Paper

Data Fixing Algorithm in Radiosonde Monitoring Process

Piotr Szuster

Faculty of Physics, Mathematics and Computer Science, Cracow University of Technology, Cracow, Poland

Abstract—Earth surface monitoring can give information that may be used in complex analysis of the air conditions, temperature, humidity etc. Data from a vertical profile of the atmosphere is also essential for accurate thunderstorm forecasting. That data is collected by radiosondes – telemetry instruments carried into the atmosphere usually by balloons. Sometimes, due to the hostile conditions of upper troposphere, incorrect data can be generated by radiosonde sensors. In this paper, a new algorithm is developed for fixing the incorrect data, i.e. missing or out of specific range values. The proposed algorithm was tested both on benchmarks and real data generated by radiosondes. About 70% of significantly damaged test data volume was recovered. Up to 100% of real data was fixed.

Keywords—aerology, data fixing, data fusion, meteorological data, monitoring, radiosonde.

1. Introduction

Radiosonde monitoring delivers important information about convective available potential energy (CAPE), storm relative helicity (SRH), lifted condensation level (LCL) or wind shears at different altitudes [1]. The combination of specific values of these parameters can indicate high probability of the development of a tornadic supercell storm (high CAPE in area of high SRH accompanied by low LCL). In Figs. 1 and 2 the most important thermodynamic and kinematic parameters related to the tornadic supercell are presented, which has happened on August 15, 2008 near Strzelce Opolskie, Poland.

Thermodynamical data generated by radiosondes can be represented in graphical Skew-T log-P diagrams (shown in Fig. 1) and Stuve diagrams [2]. The hodograph [3] is shown in Fig. 2. Hodograph is a diagram for graphical representation of the wind velocity vectors.

Information delivered by radiosonde monitoring is important for providing meteorological warnings and post incidental case studies. Upper air monitoring is also source of data for the numerical weather prediction models, which are based on the concept of the characteristics of the atmosphere as a fluid. That data is also useful in the analysis of a vertical profile of the atmosphere. The general equations of fluid dynamics and thermodynamics are used for the estimation of atmosphere states at the specific time slots [4]. Those equations are sensitive on the input data errors. The quality of that data should be as high as possible and numerical weather prediction models solve those dynamic and thermodynamic equations. The obtained numerical results are presented to forecasters in a form of maps and charts in order to aid the process of the weather forecast. The

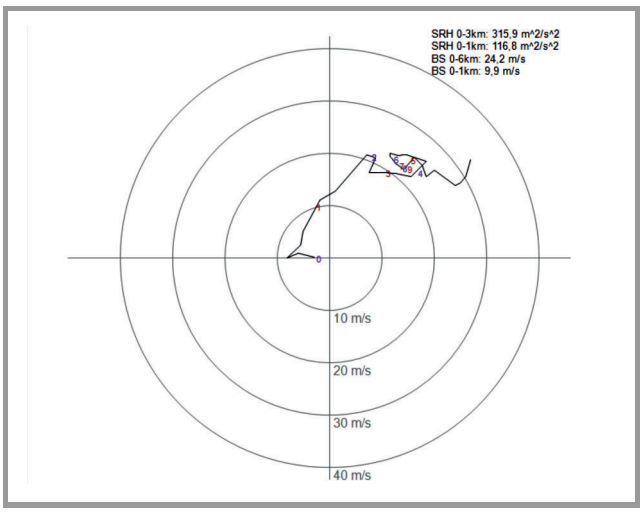

Fig. 1. Data from Poprad-Ganovce radiosonde monitoring, gathered on August 15, 2008, plotted on hodograph. Characteristic for a tornado shape of wind profile is visible.

