EARLY FAILURES DIAGNOSIS BASED ON MULTI-DIMENSIONAL ANALYSIS AND SEPARATION OF HIGH-DIMENSIONAL TEST RESULTS OF AN ELECTRONIC SYSTEM

Mohamed Denguir, Naime Denguir and Sebastian M. Sattler

Friedrich-Alexander-University Erlangen-Nuremberg (FAU)
Chair of Reliable Circuits and Systems Erlangen, Germany

ABSTRACT

When carrying out a separation of test results from an electronic system, coping with enormous high-dimensional data sets is necessary but problematic. The input of high-dimensional data, in which not a few elements of a data set are irrelevant or less relevant than others, usually lead to inadequate results. It is therefore helpful to use methods that classify the individual dimensions of the data set according to their relevance, or the deviation of the set in this dimension. Such data analysis methods are used in many areas, such as industry, medicine, biology or in the field of military reconnaissance for data reduction and classification as well as foresting and verification of electronic systems and circuits. In this paper, the Principal Component Analysis and the Linear Discriminant Analysis are presented as classification methods, their mathematical background is explained and flow diagrams of the algorithms are presented. By means of areal application example i.e. test results of an electronic system it becomes clear, how suitable these classification methods are in order to analyse of high-dimensional test results (test data) and e.g. to discover the reasons of early failure of an electronic system. Further, it is shown how useful it is to carry out both classification procedures in a certain order in order to ensure an optimal analysis and to come to the cause of an early failure more precisely.

KEYWORDS

Early Failure Analysis, Fault Diagnosis, Testing and Verification of Electronic System, Dimensional Data Reduction, Multidimensional Analysis of Test Results, Group Separation, Adaptive Multidimensional Analysis, Machine Learning, Reduction of test costs

1. INTRODUCTION

In many areas of research, the analysis of test results of a system is necessary. Huge data sets of messages and signals of countless redundant sensors of a system are characterized by criteria such as their amount, their complexity and their speed. Globally, companies and research institutes strive to discover valuable information and connections from the vast amounts of data that have so far been difficult or impossible to determine [1]. Very often, enormous, high-dimensional data sets need to be collected and analysed from experiments. However, the input of high-dimensional data, in which not a few elements of a data set are irrelevant or less relevant than others, usually lead to inadequate results [2]. Thus, it is useful to use classification methods that classify the individual dimensions of the data set according to their relevance. In this paper, two different known classification procedures will be investigated and discussed. We present the Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA). We are explaining the mathematical background for each method, using flow charts to summarize the corresponding algorithms and illustrating them by means of the same application example (database). In this example, it's about finding out cause of early failures of an electronic medicine system (device). For this purpose, we developed, with the help of the resulting flow charts, self-programmed

MATLAB codes for each method in order to determine the results of classification, generate graphical representations and provide visual insight into the capabilities of the analysis programs. We show how a specific order of the analysis methods can lead to more precise results in the matter of analysis and separation of high-dimensional test results of an electronic system. Finally, we compare the methods with each other based on criteria such as ease of use and classifying capacity and discuss the advantages, disadvantages and the possible applications of the individual procedures.

2. PRINCIPAL COMPONENT ANALYSIS

2.1. Definition

The PCA is a variable-orientated, linear classification method for data reduction. It was introduced by Karl Pearson in 1901 [3] and further developed by Harold Hotelling in the 1930s [4]. The method uses linear structures enabling the reduction and interpretation of large multivariate data sets. This method allows the user to replace a number of original variables by a smaller number and it extracts relevant information from a given data set by reducing the dimension. By means of an orthogonal transformation, a new set of uncorrelated variables, the so-called Principal Components (PCs), is generated as a transformed database [5]. The newly determined PCs are linear combinations of the original variables. The first PC is so designed to be responsible for most of the variation in the original data and thus causing the reduction of the data size [6]. If the first PC describes the majority of the data variation, than this can also reduce the dimension of the problem. Through the transformation into PCs, the data sets can be visualized graphically and interpreted better.

2.2. Mathematical derivation

Since the PCA is already well established in today's technology and is already actively used e.g. in image processing [7], in the analysis of dynamic movements [8] or even in the anomaly detection in spacecraft [9], we restrict ourselves to the required steps and their corresponding most important equations to set up the flow chart. Figure 1 describes the PCA algorithm by means of a flow chart. In it, the derivation of the PCs is represented by the mathematical formulas required. The PCA allows to obtain PCs or a transformed database (same abbreviation: PC) after entering the original database (D), with n-rows and m-columns ((n x m)-matrix) and performing five steps. First (step 1), a standardised database (S) is generated, which is column-wise mean-free, has column-wise value one as mean variation and occupies the same dimension as D. The sense of this step (standardisation) is to transform the various variables in the database so that they accept similar values and are directly comparable. Then (step 2), a correlated database (C) is generated, from which a correlation matrix ((m x m)-matrix) emerges, giving information about the relationships of variables. Further (step 3), the eigenvalues λ_j for $j = 1, 2, \ldots, m$ of the calculated correlation matrix are determined. The eigenvalues λ_j, characterizing general properties of linear images, are ordered accordingly to their size from large to small. Next (step 4), the eigenvectors V_j are determined with the help of the calculated and ordered eigenvalues λ_{sj} of correlation matrix C. Last (step 5), the subsequent multiplication of the standardised data S with the eigenvector matrix $V = (V_j)$ results in the transformed database PC. Thus, we have converted the original database D into database PC, which has the same dimension of a (n x m)-matrix [10]. Here one speaks of an orthogonal transformation or a projection of the standardised database S onto the eigenvectors V_j, which are therefore called the coefficients of the PCs. However, the corresponding vectors PC_j (n x 1-matrix) to the columns of PC are not all equivalent. They can be arranged depending on the size of the ordered eigenvalues λ_{sj} of the correlation matrix C. The information value of the variables decreases from PC_1 to PC_m. The following considerations are used to determine the variances of each PC. This should give us an idea of how the variances are related to the eigenvalues λ_{sj}. In general, the variance of PC_j can be represented by means of (1).

