



Article

# Benchmarking Daily Line Loss Rates of Low Voltage Transformer Regions in Power Grid Based on Robust Neural Network

Weijiang Wu <sup>1,2</sup>, Lilin Cheng <sup>3,\*</sup>, Yu Zhou <sup>1,2</sup>, Bo Xu <sup>4</sup>, Haixiang Zang <sup>3</sup>, Gaojun Xu <sup>1,2</sup> and Xiaoquan Lu <sup>1,2</sup>

<sup>1</sup> State Grid Jiangsu Electric Power Company Limited Marketing Service Center, Nanjing 210019, China; weijiangwu@126.com (W.W.); zhoyuudky@163.com (Y.Z.); hhuxgj@163.com (G.X.); luxiaoquandy@163.com (X.L.)

<sup>2</sup> State Grid Key Laboratory of Electric Power Metering, Nanjing 210019, China

<sup>3</sup> College of Energy and Electrical Engineering, Hohai University, Nanjing 211100, China; zanghaixiang@hhu.edu.cn

<sup>4</sup> Jiangsu Frontier Electric Technology Co., Ltd., Nanjing 211100, China; 15951865004@163.com

\* Correspondence: straw@hhu.edu.cn; Tel.: +86-173-7211-8166

Received: 28 September 2019; Accepted: 3 December 2019; Published: 17 December 2019



**Abstract:** Line loss is inherent in transmission and distribution stages, which can cause certain impacts on the profits of power-supply corporations. Thus, it is an important indicator and a benchmark value of which is needed to evaluate daily line loss rates in low voltage transformer regions. However, the number of regions is usually very large, and the dataset of line loss rates contains massive outliers. It is critical to develop a regression model with both great robustness and efficiency when trained on big data samples. In this case, a novel method based on robust neural network (RNN) is proposed. It is a multi-path network model with denoising auto-encoder (DAE), which takes the advantages of dropout, L2 regularization and Huber loss function. It can achieve several different outputs, which are utilized to compute benchmark values and reasonable intervals. Based on the comparison results, the proposed RNN possesses both superb robustness and accuracy, which outperforms the testing conventional regression models. According to the benchmark analysis, there are about 13% outliers in the collected dataset and about 45% regions that hold outliers within a month. Hence, the quality of line loss rate data should still be further improved.

**Keywords:** benchmark calculation; daily line loss rate; low voltage transformer region; robust neural network

## 1. Introduction

Line loss rate is a vital indicator in power grid as it can reflect the operation and management levels in both economic and technical aspects [1]. It is inevitable and directly impacts the profits of power-supply corporations [2]. Commonly, line loss can be classified as technical and non-technical ones. In terms of technical line loss, it is caused by the electro-thermal effect of conductors, which is unavoidable over the transmission process. The level of technical line loss depends on the structure and operation state of the power grid, conductor type, as well as the balance condition of three-phase loads [3]. Non-technical line loss usually occurs due to electricity theft, which may lead to abnormal values in metered line loss rates. Hence, it is urgently concerned by power grid operators whether the daily line loss rate values are within a reasonable range, i.e., the pass percentages of daily line loss rates. It is sometimes difficult to distinguish normal line loss rate values and outliers from a large number of collected samples. Besides the intervals, a benchmark value of daily line loss rate is also essential for

transformer regions, as it directly provides the information what a correct daily line loss rate value the region should approximately achieve, in order for operators to better know the operating condition of the region, as well to further improve the level of line loss management. In this case, an accurate calculation method is still needed nowadays to obtain the benchmark values and reasonable intervals of line loss rates, as well as to recognize outliers from those samples of line loss rates.

In the field of data mining and analysis, there are usually four approaches to calculate benchmarks and detect outliers, i.e., the empirical method, statistical method, unsupervised method and supervised method [4–7]. Firstly, empirical method utilizes practical experience to set an interval with fixed bounds, where a value out of the interval can be treated as an outlier. In the benchmarking of daily line loss rates, an empirical interval is customarily set to be  $-1\%$ ~ $5\%$ . It is noted that although line loss rate is usually non-negative, a value no less than  $-1\%$  is acceptable due to unavoidable acquisition errors. This method is simple and easy to realize, whereas a fixed interval is sometimes inaccurate and cannot reflect the influences of relevant factors. Secondly, the statistical method is aimed to research the distribution of data samples, where the outliers can be eliminated using probabilistic density functions [8] or box-plots [9,10]. Comparing to empirical methods, the interval bounds in statistical methods is able to adapt to different testing samples, while the influencing factors of line loss rates can still hardly be involved in this kind method. Thirdly, the unsupervised method, i.e., the clustering method, is also an efficient way to detect outliers [11]. In clustering, a sample of line loss rate can be treated as a data point, where the values of line loss rate and its influencing factors are the different dimensional attributes of the point. The outliers can be identified based on the distances between the data points and clustering centers [12–14]. Unsupervised methods usually outperform empirical and statistical methods, as multi-dimensional factors can be inputted and analyzed [15]. Nevertheless, unsupervised methods still face several problems. On one hand, it is sometimes difficult to design a proper distance function as those input factors are in dissimilar dimensions and units. On the other hand, some clustering methods involve all data points to update clustering centers, e.g., *k*-means and fuzzy C-means (FCM), where outliers may easily affect the centers and final clustering results. Reasonable intervals cannot be calculated by unsupervised methods as well. Finally, supervised methods utilize machine learning models to solve classification [16,17] and regression problems [18,19], which are designed for outlier detecting and benchmark calculating tasks, respectively. Classification models learn from labeled samples to distinguish normal and abnormal data. However, line loss rate samples are usually unlabeled, as it is unable to recognize whether a collected value of line loss rates is normal or not.

According to the relevant references, data-driven methods have been widely applied to estimate line loss values and loss rates, where clustering and regression methods are the two major ones [20]. In [21], FCM is adopted to select a reference value for each type of feeders, in order to calculate the limit line loss rates of feeders for distribution networks. Similarly, another cluster method, namely Gaussian mixture model (GMM), is utilized to calculate line loss rates for low-voltage transformer regions in [22]. The above two methods both calculate a fixed benchmark value for each cluster, where they are not designed for a single feeder or region. Considering this fact, regression methods are proposed to calculate a benchmark with certain inputs. Those inputs are the influencing factors of line losses from only one feeder or region. Reviewing the state-of-art methods, decision tree and its derived boosting models are the most frequent-used ones in line loss computation. In [23,24], gradient boosting decision tree (GBDT) is used to predict and estimate line losses for distribution networks and transmission lines, respectively. Among those, power flow and weather information are taken as input factors. In [25], extreme gradient boosting (XGBT) model is proposed to estimate line losses for distribution feeders, based on the characteristics of feeders.

Regression models can obtain benchmark values for line loss rate samples based on different influencing factors, where a value greatly away from benchmarks can be treated as an outlier. Nevertheless, a regression model may perform poor robustness and reliability when directly trained on samples with massive outliers, causing it critical to develop a highly robust method. In addition to

the boosting models mentioned before,  $k$ -nearest neighbors (KNN) is one of the most common-used machine learning models with great robustness [26–28]. It analyzes the similarity between the predicted sample and original training ones, in order to calculate an averaged value based on those nearest training samples. This calculating method can relieve certain impacts from outliers, but it increases the computational burden at the application stage and can hardly deal with high-dimensional inputs. Besides, support vector machine (SVM) is also frequently utilized for robust regression [29]. As the regression results of SVM are strongly correlated with support vectors, the influences of outliers are decreased. However, SVM shows low training efficiency on a large amount of samples. Due to the great number of regions in practical application, a method with both high efficiency and robustness is proposed in this study, called robust neural network (RNN). A neural network utilizes error back-propagation (BP) and mini-batch gradient descent algorithms to update its parameters, suitable for big data samples training. Besides, RNN modifies the structure of conventional neural networks, further increasing its robustness against outliers. The main contributions of this study can be summarized in detail as follows.

