


Article

# AdaBoost Algorithm Could Lead to Weak Results for Data with Certain Characteristics

Olivér Hornyák <sup>1</sup>  and László Barna Iantovics <sup>2,\*</sup> <sup>1</sup> Institute of Information Engineering, University of Miskolc, 3515 Miskolc, Hungary<sup>2</sup> Department of Electrical Engineering and Information Technology, George Emil Palade University of Medicine, Pharmacy, Science and Technology of Targu Mures, 540142 Targu Mures, Romania

\* Correspondence: barna.iantovics@umfst.ro

**Abstract:** There are many state-of-the-art algorithms presented in the literature that perform very well on some evaluation data but are not studied with the data properties on which they are applied; therefore, they could have low performance on data with other characteristics. In this paper, the results of comprehensive research regarding the prediction with the frequently applied AdaBoost algorithm on real-world sensor data are presented. The chosen dataset has some specific characteristics, and it contains error and failure data of several machines and their components. The research aims to investigate whether the AdaBoost algorithm has the capability of predicting failures, thus providing the necessary information for monitoring and condition-based maintenance (CBM). The dataset is analyzed, and the principal characteristics are presented. Performance evaluations of the AdaBoost algorithm that we present show a prediction capability below expectations for this algorithm. The specificity of this study is that it indicates the limitation of the AdaBoost algorithm, which could perform very well on some data, but not so well on others. Based on this research and some others that we performed, and actual research from worldwide studies, we must outline that the mathematical analysis of the data is especially important to develop or adapt algorithms to be very efficient.

**Keywords:** mathematical modeling; statistical analysis; statistical correlation; statistical significance; statistical analysis of experimental evaluation data; AdaBoost algorithm; smart applications; sensor data; failure data; condition-based maintenance; performance analysis of an algorithm

**MSC:** 68W40

**Citation:** Hornyák, O.; Iantovics, L.B. AdaBoost Algorithm Could Lead to Weak Results for Data with Certain Characteristics. *Mathematics* **2023**, *11*, 1801. <https://doi.org/10.3390/math11081801>

Academic Editor: Xiang Li

Received: 12 March 2023

Revised: 4 April 2023

Accepted: 7 April 2023

Published: 10 April 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Modern buildings integrate increasingly smarter applications; thus, a large amount of real-time sensor data can be collected and transmitted by the Internet of Things devices [1]. Oliveira et al. [2] presented evaluation procedures for performing forecasting based on complex spatiotemporal data. Leon and Gavrilescu [3] presented a survey on the problem of tracking and trajectory prediction methods that should be applied in autonomous driving. Haq et al. [4] studied an adapted consumption prediction model that can be applied in commercial and residential sectors.

Typical applications aim to conserve energy consumption: heating, lighting, air conditioning, ventilation, etc. Other applications belong to routine facility management such as electricity, water supply, sanitary, etc. There is immense potential to apply artificial intelligence and machine learning during the life cycle of buildings [5].

Measuring machine intelligence makes possible the comparison of the systems and applications based on their intelligence. Because the diversity of the systems is large, it is difficult to elaborate on universal intelligence metrics. The paper [6] presented the mathematical modeling of a universal black-box-based intelligence metric called MetrIntPairII, which is able to measure and compare a set of systems based on their intelligence, finally

classifying the systems into intelligence classes. In [7], a novel universal method called ExtrIntDetect for the identification of intelligent systems with extremely low and extremely high intelligence was proposed. This allowed choosing the system that has statistically much lower or higher intelligence than others to solve problems. MetrIntPairII and ExtrInt-Detect could help intelligent systems developers and at the same time users of intelligent systems in choosing systems based on problem-solving intelligence.

This paper focuses on the monitoring and condition-based maintenance (CBM) of construction facility equipment that is essential for providing flawless operation. CBM, which is defined as “preventive maintenance which includes assessment of physical conditions, analysis and the possible ensuing maintenance actions” [8] plays an important role in lifecycle engineering to improve machines’ availability and reduce maintenance costs. State-of-the-art CBM applications have been improved recently; they use data analytics processes on the data of the system being investigated. Very often, data analytics [9] use Artificial Intelligence techniques that can evaluate whether the machine operations follow a normal condition pattern or whether there is an anomaly [10]. Machine designers do not have experience with machine operational failures; the machines operate in different environments. Thus, the maintenance activities become increasingly significant [11]. A closely related term is prognostics and health management (PHM) which aims for the prediction of its reliability and the remaining useful lifetime (RUL) of machines and machines health [12].

Internet of Things (IoT) technology focuses on obtaining sensor information [13–15], then securely processing and storing it. Cloud-based data centers are widely used. The benefits of Big Data techniques [16] are also utilized in this process. Some research suggests using blockchain technology [17].

The monitoring of electricity and thermal energy systems, ventilation, and cooling/heating systems can be traced back to measuring electrical parameters. The following signals are typically measured: power consumption, and mechanical vibration. For example, increasing power consumption or higher vibration may indicate an upcoming failure. They are usually analyzed using methods such as [18] a fast Fourier transform and power spectrum analysis.

Collecting and monitoring these data can form a base to evaluate the condition of each piece of equipment to explore the characteristic and life curve state of a particular piece of equipment. The remaining useful life is the length of usable time left on an asset at a specific time [19]. The definition of what is ‘usable’ for the owner of the equipment may be specific to the asset.

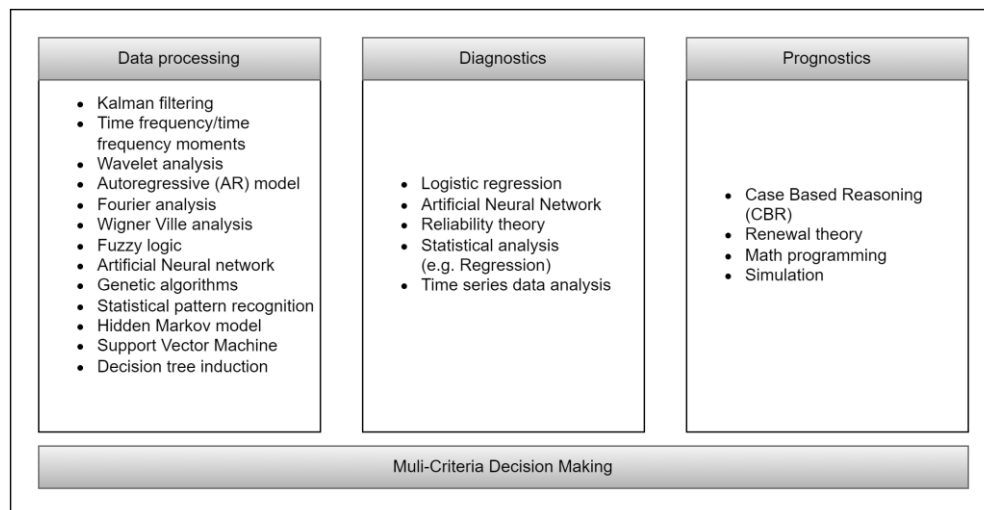
The residual life is constantly decreasing during the operation of equipment. Properly scheduled maintenance and repairs can extend the lifetime. The advantage of condition-based maintenance is that the maintenance events of the equipment need to happen only when its status characteristics justify this. The investigation of the dynamics of physical signal characteristics and performance measurements may indicate the failure of parts of the equipment so that the maintenance time and cost can be minimized with the help of the accurate prediction of forthcoming failures. Another advantage is that purchasing and storing those parts can be more efficiently scheduled.

CBM may be based on offline and online measurements. In the offline case, the measurement and data acquisition are performed periodically, followed by recurring data processing. This is a cost-effective way for monitoring equipment whose lifetime degradation is known and nearly constant.

The other type is as follows: online, real-time measurement and monitoring is feasible to use in cases where the lifetime degradation is unknown or very uncertain. Furthermore, this strategy is valuable when the loss of the equipment is great, there is a potential personal injury, or the environmental damage is greater than the cost of real-time process monitoring.

Figure 1 shows the most typical CBM techniques presented by [20]. There are three major families of techniques: data processing, diagnostics, and prognostics. Data processing focuses on handling and analyzing data or signals and transforming them into an

interpretable form. Diagnostics are responsible for fault detection, when and what kind of fault happened, and which components were affected. Prognostics attempt to predict upcoming faults.



**Figure 1.** Most common CBM techniques presented in [20].

Another comprehensive review of the available CBM models was given in [21]. The paper treats two distinct strategies: time-based maintenance and condition-based maintenance. CBM techniques are divided into prognostic methods, and machine learning approaches are also discussed such as artificial neural networks, fuzzy logic, and expert systems. Common data-driven methods include the principal component analysis, learning vector quantization, and hidden Markov models.

The structure of the rest of the paper is as follows: Section 2 describes the research work, and then a review is given of the scientific literature. Section 3 describes the characteristics of the dataset. Then, the AdaBoost algorithm is introduced to predict failures using the given dataset. Finally, in Section 4 the conclusions are formulated.

## 2. Materials and Methods

In this paper, a data-driven method is studied. The research hypothesis is that by investigating a database that contains records of machine errors and failures along with further telemetry data of those machines, this will allow the machine errors to be predicted with some expected probability. The research project focused on investigating a boosting algorithm, which will be discussed later. In our methodology, the dataset was divided into two parts. Within the available time range, the time of the investigation split the dataset into historical and future data. By using the historical data, predictions were made for the ‘future’, i.e., to the time after that point. The predictions were then compared with the real facts. The research workflow is depicted in Figure 2.

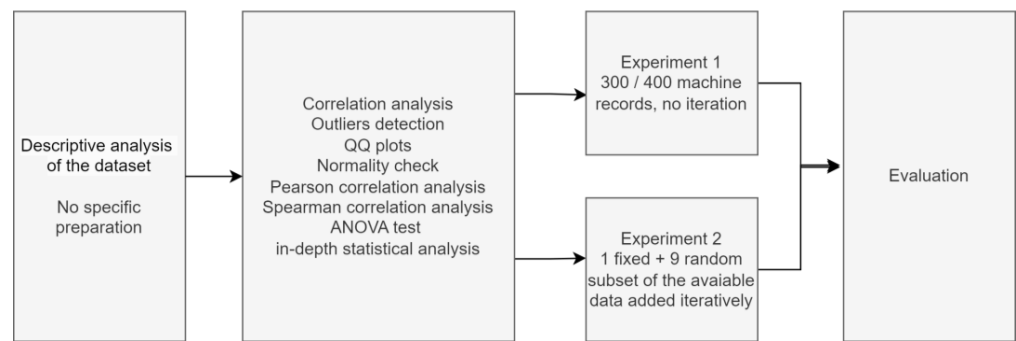


Figure 2. The workflow followed in the research.

### 2.1. Estimations of Machine Failures

We considered a constraint when an estimation should be given on the expected time of a machine failure. The following sections give an overview of the estimation methods that have been investigated in this paper.