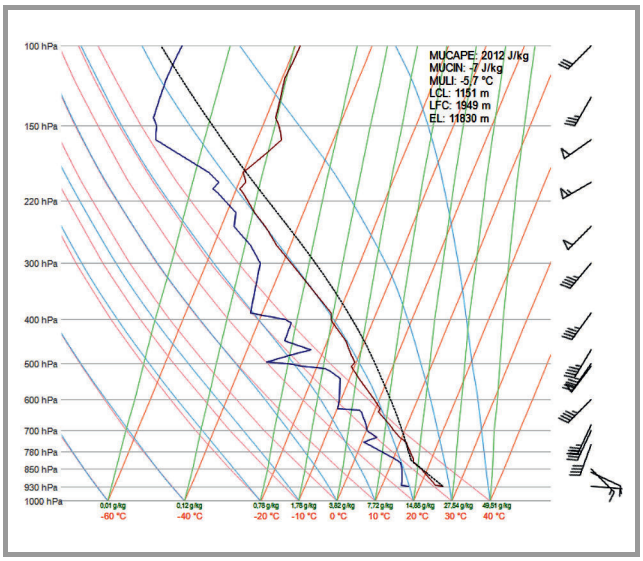

Fig. 2. Data from Poprad-Ganovce radiosonde monitoring, gathered on August 15, 2008, plotted on Skew-T log-P diagram. Large convective available potential energy is clearly visible. (See color pictures online at www.nit.eu/publications/ journal-jtit)

map presented in Fig. 3 defines the spatial distribution of the various parameters used in solving the above-mentioned analytical models.

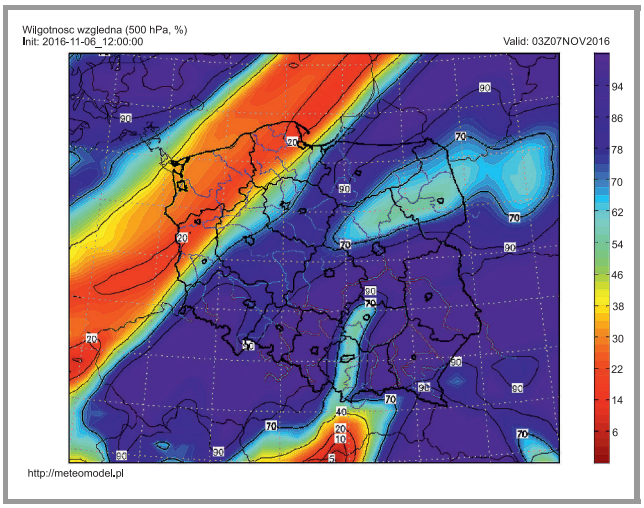

Fig. 3. Spatial distribution of relative humidity at 500 hPa height level for Poland. Numerical forecast generated by GFS model for November 7, 2016 03Z. Initial conditions taken from November 6, 2016 12Z.

There are many sources of input data (settings) in numerical prediction models. Such data is generated during the monitoring processes provided in irregularly distributed physical locations. Therefore, the generated data must be merged and analyzed based on the specified quality requirements. The major challenge is the large number of data sources, active monitoring processes executed simultaneously, and large data volumes. There is a need of the development of the new efficient algorithms and tools for data repairing and data fusion, which are necessary for providing data analysis. All those processes should be automated and independent on the administrators and all type of computational platforms.

In this paper an algorithm for detecting the damaged data records and repairing. Proposed methodology is based on fast interpolation data matrix structures stored at the data servers.

The paper is organized as follows. In Section 2 the data sources are specified together with the definition of the data matrices. In Section 3 the existing methods used in data healing are compared and in processing of the vast datasets. The computational and implementation aspects of the proposed models are discussed in Sections 4 and 5. Section 6 concludes the paper.

2. Data Characteristics

Radiosonde upper air monitoring is the most popular method of the data gathering for the analysis of weather conditions in the whole troposphere. In order to provide the proper analysis of weather condition, there is a need to obtain the complete data from such monitoring process without incorrect (missing or out of range) values. It becomes necessary to take a close look into radiosonde monitoring data that is distributed by the University Wyoming in the public repository [5]. In that repository, the aerological data is collected in most of countries. Radiosonde monitoring is often interpreted as sounding process, and it is performed twice a day. The data gathered during the monitoring can be saved as raws records, text list, and plain text. The graphical representations of such data structures are tephigrams and hodographs stored as GIF and PDF files.

The easiest format for the data analysis is the text list format presented in Table 1. In such case, the dataset is defined as a matrix. The columns represent the atmosphere attributes: pressure (P), height (H), temperature (T), dew point (D), relative humidity (RH), mixing ratio (ω), wind direction (A), wind speed (S), potential temperature (θ_a), equivalent potential temperature (θ_e) and virtual potential temperature (θ_v) [6]. In this paper, the attributes RH, ω, θ_a, θ_e and θ_v are ignored. Therefore, the number of columns of matrix is reduced to five. The exemplary values of such parameters are shown in the Table 2.