- (1) As the number of researches that focus on benchmarking daily line loss rates is limited, a supervised regression method is proposed in this study to obtain benchmark values of daily line loss rates in different transformer regions. In the proposed supervised method, various influencing factors of line loss rates are considered, where a high computation accuracy can be thus ensured.
- (2) A novel RNN model is proposed in this study. It possess a multi-path architecture with denoising auto-encoders (DAEs). Moreover, L2 regularization, dropout layer and Huber loss function are also applied in RNN. The robustness and reliability of the proposed regression model are greatly improved when compared with conventional machine learning models according to the testing datasets in the case study.
- (3) Based on the multiple outputs of the RNN, a method is proposed to calculate benchmark values and reasonable intervals for line loss rate samples. It can precisely evaluate the quality of sampled datasets and eliminate outliers of line loss rates, increasing the stability of data monitoring.

The rest of the paper is organized as follows. The utilized dataset and the proposed method based on RNN are introduced in Section 2. The comparison results and discussions are provided in Section 3. The conclusion is drawn finally in Section 4.

## 2. Materials and Methods

### 2.1. Theoretical Computation Equations of Line Losses

Theoretical equations of line losses are proposed to compute technical line losses, which are based on the equivalent resistance method. The method supposes that there is an equivalent resistance at the head of line, where the energy loss of three-phase three-wire and three-phase four-wire systems can be formulated as [30]:

$$\Delta A_b = NK^2 I_{av}^2 R_{eq} T \times 10^{-3} (\text{kWh}), \quad (1)$$

where  $\Delta A_b$  refers to the theoretical line loss with balanced three-phase load.  $N$  is the structure coefficient, which is equal to 3 at a three-phase three-wire system and equal to 3.5 at three-phase four-wire system.  $K$ ,  $I_{av}$ ,  $R_{eq}$  and  $T$  denote the shape coefficient of load curves, the average current at the head of line (A), the equivalent resistance of conductors ( $\Omega$ ) and the operating time (h), respectively. Furthermore,  $R_{eq}$  can be computed under the following equation:

$$R_{eq} = \frac{\sum_i N_i A_i^2 R_i}{N(\sum_j A_j)^2}, \quad (2)$$

where  $N_i$ ,  $A_i$  and  $R_i$  are the structure coefficient, the metered electricity power and the resistance of the  $i$ th line segment, respectively.  $A_j$  denotes the electricity power collected from the  $j$ th power meter. For a system with balanced three-phase load, the theoretical line loss should be corrected as:

$$\Delta A_{ub} = \Delta A_b \times K_{ub}, \tag{3}$$

where  $K_{ub}$  represents the correction coefficient that can be defined as:

$$K_{ub} = 1 + k\delta_I^2, \tag{4}$$

where  $k = 2$  with one phase heavy load and two phase light loads. If there are two phase heavy loads,  $k = 8$ .  $\delta_I$  denotes the unbalance level of three-phase load, which can be calculated as follows:

$$\delta_I = \frac{I_{max} - I_{av}}{I_{av}}, \tag{5}$$

where  $I_{max}$  is the current from the phase with the maximum load. Thus, the theoretical line loss defined above is an unavoidable energy loss, which is so-called technical line loss. However, the non-technical line loss caused by electricity theft is also concerned by power grid operators. As those non-technical loss situations can lead to abnormal values in the metered daily line loss rates, it is of necessity to calculate reasonable intervals for outlier discrimination, which is one of the purposes in this study.

## 2.2. Datasets

In practical application, the pass percentages of daily line loss rates in transformer regions are generally examined once a month in State Grid Corporation of China. In this case, the line loss rate dataset from July 2017 is utilized in the study, which is collected in daily intervals, in order to examine the pass percentage of line loss rates in the month. The pass percentage quota is especially important in July as it usually meets the peak load period of summer. The dataset is obtained from a total of 19,884 regions, which are located in Wuxi, Jiangsu Province, China. As a result, there are altogether 616,404 samples in this study, which satisfies the demand of big data analysis. Based on the dataset, about 80% of the samples (15,907 regions) are chosen as training ones and the others (3977 regions) are testing samples.

### 2.2.1. Data Quality Analysis

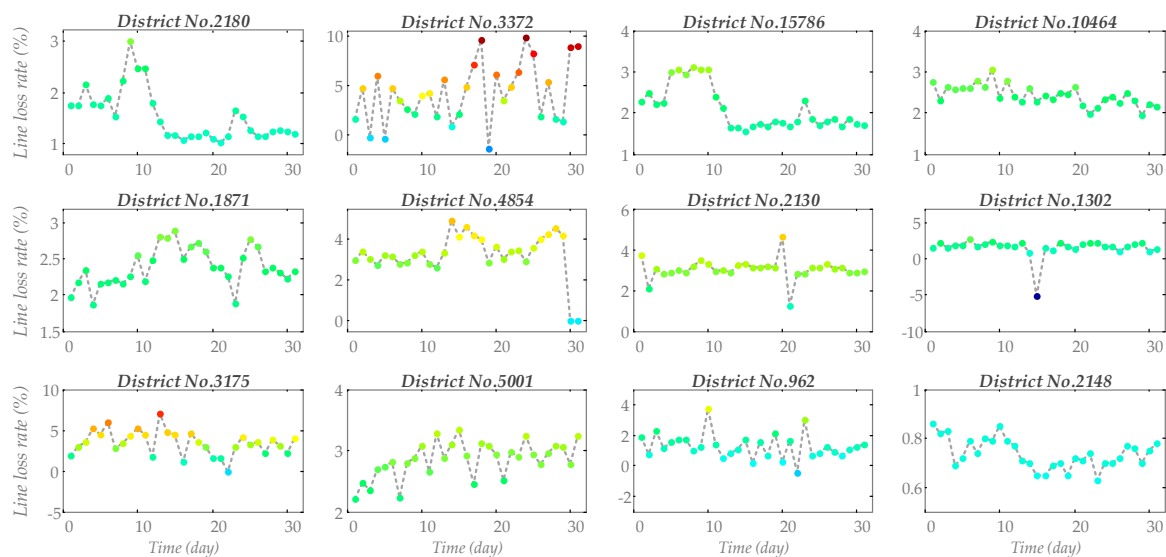
The research object in this study is daily line loss rate, some of whose curves are presented as examples in Figure 1. Besides, the 25 percentile ( $q1$ ), median ( $q2$ ), 75 percentile ( $q3$ ), maximum (max), minimum (min), mean, standard deviation (std), lower bound ( $l_a$ ) and upper bound ( $u_a$ ) values of the overall line loss rate dataset are calculated and provided in Table 1. The distribution box-plots of the line loss rates are provided in Figure 2. It is noted that lower bound ( $l_a$ ) and upper bound ( $u_a$ ) are calculated based on 25 percentile ( $q1$ ) and 75 percentile ( $q3$ ) [31], where a value out of the bounds can be treated as an outlier:

$$\begin{cases} l_a = q1 - 1.5 \times (q3 - q1) \\ u_a = q3 + 1.5 \times (q3 - q1) \end{cases} \tag{6}$$

According to the curves and quality analysis, the data characteristics of daily line loss rates can be summarized as follows:

1. The line loss rate data possess few daily regularity and show high fluctuation. From Figure 1, the curves of line loss rates in different regions change greatly over the days, where historical line loss rates can be hardly used to estimate the further values. Thus, it is vital in this study to select influencing factors of line loss rate.