#### 2.1.1. Estimation by Mean Operation Time

Let us assume that the data of some similar machines operating in a similar environment are available for investigation. It can be noticed that the machine failure will be close to the mean life (unless the usage of a given machine is different from the other machines). Let us have  $n$  machines. The start time of the machine is denoted by  $s_i$ . Let us denote the  $f_i$  as the failure time of the  $i$ th machine. The operation time of the  $i$ th machine can be calculated as:

$$t_i = f_i - s_i \tag{1}$$

The mean operation time is:

$$\bar{t} = \frac{\sum_{i=1}^n t_i}{n} \tag{2}$$

Let us assume the  $k$ th machine starts at  $s_k$ . The average time of the expected failure time can be calculated by (3):

$$s_k + \bar{t} \tag{3}$$

The precision of the estimation can be described by the distance between the expected and real failure time. The absolute value of the distance or the square of the distance is an appropriate indicator. It is expected that the more machines we have, the more precise the calculation of the expected value. The precision of the estimation can be improved incrementally. Let us assume there is an appropriate number of data samples to calculate the average operation time. When a new failure is noted, the value of the mean operation time can be updated by (2).

#### 2.1.2. Estimation Based on the Relative Frequency

When the estimation targets a fixed time interval, we can count the occurrences of the failures of the previous periods, and the relative frequency can be considered as the probability of occurrence. This is a calculation that is not computationally expensive and provides an approximate value for future occurrences without considering the distribution of the failures. An advantage of the method is that adding new samples to the dataset will make the estimation more precise. The disadvantage is that the entire time interval must be considered. Properly selected aggregation methods can eliminate the linear growth of computing time. This method cannot 'forget', i.e., all the data samples are taken into account. We may want the model not to consider failure data too old, or 'refresh' memory after a repair of the machines. Applying a sliding time window can improve this method. Another characteristic of the model is that certain time windows contain no failure data at all; therefore, the predicted probability of failure is zero.

### 2.1.3. Estimation by the Expected Value

Since time is a continuous value, the likelihood is 0 that the failure will occur exactly at the estimated time. For this reason, it is expedient to estimate for a period. Let us utilize Markov's inequality [22], which looks as follows:

$$P(\xi > \delta \cdot E(\xi)) \leq \frac{1}{\delta} \tag{4}$$

where  $\xi$  is a probability variable and  $\delta \in R$ . In our case,  $\xi$  refers to the lifetime of the equipment. The expected value  $E(\xi)$  can be predicted by the aforementioned  $\bar{t}$  value. In the case of a homogeneous sampling, when  $n \rightarrow \infty$  the value of  $\bar{t}$  will approach the expected value. If there is an estimation for the expected value, any arbitrary  $\delta > 0$  value can be used to estimate whether a specific machine would fail within a specified period. Let us denote this period  $h \in R$ . For the estimation, let us use (5):

$$1 - P(\xi \leq \delta \cdot E(\xi)) \leq \frac{1}{\delta} \tag{5}$$

from which

$$1 - P(\xi \leq h) \leq \frac{1}{\delta} \tag{6}$$

as

$$\delta = \frac{h}{E(\xi)} \tag{7}$$

Rearranging the equation results:

$$P(\xi \leq h) \leq 1 - \frac{E(\xi)}{h} \approx 1 - \frac{\bar{t}}{h} \tag{8}$$

Note that  $h \geq E(\xi)$  must be true.

It is important to note that this estimation does not take into account how long the machine has been working. The estimated probability can be calculated when the machine is started. From a practical point of view, it is more advantageous to give an estimate for a specified interval such as one day, one week, or one month forward. A threshold for the estimated probability of the failure occurring in the interval can be calculated based on a metric. Such a metric can be the proportion of correct and incorrect failures. In order to calculate this, a known period must be investigated and the expected and obtained results should be compared. Frequencies and relative frequencies can be used. The advantage of using the frequency is that it contains the size of the samples as meta information. The use of relative frequencies makes the result easier to compare with other calculations. To evaluate the method by investigating the number of failures, we can use:

- The absolute value of the difference between the observed and calculated results;
- The square of the difference between the observed and calculated results;
- And p-norm as a generic case. Let  $\Delta$  be the vector of differences, and  $p \in R, p \geq 1$ . p-norm [23] is defined as:

$$\|\Delta\|_p = \left( \sum_{i=1}^n |\Delta_i|^p \right)^{\frac{1}{p}} \tag{9}$$

where  $n$  is the number of discrete times considered for the estimation.

If the question is only whether the machine operates with or without failure during the period, then the problem can be considered as a classification problem. The simplest case is binary classification: to have one class for the failures and one for no failure. If the failure count is an important factor, then additional classes can be introduced to represent the number of failures.

### 2.1.4. Estimation by the Expected Value

With Markov inequality [22], we can only provide a very rough probability of failures. Let us have a  $\zeta$  probability value, whose expected value is  $E$  and its standard deviation is  $D$ . For an arbitrary  $\varepsilon \in R, \varepsilon > 0$ , the following is true:

$$P(|\zeta - E(\zeta)| \geq \varepsilon) \leq \frac{D^2(\zeta)}{\varepsilon^2} \tag{10}$$

Therefore, the inequality (10) shows the probability of the values that are likely to fall into the  $\varepsilon$  radius of the expected value. In practice, this can be used as follows:

- Based on the available samples, give an estimation for the expected average value and the standard deviation;
- Specify an  $\varepsilon$  value that will select a time interval of  $2\varepsilon$  length on the time axis, with the expected value in the center;
- Give an estimation of the probability of an item being inside/outside of the interval;
- Often, the result can be interpreted as giving the probability of the failure to fall into the examined interval. With the rearrangement of the equation, this can be calculated as:

$$P(|\zeta - E(\zeta)| < \varepsilon) \geq 1 - \frac{D^2(\zeta)}{\varepsilon^2} \tag{11}$$

if

$$1 - \frac{D^2(\zeta)}{\varepsilon^2} > 0 \Rightarrow 1 > \frac{D^2(\zeta)}{\varepsilon^2} \Rightarrow \varepsilon^2 > D^2(\zeta) \Rightarrow \varepsilon > D(\zeta) \tag{12}$$

It can also be noticed that the increase in the value of  $\varepsilon$ , the size of the period, and the probability of a failure to fall into the period will increase.

It is possible to use inequality in another form. If the probability  $p$  is known and the size of the period is the question, then the inequality can be rearranged as follows:

$$p = 1 - \frac{D^2(\zeta)}{\varepsilon^2} \tag{13}$$

To express  $\varepsilon$ :

$$\varepsilon = \sqrt{\frac{D^2(\zeta)}{1 - p}} \tag{14}$$

The higher the probability  $p$ , the higher  $\varepsilon$  will be. Note that the equalities refer to known theoretically expected values and standard deviations. In most cases, however, they are not available for the equipment.

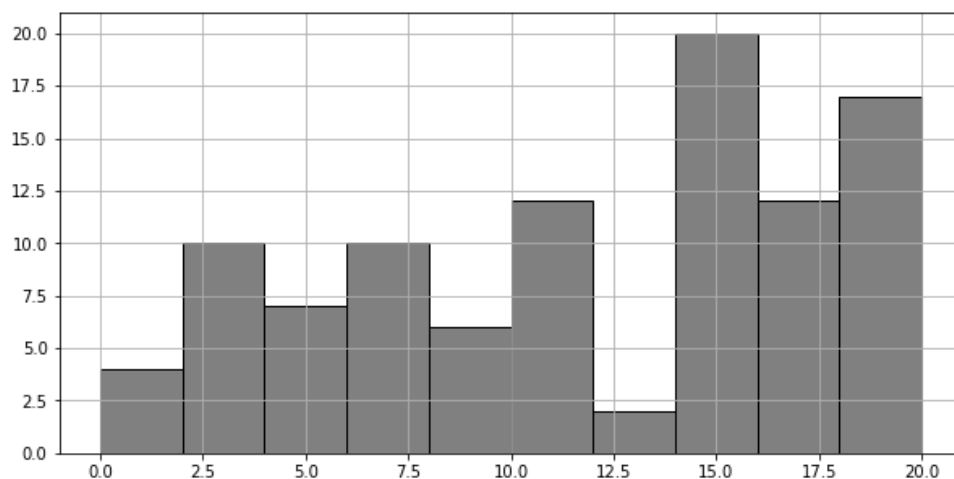
### 2.2. Analysis of the Dataset

In this section, the dataset used is described. The target of the investigations made in this paper was acquired with a real dataset (see data availability statement at the end of the paper), which contained data from 100 machines, see Table 1. A brief overview of the information gained about the machine’s lifetime is as follows:

- The lifetime of the machines is between 0 and 20 years;
- No typical distribution is identified for the age of the machines;
- The average lifetime of the machines is 11.33 years;
- The age of the machines is approximately 5.8 years;
- No 13-year-old machines are in the set—all other ages in the range of [0, 20] can be seen; see Figure 3;
- The most frequent is the 14-year-old machines with the count 14. The second most common age is 10 years, with 10 instances.

**Table 1.** Machine model and age data.

Machine ID	Model	Age
1	model <sub>3</sub>	18
2	model <sub>4</sub>	7
3	model <sub>3</sub>	8
	...	
98	model <sub>2</sub>	20
99	model <sub>1</sub>	14
100	model <sub>4</sub>	5



**Figure 3.** The histogram of machine ages in the dataset.

Some remarks on model types:

- The distribution of the number of similar machines is not uniform. The individual model counts are 16, 17, 35, and 32;
- The most common machine of type #3 is 14 years old having 6 instances;
- The average age of type #4 is outstanding: the average ages of other types are 12.25, 12.76, and 12.03, respectively, while type #4 is about 9.34 years old on average;
- The highest standard deviation regarding ages is for type #4;
- The maximum lifetime for each type is 20 years. The minimum lifetime is between 0 and 2 years.

Figure 3 depicts the histogram of the machine ages in the dataset.

### 2.3. Performed Statistical Analysis

In this section, we perform a specific statistical analysis.

#### Correlation Analysis

Initially, some quantitative indicators were determined. In the sample dataset, the *nr\_failures* (count of failures), *nr\_errors* (count of error statements), and *nr\_maintenance* (count of maintenance activities) were identified for the analysis.

For the data analysis, we performed a verification of the outliers first, followed by the verification of the data normality.