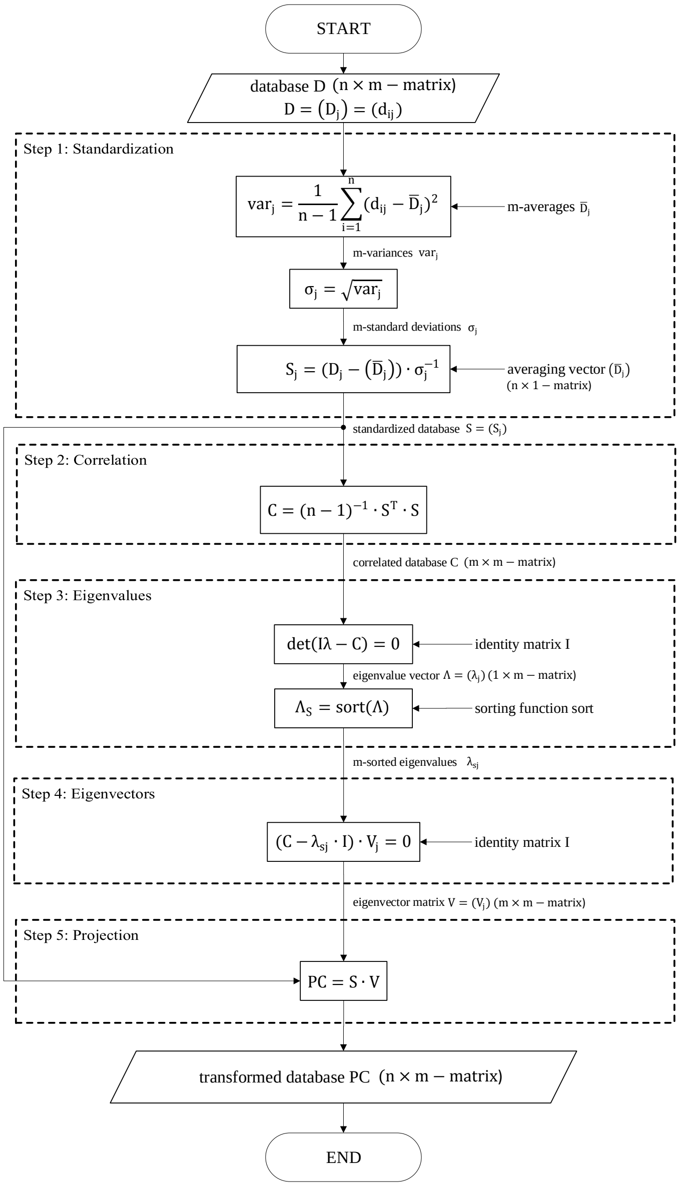

START
database D (n × m − matrix)
D = (D_j) = (d_ij)
Step 1: Standardization
var_j = 1/(n − 1) Σ_{i=1}^{n} (d_ij − D̄_j)²
m-averages D̄_j
m-variances var_j
σ_j = √var_j
m-standard deviations σ_j
S_j = (D_j − (D̄_j)) · σ_j^{−1}
averaging vector (D̄_j) (n × 1 − matrix)
standardized database S = (S_j)
Step 2: Correlation
C = (n − 1)^{−1} · S^T · S
correlated database C (m × m − matrix)
Step 3: Eigenvalues
det(Iλ − C) = 0
identity matrix I
eigenvalue vector Λ = (λ_j) (1 × m − matrix)
Λ_S = sort(Λ)
sorting function sort
m-sorted eigenvalues λ_sj
Step 4: Eigenvectors
(C − λ_sj · I) · V_j = 0
identity matrix I
eigenvector matrix V = (V_j) (m × m − matrix)
Step 5: Projection
PC = S · V
transformed database PC (n × m − matrix)
END

Figure 1. Flow chart of the PCA

In the following equations mentioned in this paper the abbreviation for variance and transposed will be "var" and "T". The matrix S is mean-free, i.e. the averaging vector $\overline{S} = (0)$. It follows that $\overline{PC}_j = \overline{S}\cdot(\overline{V}_j)$ equals zero and thus the variance of PC_j can further be calculated as defined in (2).