2. The deviations of outliers in the dataset are sometimes extremely away from normal values, indicating the low dependability of the acquisition and communication equipment. According to Table 1 and Figure 2, the lower and upper bounds of the original dataset in the box-plot are  $-1.57\%$  and  $5.22\%$ , respectively, which is quite close to the project standard ( $-1\%$  and  $5\%$ ). However, the maximum and minimum of the collected line loss rates are  $100\%$  and  $-1.69 \times 10^6\%$ , respectively; greatly different from the bounds. In this case, benchmarking line loss rates is still necessary nowadays in practicable applications.
3. The quality of the dataset is poor to be used directly. As the component analysis of the dataset is presented in Figure 3, there are a large number of outliers and missing values, constituting  $8.67\%$  and  $6.72\%$  of the overall dataset. In this study, the spline interpolation method is utilized to fill the missing values. From Table 1 and Figure 2, the dataset after interpolation holds a similar distribution comparing to that of the original dataset. On the contrary, although the outliers can be eliminated directly based on  $l_a$  and  $u_a$ , the distribution will change and it will be difficult to calculate an accurate reasonable intervals.



**Figure 1.** Examples of daily line loss rates in July 2018 from different regions in Wuxi, Jiangsu Province, China.

**Table 1.** The data quality analysis based on the overall line loss rate dataset.

Indicator	Original Dataset	Original Dataset (Without Outliers)	Dataset after Interpolation
mean (%)	-5.16	1.78	-10.96
std (%)	$2.47 \times 10^3$	1.16	$3.19 \times 10^3$
min (%)	$-1.69 \times 10^6$	-1.57	$-1.69 \times 10^6$
max (%)	100	5.22	100
$l_a$ (%)	-1.57	-1.10	-1.50
$u_a$ (%)	5.22	4.57	5.26
q1 (%)	1.02	1.02	1.03
q2 (%)	1.74	1.67	1.76
q3 (%)	2.70	2.44	2.72

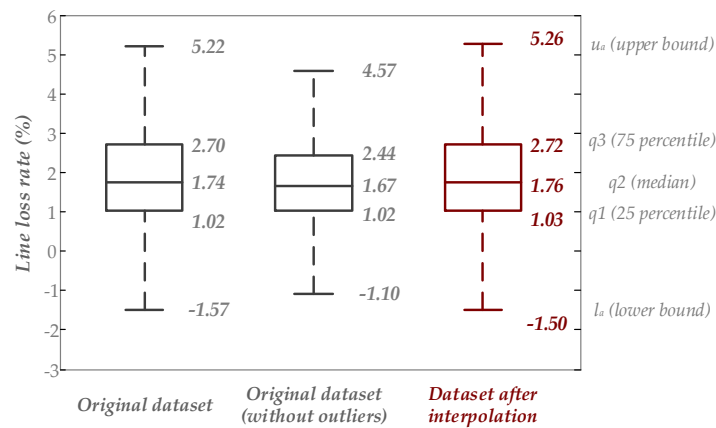


Figure 2. The distribution box-plots of original dataset and the dataset after interpolation operation.

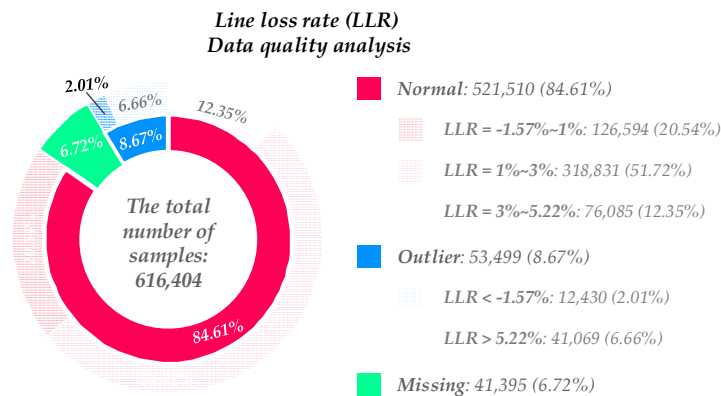


Figure 3. The component result of data quality analysis (84.61% normal values, 8.67% outliers and 6.72% missing values).

### 2.2.2. Influencing Factors of Line Loss Rate

Taking into account of both the possible influencing factors and the information recorded, a total of twelve factors are selected as inputs of the regression models, as shown in Table 2. Among those, the third and fourth factors are one-bit codes, while the others are numerical values.

Table 2. Influencing factors of line loss rate utilized in this study.

No.	Factor	Remarks
1	Date	-
2	Capacity of the transformer	-
3	Type of the transformer	Public transformer (=0); special transformer (=1)
4	Type of the grid that the region belongs to	City grid (=0); country grid (=1)
5	Monthly line loss rate	Three inputs, including the last three months
6	Daily load rate	Daily load rate = daily power supply/(transformer capacity × 24)
7	Daily maximum load rate	-
8	Daily average power factor	-
9	Number of customers	-
10	Average transformer capacity per customer	Average capacity = transformer capacity/number of customers
11	Rate of residential capacity	Rate = residential capacity/transformer capacity
12	Power supply duration	-

### 2.3. Calculation of Benchmark Values and Reasonable Intervals

According to the data quality analysis, the original datasets contain a large quantile of outliers, which are far from the rational range, causing it much difficult to obtain an accurate result. Therefore, the task of this study is to utilize robust learning strategy and achieve a stable regression result away from outliers, as shown in Figure 4.



**Figure 4.** Conventional learning methods may be easily affected by outliers. (a) Common condition; (b) affected by outliers.

Commonly, there is a robust learning solution that one can set thresholds manually and delete outliers from the dataset according to those thresholds, where the rest of the dataset can be applied to train a machine learning model. However, it still remains a problem how to decide precise thresholds. Besides, computed bounds of the reasonable interval from the learning model may be close to the manual thresholds, which breaks the distribution of the original dataset and makes it meaningless to train a probabilistic learning model. In this case, the calculation method based on RNN is proposed, as shown in Figure 5, which consists of the following steps:

1. Build a RNN. In order to fully expand its robustness, DAE, multi-path architecture, L2 regularization, dropout layer and Huber loss function are applied. It is noted that a RNN possesses ten output nodes, where each node is connected to one layer with a dissimilar dropout rate (from 0.05 to 0.50).
2. Calculate an average value according to the ten different outputs, which is the final benchmark value of line loss rate, as:

$$\tilde{y}_i = \frac{1}{10} \sum_{n=1}^{10} y_i^n, \tag{7}$$

where  $\tilde{y}_i$  is the  $i$ th benchmark value, and  $y_i^n$  is the  $n$ th output of the  $i$ th line loss rate.

3. Operate error analysis to acquire a reasonable interval. Not only the absolute error between the benchmark value and the actual line loss rates is computed, the variance of different outputs is calculated as well. According to the interval result, data points that are not involved between the bounds of the interval are considered as outliers. The operation can be described under the following equations:

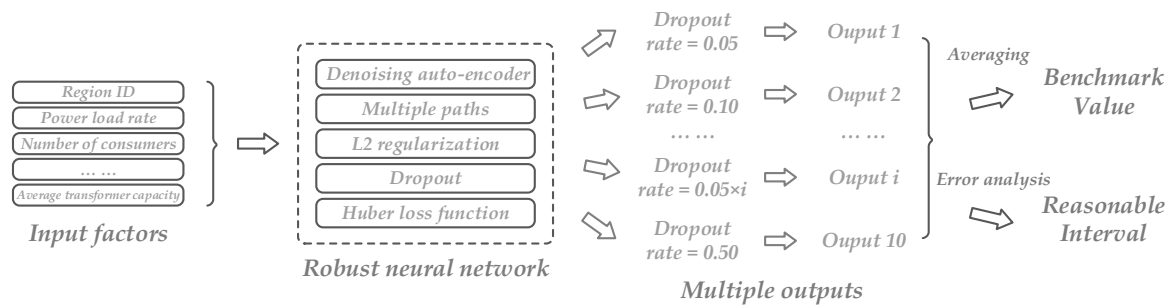
$$e_1 = \sqrt{\frac{1}{n_s} \sum_{i=1}^{n_s} (\tilde{y}_i - y_i^*)^2}, \tag{8}$$