An outlier that could appear in a dataset is an extreme that is statistically significantly much higher or lower than the other values from the same dataset. The appearance of outliers in datasets can influence to a high degree the evaluation results. For instance, if the mean must be calculated, a very high or low outlier influences the result to a high degree. We have verified applying the two-tailed Grubbs outliers detection test [24] at the significance level  $\alpha_{GR} = 0.05$  for identifying outliers, in all the datasets, *nr\_errors*, *nr\_failures*, and *nr\_maintenance*. The Grubbs test applied to the number of errors did not

detect any outlier, just the value 60 that was statistically furthest from the rest. The Grubbs test applied to the number of failures did not detect any outlier, just the 19 value that was statistically furthest from the rest. The Grubbs test applied to the number of maintenances did not detect any outlier, just the 25 value that was statistically furthest from the rest. In the following analyses, we have considered even these furthest from the rest of the values.

There are diverse goodness-of-fit tests that can be applied for the verification of data normality. Among the most frequently used goodness-of-fit normality tests can be mentioned the Kolmogorov–Smirnov (KS) test [25,26]; Lilliefors (Lill) test [27], which is a Kolmogorov–Smirnov test with a specific Lilliefors correction; Anderson Darling test [28] and Shapiro–Wilk (SW) test [29]. According to [30], from the previously mentioned tests, the SW test has the highest power, but it has disadvantages as well. The SW test is recommended to be applied for small samples (where the sample size  $\leq 30$ ). In the case of a large sample size, we recommend the application of the Lill test. As the significance level of the Lill normality test, we considered in most of the cases  $\alpha_{norm} = 0.05$ . The  $p$ -value of the normality test can be interpreted if  $p\text{-value} < \alpha_{norm}$ , then the Null hypothesis (H0) should be rejected, and H1 accepted as the alternative hypothesis; the data failed to pass the normality assumption at the  $\alpha_{norm}$  significance level. Elsewhere, if  $p\text{-value} \geq \alpha_{norm}$ , H0 could not be rejected, and the data passed the normality assumption at the  $\alpha_{norm}$  significance level.

For additional visual validation of the result of the Lill normality test, we recommend the use of the quantile-quantile plot (Q-Q plot) visual representation [31]. The Q-Q plot is a scatterplot appropriate for normality visual appreciation. If the data are normally distributed data, the points should fall approximately along this reference line. The larger the departure from the reference line, the greater the evidence for the conclusion that the data failed the normality assumption.

We considered that the data sample normality was an influencing factor that must be analyzed for the data characterization. In the following, we have verified the assumption of normality of the variables nr\_errors, nr\_failures, and nr\_maintenance.

For the verification of normality, based on the fact that the sample size was larger by 30, we applied the Kolmogorov–Smirnov test with the Lilliefors improvement (Lill test) [29] at the  $\alpha_{norm} = 0.05$  significance level. Table 2 presents the obtained results by applying the Lill test, with the considered  $\alpha_{norm} = 0.05$  significance level.

**Table 2.** Verification of data normality assumption using the Lill test,  $\alpha_{norm} = 0.05$ .

	nr_Errors	nr_Failures	nr_Maintenance
Test statistics	0.095	0.087	0.120
$p$ -value	0.026	0.058	0.001
$p\text{-value} \geq \alpha_{norm}$	No	Yes	No
Normality assumption passed (No reason for rejection of H0)	No	Yes	No

For additional visual validation of the numerical normality analysis results, we created the Q-Q plots corresponding to the variables, nr\_errors (Figure 4), nr\_failures (Figure 5), and nr\_maintenance (Figure 6). Based on the obtained numerical results and the visual validations, it can be concluded that the number of failures passed the normality assumption. The number of errors failed to pass the normality assumption. The number of maintenances failed to pass the normality assumption. Figure 7 depicts the relation between error statements, failures, and maintenance count.

For calculating the correlation coefficient of two variables, we recommend the Pearson [7,32] or Spearman [32,33] correlation coefficient as being the most appropriate in many cases. We have defined the decision rule for choosing between the Pearson or Spearman correlation coefficient based on the normality of the variables whose correlation was studied. The Spearman correlation coefficient is more appropriate in the nonparametric case when the data (variable) fail to pass the normality assumption. When the data (variable) normality assumption passes, we recommend the Pearson correlation coefficient [32] to be used.



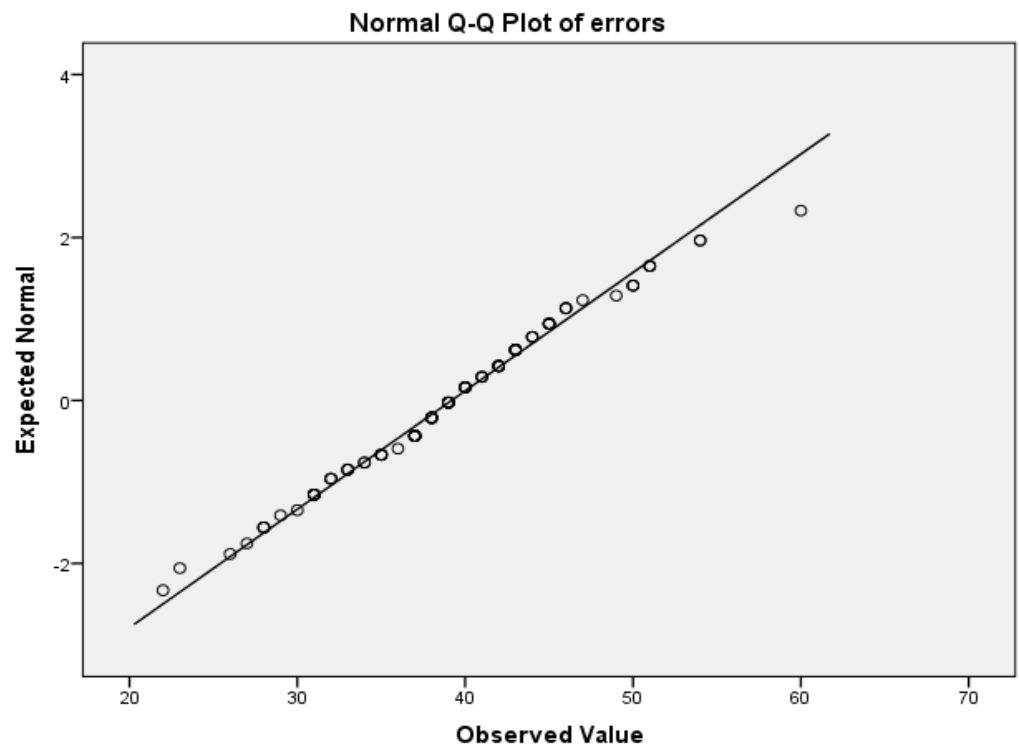


Figure 4. Q-Q plot for the number of errors.

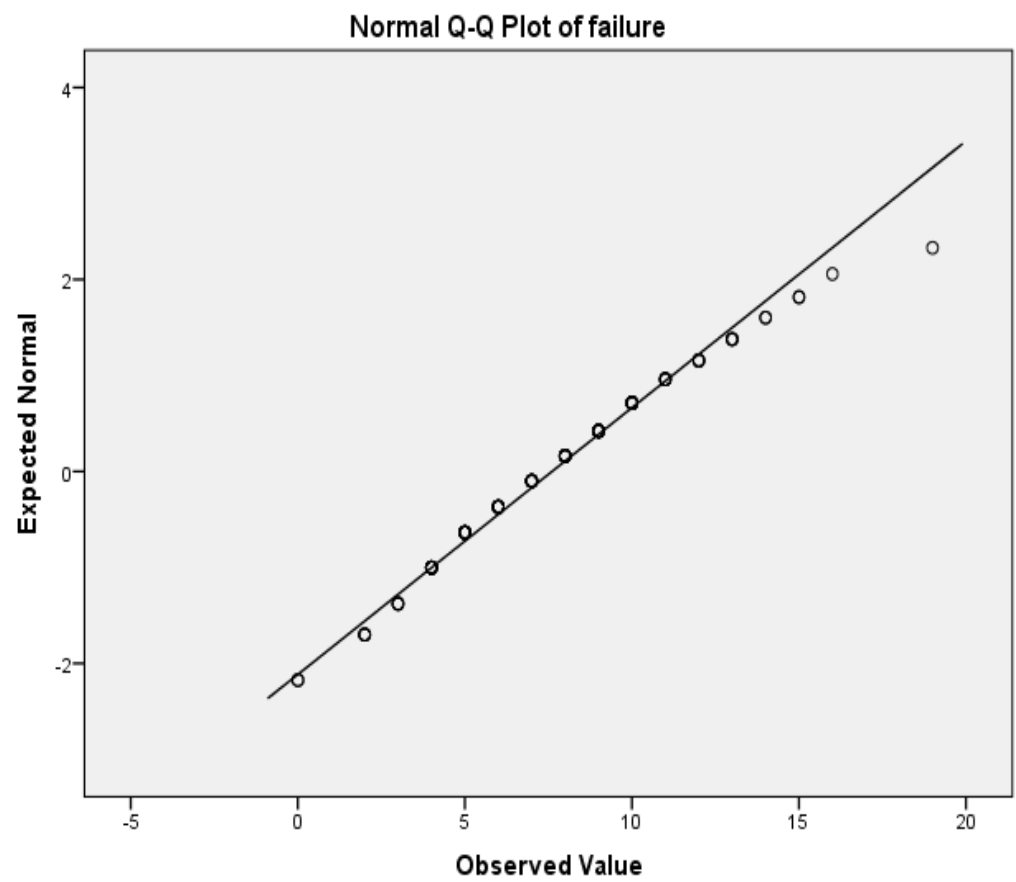


Figure 5. Q-Q plot for the number of failures.

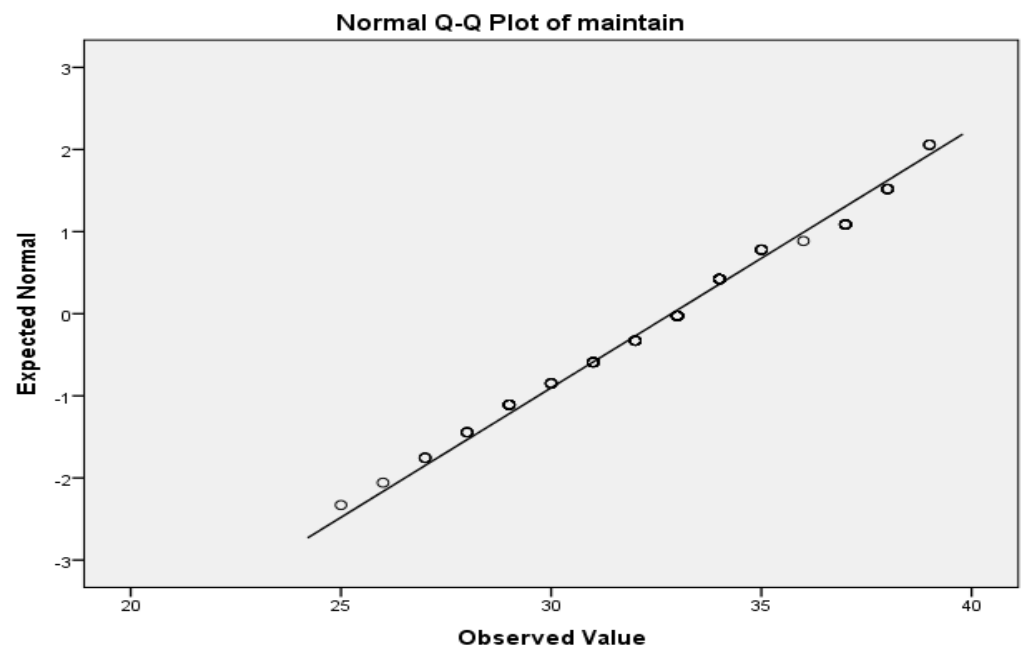


Figure 6. Q-Q plot for the number of maintenances.