Table 1
The example of soundings data format after unnecessary cells reduction

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
925.0	888	3.6	–15.4	0	0
897.0	1141	9.8		150	16
700.0	3167	–0.1	–5.0	165	16
500.0	5760			300	50
300.0	9290	–49.3	0.0	65	10
100.0	16110	–66.3		310	16

Table 2
The example of sounding data values after correction performed by proposed algorithm

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
925.0	888.0	3.6	–15.4	0	0
897.0	1141.0	9.8	–14.1	150.0	16.0
700.0	3167.0	–0.1	–5.0	165.0	16.0
500.0	5760.0	–10.2	–10.2	300.0	50.0
300.0	9290.0	–49.6	–49.6	65.0	10.0
100.0	16110.0	–66.3	–66.3	310.0	16.0

Each row of the data matrix defines different altitude of the measurement performed by the ascending radiosonde. Rows are sorted in ascending order based on the altitude parameters and in descending order based on the pressure

parameters. Table 2 presents exemplary seven rows of the matrix with six cells each. The first row contains complete data from the monitoring and the range of the data values is correct. The second row has missing data in the fourth cell (dew point). The dew point data at the fourth and sixth rows are also missing. The fifth row does not contain both the temperature and the dew point parameters. The fourth row has missing temperature parameter.

2.1. Data Corruption

In existing weather forecasting systems, three main methods are implemented for the healing of the missing data of sounding, namely:

- the elimination of rows with incorrect values,
- manual repair of the damaged data,
- interpolation of the damaged data.

The correct data can be lost if all the rows, which contain incorrect values are removed. The rows can consist of both damaged and correct data.
A simple example of sounding with missing data is presented in [7].

2.2. Data Volume and Velocity

Data generated by different radiosondes is considered as representation of the atmosphere states at different time slots and different physical locations. In order to provide the fair analysis of the atmosphere in Poland, there is a need to collect the data from radiosonde monitoring provided in the neighbor countries, i.e. in Germany, Czech Republic, Slovakia, Ukraine, Belarus and Russia. The generated data matrix for such monitoring consists of up to 130 rows. Each row has six significant cells.
In most cases, the monitoring at aerological stations is provided twice a day (at 00Z and 12Z). By committing a single query data from one station, gathered within one month (maximum 31 days) can be requested. At time up to 62 soundings containing approximately 8060 rows can be obtained from Wyoming University's database. Finally, there are typically 62 soundings (about 50,000 values to check). Assuming that 25% of the values are damaged, there are approximately 12,500 cells to cure in order to heal soundings from a whole month.
Manual data repairing is a time-consuming process. In that case, every corrupted data record should be detected corrected. Software tools designed for manual data repairing require the involvement of the users and administrators who must assist the whole data healing. For large soundings with data matrices (over 100 rows), such process must be automated. The efficient method data healing can be based on the interpolation method.

3. Data Fixing Algorithms

In [8] and [9], the authors analyze the reasons of missing data and define the following three possible "data missingness" scenarios:

- missingness completely at random (MCAR),
- missingness at random (MAR),
- missingness not at random (MNAR).

Statistical analysis can be useful in solving the missing data problems [9]. The statistical approach shown in [9] can be classified according to the different criteria. First criterion is data shifting, which can be realized as data discarding or data retaining.
Three other methods defined as data discarding analysis:

- complete-case analysis,
- available-case analysis,
- non-response weighting.

In complete-case analysis, all the rows or columns of radiosonde data matrix with missing or incorrect data are removed. This will result in the loss of correct information because if there is a row with one missing data, such row is removed from the data matrix.
In the available-case, the columns of data matrix are grouped into two categories: damaged and correct, and next the damaged values are removed. The remaining columns are suitable for the further analysis. In non response weighting, the columns classified as "correct" in complete-case analysis are reweighted (i.e. the values of the parameters are reweighted). Reweighting is made in order to restore the proper representation of the parameters. In most of the cases, the loss of data is very undesirable and data retaining methods can be more effective. Such methods are based on imputation – filling gaps in data records. The most popular imputation methods include:

- mean imputation,
- last value carried forward,
- information from related monitoring dependent,
- using indicator variables for missingness,
- imputation based on logical rules, and others.