$$e_2 = \sqrt{\frac{1}{10-1} \sum_{i=1}^{10} (\tilde{y}_i - y_i^n)^2}, \tag{9}$$

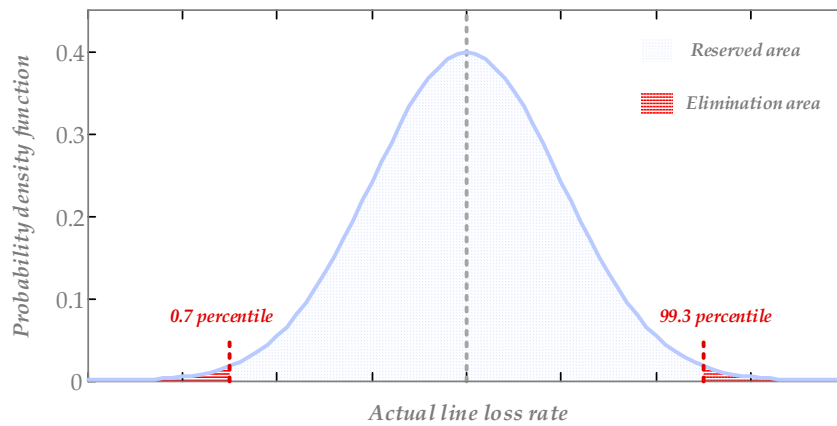
$$\begin{cases} l_i = \tilde{y}_i - e_1 - e_2 \\ u_i = \tilde{y}_i + e_1 + e_2 \end{cases}, \tag{10}$$

where  $e_1$  and  $e_2$  are the results of error analysis.  $e_1$  is a constant and  $e_2$  changes according to the number  $i$ .  $n_s$  is the number of training samples;  $y_i^*$  is the  $i$ th actual line loss rate;  $l_i$  and  $u_i$  are the lower and upper bounds of the reasonable interval, respectively. Furthermore, as outliers exist in the actual line loss rate values, which may influence the result of  $e_1$ , a two-tailed test is utilized to

eliminate the possibly abnormal  $y_i^*$  values that are smaller than 0.7 percentile and bigger than 99.3 percentile, as shown in Figure 6:



**Figure 5.** The flowchart of the proposed method in this study for calculating benchmark values and reasonable intervals based on robust neural network (RNN).



**Figure 6.** A two-tailed test to eliminate possibly abnormal line loss rate values.

#### 2.4. Robust Neural Network

As mentioned before, an RNN is used in this study for robust learning, whose architecture is provided in Figure 7. It is made up of three main paths which are combined by concatenation, where there is a DAE on each main path. The concatenated output nodes are placed in a same layer, which represent high-order features extracted from original inputs. For the further improvement of robustness, L2 regularization is utilized in this layer to limit the output values of those nodes. Then, ten dropout layers with different dropout rate are stacked after the high-order feature layer, where ten outputs can be obtained. The ten outputs are analyzed to calculate the benchmark value and reasonable interval, as introduced above in Section 2.3.



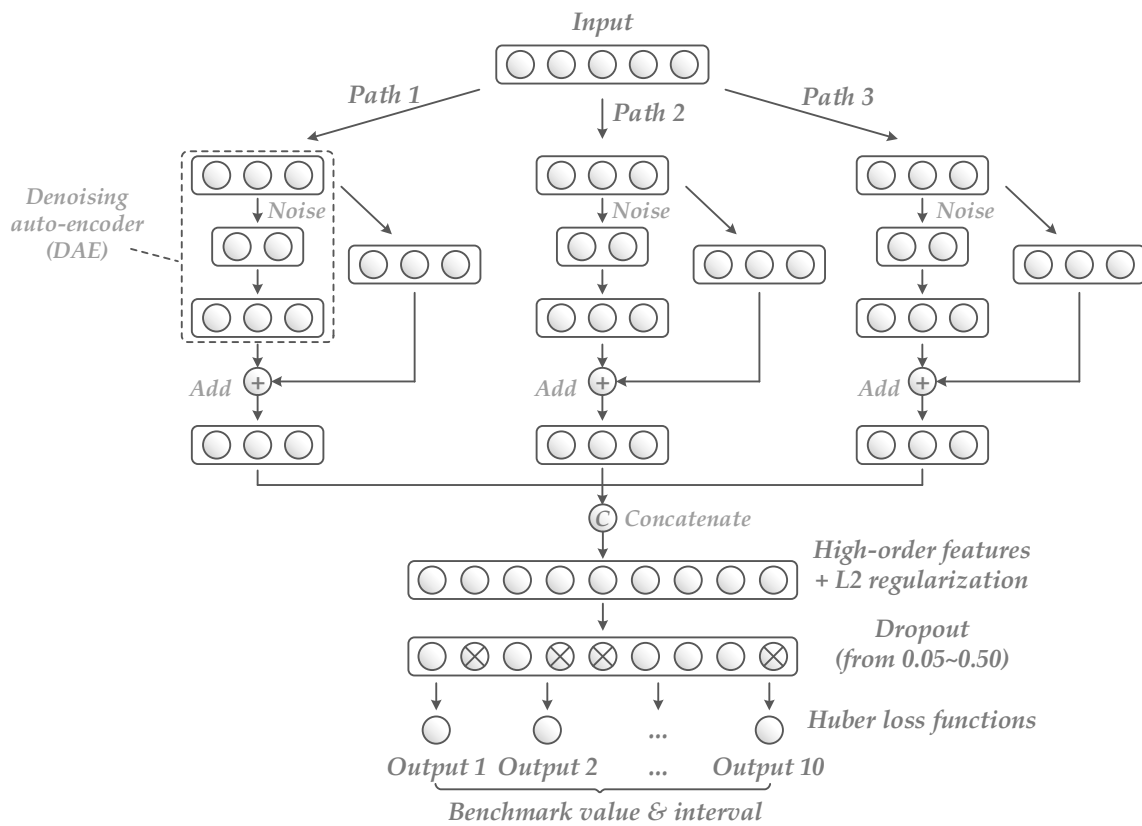


Figure 7. The architecture of the proposed robust neural network (RNN).

### 2.4.1. Denoising Auto-Encoder

The architecture of DAE is provided in Figure 8. It is a robust variant of auto-encoder, which possesses a noise layer before encoder [32], such as a normal (Gaussian) noise layer:

$$x_{i,n} = x_i + N(0, \sigma^2), \tag{11}$$

where  $x_i$  and  $x_{i,n}$  are the  $i$ th input and the  $i$ th output of the noise layer, respectively.  $N(0, \sigma^2)$  is a normal distribution with a mean value of 0 and a variance value of  $\sigma^2$ . In this study,  $\sigma$  is set to be 0.05 when the inputs are normalized into [0, 1].

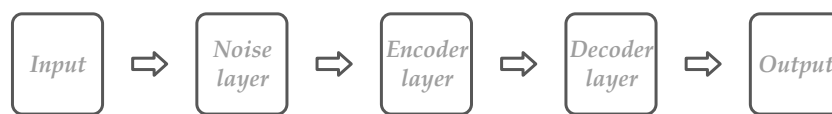


Figure 8. The architecture of denoising auto-encoder (DAE).

Besides, the encoder and decoder layers in DAE are both made up of conventional fully-connected (FC) layers, whose equation can be expressed as:

$$y_i^l = w_{ij}x_j^{l-1} + b_j, \tag{12}$$

where  $y_i^l$  and  $x_j^{l-1}$  are the  $i$ th output and the  $j$ th input of the  $l$ th layer, respectively;  $w_{ij}$  and  $b_j$  are the weight and bias of the FC layer that connect the  $j$ th input and the  $i$ th output. For the encoder layer, the number of outputs is smaller than that of inputs, i.e.,  $i < j$ ; and the number of outputs in the decoder layer is equal to that of the original inputs. Hence, after the computation of DAE, the dimension and size of inputs remain unchanged, whereas the robustness of features increases as they can resist a certain degree of noise interference.