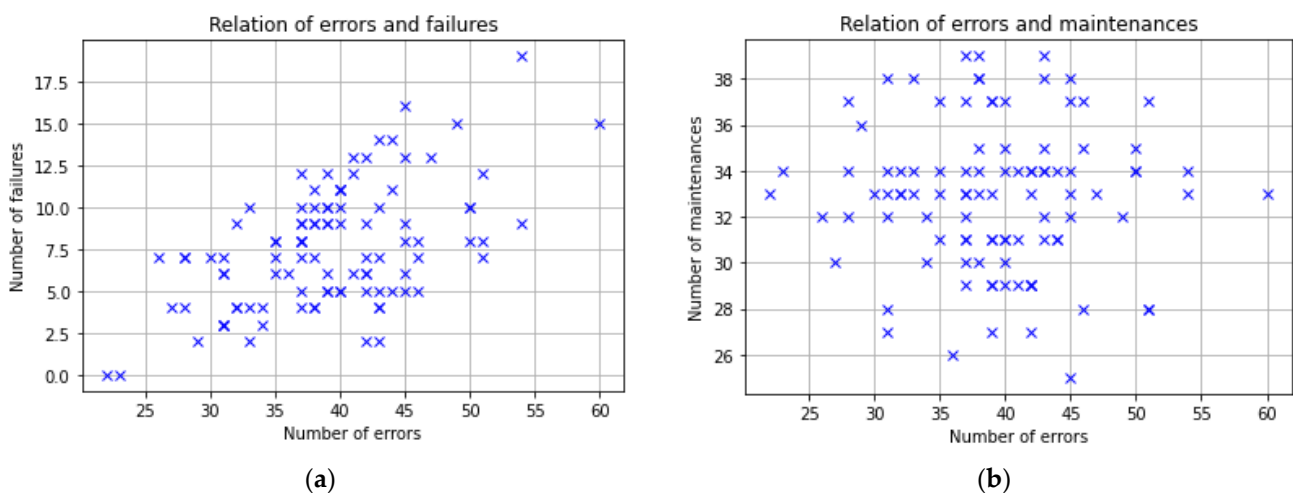


Figure 7. Relation between error statements, failures, and maintenance count. (a) The number of failures; (b) the number of maintenances.

Based on the results of the normality analysis, we decided to use the Spearman correlation coefficient  $r$  [33,34] between the number of errors and the number of failures. The obtained Spearman  $r = 0.448$  numerical value indicated a moderate correlation. For verification, if  $r$  was statistically significant, we have calculated the 95% confidence interval (CI) obtaining [0.2703, 0.5961]. We found that  $r > 0$  and  $0.2703 > 0$ , which proves that the correlation was statistically significant. Additionally, we have applied the statistical ANOVA test at the significance level  $\alpha_{ANOVA} = 0.05$  for verification of the hypothesis that  $r$  was statistically significantly different from 0. The obtained  $p$ -value of 0.0001 indicated that the difference was statistically significant ( $0.0001 < \alpha_{ANOVA}$ ).

We calculated the Spearman correlation coefficient  $r$  between the number of failures and the number of maintenances obtaining a very small  $r = -0.04585$ . For verification, if  $r$  was statistically significant, we have calculated the 95%CI obtaining  $[-0.2457, 0.1577]$ . Therefore,  $r < 0$  and  $0.1577 > 0$  proved that there was no correlation. Additionally, we have applied the statistical ANOVA test at the  $\alpha_{ANOVA} = 0.05$  significance level for verification

of the hypothesis that  $r$  was statistically significantly different from 0. The obtained  $p$ -value of 0.6506 indicated that the difference was not statistically significant ( $0.6506 > \alpha_{ANOVA}$ ).

We calculated the Spearman correlation coefficient  $r$  between the number of errors and the number of maintenances, obtaining a very small  $r = -3.028 \times 10^{-6}$ . For verification, if  $r$  was statistically significant, we have calculated the 95% CI obtaining  $[-0.2021, 0.2021]$ .  $r < 0$  and  $0.2021 > 0$  proved that there was no correlation. Additionally, we have applied the statistical ANOVA test at the 0.005 significance level for verification of the hypothesis that  $r$  was statistically significantly different from 0. The obtained  $p$ -value of 0.9999 indicated that the difference was not statistically significant ( $0.9999 > \alpha_{ANOVA}$ ).

In conclusion, the correlation analysis was realized using the Spearman correlation coefficient on the number of error statements and the number of maintenance events. The correlation analysis has provided the following results:

- There was a moderate correlation between error statements and failures, with a value of 0.448. This was only  $-3.028 \times 10^{-6}$  between error statements and maintenance, while in the case of failures and maintenance, this was  $-0.04585$ ;
- Our initial expectation was that the more failures there were, the more maintenance there was. In fact, due to regular /predictive maintenance, the correlation between the two types of events was not significant.
- There was no relationship between error statements and maintenance.

#### 2.4. Discussion on Error Statements by Machines

It can be assumed that if there are many similar types of machines in the supervised system, the phenomena observed on a machine could happen with similar machines as well. If we can provide a good estimation of the distribution of the error statements per machine, the error statement itself can be predicted with greater reliability.

For a fixed period, the number of error statements can be counted. From the distribution of these (see Figure 8), the following conclusions can be drawn:

- The expected value can be estimated by the average values in the period;
- The standard deviation indicates how similar the machines are.

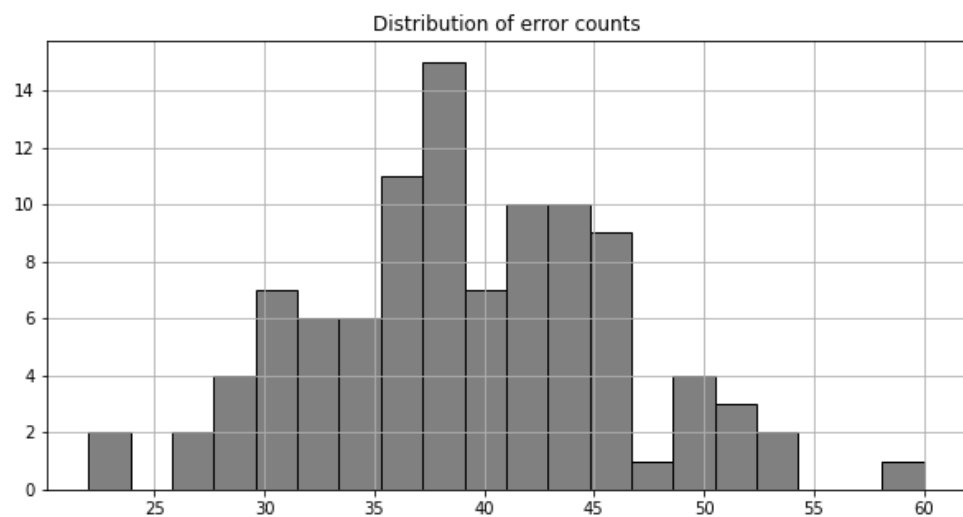
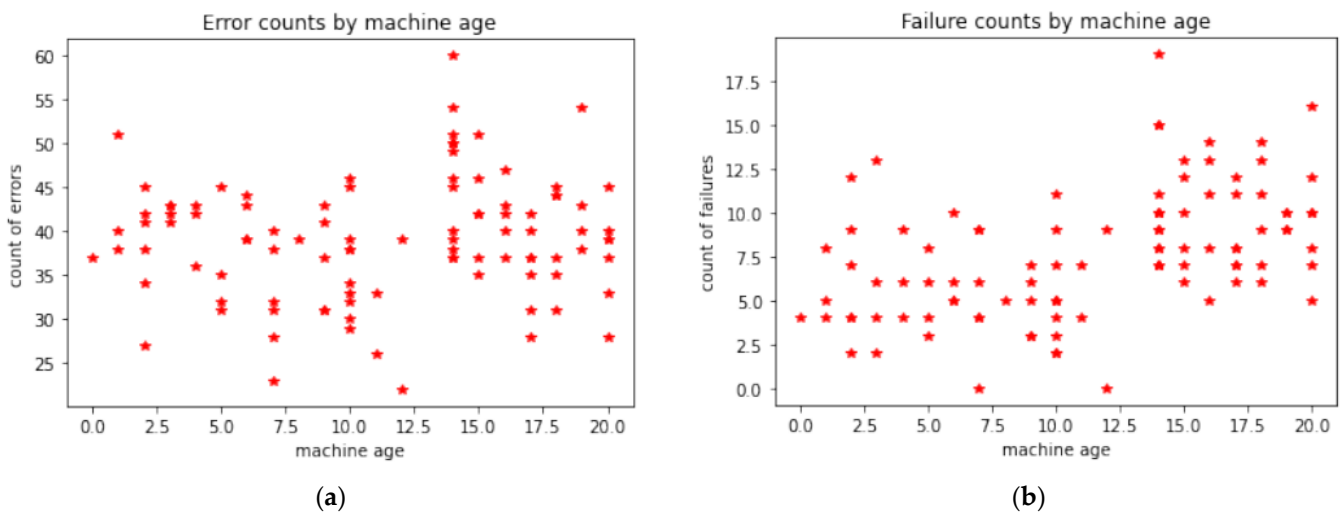


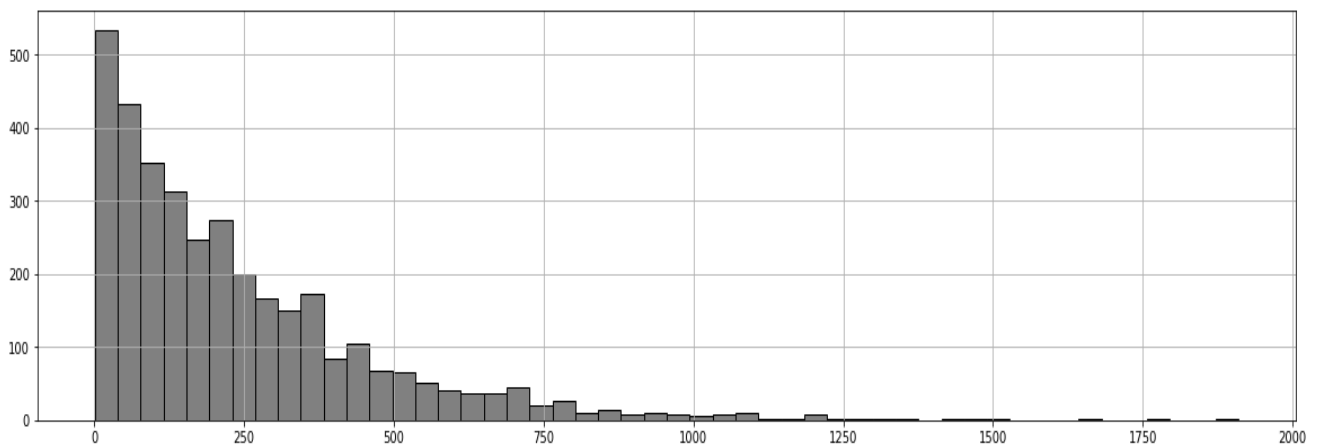
Figure 8. Distribution of error counts.