In mean imputation, the missing data values are filled with mean values of remaining data. In last value carried forward, the gaps in data records are filled with the data from predecessor cells. In order to fix data information from related monitoring also could be used, but there is a need to recognize relations between them and a degree of misrepresentation. Those methods could be called information from related monitoring dependent. Methods that are using indicator variables are based on concept of creation of

extra category, that has an information about missingness. Some solution also come from logical rules, for example if there is no measurement, its value is set to zero.
The problem is more complex when more than one data variable is missing. One joint general model of imputation can be defined for all the variables. In addition, different models for each column could be applied.
All the methods mentioned above are based on the statistical subsets and performed in order to preserve the proper representation of the set and original probability distribution.
In the single sounding case, probability distribution of different values cannot be considered. There is only vertical distribution so all the methods are unsuitable. For example, mean imputation of filling missing values with mean can lead to nonsense. If mean of temperature during day is equal to -5°C, the missing value occurs at ground level and there is a summer, which is usually warm, it is leading to nonsense.

3.1. Traditional Interpolation Methods

Traditional interpolation methods can be also useful in data healing. In that case, reference points that have correct values are defined as interpolation nodes. The useful in data healing interpolation methods include:

- piecewise linear interpolation,
- Lagrange interpolation,
- spline interpolation.

The easiest way to interpolate data in tabular set is to replace a real function that delivers value of each pair of the set with a linear function [10]. In this case, there are two discrete sets: X – set of n arguments and Y – set of n values. Hence, n following samples can be generated:

$$(x_i, y_i): \quad x_i \in X, y_i \in Y, i \in 1, \ldots, n. \tag{1}$$

Let us define the function $f(x_i) = y_i$, which can be approximated by the linear function $g(x)$:

$$g(x) = ax + b. \tag{2}$$

For each set of the following parameters $x_i, y_i, x_{i+1}, y_{i+1}$ for $i \in 1, \ldots, n-1$, the following conditions are specified:

$$g(x_i) = y_i, \tag{3}$$

$$g(x_{i+1}) = y_{i+1}. \tag{4}$$

The function $g(x)$ can approximate $f(x)$ within the range $[x_i, x_{i+1}]$. Now the analytic formulas for a and b in the following way can be defined:

$$a = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \tag{5}$$

$$b = y_i - a \cdot x_i. \tag{6}$$

The main advantages of using the piecewise linear interpolation can be formulated as:

- it requires only two complete samples to create the approximation for the range between them,
- it is of a low computing task and power efficient,
- it is easy for the implementation.

The main disadvantages of using the piecewise linear interpolation is:

- it is not smooth,
- for sparse sets and complex functions there can be large error occurring.

In the Lagrange interpolation [11], in order to define the n-th Lagrange interpolation polynomial, we have to:

- define the basis polynomials,
- define the Lagrange interpolation polynomial.

For $n+1$ discrete points (x_i, y_i) for $i \in 0, \ldots, n$ the basis polynomial $B_i(x)$ is defined as:

$$B_i(x) = \prod_{\substack{0 \le k \le n \\ k \ne i}} \frac{x - x_k}{x_i - x_k}. \tag{7}$$

Lagrange interpolation polynomial is defined in the following way:

$$L(x) = \sum_{i=0}^{n} y_i B_i(x). \tag{8}$$

The Lagrange interpolation is numerically unstable method due to Runge's phenomenon [12]. The computational complexity of that method is $O(n^2)$. All those makes this method rather ineffective in solving data-healing problems.
The third considered interpolation method is spline interpolation. It is based on the polynomials that have degrees lower than the number of samples. There are $n+1$ discrete points generated (x_i, y_i) for $i \in 0, \ldots, n$. The interpolation between all the pairs of the set is defined by polynomials $q_i(x)$ for $i \in 1, \ldots, n$. The following conditions should be satisfied:

$$\begin{cases} q_i'(x_i) = q_{i+1}'(x_i) \\ q_i''(x_i) = q_{i+1}''(x_i) \end{cases}. \tag{9}$$

The classical approach is to use cubic splines with third degree polynomials [13]. A spline interpolation is a very smooth and avoids Runge's phenomenon. In case of interpolation of aerological data smoothness is not desirable. For example large error can exist if there is sudden and large temperature inversion. In this case, conditions shown in Eq. (9) can lead to a large error.

In case of soundings there is a multivariate interpolation problem. The simplest and most promising way seems to be a usage of the piecewise linear interpolation.