$$\mathrm{var}(PC_j) = \frac{1}{n - 1}(PC_j - (\overline{PC_j}))^T\cdot(PC_j - (\overline{PC_j})) \tag{1}$$

$$\mathrm{var}(PC_j) = \frac{1}{n - 1}\ (PC_j^T\cdot PC_j) = \frac{1}{n - 1}(S\cdot V_j)^T\cdot(S\cdot V_j) \tag{2}$$

According to general mathematical matrix rules $(S\cdot V_j)^T = V_j^T\cdot S^T$ follows and thus (3). After conversion of the formula from step 2 (Correlation) from the flow chart (Figure 1) one obtains (4). Substituting (4) into (3) results in (5).

$$\mathrm{var}(PC_j) = \frac{1}{n - 1}(V_j^T\cdot S^T\cdot S\cdot V_j) \tag{3}$$

$$S^T\cdot S = (n - 1)\cdot C \tag{4}$$

$$\mathrm{var}\left(\mathrm{PC_j}\right) = \frac{1}{\mathrm{n - 1}}(\mathrm{V_j^T} \cdot (\mathrm{n - 1}) \cdot \mathrm{C} \cdot \mathrm{V_j}) \tag{5}$$

Since correlation matrices are in general symmetrically and square, the eigenvectors V_j of correlation matrix C are orthogonal [10], i.e. $V^T = V^{-1}$, which is why (6) results the following way. After reducing the constant (n-1) in (6), finally (7) results for the variance of a j-th PC.

$$\mathrm{var}\left(\mathrm{PC_j}\right) = \frac{1}{\mathrm{n - 1}}(\mathrm{V_j^{-1}} \cdot (\mathrm{n - 1}) \cdot \mathrm{C} \cdot \mathrm{V_j}) \tag{6}$$

$$\mathrm{var}\left(\mathrm{PC_j}\right) = \left(\mathrm{V_j^{-1}} \cdot \mathrm{C} \cdot \mathrm{V_j}\right) = \lambda_{\mathrm{sj}} \tag{7}$$

So mathematically, it can be shown that the variance of a j-th PC equals the j-th eigenvalue of the correlated database. Essentially, the PCA corresponds to a rotation of the coordinate system in the direction of maximum variance [10]. The first PC shows the greatest variance, since within the analysis the eigenvalues were arranged according to their size. Equation (8) follows accordingly, which reproduces the proportion of shared variance of the data.

$$\frac{1}{\mathrm{m}} \cdot \lambda_{\mathrm{sj}} = \frac{1}{\mathrm{m}} \cdot \mathrm{var}(\mathrm{PC_j}) \tag{8}$$

Thus PCs with great variance represent interesting dynamics while PCs with low variance represent low noise and therefore not much of information of the original database gets lost, when PCs with low variance are ignored [11]. The following application example eases the understanding of the theory and mathematics of the PCA discussed so far.

2.3. Application example

In many areas of research, it is necessary to detect errors and thus search out the causes of e.g. early failure. For that, PCA is a very useful analysis tool. Let us say a certain company produces and sells an electronic product, which consists of many digital and analog subsystems. Often their product "breaks" before the warranty period. The reasons for the early failure must be identified in order to achieve improvements in product production. Meanwhile, in many products an integrated chip stores important information about user and product behaviour. Engineers can use these information as a database and filter the most important user variables, responsible for the early failure, by using the PCA. For this, the knowledge about user variables of functional, not early failed products is necessary to enable a separation of the variables. In this case, the user variables e.g. voltage, current, temperature, etc. are the eigenvectors and the products are the PCs. We demonstrate the PCA analysis through the following study: We consider a database of about 2300 data sets or objects and 68 features (user variables) that should represent 2300 different devices of the same product of a company (Data for reasons of data protection not explicitly shown). They are sorted according to their lifespan (Figure 2), so that the first 450 represent early failed products (red) and the last 450 represent late failed products (green). The remaining data in black are in the line between early and late failure, so will not be included in the PCA and in the entire article. We concentrate ourselves on similar numbers of early and late failed products for reasons of clarity and accuracy. After performing the PCA algorithm (Figure 1) on the 900 sorted data sets, 68 PCs are calculated (PC_1 to PC_{68}). Since the method bases on matrices, we used a self-written program in MATLAB. For each PC, the percent of the variance (per_of_var) is calculated using (9) and the results are shown in Table 1. Table 1 shows that the four first PCs have the largest eigenvalues and cover over 69% of the variance. The number of relevant PCs depends on the point at which the remaining eigenvalues are relatively small and approximately all equally large. As a result of the PCA, it is clear that the first four PCs are responsible for about 70% of data information, while the remaining PCs are contributing to 4% or less. In addition, a visual representation of the eigenvalues against the PC number (Figure 3) is also helpful for the determination of the relevant PCs. Next, it is useful to display the object distribution in a plot with respect to the PCs.