In the following, we have studied how the age of machines affected the count of error statements. This can be seen in Figure 9; the visual analysis indicates no correlation. This can be interpreted as the maintenance work performed on the machines reducing the significance of the machines' ages.



**Figure 9.** Machine age analysis. (a) Error counts by the machine’s age; (b) failures by the machine’s age.

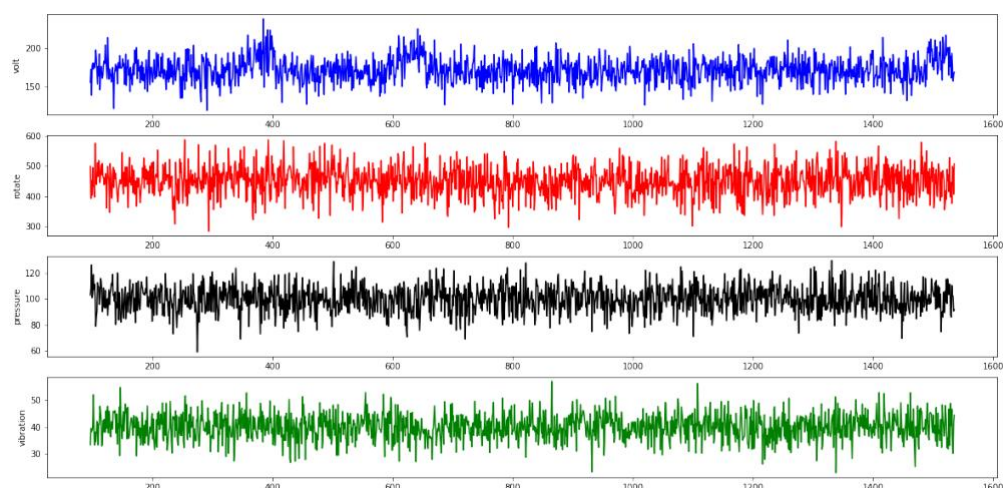
The histogram of elapsed time between errors can be seen in Figure 10. To be visually more illustrative, we have considered classes.



**Figure 10.** Histogram of elapsed time between errors.

The dataset contained telemetry data as well. According to our assumption, telemetry data may have a relationship between the signals and the failures/error statements. It is assumed that telemetry data may be used to forecast future errors of the machine, and thus it may indicate the need for (condition-based) maintenance. For example, vibration and resonance may cause degradation of the lifetime of a rotating component, thus leading to mechanical failures. This research aimed to verify if recognizing patterns in telemetry data can be used to predict forthcoming failures, and thus form a base of CBM.

As discussed before, the dataset investigated consisted of a telemetry dataset of various machines. For example, the telemetry data of machine 1, sample 0 are plotted in Figure 11 as follows:



**Figure 11.** Telemetry data for vibration and pressure.

### 3. The Study of the AdaBoost Algorithm for Prediction

In this section, the details of the study are presented. In the previous sections, the goal of the research and the available dataset were discussed. First, the AdaBoost algorithm will be shown. After that, an overview will be given of the existing application areas of the algorithm. Finally, two experiments will be evaluated on the dataset.

#### 3.1. A Summary of the AdaBoost Algorithm

The AdaBoost algorithm [35,36] is appropriate for accelerating machine learning algorithms and increasing their performance by making a strong classification as a linear combination of weak classifications with appropriate weights. The AdaBoost algorithm belongs to the class of boosting algorithms [37]. The quality of the resulting classification is influenced to a high degree by the definition of initial weak classifiers. The studies [38,39] proved that combining weak learners may form a strong learner. Some enhancements of the original algorithm have been evaluated when running the method. The authors of [39] proposed a weight allocation scheme to enhance the generalization effect. By selecting and combining information in the dataset, different prediction models can be created.

The AdaBoost algorithm is a special case of combined classification methods. Combined classifiers are created by mixing multiple classes of classifiers and methods for making forecasts. This way, they increase the accuracy of the classification. During the learning phase, base classifiers are created. The classification itself relies on voting. The algorithm in its general form is described by the following pseudo-code: Algorithm AdaBoost (Algorithm 1).

---

#### Algorithm 1: Algorithm AdaBoost

---

**Input:** X: domain dataset; Y: label dataset; T: number of steps;  
**Output:**  $H = C$ , the final hypothesis  
 Create  $D_0$  initial distribution  
**for**  $t = 1$  **to** T **do**  
     Compute  $D_{t+1}$  from  $D_t$ ;  
     Construct a  $C_t$  classifier from  $D_t$ ;  
**end for**  
**for** (all  $x_i \in S$  test records) **do**  
      $C(x_i) = \text{Vote } C_1(x_1), C_2(x_2), \dots, C_m(x_m)$ ;  
**end for**  
 End AdaBoost

---

The explanation of the algorithm is as follows. The training dataset consists of suggestions:  $(x_1, y_1) \dots (x_m, y_m)$  where  $x_i$  belongs to the domain space  $X$ , and  $y_i$  is the corresponding label. In our research, we assumed  $Y = \{-1, 1\}$ .  $(x_i, y_i)$  is the experimental classification of phenomenon  $i$ , which can be inaccurate.  $D_t$  is the distribution on round  $t$ , in other words, a weight that describes the importance of the classification of  $x_i$ . In this algorithm, the weight is increased so that it acquires more focus when the classification is incorrect. Initially, the weights are equal (see Equation (15)). At each iteration step, an error is calculated, see Equation (16), and the distribution is updated and normalized (see Equations (17)–(19)). The output is the final hypothesis, which is a result of a voting mechanism. In this paper, a linear combination was used (20), but there can be other voting algorithms as well.

Boosting algorithms [40] are sequential in the sense that the creation of consecutive classification models depends on the data of the previous model. In the first model, we took into account each object of the training dataset with the same weight to see whether the model classified each instance correctly or incorrectly. In the former case, we reduced the weights in the latter to pay more attention to the improperly classified objects in the next round. During the procedure, this step was repeated through several iterations, with the purpose that the algorithm at a certain iteration will be correctly classified as the appropriate class.

There are several variants of the general boosting algorithm depending on how the parameters are used (how to apply weights on instances, what kind of basic classifier to use, and how to combine the models obtained during the process). The most widespread of these is the state-of-the-art AdaBoost algorithm [41–43]. Ref. [44] compared the prediction capabilities of the AdaBoost algorithm with the backpropagation neural network (BPNN), regression classifier, support vector machine (SVM), and support vector regression (SVR).

### 3.2. Applications of the AdaBoost Algorithm

Various application fields of the AdaBoost algorithm can be found in the scientific literature. Ref. [45] used the AdaBoost algorithm in the road engineering field to predict performance indicators of asphalt concrete roads. Leaf nitrogen concentration was estimated by the AdaBoost-based machine learning algorithm in [46]. Vertical total electron content forecasting of the Earth's ionosphere was discussed in [47]. COVID-19 significantly changed the financial performance of enterprises. An AdaBoost-based intelligent driving algorithm for heavy-haul trains was described in [48] to realize the intelligent control of the air brake, optimized based on two aspects: the extraction method of the training sample subset and the voting weight. A neurobiological application can be found in [49], which described a model to improve dementia prediction accuracy; in the paper, an intelligent learning system was described. AdaBoost was used for electroencephalogram epileptic signals investigation [50]. An energetical application was described in [51]: the state-of-charge prediction of lead-acid batteries by AdaBoost. An online sequential extreme learning machine model was proposed in [52] where AdaBoost and the recurrent neural network models were used for lithium batteries' state-of-charge estimation. The architectural engineering application of the AdaBoost algorithm was reported in [53] to predict some mechanical properties of surrounding rock in tunneling. The authors of [54] presented a video-based fire smoke detection using robust AdaBoost. AdaBoost was used to increase the accuracy and reliability of a framework for daily activities and environment recognition using mobile device data [55].

### 3.3. Implemented and Evaluated Version of the AdaBoost Algorithm

#### 3.3.1. AdaBoost Algorithm for Condition-Based Maintenance

In this paper, the AdaBoost algorithm will be applied to a binary classification problem in the field of condition-based maintenance, so this version is reviewed. The algorithm is capable of executing both binary classification and multiclass problems [56]. In this case,

the input dataset consists of  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  pairs, where  $x_i$  is the property vector of the specified entity,  $y_i = \{0, 1\}$  is a label, and  $i = \{1, 2, \dots, N\}$ .

Suppose teaching consists of  $t$  turns. The AdaBoost algorithm calls  $h_t$  a weak classification procedure in each step. Most often, the decision chunk algorithm is used, which divides the teaching patterns along a single attribute value and cuts at a threshold  $d$ . The procedure is similar to a single-level decision tree model, i.e., one leaf of the tree will have an attribute value smaller than  $d$ , while the other leaf is larger than  $d$ .