3.2. Modeling Using Soft Computing

Another approach to solve missing data problem is to use soft computing components such as Support Vector Machine (SVM) or fuzzy logic.
The use of SVM model is described in [14]. General idea consists of few steps: at first we have to choose some measurements that do not have missing values, then select input, output attributes and decision attributes and finally apply regression model.
In [15] the authors proposed other methods of data imputation, i.e. *k*-nearest neighbor imputation (KNNI) that uses data from the most similar neighbors by imputing mean value from their monitoring. The second interesting way is to use value that comes from estimated distribution of missing value. That method is called hot deck imputation. The third manner of doing data curing is to use predictive model to estimate values that are missing. In that way attributes of missing value are used as the response attribute. The rest of them are used to create the model as an input. The authors also proposed a use of decision trees by using built-in approaches.
The soft computing methods mentioned above are more complex than the interpolation methods. Creating different predictive or supervised learning model for each sounding is in fact an ineffective way of data healing.

4. Proposed Methodology

The proposed algorithm use sounding in text list format. Then two stages correction process that consists is performed.

4.1. Validation Stage

At first stage validation is performed. Described in the first section data structure (matrix like) input is divided into m lines l_m. There has to be lexicographical order between all the lines in the set. Each line $l_k \quad k = 1,\ldots,m$ is split into n cells c_n. Cells that are not necessary (relative humidity, mixing ratio, thta, thte, thtv) are now removed so each line l_k consists of $n-5$ cells. That is lexical analysis part. In the next step each cell is verified if it has a value and if the value consists of permissible characters. That is syntax analysis. At this time the matrix of validity $X = m \times n-5$, the matrix of values $V = m \times n-5$ and $\vec{x}$ of invalid rows indexes are created:

$$X_{k,o} = S(l_k, o)\,, \tag{10}$$

where S is a syntax error function, l_k is k-th line and o is o-th cell in k-th line. When

$$l_k = [c_1,\ldots,c_{n-5}]\,, \tag{11}$$

then

$$S(l_k,o) = \begin{cases} 2 & \text{if } c_o \text{ consists of illegal characters} \\ 1 & \text{if } c_o \text{ is empty} \\ 0 & \text{otherwise} \end{cases}. \tag{12}$$

$V_{k,o}$ is set to zero if $S(l_k,o)$ is not equal to zero else $V_{k,o} = l_{k_o}$ then $X_{k,o} = R_o(V_{k,o})$ where R_o is o-th range error function. The o-th means that for different column of value matrix different correct range $[r_{o_l}, r_{o_u}]$ of values is defined.

$$R_o(V_{k,o}) = \begin{cases} 0 & \text{if } V_{k,o} \in [r_{o_l}, r_{o_u}] \\ 3 & \text{otherwise} \end{cases}, \tag{13}$$

For each k-th row of matrix X following sum is calculated:

$$\sum_{o=1}^{n-5} X_{k,o}\,, \tag{14}$$

and if the sum is greater than zero to vector $\vec{x}$ at the end element equals k is added.

4.2. Curing Stage

Each element of vector $\vec{x}$ contains index of the values matrix V row that is composed of cells that have damaged values. At this stage for each index stored in $\vec{x}$ algorithm is finding two nearest rows (curing rows) that have correct values that are damaged in cured row and at least one value that is correct in both cured and curing rows. The vector $\vec{u} = X_k$ is the nearest to $\vec{b} = X_{x_i}$ when difference between k and x_i is minimal.
Because there is the matrix of validity X that was created in previous stage, for each value x_i stored in $\vec{x}$ algorithm is analyzing each row of matrix of validity X:

$$\vec{b} = X_{x_i}\,. \tag{15}$$

Each value $\vec{b}_j$ of row $\vec{b}$ cells is now converted to Boolean form:

$$\vec{c}_j = B(\vec{b}_j) = \begin{cases} \text{true} & \text{if} \quad \vec{b}_j = 0 \\ \text{false} & \text{otherwise} \end{cases}. \tag{16}$$