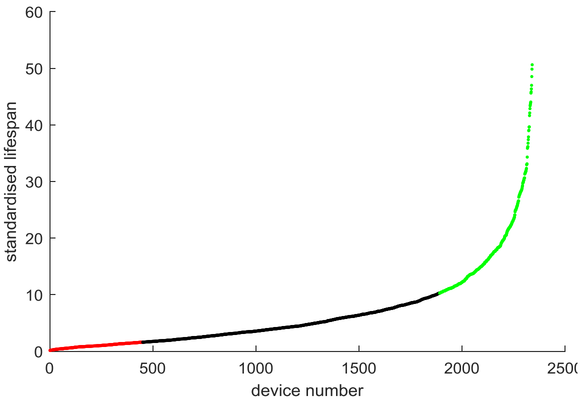


Figure 2. Life curve for 2300 devices of the same product

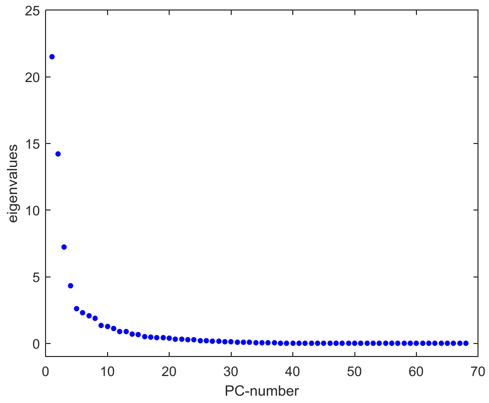


Figure 3. Representation of the eigenvalues over the PC number

Table 1. Eigenvalues and percentages of variances

PC	λs	percent of variance	PC	λs	percent of variance	PC	λs	percent of variance	PC	λs	percent of variance
1	21,4944	31,6094 %	**18**	0,4357	0,6407 %	**35**	0,0433	0,0637 %	**52**	0,0012	0,0018 %
2	14,2059	20,8910 %	**19**	0,4142	0,6091 %	**36**	0,0350	0,0515 %	**53**	0,0006	0,0009 %
3	7,2234	10,6226 %	**20**	0,3992	0,5871 %	**37**	0,0310	0,0456 %	**54**	0,0003	0,0004 %
4	4,3277	6,3643 %	**21**	0,3231	0,4751 %	**38**	0,0294	0,0432 %	**55**	0,0003	0,0004 %
5	2,6031	3,8281 %	**22**	0,3039	0,4469 %	**39**	0,0232	0,0341 %	**56**	0,0002	0,0003 %
6	2,2873	3,3637 %	**23**	0,2760	0,4059 %	**40**	0,0165	0,0243 %	**57**	0,0002	0,0003 %
7	2,0789	3,0572 %	**24**	0,2642	0,3885 %	**41**	0,0127	0,0187 %	**58**	0,0002	0,0003 %
8	1,8859	2,7734 %	**25**	0,2190	0,3221 %	**42**	0,0117	0,0172 %	**59**	0,0001	0,0001 %
9	1,3294	1,9550 %	**26**	0,1871	0,2751 %	**43**	0,0083	0,0122 %	**60**	0,0001	0,0001 %
10	1,2881	1,8943 %	**27**	0,1780	0,2618 %	**44**	0,0071	0,0104 %	**61**	0,0001	0,0001 %
11	1,1177	1,6437 %	**28**	0,1630	0,2397 %	**45**	0,0059	0,0087 %	**62**	0,0001	0,0001 %
12	0,9086	1,3362 %	**29**	0,1319	0,1940 %	**46**	0,0053	0,0078 %	**63**	0,0001	0,0001 %
13	0,8752	1,2871 %	**30**	0,1250	0,1838 %	**47**	0,0039	0,0057 %	**64**	0	0 %
14	0,7163	1,0534 %	**31**	0,1025	0,1507 %	**48**	0,0031	0,0046 %	**65**	0	0 %
15	0,6712	0,9871 %	**32**	0,0894	0,1315 %	**49**	0,0027	0,0040 %	**66**	0	0 %
16	0,5224	0,7682 %	**33**	0,0773	0,1137 %	**50**	0,0017	0,0025 %	**67**	0	0 %
17	0,4631	0,6810 %	**34**	0,0658	0,0968 %	**51**	0,0016	0,0024 %	**68**	0	0 %

$$\text{per\_of\_var} = \frac{\text{var}(\text{PC}_j)}{\sum_{j=1}^{68}\text{var}(\text{PC}_j)} \cdot 100\% = \frac{\lambda_{sj}}{\sum_{j=1}^{68}\lambda_{sj}} \cdot 100\% \qquad (9)$$

For the graphic representation, the coordinates of the 900 objects (450 red and 450 green), sorted according to lifespan, are plotted with respect to PC_1 and PC_2 in a coordinate system with PC_1 as x-axis and PC_2 as y-axis and we obtain Figure 4. This figure shows something interesting: From this simple representation, a first separation of the data between early and late failed can be observed. Although not all objects can be separated to a hundred percent and overlapping being avoided, a large part of the object distribution is specific. In order to display the features graphically, their coordinates have to be plotted. These are listed in eigenvectors, which are defined as coefficients of the PCs. We seek for 2D-plots and thus choosing the first eigenvectorV_1 as x and the second eigenvectorV_2 as y-axis. In addition, the individual points in the plot are linked to the origin in order to obtain vectors of features M_j for j = 1 to 68 and thus better represent their location in the coordinate system (Figure 5). It is obvious, that some features as e.g. M1, M7, M8 and M9 cannot be seen clearly, since overlapping occurs, which arises through same coordinates in the eigenvectors. We now want to investigate what the causes of early and late failure are. To answer this question, we overlay the representations of object data distribution and vectors of features (Figure 4 and Figure 5) in Figure 6. This allows to see, which features are in which areas of the objects and thus possibly influencing the behaviour of the devices. It is clear, that the features M1, M6, M7, M8, M9, M10, M24 and M25 clearly correlate with red objects (early failure) whereas e.g. features M4, M56, M57, M59, M65 and M66 are correlating with the green object cloud (late failure).