Let us denote the weight of the  $i$ -th learning dataset of step  $t$   $D_t(i)$ . Initially, all the weights are calculated by (15):

$$D_t(0) = \begin{cases} \frac{1}{2^m} & \text{if } y_t = 0 \\ \frac{1}{2^l} & \text{if } y_t = 1 \end{cases} \tag{15}$$

where  $m$  and  $l$  are the counts of negative and positive examples, respectively. In each iteration, the algorithm re-divides the weight of the training samples by increasing the weight of poorly classified patterns. As a result, poorly performing learning classifiers will receive more attention. The error [57] is evaluated as follows. Let us denote  $X$  as the set of property vectors of the data also known as a weak hypothesis:  $h_i: X \rightarrow \{-1, 1\}$ . In step  $t$ , the error  $\epsilon_t$  of the hypothesis can be calculated as the weighted sum of the wrongly classified entities, in other words, where the predicted  $h_t(x_i)$  is not equal to the training sample  $y_i$ :

$$\epsilon_t = \sum_{i:h_t(x_i) \neq y_i}^N D_t(i) \tag{16}$$

To calculate the distribution in the next step, we need an update parameter. Ref. [40] suggested choosing an  $\alpha$  update parameter as:

$$\alpha = \frac{1}{2} \cdot \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{17}$$

In step  $t + 1$ , the weights are calculated as

$$D_{t+1}(i) = D_t(i) \cdot e^{s_t(i)\alpha_i h_t(x_i)} \tag{18}$$

$s_t(i)$  is a sign which is  $-1$  if the hypothesis  $i$  is correct,  $+1$  otherwise:

$$s_t(i) = \begin{cases} -1 & \text{if } h_t(x_i) = y_i \\ +1 & \text{if } h_t(x_i) \neq y_i \end{cases} \tag{19}$$

As  $D_{t+1}(i)$  is a distribution, we need to apply a normalization in each step using a normalization factor  $Z_t$ .

$$D_{t+1}(i) = \frac{D_{t+1}(i)}{Z_t} \tag{20}$$

The final hypothesis is the linear combination of the individual hypotheses [40]:

$$H(x_i) = \text{sign} \sum_{t=1}^T \alpha_t \cdot h_t(x_i) \tag{21}$$

### 3.3.2. Experimental Evaluation of the AdaBoost Algorithm

To cover a larger diversity of situations, we have performed more representative experiments.

Experiment 1—using no iteration

In the first experiment, the AdaBoost algorithm was executed to predict error statements and failures. There were one hundred machines in the dataset, and four telemetry types for each, so a total of four hundred records were available. The goal was to process

three-quarters of the data to predict the failures in the fourth quarter. The input of the algorithm contained the number of maintenance events in the known quarters. Two types of weak classifiers were used, and the experiments were run ten times as the weight calculation had a random factor. Table 3 presents a screenshot of the obtained experimental evaluation results.

**Table 3.** Evaluation results for experiment 1.

Nr. of Experimental Evaluation	PDS	PSV
1	0.675	0.06083
2	0.675	0.625
3	0.7333	0.6083
4	0.7083	0.6917
5	0.7583	0.6583
6	0.675	0.6167
7	0.675	0.65
8	0.7333	0.633
9	0.675	0.65
10	0.675	1.575

In order to formulate accurate conclusions, we performed an in-depth statistical analysis of the results obtained. For both precision with decision stump (PDS) and precision with support vector classifier (PSV) as a first step, we have verified, using the Grubbs outliers detection test at the  $\alpha_{GR} = 0.05$  significance level, whether outlier values could be detected that were statistically significantly different from those others. By applying the Grubbs test to the PDS, it was identified that the value 0.7583 was the furthest from the rest, but not a significant outlier (the obtained  $p$ -value,  $p$ -value  $> \alpha_{GR}$ ). By applying the Grubbs test, at the  $\alpha_{GR}$  significance level, to PSV, it was identified that a value of 0.6917 was furthest from the rest, but not a significant outlier (the obtained  $p$ -value,  $p$ -value  $> \alpha_{GR}$  0.05).

As a next step, we have made a descriptive statistic for both PSD and PSV. For measuring the variability, we calculated the standard deviation (SD). Furthermore, we calculated the minimum, maximum, range, mean, 95 CI% of the mean, and median. For measuring the data homogeneity-heterogeneity, the coefficient of variation (CV)  $CV = \frac{SD}{mean} \times 100$  was calculated, where  $CV < 10$  indicates a homogeneous dataset;  $CV \in [10, 20)$  indicates a relatively homogeneous dataset;  $CV \in [20, 30)$  indicates a relatively heterogeneous dataset;  $CV \geq 30$  indicates a heterogeneous dataset. Table 4 presents the obtained experimental evaluation results and descriptive statistical characterization.

**Table 4.** Results of the performed descriptive statistics.

Type of Characterization	PDS	PSV
Minimum	0.675	0.575
Maximum	0.7583	0.6917
Range	0.0833	0.1167
Mean	0.70333	0.6317
95 CI% of the mean	[0.68, 0.7263]	[0.6083, 0.655]
SD	0.032189	0.032591
CV	4.58	5.16
Median	(4.58 < 10, homogeneous data)	(5.16 < 10, homogeneous data)
	0.6917	0.6292

Further, we analyzed the data normality of PDS and PSV. Based on the fact that the sample sizes were very small  $10, 10 < 30$  (where 30 can be indicated as a threshold), we have chosen the Shapiro–Wilk (SW) test of normality. The SW test has higher power [29] compared with other very frequently applied statistical tests such as the Kolmogorov–Smirnov, Lilliefors (Kolmogorov–Smirnov test with Lilliefors correction), and Anderson–Darling tests. Table 5 presents the results of the SW test applied with the  $\alpha_{norm} = 0.05$

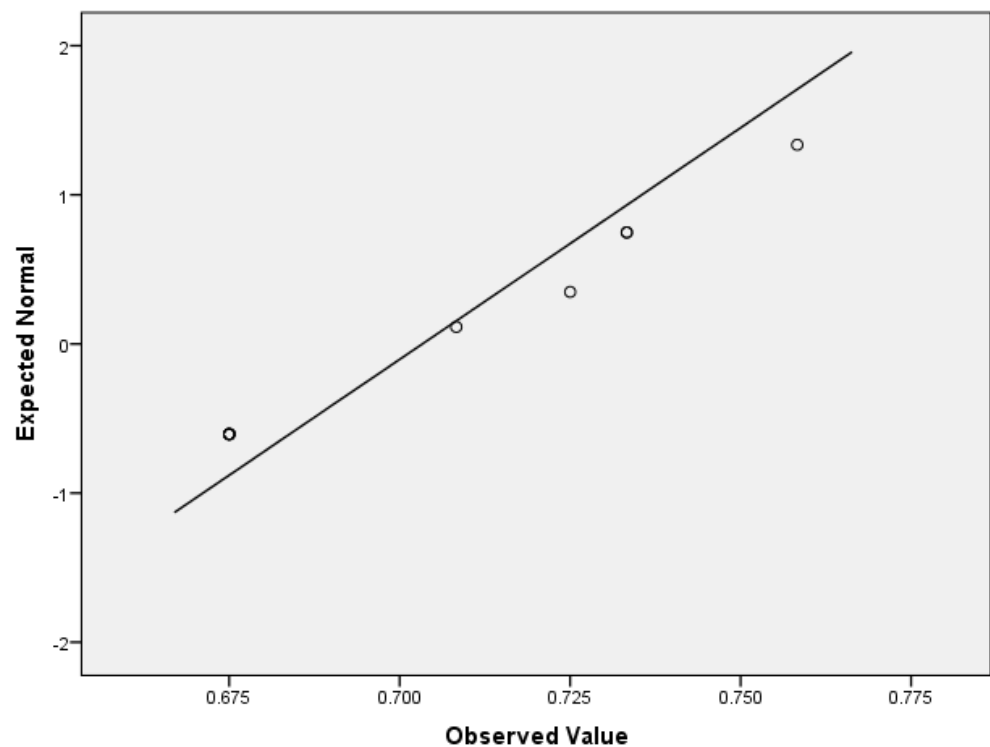


significance level. The obtained results indicate that PDS did not pass the normality assumption, and just PSV passed the normality assumption.

**Table 5.** Verification of data normality assumption using the SW test,  $\alpha_{norm} = 0.05$ .

	PDS	PSV
Test statistics	0.811	0.979
$p$ -value	0.019	0.962
$p$ -value $\geq \alpha_{norm}$	No	Yes
Normality assumption passed	No	Yes

For additional visual validation of the normality assumption analysis results, we have plotted the Q-Q plot for PSD (Figure 12) and PSV (Figure 13). The visual interpretation of Figures 12 and 13 led to the same conclusions as the SW test results (Table 5).



**Figure 12.** Q-Q-plot for PSD.

Based on the results regarding the normality analysis in a nonparametric case, we have noticed that the median was a more appropriate indicator of the central performance tendency than the mean. For the verification of the null hypothesis ( $H_0$ ) that between the medians of PSD and PSV there was no statistically significant difference, we have applied the nonparametric Mann–Whitney test [58] at the  $\alpha_{MW} = 0.05$  significance level. As the result of the test application, it was obtained that the  $p$ -value denoted  $p_{MW} = 0.0002$ , with  $p_{MW} < \alpha_{MW}$  indicating that the medians of PDS and PSV were statistically significantly different. ( $H_0$  should be rejected and  $H_1$ , the alternative hypothesis, must be accepted). Based on the performed analyses, it can be concluded that the decision stump classifier runs better than the support vector classifier.

Experiment 2 with an iteratively increased dataset

In the second experiment, the training set was iteratively increased. It always contained those 3616 records, where the output was 1, and it was extended by  $1 \times 3616$ ,  $2 \times 3616, \dots, 10 \times 3616$  other records randomly. This increased training data produced a

better hit ratio (see Table 6). However, by looking at the confusion matrix of the 10th run (22), which provided the most precise result, you could see:

$$\begin{bmatrix} 10,780 & 8 \\ 1143 & 2 \end{bmatrix} \tag{22}$$

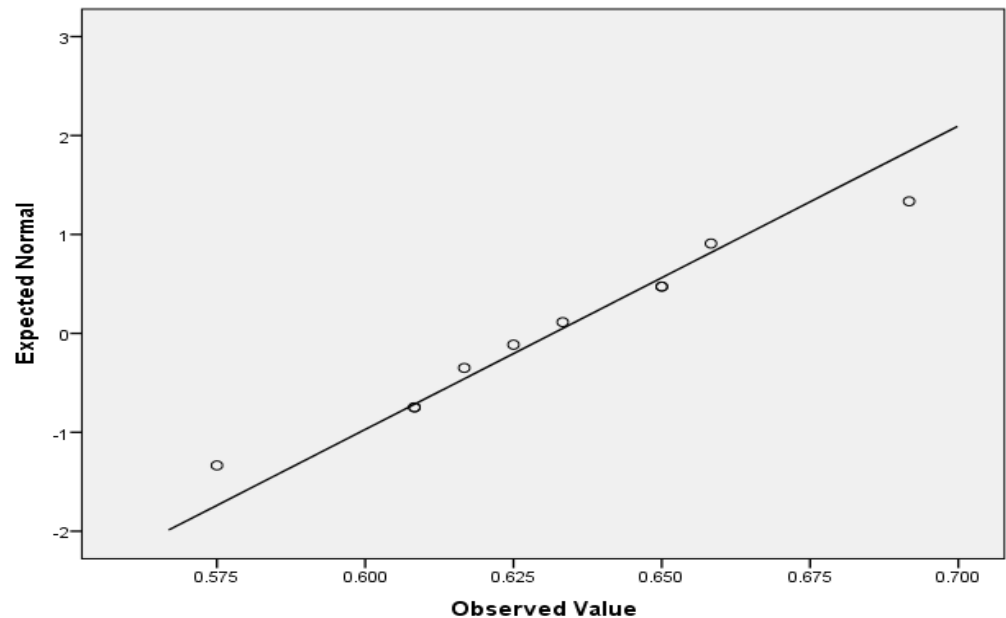


Figure 13. Q-Q-plot for PSV.

