (val_df - train_mean) / train_std
test_df = (test_df - train_mean) / train_std
```

Fig. 7. Data standardization.

Now let us look at the distribution of data before and after the application of standardization. Graphs of the violin plot type were built for this purpose, showing the probability density of data distribution. The width of the violin shows how often values with a certain value occur (Fig.).



```
data_before = data
plt.figure(figsize=(18, 16))
data_before = data_before.melt(var_name='Параметры', value_name='Нормализованные данные')
ax = sns.violinplot(x='Параметры', y='Нормализованные данные', data=data_before)
_ = ax.set_xticklabels(data.keys(), rotation=90)
```

```
df_std = (data - train_mean) / train_std
df_std = df_std.melt(var_name='Параметры', value_name='Нормализованные данные')
plt.figure(figsize=(18, 16))
ax = sns.violinplot(x='Параметры', y='Нормализованные данные', data=df_std)
_ = ax.set_xticklabels(data.keys(), rotation=90)
```

Fig. 8. Violin plot. Data before a) and after b) normalization.

As a result of reduction to a single dimensionless form, all features before the start of training become equal in their possible influence on the object, which makes it possible to improve the quality of the machine learning algorithms applied to them.

6 Conclusion

Thus, the initial datasets were prepared for their use in machine learning methods, such as neural networks: filtering and scaling were performed, missing values were processed.

References

1. M.V. Dvoryanchikov, L.D. Pavlova, *Modeling and high-tech information technologies in technical and socio-economic systems*, in Proceedings of the V International Scientific and Practical Conference, 14-16 April, Novokuznetsk, Russia (2021)
2. M.V. Dvoryanchikov, L.D. Pavlova, Science-intensive technologies for the development and use of mineral resources, **6**, 241-244 (2020)
3. N. Buduma, N. Lokaskio, Fundamentals of Deep Learning (2017)
4. D.M. Shprekher, G.I. Babokin, E.B. Kolesnikov, Bulletin of TSU, 5, 46-57 (2020)
5. M. Tutak, J. Brodny, IJERPH, **16(8)**, 1406 (2019).
6. S. García, Data Preprocessing in Data Mining (Springer Int. Pub., 2015)