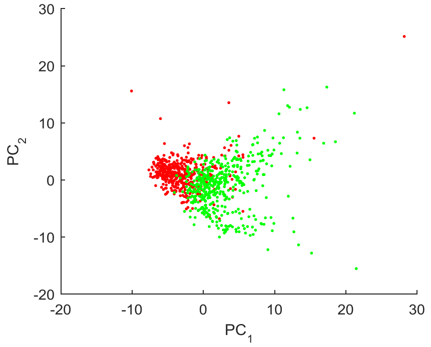


Figure 4. Representation of the object data distribution for the first two PCs

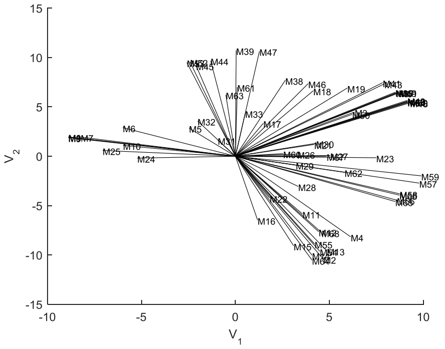


Figure 5. Representation of the vectors of all 68 features M for the first two PCs

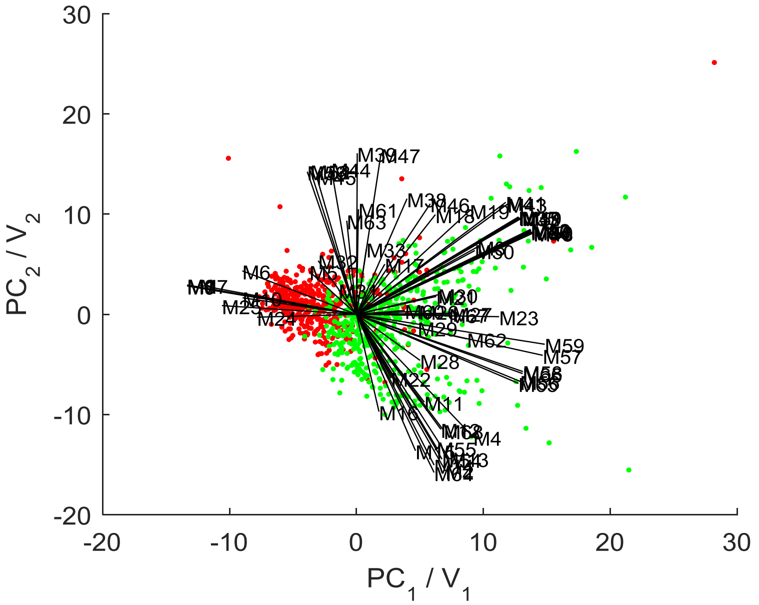


Figure 6. Representation of the object data distribution and vectors of all 68 features M

In order to confirm and investigate more precisely the above-observed correlation of the mentioned features with the object clouds, the entire database is reduced to some of these features. This is followed by a re-execution of the PCA on the reduced database. As a result, the statements made in Figure 6 are confirmed in Table 2 and in the corresponding Figure 7, analogous to Table 1 and Figure 6, for the features M1, M6, M7, M8, M9, M10, M24 and M25.

Table 2. Eigenvalues and percent of variances for all PCs of reduced database (M1, M6, M7, M8, M9, M10, M24 and M25)

PC	λ_s	percent of variance	cumulative
1	5,5473	69,341 %	69,341 %
2	1,1049	13,811 %	83,152 %
3	0,5818	7,2731 %	90,425 %
4	0,4284	5,3552 %	95,78 %
5	0,3367	4,2085 %	99,989 %
6	0,0006	0,0084 %	99,997 %
7	0,0002	0,0025 %	100 %
8	0,0001	0,0002 %	100 %