Then for each false value $\vec{c}_j$ in $\vec{c}$ algorithm searches for the two nearest rows of X $\vec{f} = X_k, \vec{l} = X_l \wedge k \neq l \wedge k, l \in 1,\ldots,m$ that fits criteria:

$$\vec{f}_j = 0 \wedge \vec{l}_j = 0\,. \tag{17}$$

And after creation of vector of comparison $c\vec{m}p$ such as:

$$c\vec{m}p_j = \vec{c}_j \wedge B(\vec{f}_j) \wedge B(\vec{l}_j)\,. \tag{18}$$

The $c\vec{m}p$ has at least one true value. Two first vectors $\vec{f}, \vec{l}$ that are meeting mentioned requirements are chosen to be curing vectors.
For the chosen curing vectors $\vec{f} = X_k, \vec{l} = X_l$ algorithm is getting corresponding values vectors from V: $\vec{o} = V_k, \vec{p} = V_l$. Then if $l > k$ swapping is performed:

$$\vec{o} = \vec{p} \qquad \vec{p} = \vec{o}, \tag{19}$$

in order to maintain the ascending order. Then index z of the first cell of $c\vec{m}p$ that has true value is obtained and finally:

$$V_{x_i,j} = Vx_i,z \cdot \frac{\vec{p}_j - \vec{o}_j}{\vec{p}_z - \vec{o}_z} + \vec{o}_j - \vec{o}_z \cdot \left(\frac{\vec{p}_j - \vec{o}_j}{\vec{p}_z - \vec{o}_z}\right). \quad (20)$$

The missing value is interpolated by using of linear interpolation.

4.3. Algorithm Limitations

Developed algorithm has some limitations. The first limitation is that each damaged row of sounding has to have at least one correct value in order to make curing possible. The second limitation is that each sounding column has to have at least two correct values. At least one column row has to have lexicographical order.
Assuming that the number of damaged lines m will be significantly lower than the number of whole lines of sounding n $(m << n)$ the theoretical computational complexity will be $O(mn^2)$ in the worst case. The typical sounding gathered in the USA has up to 130 lines.

5. Algorithm Implementation

The developed algorithm was implemented in C# language and was integrated with Wyoming University Data Repository.

5.1. Validation Stage

The first step is to split each line into cells and to remove those that are not necessary.

```
p_ = line.Substring(1, 6);
p_ = p_.Replace(" ", String.Empty);
h_ = line.Substring(9, 5);
h_ = h_.Replace(" ", String.Empty);
t_ = line.Substring(16, 5);
t_ = t_.Replace(" ", String.Empty);
d_ = line.Substring(23, 5);
d_ = d_.Replace(" ", String.Empty);
a_ = line.Substring(46, 3);
a_ = a_.Replace(" ", String.Empty);
v_ = line.Substring(52, 4);
v_ = v_.Replace(" ", String.Empty);
```

The second step is to perform syntax analysis.

```
vp = tryParseDouble(p_, out p);
vh = tryParseDouble(h_, out h);
vt = tryParseDouble(t_, out t);
vd = tryParseDouble(d_, out d);
va = tryParseDouble(a_, out a);
vv = tryParseDouble(v_, out v);
```

Then correct range is checked and matrix of validity is created.

```
if (p < 0)
lineStatus.Add(LineStatus.PRES_ERR);
if (t < -273.15 || t > 100.0)
lineStatus.Add(LineStatus.TEMP_ERR);
if (d > t || d < -273.15 || d > 100.0)
lineStatus.Add(LineStatus.DEW_ERR);
if (a < 0 || a > 360)
lineStatus.Add(LineStatus.ANG_ERROR);
if (v < 0)
lineStatus.Add(LineStatus.VEL_ERROR);
if (!vp)
lineStatus.Add(LineStatus.PRES_MISSING);
if (!vh)
lineStatus.Add(LineStatus.HEI_MISSING);
if (!vt)
lineStatus.Add(LineStatus.TEMP_MISSING);
if (!vd)
lineStatus.Add(LineStatus.DEW_MISSING);
if (!va)
lineStatus.Add(LineStatus.ANG_MISSING);
if (!vv)
lineStatus.Add(LineStatus.VEL_MISSING);
```

5.2. Curing Stage

When the validation stage is completed then for each damaged cell in each damaged row indexes of curing rows are set to be out of range, index of healed row is copied to temporary variable and row of matrix of validity is converted to Boolean form.

```
bool[] cmp = null;
int np = -ylen-1; int nnp = -ylen-1;
int dmgInd = j;
bool[] cdmg;
cdmg = damagedColumns(sList[dmgInd]);
```