Table 6. Experiment 2 evaluation results.

	Training Dataset Size	Precision
1	7232	0.5442
2	10,848	0.6691
3	14,464	0.7486
4	18,080	0.7871
5	21,696	0.8298
6	25,312	0.8540
7	28,928	0.8692
8	32,544	0.8860
9	36,160	0.8965
10	39,776	0.9035

The algorithm correctly classified 10,780 of the 11,923 test input values. However, only two failures of 1143 were classified correctly. This rate is very poor [59] for failure prediction capabilities, and its root is probably the lack of failure records in the database. However, there are domains where the sensitivity is extremely important, i.e., to not miss true positives. There are two reasons for this. On the one hand, the number of normal operation samples is almost ten times bigger than the number of failure samples. On the other hand, there might be no relationship between the measured signals and the observed failures. Some of the referenced applications for the AdaBoost algorithm reported better performances, but their dataset might be better for their purpose.

Further indicators have been calculated according to [60]. The true positive rate (TPR), also known as the sensitivity hit rate:

$$TPR = \frac{TP}{(TP + FN)} = 0.9041 \tag{23}$$

The true negative rate (*TNR*), also known as specificity or selectivity:

$$TNR = \frac{TN}{(TN + FP)} = 0.2 \quad (24)$$

The positive predictive value (*PPV*), also known as precision:

$$PPV = \frac{TP}{(TP + FP)} = 0.9992 \quad (25)$$

The negative predictive value (*NPV*):

$$NPV = \frac{TN}{(TN + FN)} = 0.0017 \quad (26)$$

The false negative rate (*FNR*), also known as the miss rate:

$$FNR = 1 - TPR = 0.0959 \quad (27)$$

The false positive rate (*FPR*), also known as fall-out:

$$FPR = 1 - TNR = 0.8 \quad (28)$$

The false discovery rate (*FDR*):

$$FDR = 1 - PPV = 0.007 \quad (29)$$

The false omission rate (*FOR*):

$$FOR = 1 - NPV = 0.9982 \quad (30)$$

The positive likelihood ratio (*LR*<sup>+</sup>):

$$LR^+ = \frac{TPR}{FPR} = 1.1301 \quad (31)$$

The negative likelihood ratio (*LR*<sup>-</sup>):

$$LR^- = \frac{FNR}{TNR} = 0.4793 \quad (32)$$

The prevalence threshold (*PT*):

$$PT = \frac{\sqrt{TPR}}{(\sqrt{TPR} + \sqrt{FPR})} = 0.4847 \quad (33)$$

The threat score (*TS*) or critical success index (*CSI*):

$$TS = \frac{TP}{(TP + FN + FP)} = 0.9035 \quad (34)$$

The prevalence:

$$Prevalence = \frac{(TP + FN)}{(TP + FN + TN + FP)} = 0.9991 \quad (35)$$

The accuracy (*ACC*):

$$ACC = \frac{(TP + TN)}{(TP + FN + TN + FP)} = 0.9035 \quad (36)$$

The balanced accuracy (*BA*):

$$BA = TPR + TNR = 0.5520 \tag{37}$$

The *F1* score, which is the harmonic mean of precision and sensitivity:

$$F1 = \frac{2TP}{(2TP + FP + FN)} = 0.9035 \tag{38}$$

Matthew’s correlation coefficient (*MCC*):

$$MCC = \frac{(TP + TN) - (FP + FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = 0.0013 \tag{39}$$

The Fowlkes–Mallows index (*FM*):

$$FM = \sqrt{(PPV \cdot TPR)} = 0.9505 \tag{40}$$

Bookmaker informedness (*BM*):

$$BM = TPR + TNR - 1 = 0.1041 \tag{41}$$

Markedness (*MK*):

$$MK = PPV + NPV - 1 = 0.0010 \tag{42}$$

The diagnostic odds ratio (*DOR*):

$$DOR = \frac{LRP}{LRN} = 2.3578 \tag{43}$$

We have analyzed the presence of outliers in the training dataset size (Table 6). The value 7232 was identified as furthest from the rest, but no significant outlier was detected (*p*-value > 0.05).

The precision data from Table 7 was analyzed in-depth. The Grubbs identified that the value 0.5442 was furthest from the rest, but not a significant outlier (*p*-value > 0.05).

**Table 7.** Descriptive statistics for precision (data from Table 6).

Name	Value
minimum	0.5442
maximum	0.9035
range	0.3593
mean	0.7988
95 CI% of the mean	[0.7157, 0.8819]
SD	0.11613
CV	15.54 (relatively homogeneous data)
median	0.8419

Based on the fact that the sample size [61] was 10, which is very small, we have chosen the application of the Shapiro–Wilk test (SW) of normality at the  $\alpha_{norm} = 0.05$  significance level. We obtained, as the results, the test statistics 0.851 and the *p*-value 0.06 ( $0.06 \geq \alpha_{norm}$ ), which indicated that the data passed the normality assumption. Based on this finding, the best indicator of the central tendency was the mean = 0.7988. The results also indicated that the 95 CI% of the mean [0.7157, 0.8819] must be reported.

#### 4. Conclusions and Future Directions

The main purpose of this study was to assess the performance of the AdaBoost algorithm for researchers who intend to use it for problem-solving for diverse data. The study presented an approach to using sensor data to predict the probability of machine failures to track the remaining useful life, so that the proposed algorithm applied machine learning in the field of condition-based maintenance. The AdaBoost algorithm can analyze sensor data from equipment to predict when a machine is likely to fail. This enables maintenance personnel to schedule maintenance in advance, reducing downtime and maintenance costs. By analyzing sensor data and predicting equipment failure, the AdaBoost algorithm can help operation management to move from reactive maintenance to proactive maintenance, thus improving productivity and reducing costs.

Two models were implemented and evaluated by experiments. In the first experiment (300 samples, random factor of weight calculation), two types of classifiers were used. In the decision stump method, the precision of the classification fell into 65–75%, while the support vector classifier's basic classifiers generally were 5–10% worse. The forecast accuracy (especially for the first version) can be considered good, but not excellent. In the second experiment (the training set was iteratively increased), an attempt was made to predict error statements by the four physical telemetry signals. Since less than 0.42% of the dataset reported errors, only some proportion of the entire dataset was taken into account during the training.

The main purpose of this study was to show the limitations of the AdaBoost algorithm for researchers who intend to use it for diverse data. Unfortunately, the AdaBoost algorithm did not produce the expected forecasting capability on the data used, as proven by the confusion matrix: the overall accuracy was not satisfactory. The reason for this could be that the classifiers were not detailed enough or the fact that there was no relationship between the telemetry data and error statements in the dataset we investigated. The authors also ran the AdaBoost algorithm on simulated data, where it performed better. Unfortunately, there were no other real-world data in the field of CBM to test the performance of the algorithm against.

Our research included, among others, a correlation analysis for which we have proposed a decision rule for choosing between the Pearson and Spearman correlation based on data (variable) normality. For a comparison of the performance of the two algorithms, the decision rule for establishing the appropriate central performance tendency indicator was presented. For an accurate comparison of the central performance tendency of the two algorithms, an in-depth statistical analysis was presented that considered the experimental evaluation results' variability and the existence of outlier experimental evaluation results.

As future work, the authors plan to build a sensor set with collective intelligence in a real environment able to collect and intelligently analyze data. The authors expect that more reliable estimations can be made using data whose collection is based on prior knowledge. Furthermore, sensor data may be used to show the opportunity for optimization of the processes. Another direction of future work could be to compare AdaBoost against other machine learning algorithms. Another research direction is to study the long short-term memory model using a diverse dataset, including the dataset used in the research presented in this paper and its comparison with the Adaboost algorithm.

**Author Contributions:** Conceptualization, O.H. and L.B.I.; methodology, O.H. and L.B.I.; software, O.H.; validation, O.H. and L.B.I.; formal analysis, O.H. and L.B.I.; investigation, O.H. and L.B.I.; resources, O.H. and L.B.I.; data curation, O.H.; writing—original draft preparation, O.H.; writing—review and editing, L.B.I.; visualization, O.H. and L.B.I.; supervision, O.H.; project administration, O.H.; funding acquisition, O.H. and L.B.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by National Research Development and Innovation Office, Hungary, grant number 2020-1.1.2-PIACI-KFI-2020-00147.

**Data Availability Statement:** This report is based on the data for predictive maintenance, which are available for download at <https://www.kaggle.com/arnabbiswas1/microsoft-azure-predictive-maintenance/code>, accessed on 10 December 2022.

**Acknowledgments:** This research was realized in consultancy with members of SOON project developed in the framework of the CHIST-ERA program supported by the Future and Emerging Technologies (FET) program of the European Union through the ERA-NET Cofund funding scheme under the grant agreements, title: Social Network of Machines (SOON). This research was partially supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI-UEFISCDI, project number 101/2019, COFUND-CHISTERA-SOON, within PNCDI III. Also, we would like to thank the Research Center on Artificial Intelligence, Data Science, and Smart Engineering (Artemis) for the support.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

### Abbreviations

CBM	Condition-Based Maintenance
IoT	Internet of Things
PHM	Prognostics and Health Management
RUL	Remaining Useful Lifetime
Q–Q plot	Quantile–Quantile plot
CI	Confidence Interval
ANOVA	Analysis of Variance
BPNN	Backpropagation Neural Network
SVM	Support Vector Machine
SVR	Support Vector Regression
PDS	Precision with Decision Stump
PSV	Precision with Support Vector Classifier
SD	Standard Deviation
CV	Coefficient of Variation
SW	Shapiro–Wilk Test of Normality
MW	Mann–Whitney Test

### References

- Vijayan, D.S.; Rose, A.L.; Arvindan, S.; Revathy, J.; Amuthadevi, C. Automation systems in smart buildings: A review. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *1–13*. [[CrossRef](#)]
- Oliveira, M.; Torgo, L.; Costa, V.S. Evaluation Procedures for Forecasting with Spatiotemporal Data. *Mathematics* **2021**, *9*, 691. [[CrossRef](#)]
- Leon, F.; Gavrilescu, M. A Review of Tracking and Trajectory Prediction Methods for Autonomous Driving. *Mathematics* **2021**, *9*, 660. [[CrossRef](#)]
- Haq, I.U.; Ullah, A.; Khan, S.U.; Khan, N.; Lee, M.Y.; Rho, S.; Baik, S.W. Sequential Learning-Based Energy Consumption Prediction Model for Residential and Commercial Sectors. *Mathematics* **2021**, *9*, 605. [[CrossRef](#)]
- Alanne, K.; Sierla, S. An overview of machine learning applications for smart buildings. *Sustain. Cities Soc.* **2022**, *76*, 103445. [[CrossRef](#)]
- Iantovics, L.B. Black-Box-Based Mathematical Modelling of Machine Intelligence Measuring. *Mathematics* **2021**, *9*, 681. [[CrossRef](#)]
- Iantovics, L.B.; Kountchev, R.; Crişan, G.C. ExtrIntDetect-A New Universal Method for the Identification of Intelligent Cooperative Multiagent Systems with Extreme Intelligence. *Symmetry* **2019**, *11*, 1123. [[CrossRef](#)]
- BS EN 13306; Maintenance-Maintenance Terminology. British Standards Institution: London, UK, 2017.
- Hiruta, T.; Uchida, T.; Yuda, S.; Umeda, Y. A design method of data analytics process for condition based maintenance. *CIRP Ann.* **2019**, *68*, 145–148. [[CrossRef](#)]
- Ahmad, R.; Kamaruddin, S. An overview of time-based and condition-based maintenance in industrial application. *Comput. Ind. Eng.* **2012**, *63*, 135–149. [[CrossRef](#)]
- Gouriveau, R.; Medjaher, K.; Zerhouni, N. *From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics*; John Wiley & Sons: Hoboken, NJ, USA, 2016; pp. 67–135.
- Muhonen, T. *Standardization of Industrial Internet and Iot (Iot–Internet of Things)–Perspective on Condition-Based Maintenance*; University of Oulu: Oulu, Finland, 2015.