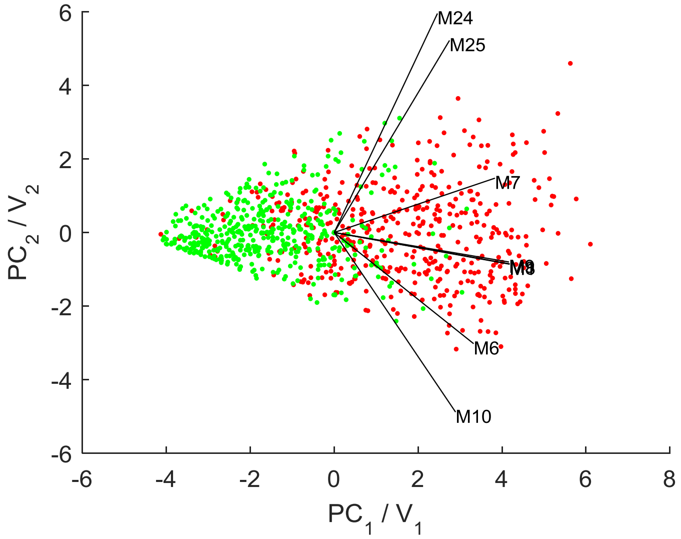


Figure 7. Representation of the object data distribution and vectors of features of reduced database (M1, M6, M7, M8, M9, M10, M24 and M25)

The first two PCs cover almost 83% of the variance and thus the data. So more than compared to the entire database, where the first two PCs cover only 50% of the data (Table 1). The correlation between the features in the reduced database is stronger, or there is a greater linear dependency between the features and the late-failed objects. Figure 7 shows the object data distribution and the vectors of features. To the right there is a red object cloud (early failed devices) and all eigenvectors or features show in the same direction (orientation). Here too, an overlapping between the features M1, M8 and M9 occurs. This figure can be used to indicate which features are more important or play a major role for an early failure than others. M1, M6, M7, M8 and M9 correlate more than M10, M24 and M25 because they accumulate stronger with the red object cloud. In addition to the direction, the magnitude of the vectors also has an information content. The longer the vector, the more often the corresponding feature appears, and the more important is this user variable. In our case (Figure 7) no further information are obtained from the magnitudes since all vectors are approximately equal in length. Next, PCA is applied to the features M4, M23, M56, M57, M59, M64, M65 and M66 and we obtain Table 3 and Figure 8 as result. Table 3 shows that the first two PCs cover over 86% of the data, and the first four together over 97%. This indicates a strong correlation of the features with late failed devices. This can be seen better in Figure 8, a green cloud to the right with eigenvectors in the same direction. Thus the statements made in Figure 6 are confirmed. Here too, some features like M56, M57, M59, M65 and M66 correlate more than M4, M23 and M64 because of the stronger accumulation with the green object cloud. The interpretation of the PCA results can be carried out not only by means of graphical representations, but also very well and more precisely by calculation, e.g. by means of the mathematical concept of the correlation coefficient.

Table 3. Eigenvalues and percent of variances for all PCs of reduced database (M4, M23, M56, M57, M59, M64, M65 and M66)

PC	λ_s	percent of variance	cumulative
1	5,9615	74,519 %	74,519 %
2	0,9642	12,052 %	86,572 %
3	0,6683	8,3542 %	94,926 %
4	0,2210	2,7619 %	97,688 %
5	0,1137	1,4222 %	99,110 %
6	0,0432	0,5402 %	99,650 %
7	0,0276	0,3442 %	99,994 %
8	0,0005	0,0057 %	100 %

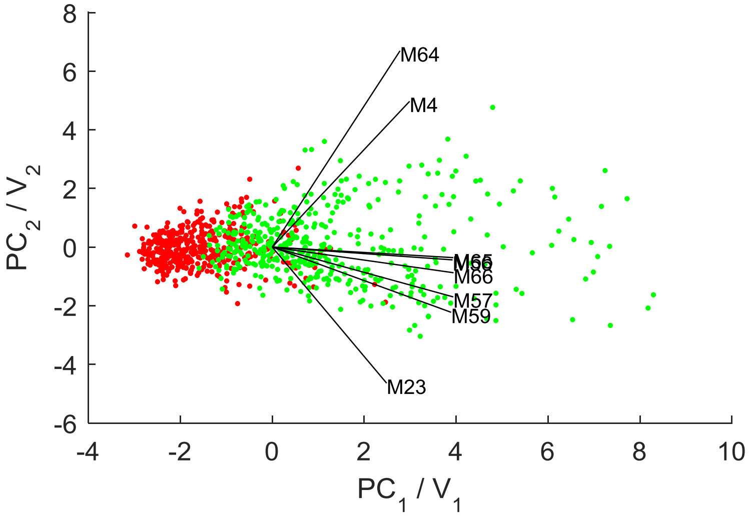


Figure 8. Representation of the object data distribution and vectors of features of reduced database (M4, M23, M56, M57, M59, M64, M65 and M66)

The measure of a linear relationship of two interval-scaled features can be described by the dimensionless correlation coefficient. Let the correlation coefficient of vector X = (x_1, x_2, ...,x_n) and vector Y = (y_1, y_2, ..., y_n) be defined by (10). Correlation coefficients may take on values between -1 and 1, where a positive value implies a relationship such that high values of characteristic x are associated with high values of characteristic y while negative values describe an opposite relationship. A value of zero implies an uncorrelated relationship between the features. Through (10), the correlation coefficients of PC_1 and PC_2 for the red as well as for the green objects from the reduced database M1, M6, M7, M8, M9, M10, M24 and M25 (Figure 7) were determined in (11) and (12).