### 2.4.2. Multiple Paths Combined by Addition and Concatenation

There are altogether three main paths in RNN, which have similar layers and whose outputs are combined under a concatenation operation:

$$\begin{cases} \mathbf{y}_c = C[f_k(\mathbf{x}, \{\mathbf{w}_k^{n_w}, \mathbf{b}_k^{n_b}\})] \\ \mathbf{y}_{k,mp} = f_k(\mathbf{x}, \{\mathbf{w}_k^{n_w}, \mathbf{b}_k^{n_b}\}) \end{cases}, k = 1, 2, 3, \quad (13)$$

where  $C(\cdot)$  is the concatenation operation, which combines output nodes from different layers into an entire layer;  $\mathbf{y}_c$  and  $\mathbf{y}_{k,mp}$  are the output vector of the concatenation and the output vector of the  $k$ th main path, respectively;  $f_k(\mathbf{x}, \{\mathbf{w}_k^{n_w}, \mathbf{b}_k^{n_b}\})$  is the computation result of the  $k$ th main path;  $\mathbf{x}$  denotes the input vector of RNN;  $\mathbf{w}_k^{n_w}$  and  $\mathbf{b}_k^{n_b}$  are the weight matrixes and bias vectors of the  $k$ th path, respectively;  $n_w$  and  $n_b$  are the numbers of weights and biases, respectively.

Furthermore, a main path is formed by two sub-paths, i.e., a DAE sub-path and a FC layer sub-path. The outputs of the two sub-paths are added as the output of a main path, as follows:

$$\begin{aligned} \mathbf{y}_{k,mp} &= f_k(\mathbf{x}, \{\mathbf{w}_k^{n_w}, \mathbf{b}_k^{n_b}\}) \\ &= g_k(\mathbf{x}, \{\mathbf{w}_k^{n_w-1}, \mathbf{b}_k^{n_b-1}\}) + (\mathbf{w}_k^{sp} \mathbf{x} + \mathbf{b}_k^{sp}) \end{aligned} \quad (14)$$

where  $g_k(\cdot)$  represents the computation of DAE on the  $k$ th main path;  $\mathbf{w}_k^{sp}$  and  $\mathbf{b}_k^{sp}$  are the weight matrix and bias vector of the FC layer sub-path on the  $k$ th main path, respectively.

### 2.4.3. Dropout

Dropout is a special kind of layer, which is efficient to prevent over-fitting [33]. The procedure of dropout can be summarized into two steps, i.e., the training step and application step. For a conventional FC layer as expressed in Equation (12), there are  $j$  input nodes. In the training step, the input nodes will be abandoned under a probability  $p$  ( $0 < p < 1$ ), where those abandoned nodes cannot be connected to outputs [34], as shown in Figure 9. After training, all the input nodes are operating in the application step, whereas the weight values are multiplied by the probability  $p$ , which can be described as:

$$y_i^l = p \cdot w_{ij} x_j^{l-1} + b_j, \quad (15)$$

where  $p$  is the probability, called dropout rate. It is a hyper-parameter and is set from 0.05 to 0.50, with a step of 0.05, in order to obtain ten different outputs in this study.

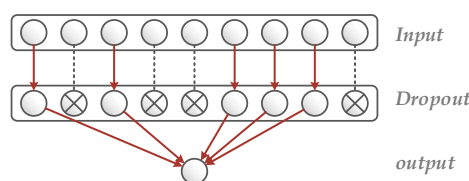


Figure 9. The principle of dropout during the training step.

### 2.4.4. Huber Loss Function

The training process of neural network is to set a loss function and utilize back-propagation (BP) gradient descent algorithm to update the parameters layer by layer. One of the most common-used function is mean squared error (MSE):

$$MSE = \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i - y_i^*)^2, \quad (16)$$

where  $n_s$  is the number of training samples;  $y_i$  and  $y_i^*$  are the  $i$ th predicted and the  $i$ th actual outputs, respectively. Besides, mean absolute error (MAE) is another conventional function, which can be defined as follows:

$$\text{MAE} = \frac{1}{n_s} \sum_{i=1}^{n_s} |y_i - y_i^*|, \tag{17}$$

where MSE and MAE can be also called L1 loss and L2 loss, as linear term and quadratic term are used in the MSE and MAE, respectively.

Comparing MSE and MAE, MSE holds a smoother derivative function, which can benefit the calculation of gradient descent algorithm, whereas a small difference in MAE may cause a huge change in parameter updating. On the contrary, MAE demonstrates a better capability than MSE when fighting against outliers [35]. In this case, a Huber loss function is applied in this study that combines the merits of both MSE and MAE [36], as shown in Figure 10:

$$\text{Huber} = \frac{1}{n_s} \sum_{i=1}^{n_s} \begin{cases} \frac{1}{2}(y_i - y_i^*)^2, & |y_i - y_i^*| \leq \delta \\ \delta|y_i - y_i^*| - \frac{1}{2}\delta^2, & |y_i - y_i^*| > \delta \end{cases}, \tag{18}$$

where  $\delta$  is a hyper-parameter that needs to be set manually. It is set to be 10% in this study.

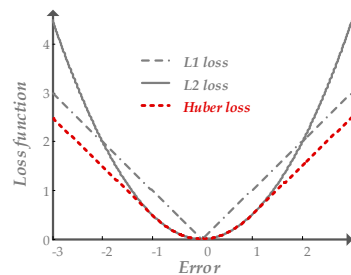


Figure 10. The principle of Huber loss function.

### 2.4.5. L2 Regularization

L2 regularization in this study is aimed to set a penalty term for the nodes that hold large activation outputs, in order to prevent over-fitting and increase robustness of the neural network. The regularization works during the training phase, where a two-norm penalty term is added to the training loss function [37], which can be expressed as follows:

$$\begin{aligned} L &= \lambda \|\mathbf{y}_c\|_2 + \text{Huber} \\ &= \lambda \|\mathbf{y}_c\|_2 + \frac{1}{n_s} \sum_{i=1}^{n_s} \begin{cases} \frac{1}{2}(y_i - y_i^*)^2, & |y_i - y_i^*| \leq \delta \\ \delta|y_i - y_i^*| - \frac{1}{2}\delta^2, & |y_i - y_i^*| > \delta \end{cases} \end{aligned} \tag{19}$$

where  $L$  represents the final loss function for model training; and  $\lambda$  is a hyper-parameter of the penalty term, which is set to be 0.001 in this study.

## 3. Results and Discussion

The architecture and hyper-parameters of the proposed RNN are presented in Table 3. Moreover, considering the fact that there are a large number of training samples,  $k$ -nearest neighbors (KNN), decision tree regression (DTR) and single-hidden-layer artificial neural network (ANN) are established for comparisons, which hold relatively high training efficiency on big datasets. The training of the deep RNN model is conducted based on a personal computer with NVIDIA GTX 1080 GPU using Python 3.5 and Tensorflow 1.4. The calculation results and discussions are provided in detail as follows. It should be noted that all the hyper-parameters and training configurations of RNN, along with the hyper-parameters mentioned in Section 2.4 (i.e.,  $\sigma$ ,  $\delta$  and  $\lambda$ ), are chosen via grid search with three-fold

cross-validation based on the overall training dataset. The searching spaces and final results of those parameters are listed in Table 4.