13. Jo, O.; Kim, Y.-K.; Kim, J. Internet of things for smart railway: Feasibility and applications. *IEEE Internet Things J.* **2017**, *5*, 482–490. [[CrossRef](#)]
14. Xu, X.; Chen, T.; Minami, M. Intelligent fault prediction system based on internet of things. *Comput. Math. Appl.* **2012**, *64*, 833–839. [[CrossRef](#)]
15. Fumeo, E.; Oneto, L.; Anguita, D. Condition based maintenance in railway transportation systems based on big data streaming analysis. *Procedia Comput. Sci.* **2015**, *53*, 437–446. [[CrossRef](#)]
16. Kumar, A.; Shankar, R.; Thakur, L.S. A big data driven sustainable manufacturing framework for condition-based maintenance prediction. *J. Comput. Sci.* **2018**, *27*, 428–439. [[CrossRef](#)]
17. Idé, T. Collaborative anomaly detection on blockchain from noisy sensor data. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 120–127.
18. Cerna, M.; Harvey, A.F. *The Fundamentals of FFT-Based Signal Analysis and Measurement*; Application Note 041; National Instruments: Austin, TX, USA, 2000.
19. Si, X.S.; Wang, W.; Hu, C.H.; Zhou, D.H. Remaining useful life estimation—A review on the statistical data driven approaches. *Eur. J. Oper. Res.* **2011**, *213*, 1–14. [[CrossRef](#)]
20. Prajapati, A.; Bechtel, J.; Ganesan, S. Condition based maintenance: A survey. *J. Qual. Maint. Eng.* **2012**, *18*, 384–400. [[CrossRef](#)]
21. Peng, Y.; Dong, M.; Zuo, M.J. Current status of machine prognostics in condition-based maintenance: A review. *Int. J. Adv. Manuf. Technol.* **2010**, *50*, 297–313. [[CrossRef](#)]
22. Wilhelmsen, D.R. A Markov inequality in several dimensions. *J. Approx. Theory* **1974**, *11*, 216–220. [[CrossRef](#)]
23. Gentile, C.; Littlestone, N. The robustness of the p-norm algorithms. In Proceedings of the Twelfth Annual Conference on Computational Learning Theory, Santa Cruz, CA, USA, 6–9 July 1999; pp. 1–11.
24. Stefansky, W. Rejecting outliers in factorial designs. *Technometrics* **1972**, *14*, 469–479. [[CrossRef](#)]
25. Lilliefors, H. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *J. Am. Stat. Assoc.* **1967**, *62*, 399–402. [[CrossRef](#)]
26. Lilliefors, H. On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown. *J. Am. Stat. Assoc.* **1969**, *64*, 387–389. [[CrossRef](#)]
27. Dallal, G.E.; Wilkinson, L. An analytic approximation to the distribution of Lilliefors’s test statistic for normality. *Am. Stat.* **1986**, *40*, 294–296.
28. Stephens, M.A. EDF Statistics for Goodness of Fit and Some Comparisons. *J. Am. Stat. Assoc.* **1974**, *69*, 730–737. [[CrossRef](#)]
29. Shapiro, S.S.; Wilk, M.B. An analysis of variance test for normality (complete samples). *Biometrika* **1965**, *52*, 591–611. [[CrossRef](#)]
30. Razali, N.; Wah, Y.B. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *J. Stat. Model. Anal.* **2011**, *2*, 21–33.
31. Wilk, M.B.; Gnanadesikan, R. Probability plotting methods for the analysis of data. *Biometrika* **1968**, *55*, 1–17. [[CrossRef](#)] [[PubMed](#)]
32. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
33. Stigler, S.M. Francis Galton’s Account of the Invention of Correlation. *Stat. Sci.* **1989**, *4*, 73–79. [[CrossRef](#)]
34. Iantovics, L.B.; Enăchescu, C. Method for Data Quality Assessment of Synthetic Industrial Data. *Sensors* **2022**, *22*, 1608. [[CrossRef](#)]
35. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; Publisher CRC Press: Boca Raton, FL, USA, 2012; pp. 23–44.
36. Schapire, R.E. Explaining adaboost. In *Empirical Inference*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52.
37. Freund, Y.; Schapire, R.E.; Abe, N.A. A short introduction to boosting. *J. Jpn. Soc. Artif. Intell.* **1999**, *14*, 771–780.
38. Schapire, R.E. The strength of weak learnability. *Mach. Learn.* **1990**, *5*, 197–227. [[CrossRef](#)]
39. Ding, Y.; Zhu, H.; Chen, R.; Li, R. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. *Appl. Sci.* **2022**, *12*, 5872. [[CrossRef](#)]
40. Schapire, R.E.; Freund, Y. Boosting: Foundations and algorithms. *Kybernetes* **2013**, *42*, 164–166. [[CrossRef](#)]
41. Freund, R.M.; Grigas, P.; Mazumder, R. A new perspective on boosting in linear regression via subgradient optimization and relatives. *Ann. Stat.* **2017**, *45*, 2328–2364. [[CrossRef](#)]
42. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the Machine Learning: Thirteenth International Conference, Bari, Italy, 3–6 July 1996; pp. 148–156.
43. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [[CrossRef](#)]
44. Tsai, J.K.; Hung, C.H. Improving AdaBoost classifier to predict enterprise performance after COVID-19. *Mathematics* **2021**, *9*, 2215. [[CrossRef](#)]
45. Wang, C.; Xu, S.; Yang, J. AdaBoost Algorithm in Artificial Intelligence for Optimizing the IRI Prediction Accuracy of Asphalt Concrete Pavement. *Sensors* **2021**, *21*, 5682. [[CrossRef](#)]
46. Wang, J.; Xue, W.; Shi, X.; Xu, Y.; Dong, C. AdaBoost-Based Machine Learning Improved the Modeling Robust and Estimation Accuracy of Pear Leaf Nitrogen Concentration by In-Field VIS-NIR Spectroscopy. *Sensors* **2021**, *21*, 6260. [[CrossRef](#)]
47. Natras, R.; Soja, B.; Schmidt, M. Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting. *Remote Sens.* **2022**, *14*, 3547. [[CrossRef](#)]
48. Wei, S.; Zhu, L.; Chen, L.; Lin, Q. An AdaBoost-Based Intelligent Driving Algorithm for Heavy-Haul Trains. *Actuators* **2021**, *10*, 188. [[CrossRef](#)]

49. Javeed, A.; Dallora, A.L.; Berglund, J.S.; Anderberg, P. An Intelligent Learning System for Unbiased Prediction of Dementia Based on Autoencoder and AdaBoost Ensemble Learning. *Life* **2022**, *12*, 1097. [[CrossRef](#)]
50. Al-Hadeethi, H.; Abdulla, S.; Diykh, M.; Green, J.H. Determinant of Covariance Matrix Model Coupled with AdaBoost Classification Algorithm for EEG Seizure Detection. *Diagnostics* **2021**, *12*, 74. [[CrossRef](#)]
51. Sun, S.; Zhang, Q.; Sun, J.; Cai, W.; Zhou, Z.; Yang, Z.; Wang, Z. Lead–Acid Battery SOC Prediction Using Improved AdaBoost Algorithm. *Energies* **2022**, *15*, 5842. [[CrossRef](#)]
52. Li, R.; Sun, H.; Wei, X.; Ta, W.; Wang, H. Lithium Battery State-of-Charge Estimation Based on AdaBoost.Rt-RNN. *Energies* **2022**, *15*, 6056. [[CrossRef](#)]
53. Zhao, H.; Zhang, L.; Ren, J.; Wang, M.; Meng, Z. AdaBoost-Based Back Analysis for Determining Rock Mass Mechanical Parameters of Claystones in Goupitan Tunnel, China. *Buildings* **2022**, *12*, 1073. [[CrossRef](#)]
54. Wu, X.; Lu, X.; Leung, H. A video based fire smoke detection using robust AdaBoost. *Sensors* **2018**, *18*, 3780. [[CrossRef](#)] [[PubMed](#)]
55. Ferreira, J.M.; Pires, I.M.; Marques, G.; Garcia, N.M.; Zdravevski, E.; Lameski, P.; Flórez-Revuelta, F.; Spinsante, S. Identification of daily activities and environments based on the AdaBoost method using mobile device data: A systematic review. *Electronics* **2020**, *9*, 192. [[CrossRef](#)]
56. Ying, C.; Qi-Guang, M.; Jia-Chen, L.; Lin, G. Advance and prospects of AdaBoost algorithm. *Acta Autom. Sin.* **2013**, *39*, 745–758.
57. WANG, R. AdaBoost for feature selection, classification and its relation with SVM, a review. *Phys. Procedia* **2012**, *25*, 800–807. [[CrossRef](#)]
58. Fay, M.P.; Proschan, M.A. Wilcoxon–Mann–Whitney or *t*-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Stat. Surv.* **2010**, *4*, 1–39. [[CrossRef](#)]
59. Wang, W.; Zhang, W. An asset residual life prediction model based on expert judgments. *Eur. J. Oper. Res.* **2008**, *188*, 496–505. [[CrossRef](#)]
60. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]
61. Bonett, D.G.; Wright, T.A. Sample size requirements for Pearson, Kendall, and Spearman correlations. *Psychometrika* **2000**, *65*, 23–28. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.