$$\mathrm{corr(X,Y)}= \frac{\frac{1}{1-n}\sum_{i=1}^{n}(x_i - \overline{x})\cdot\left(y_i - \overline{y}\right)}{\sqrt{\frac{1}{1-n}\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\frac{1}{1-n}\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}} \tag{10}$$

$$\mathrm{corr}\left(PC_{1\_red}, PC_{2\_red}\right) = 0{,}332 \tag{11}$$

$$\mathrm{corr}\left(PC_{1\_green}, PC_{2\_green}\right) = -0{,}382 \tag{12}$$

On the basis of the two correlation values for green and red objects, it can be said that the two groups are partially separated because the correlation values are nearly the same according to their amounts but have different signs. Thus, the two groups show in the opposite direction. In order to show the direction in which the eigenvectors or the features show, the correlation coefficient of the first two eigenvectors has to be determined. Table 4 shows the resulting two first eigenvectors V_1 and V_2, as coordinates of the chosen features, with which the correlation can be calculated in (13). It can thus be proved that the features M1, M6, M7, M8, M9, M10, M24 and M25 are in the same direction as the red cloud. This computational method after the PCA implementation, which is based on the calculation of correlation coefficients of the PCs, can be automated. In doing so, other methods can be additionally integrated, e.g. a calculation of the magnitudes of the eigenvectors with the aid of Pythagoras. This can be extended to more than two PCs in order to cover as much data as possible, thus minimizing the loss of information. Graphical representation will further confirm the calculations and the interpretation. In the next chapter, another multidimensional analysis for data separation is explained. The mathematical background of the LDA and the analysis method is explained using the same application example or database. In addition, it is shown how more accurate LDA results can be generated by means of a prior PCA execution and generally a better separation of the data is made possible.

Table 4. Coordinate of the chosen features

feature	V_1	V_2
M1	4,285	1,061
M6	3,979	2,517
M7	4,230	-1,937
M8	4,285	1,061
M9	4,285	1,061
M10	3,097	5,978
M24	2,925	-6,531
M25	3,211	-4.949

$$\mathrm{corr}(V_1, V_2) = 0{,}3023 \tag{13}$$

3. LINEAR DISCRIMINANT ANALYSIS

3.1. Definition

The LDA is a multivariate method for the analysis of groups or class differences, with which it is possible to examine and analyse groups with consideration of several variables (features). In principle, several variables are combined to one variable by a discriminant function (separation function) through linear combination under minimal loss of information. R. A. Fischer first described the discriminant analysis in 1936 in "The use of multiple measurements in taxonomic problems"[12]. Nowadays this method of analysis is used in fields such as image processing [13] and pattern recognition [14] [15] and serves as a classifier and method for dimensional reduction.

3.2. Formulation of the discriminant function

Since the LDA is already well established in today's technology and is already actively used, we restrict ourselves to the required steps and their corresponding most important equations to understand the method better and to set up the flow chart (analogous to PCA). An appropriate function for optimal group separation has to be determined. In the further progress of this paper, we limit our consideration to two groups, analogous to PCA. In Figure 9, the frequency distributions of two groups A (triangle) and B (circles) were each projected onto an x_1 and x_2-axis. Relatively large intersection regions are visible in which the values are assigned to group A or group B, i.e. the values in the overlap range can not be unambiguously assigned to either group. The two axes x_1 and x_2 are not suitable as a separation function. In Figure 10 larger overlapping areas are recognizable for the functions y* and y** as discriminant axes and thus no clean separation of the two groups is possible.

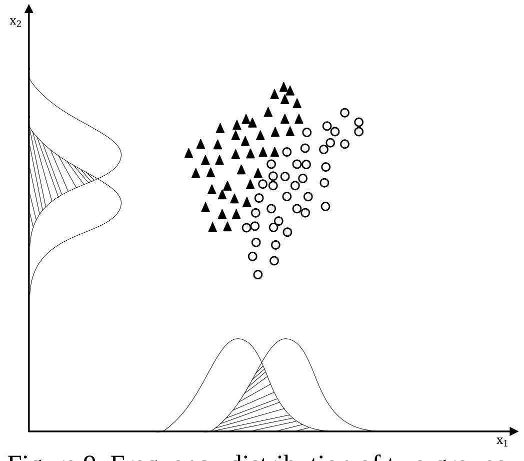


Figure 9. Frequency distribution of two groups. Separation by the variables x1 and x2. Changed according to [16] page 319

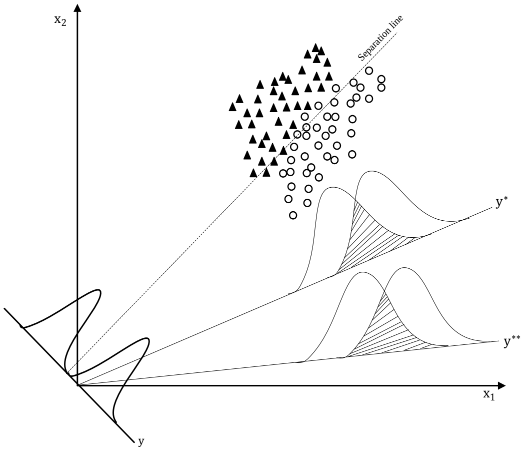


Figure 10. Separation by different discriminant axes y, y* and y**. Changed according to [16] page 321