Then the first curing row is chosen to be checked.

```
for
(k = 0; k < soundingList[i].Count; k++){
bool[] dmg = null;
bool[] dmg2 = null;
if (k!=dmgInd){
dmg = damagedColumns(sList[k]);
```

Then the second curing row is chosen to be checked.

```
for
(l = 0; l < soundingList[i].Count; l++){
if (l != k&&l!=dmgInd){
dmg2 = damagedColumns(sList[k]);
int ccp;
```

And next comparison is the made.

```
int ccp;
bool[] comparison =
compareDamaged(dmg, dmg2,
cdmg, out ccp, i);
```

If both chosen rows meet requirements.

```
if (ccp >= 1 &&
Math.Abs(dmgInd - k)
< Math.Abs(dmgInd - np)
&& Math.Abs(dmgInd - l)
< Math.Abs(dmgInd - nnp))
```

Rows are marked as curing rows.

```
np = k;
nnp = l;
cmp = comparison;
```

Then swapping is performed.

```
int f, n;
if (np < nnp){
    f = np; n = nnp;
}
else{
    f = nnp; n = np;
}
```

Appropriate values are chosen to cure the damaged one.

```
double y1 = soundingList[i][f];
double y2 = soundingList[i][n];
double x1 = 0;
double x2 = 0;
double gx = 0;
```

Then the argument of linear function is chosen basing on information from vector of comparison.

```
for (int m = 0; m < cmp.Length; m++){
if (m != i && cmp[m] == true){
x1 = soundingList[m][f];
x2 = soundingList[m][n];
gx = soundingList[m][dmgInd];
break;
}
}
```

Finally, an interpolation is made and the value is inserted into values matrix.

```
double val =
MathHelper.interpolateLinear
(x1, y1, x2, y2, gx);
soundingList[i][j] = val;
```

Note that *i* is index of column and *j* is row index.

5.3. Tests

The proposed algorithm tests were divided into two stages. At the first stage they were performed on test data that was prepared for it (Table 3). Tests were performed as unit tests.

Table 3
The example of soundings dataset prepared for testing

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
1014.0	96.0	1.0	0.3	250.0	8.0
850.0	1499.0	–6.3	–6.3	295.0	25.0
700.0	3002.0		–13.9	320.0	27.0
500.0	5520.0	–24.5		335.0	43.0
400.0	7100.0	–37.7	–41.1	340.0	45.0
300.0	9020.0		–57.4	345.0	56.0
200.0	11550.0	–58.9		330.0	39.0
100.0		–60.5	–87.5	330.0	31.0

Table 4
The example of soundings data set prepared for testing after correction made by proposed algorithm

P [hPa]	H [m]	T [°C]	D [°C]	A [°]	S [kt]
1014.0	96.0	1.0	0.3	250.0	8.0
850.0	1499.0	–6.3	–6.3	295.0	25.0
700.0	3002.0	–13.0	–13.9	320.0	27.0
500.0	5520.0	–24.5	–24.5	335.0	43.0
400.0	7100.0	–37.7	–41.1	340.0	45.0
300.0	9020.0	–50.9	–57.4	345.0	56.0
200.0	11550.0	–58.9	–73.7	330.0	39.0
100.0	14080.0	–60.5	–87.5	330.0	31.0

During the second stage of testing algorithm was connected to component that was responsible for the acquiring data from Wyoming University's website, so tests were performed on real data from radiosondes (Table 4). To make an outcome comparison, the algorithm was also connected to components of data edition and diagram generation created by the author. Also some vital thermodynamical and kinematical indices were calculated. The test sets for that stage were made from well-known soundings, which are related to famous severe weather situations. In that manner the comparison between actual outcome and expected values can be performed.

5.4. Test Results

The most of the real data examples were slightly damaged so the proposed algorithm recovered 100% of missing or damaged data in most cases. Test examples prepared for the first phase of testing were significantly damaged by the author. For example, the whole column was missing or some rows have only one correct value. There were also combinations of the missing columns and rows. About 70% of prepared data set examples were able to be used in processing, e.g. creating diagrams, computing thermodynamical indices, etc. "The example was able to be used"

means that after a correction, the outcome from processing was not misleading user – it does not result in incorrect assessment of the weather situation. In all the test cases runtime of algorithm is significantly lower than runtime of processing routines. In case of remote data usage, the time of the whole procedures was dominated by remote data obtaining time, thus confirmed proposed model's efficiency.