**Table 3.** The architecture and hyper-parameters of the proposed robust neural network (RNN).

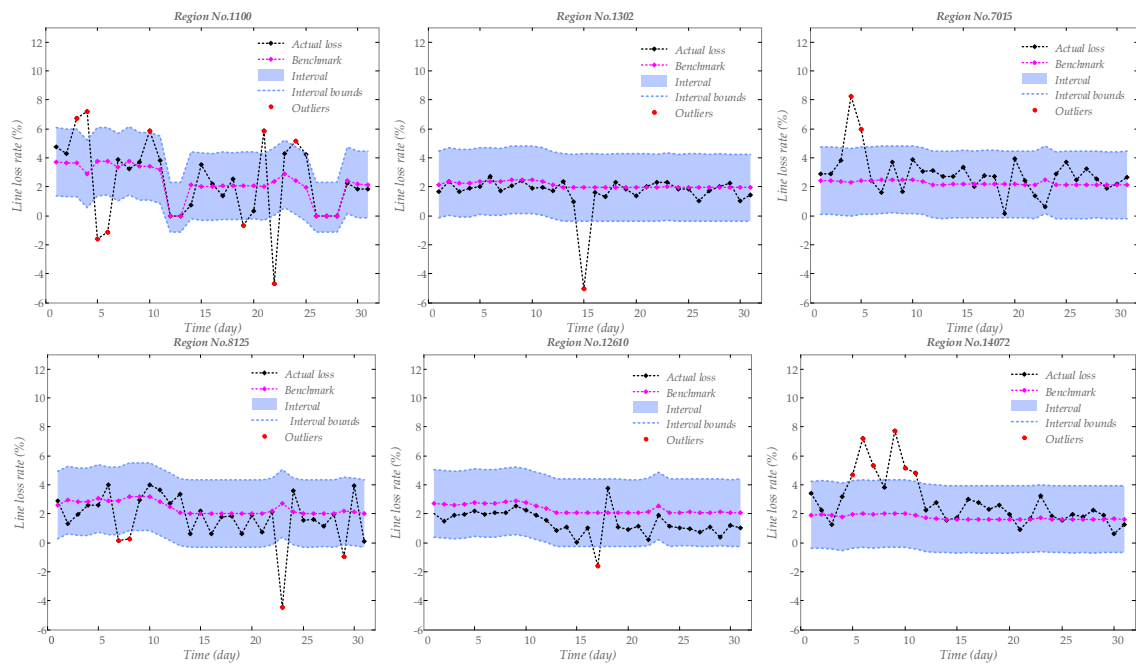
Path	Layer	Hyper-Parameter
DAE sub-path (1~3)	FC layer 0	Node number: 64 Activation: ReLU (Rectified Linear Unit)
DAE sub-path (1~3)	Noise layer	Standard deviation: 0.05
DAE sub-path (1~3)	FC layer 1 (encoder)	Node number: 8 Activation: ReLU
DAE sub-path (1~3)	FC layer 2 (decoder)	Node number: 64
Fully connected (FC) sub-path (1~3)	FC layer 3	Node number: 64
Main path (1~3)	Add layer	Inputs: FC layer 2 and FC layer 3
-	Concatenate layer	Inputs: Add layers Node number: 192 Activation: ReLU Activity regularization: L2, $\lambda = 1 \times 10^{-3}$
-	Dropout layer (1~10)	Dropout rate: 0.05~0.50 (0.05 step)
-	FC layer 4 (output 1~10)	Node number: 1 Activation: ReLU

**Table 4.** The searching spaces of selected hyper-parameters in RNN.

Hyper-Parameters	Searching Space	Result
$\sigma$ (standard error of noise in DAE)	[0.00:0.05:0.50]	0.05
$\delta$ (coefficient in Huber loss)	[0.00:0.10:0.50]	0.10
$\lambda$ (penalty coefficient of L2)	$1 \times 10^{[-4:1:0]}$	$1 \times 10^{-3}$
Number of nodes in a sub-path	{16, 32, 64, 128}	64
Number of sub-paths	[2:1:5]	3
Training optimizer	{Adadelta, Adam}	Adam
Number of epochs	[50:50:200]	100
Batch size	{64, 128, 256, 512, 1024, 2048}	1024
Learning rate	$1 \times 10^{[-5:1:0]}$	$1 \times 10^{-4}$

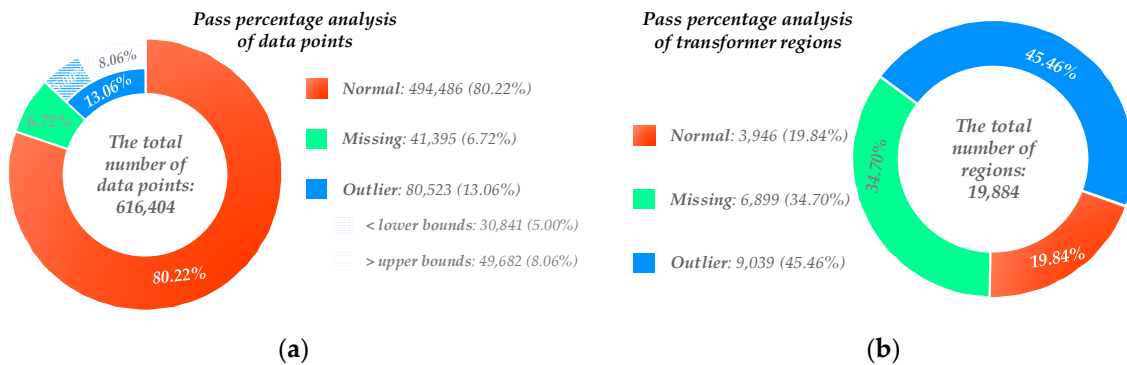
### 3.1. Calculation Results

In the case study, six regions from testing samples are chosen randomly as examples for exhibition, which are shown in Figure 11. Their region IDs are 1100, 1302, 7015, 8125, 12,610 and 14,072, respectively. From the results, the bounds of reasonable intervals can adjust adaptively according to multiple input factors, especially in Region No. 1100 and Region No. 8125. Some outliers that are far away from benchmarks are efficiently picked out, although those outliers may stay between -1% and 5%. Hence, the results of reasonable intervals are better than a fixed interval between -1% and 5%. Besides, the benchmark values show less fluctuation the actual line loss rate values, indicating a high reliability to estimate daily line loss rates. Rather than a mean value or a median value calculated from the original dataset, benchmark values are able to adaptively reflect the daily operating conditions in transformer regions according to the changes of relevant factors.



**Figure 11.** The results of benchmark values and reasonable intervals in six testing regions.

Based on the proposed RNN, the pass percentage results of line loss rates can be analyzed, which are shown in Figure 12. For the data point analysis of line loss rates, the number of outliers are more than that in Figure 3, as the proposed method is able to precisely identify those outliers that are greatly different from the benchmark values. Furthermore, although the percentages of missing values and outliers are not relatively big in all data points, which are 6.72% and 13.06%, respectively, the regions that possess no missing and abnormal values within a month only occupy 19.84% of the whole dataset in the study, indicating a low reliability of the current acquisition equipment.



**Figure 12.** Pass percentage analysis of the line loss rates based on robust neural network (RNN). (a) Analysis of data points; (b) analysis of transformer regions.

### 3.2. Comparisons and Discussion

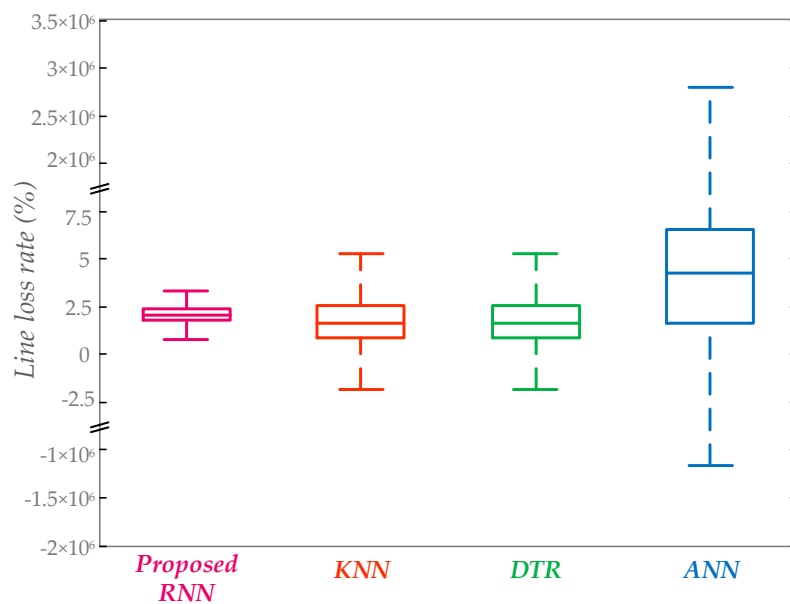
In order to evaluate the performance of the proposed method, including robustness and accuracy, comparison studies are conducted in this section. The hyper-parameters of the established KNN, DTR and ANN are provided in Table 5. The comparison results are presented and discussed in detail as follows.

**Table 5.** The hyper-parameters of the established *k*-nearest neighbors (KNN), decision tree regression (DTR) and artificial neural network (ANN).

Model	Hyper-Parameter
KNN	Number of neighbors: 10
DTR	Method of weighting: based on distance
ANN	Method of splitter: choose the best spilt
	Node number in the hidden layer: 64
	Activation: logistic

### 3.2.1. The Robustness of the Proposed Method

In order to evaluate the robustness of the proposed method, the distributions of the calculated benchmark values based on different testing models are analyzed, as shown in Figure 13. The detailed values of the distribution indicators are presented in Table 6. From the results, the testing ANN model shows the worst performance and is totally unable to calculate a valid benchmark value. The maximum and minimum values from the ANN are  $4.49 \times 10^6\%$  and  $-8.26 \times 10^7\%$ , respectively, which can hardly be applied as benchmarks. KNN and DTR obtain similar results according to the distributions. They both utilize massive training samples that are close to the situation of an unknown testing region to decide new benchmark values. Thus, they are able to receive a better robustness than ANN in the study and are practicable at most of the testing regions. However, the minimum benchmark of the two models is  $-8.13 \times 10^4\%$ , which is still not a reasonable value. Furthermore, the proposed RNN achieve the best result among all the four testing models, where the calculated benchmark values are within a reasonable range. The standard deviation of the benchmark values calculated by RNN is only 0.80%, indicating a stable and robust result obtained via the proposed method.



**Figure 13.** The distributions of the calculated benchmark values based on different testing models.

**Table 6.** The robustness analysis results of different testing models.

Indicator	Proposed RNN	KNN	DTR	ANN
mean (%)	2.03	1.03	−4.81	$-5.83 \times 10^6$
std (%)	0.80	185.69	194.78	$1.87 \times 10^7$
min (%)	0.00	$-8.13 \times 10^4$	$-8.13 \times 10^4$	$-8.26 \times 10^7$
max (%)	33.61	100.00	100.00	$4.49 \times 10^6$
$l_a$ (%)	0.81	−1.80	−1.80	$-1.17 \times 10^6$
$u_a$ (%)	3.36	5.25	5.24	$2.80 \times 10^6$
$q1$ (%)	1.76	0.84	0.84	$3.23 \times 10^5$
$q2$ (%)	2.09	1.65	1.65	$8.47 \times 10^5$
$q3$ (%)	2.40	2.60	2.60	$1.32 \times 10^6$

### 3.2.2. Accuracy Analysis

Besides the performance of robustness, accuracy is another important criterion in the study. As a result, three loss indicators are utilized to compare the four testing models, i.e., MAE, MSE and Huber loss, as their equations are discussed in Section 2.4.4. The outliers in the testing samples are eliminated before loss calculation using the two-tailed test, which is introduced above in Figure 6. The comparison result is shown in Table 7. According to the result, ANN performs the worst as its three loss indicators are much more than those of other models. Hence, the testing ANN is inapplicable when directly trained on samples with extreme outliers. Besides, although KNN and DTR show similar robustness, their accuracy indicators are quite different. KNN obtains the best MAE indicator, whereas the MSE value of KNN is bigger than that of the proposed RNN, due to the small number of outliers calculated by KNN. Comparing those indicators comprehensively, the proposed RNN shows the highest performance, as it achieves the best MSE and Huber loss indicators with a small MAE value as well.

**Table 7.** The accuracy analysis results of different testing models.

Indicator	Proposed RNN	KNN	DTR	ANN
Mean absolute error (MAE, %)	1.46	0.70	5.01	$7.26 \times 10^6$
Mean squared error (MSE, % <sup>2</sup> )	13.70	574.58	$3.90 \times 10^3$	$3.54 \times 10^{14}$
Huber loss	4.44	5.90	48.19	$7.26 \times 10^7$

## 4. Conclusions

Daily line loss rate is a critical indicator for power-supply corporations as it greatly affects their profits. In order to better manage the level of line losses and provide guidance for the construction and operation of low-voltage transformer regions, it is concerned with developing an efficient method to compute benchmark values for daily line loss rates. Based on the benchmarks, reasonable intervals of the daily line loss rates can be further obtained, which are able to help discover abnormal line loss rate values, as well to help operators to check and confirm those irregular operating conditions. However, the number of studies is limited that research on calculating benchmark values of daily line loss rates and eliminating outlier from collected line loss rate data. As a result, a regression calculation method based on RNN is proposed in this study. It consists of DAE, three main paths, dropout layers, Huber loss function, L2 regularization and ten outputs. The benchmarks are calculated according to the mean values of the ten outputs. After error analysis, reasonable intervals can be obtained to detect outliers of original line loss rate samples.

From the case study and comparison results, conventional ANN model fails to calculate results of benchmarks, as it cannot deal with outliers. KNN, DTR and the proposed RNN are proved applicable in the case study, where the proposed RNN outperforms the other two models. It shows both high accuracy and robustness among all the testing models. Furthermore, according to the final result obtained from the proposed RNN, there are about 13% outliers in the overall data points.

The percentage of regions only occupy about 20% that hold no missing and abnormal values of line loss rates within a month, indicating a low dependability of the acquisition equipment. Therefore, a reliable monitoring and management system for line loss data is still necessary nowadays in power grid.

**Author Contributions:** Writing original draft preparation, W.W. and L.C.; reviewing and editing, Y.Z., B.X. and H.Z.; supervision, G.X. and X.L.

**Funding:** This research is supported by Science and Technology Project of Jiangsu Frontier Electric Technology Corporation, called “Research on key techniques of line loss benchmark value modeling in low-voltage transformer area by data-driven model” (project ID. 0FW-18553-WF).