6. Conclusions

The developed algorithm that bases on linear interpolation in the most cases repairs all the available data. It can be classified as an imputation algorithm. It applies piecewise linear interpolation into multivariate model. During curing two the most proper values are chosen in order to heal a damaged cell so it is also related to the nearest neighbor idea. Due to its low runtime and easy understandable concept, it can be adapted to other fields that use data structures similar to that described in Section 2. The author is interested in data fusion, especially in fusion of meteorological data, so the algorithm will be used in cases that will require of a use of soundings data. The algorithm will be also used in the next versions of Sounding Decoder – the author's software solution dedicated to sounding processing.

References

[1] R. Thompson, "Explanation of SPC severe weather parameters", Storm Prediction Center, National Oceanic and Atmospheric Administration [Online]. Available: http://www.spc.noaa.gov/sfctest/help/sfcoa.html

[2] D. Dempsey, "The Stuve and Skew-T/Log-P thermodynamic diagrams", Dept. of Geosciences, San Francisco State University, 2009 [Online]. Available: http://funnel.sfsu.edu/courses/metr201/S12/handouts/stuve_skewt.html

[3] T. A. Apostolatos, "Hodograph: A useful geometrical tool for solving some difficult problems in dynamics", *Amer J. of Phys.*, vol. 71, no. 3, pp. 261–266, 2003 [Online]. Available: http://users.uoa.gr/∼pjioannou/mech1/SITES/hodograph.pdf (doi: 10.1119/1.1527948).

[4] W. C. Skamarock *et al.*, "A Description of the Advanced Research WRF Version 3", National Center for Atmospheric Research Boulder, CO, USA, Jun. 2008 [Online]. Available: http://www2.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf

[5] "Atmospheric Soundings", University of Wyoming [Online]. Available: http://weather.uwyo.edu/upperair/sounding.html

[6] "Description of Sounding Columns", University of Wyoming [Online]. Available: http://weather.uwyo.edu/upperair/columns.html

[7] "33791 Kryvyi Rih monitoring at 12Z 01 Dec 2016", University of Wyoming [Online]. Available: http://weather.uwyo.edu/cgi-bin/sounding?region=europe&TYPE=TEXT%3ALIST&YEAR=2016&MONTH=12&FROM=0112&TO=0112&STNM=33791.

[8] T. D. Pigott, "A review of methods for missing data", *J. of. Educational Res. and Eval.*, vol. 7, no. 4, pp. 353–383, 2001.

[9] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, 1st ed. Cambridge University Press, 2006, pp. 529–543.

[10] T. Blu, "Linear interpolation revitalized", *IEEE Trans. on Image Process.*, vol. 13, no. 5, pp. 710–719, 2004 (doi: 10.1109/TIP.2004.826093).

[11] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation", *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, 2004.

[12] C. Runge, "Über empirische Funktionen und die Interpolation zwischen aquidistanten Ordinaton", *Zeitschrift für Mathematik und Physik*, vol. 46, pp. 224–243, 1902 (in German).

[13] G. Wolberg, "Cubic spline interpolation: A review", Tech. Rep. CUCS-389-88, Columbia University, Computer Science Reports, New York, USA, 1988.

[14] F. Honghai *et al.*, "A SVM Regression Based Approach to Filling in Missing Values", in *Knowledge-Based Intelligent Information and Engineering Systems*, R. Khosla, R. J. Howlett, and L. C. Jain, Eds. *LNCS*, vol. 3683, pp. 581–587. Springer, 2005.

[15] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect in the classifier accuracy", in *Classification, Clustering, and Data Mining Applications*, D. Banks *et al.*, Eds. Springer, 2004 [Online]. Available: http://sci2s.ugr.es/sites/default/files/files/TematicWebSites/MVDM///IFCS04r.pdf

Piotr Szuster graduated in Computer Science at Cracow University of Technology, Poland, in 2016. Currently, he is a research and teaching assistant at Cracow University of Technology and a Ph.D. student at AGH University of Science and Technology. The main topics of his research are Big Data, Data Fusion and Internet of Things.

E-mail: Piotr.Szuster@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

AGH University of Science and Technology
Mickiewicza av. 30
30-059 Cracow, Poland