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

## References

- Chen, B.J.; Xiang, K.L.; Yang, L.; Su, Q.M.; Huang, D.S.; Huang, T. Theoretical Line Loss Calculation of Distribution Network Based on the Integrated electricity and line loss management system. In Proceedings of the China International Conference on Electricity, Tianjing, China, 17–19 September 2018; pp. 2531–2535.
- Yang, F.; Liu, J.; Lu, B.B. Design and Application of Integrated Distribution Network Line Loss Analysis System. In Proceedings of the 2016 China International Conference on Electricity Distribution (CICED), Xi’an, China, 10–13 August 2016.
- Hu, J.H.; Fu, X.F.; Liao, T.M.; Chen, X.; Ji, K.H.; Sheng, H.; Zhao, W.B. Low Voltage Distribution Network Line Loss Calculation Based on The Theory of Three-phase Unbalanced Load. In Proceedings of the 3rd International Conference on Intelligent Energy and Power Systems (IEPS 2017), Hangzhou, China, 10 October 2017; pp. 65–71. [\[CrossRef\]](#)
- Campos, G.O.; Zimek, A.; Sander, J.; Campello, R.J.G.B.; Micenkova, B.; Schubert, E.; Assent, I.; Houle, M.E. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. Knowl. Disc.* **2016**, *30*, 891–927. [\[CrossRef\]](#)
- Paulheim, H.; Meusel, R. A decomposition of the outlier detection problem into a set of supervised learning problems. *Mach. Learn.* **2015**, *100*, 509–531. [\[CrossRef\]](#)
- Daneshpazhouh, A.; Sami, A. Entropy-based outlier detection using semi-supervised approach with few positive examples. *Pattern Recogn. Lett.* **2014**, *49*, 77–84. [\[CrossRef\]](#)
- Bhattacharya, G.; Ghosh, K.; Chowdhury, A.S. Outlier detection using neighborhood rank difference. *Pattern Recogn. Lett.* **2015**, *60–61*, 24–31. [\[CrossRef\]](#)
- Domingues, R.; Filippone, M.; Michiardi, P.; Zouaoui, J. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recogn.* **2018**, *74*, 406–421. [\[CrossRef\]](#)
- Dovoedo, Y.H.; Chakraborti, S. Boxplot-Based Outlier Detection for the Location-Scale Family. *Commun. Stat.-Simul. Comput.* **2015**, *44*, 1492–1513. [\[CrossRef\]](#)
- Pranatha, M.D.A.; Sudarma, M.; Pramaita, N.; Widyantara, I.M.O. Filtering Outlier Data Using Box Whisker Plot Method For Fuzzy Time Series Rainfall Forecasting. In Proceedings of the 2018 4th International Conference on Wireless and Telematics (ICWT), Nusa Dua, Indonesia, 12–13 July 2018.
- Campello, R.J.G.B.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 5. [\[CrossRef\]](#)
- Huang, J.L.; Zhu, Q.S.; Yang, L.J.; Cheng, D.D.; Wu, Q.W. A novel outlier cluster detection algorithm without top-n parameter. *Knowl.-Based Syst.* **2017**, *121*, 32–40. [\[CrossRef\]](#)
- Jiang, F.; Liu, G.Z.; Du, J.W.; Sui, Y.F. Initialization of K-modes clustering using outlier detection techniques. *Inf. Sci.* **2016**, *332*, 167–183. [\[CrossRef\]](#)
- Todeschini, R.; Ballabio, D.; Consonni, V.; Sahigara, F.; Filzmoser, P. Locally centred Mahalanobis distance: A new distance measure with salient features towards outlier detection. *Anal. Chim. Acta* **2013**, *787*, 1–9. [\[CrossRef\]](#)
- Jobe, J.M.; Pokojovy, M. A Cluster-Based Outlier Detection Scheme for Multivariate Data. *J. Am. Stat. Assoc.* **2015**, *110*, 1543–1551. [\[CrossRef\]](#)
- An, W.J.; Liang, M.G.; Liu, H. An improved one-class support vector machine classifier for outlier detection. *Proc. Inst. Mech. Eng. C J. Mech.* **2015**, *229*, 580–588. [\[CrossRef\]](#)



17. Chen, G.J.; Zhang, X.Y.; Wang, Z.J.; Li, F.L. Robust support vector data description for outlier detection with noise or uncertain data. *Knowl.-Based Syst.* **2015**, *90*, 129–137. [[CrossRef](#)]
18. Zou, C.L.; Tseng, S.T.; Wang, Z.J. Outlier detection in general profiles using penalized regression method. *IIE Trans.* **2014**, *46*, 106–117. [[CrossRef](#)]
19. Peng, J.T.; Peng, S.L.; Hu, Y. Partial least squares and random sample consensus in outlier detection. *Anal. Chim. Acta* **2012**, *719*, 24–29. [[CrossRef](#)]
20. Ni, L.; Yao, L.; Wang, Z.; Zhang, J.; Yuan, J.; Zhou, Y. A Review of Line Loss Analysis of the Low-Voltage Distribution System. In Proceedings of the 2019 IEEE 3rd International Conference on Circuits, Systems and Devices (ICCS), Chengdu, China, 23–25 August 2019; pp. 111–114.
21. Yuan, X.; Tao, Y. Calculation method of distribution network limit line loss rate based on fuzzy clustering. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *354*, 012029. [[CrossRef](#)]
22. Bo, X.; Liming, W.; Yong, Z.; Shubo, L.; Xinran, L.; Jinran, W.; Ling, L.; Guoqiang, S. Research of Typical Line Loss Rate in Transformer District Based on Data-Driven Method. In Proceedings of the 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), Chengdu, China, 21–24 May 2019; pp. 786–791.
23. Yao, M.T.; Zhu, Y.; Li, J.J.; Wei, H.; He, P.H. Research on Predicting Line Loss Rate in Low Voltage Distribution Network Based on Gradient Boosting Decision Tree. *Energies* **2019**, *12*, 2522. [[CrossRef](#)]
24. Zhang, S.; Dong, X.; Xing, Y.; Wang, Y. Analysis of Influencing Factors of Transmission Line Loss Based on GBDT Algorithm. In Proceedings of the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 5–7 July 2019; pp. 179–182.
25. Wang, S.X.; Dong, P.F.; Tian, Y.J. A Novel Method of Statistical Line Loss Estimation for Distribution Feeders Based on Feeder Cluster and Modified XGBoost. *Energies* **2017**, *10*, 2067. [[CrossRef](#)]
26. Radovanovic, M.; Nanopoulos, A.; Ivanovic, M. Reverse Nearest Neighbors in Unsupervised Distance-Based Outlier Detection. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 1369–1382. [[CrossRef](#)]
27. Yosipof, A.; Senderowitz, H. k-Nearest Neighbors Optimization-Based Outlier Removal. *J. Comput. Chem.* **2015**, *36*, 493–506. [[CrossRef](#)]
28. Wang, X.C.; Wang, X.L.; Ma, Y.Q.; Wilkes, D.M. A fast MST-inspired kNN-based outlier detection method. *Inf. Syst.* **2015**, *48*, 89–112. [[CrossRef](#)]
29. Yang, J.H.; Deng, T.Q.; Sui, R. An Adaptive Weighted One-Class SVM for Robust Outlier Detection. *Lect. Notes Electr. Eng.* **2016**, *359*, 475–484. [[CrossRef](#)]
30. Feng, N.; Jianming, Y. Low-Voltage Distribution Network Theoretical Line Loss Calculation System Based on Dynamic Unbalance in Three Phases. In Proceedings of the 2010 International Conference on Electrical and Control Engineering, Wuhan, China, 25–27 June 2010; pp. 5313–5316.
31. Hubert, M.; Vandervieren, E. An adjusted boxplot for skewed distributions. *Comput. Stat. Data Anal.* **2008**, *52*, 5186–5201. [[CrossRef](#)]
32. Lamb, A.; Binas, J.; Goyal, A.; Serdyuk, D.; Subramanian, S.; Mitliagkas, I.; Bengio, Y. Fortified Networks: Improving the Robustness of Deep Networks by Modeling the Manifold of Hidden Representations. *arXiv* **2018**, arXiv:1804.02485.
33. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
34. Cheng, L.L.; Zang, H.X.; Ding, T.; Sun, R.; Wang, M.M.; Wei, Z.N.; Sun, G.Q. Ensemble Recurrent Neural Network Based Probabilistic Wind Speed Forecasting Approach. *Energies* **2018**, *11*, 1958. [[CrossRef](#)]
35. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
36. Esmaeili, A.; Marvasti, F. A Novel Approach to Quantized Matrix Completion Using Huber Loss Measure. *IEEE Signal Proc. Lett.* **2019**, *26*, 337–341. [[CrossRef](#)]
37. Shah, P.; Khankhoje, U.K.; Moghaddam, M. Inverse Scattering Using a Joint L1-L2 Norm-Based Regularization. *IEEE Trans. Antennas Propag.* **2016**, *64*, 1373–1384. [[CrossRef](#)]