However, the function y does not have any overlapping areas of the frequency distributions, which we use to draw in a separation line, which separates the groups optimally. The desired discriminant function, here y, can now be expressed as the linear function of the two variables of features x_1 and x_2 through (14) [16].

$$y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 \tag{14}$$

Generalized for m-features, the function equation is as in (15).

$$y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_m \cdot x_m \tag{15}$$

The equation corresponds to a linear function with the discriminant variable y and the variables of features x_j for j = 1, 2, ..., m. The coefficients a_1, a_2 to a_m are termed as discriminant coefficients and forming together the discriminant coefficient vector a with $a^T = (a_1, a_2, ..., a_m)$. a_0 is called constant term. The discriminant function is thus a mapping from the m-dimensional space of features into the one-dimensional space of the discriminant variables. In order to obtain the canonical discriminant function, the unknown coefficients a_j must be estimated in the first step of the LDA in a way of maximisation of the discriminant degree (detailed consideration in section 3.3.1). Then an estimator for the constant term a_0 must be determined under condition: $\bar{y} = 0$ [17]. In this case, $\bar{y}$can be calculated using (16).

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=0}^{n}(a_0+a_1 \cdot x_{1i}+a_2 \cdot x_{2i}+\ldots+a_m \cdot x_{mi})$$

$$\bar{y} = \frac{1}{n} n \cdot a_0 + \frac{1}{n}\sum_{i=1}^{n} a_1 \cdot x_{1i} + \frac{1}{n}\sum_{i=1}^{n} a_2 \cdot x_{2i} + \ldots + \frac{1}{n}\sum_{i=1}^{n} a_m \cdot x_{mi}$$

$$\bar{y} = a_0 + a_1 \frac{1}{n}\sum_{i=1}^{n} x_{1i} + a_2 \frac{1}{n}\sum_{i=1}^{n} x_{2i} + \ldots + a_m \frac{1}{n}\sum_{i=1}^{n} x_{mi} = a_0 + a^T \cdot \overline{X} \tag{16}$$

From (16) and with the aid of the condition ($\bar{y} = 0$) it follows that the constant term $a_0 = -a^T \cdot \overline{X}$. Here, $\overline{X}$ is the mean value vector of both groups, which together consist of n objects.

3.3. Estimation of the discriminant coefficients

Good separation results in the smallest possible overlapping range of the frequency distributions on the discriminant axis. The following discriminant criterion test size (TS) applies as a measure of the distinguishability of groups [17].

$$TS = \frac{MQSA}{MQSE} = \frac{\frac{1}{G-1} QSA}{\frac{1}{n-G} QSE} \rightarrow MAX \tag{17}$$

QSA describes the mean variation between the groups expressed as square deviation of the mean values from the total mean value of the groups (18). QSE stands for the mean variation within the groups expressed as total square deviation from the mean values in the groups (18).

$$QSA = \sum_{g=1}^{G} n_g \cdot (\bar{y}_g - \bar{y})^2 \quad \text{and} \quad QSE = \sum_{g=1}^{G}\sum_{i=1}^{n_g}(y_{gi} - \bar{y}_g)^2 \tag{18}$$

Thus, MQSA and MQSE in (17) represent the middle sum of squares. In our case, it is assumed that G = 2, thus two groups are separated by the discriminant function y. Both groups consist of $n = n_1 + n_2$ samples (objects) and the value y is the discriminant function value. By maximising (17), or respectively by maximising the distance between the groups (QSA) and minimizing the distances within the groups (QSE), the conditions for a minimum overlap range are fulfilled.

3.3.1 Mathematical derivation of the discriminant function

Let two independent groups A and B be given with n_A and n_B samples, to which a number m of features belong. The matrices $X_A = (X_{A1}, X_{A2}, ..., X_{Am})$ and $X_B = (X_{B1}, X_{B2}, ..., X_{Bm})$ with $X_{Aj}= (x_{A1j}, x_{A2j}, ..., x_{Anj})^T$ and $X_{Bj} = (x_{B1j}, x_{B2j} , ..., x_{Bnj})^T$ for j = 1, 2, ..., m, describe the corresponding groups. The arithmetic mean value vectors $\overline{X}_A = (\overline{X}_{A1}, \overline{X}_{A2}, ..., \overline{X}_{Am})^T$ and $\overline{X}_B = (\overline{X}_{B1}, \overline{X}_{B2}, ..., \overline{X}_{Bm})^T$ are obtained. The mean values $\overline{X}_A$and $\overline{X}_B$ can be determined according to (19).

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{19}$$

We now seek a discriminant function y or respectively a discriminant value vector Y in the form of (23), which optimally separates the two groups. Let us consider here, that $a_0 = 0$.

$$Y = X\cdot a = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_A 1} & x_{n_A 2} & \cdots & x_{n_A m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} a_1x_{11}+a_2x_{12}+\ldots + a_m x_{1m} \\ a_1x_{21}+a_2x_{22}+\ldots + a_m x_{2m} \\ a_1x_{31}+a_2x_{32}+\ldots + a_m x_{3m} \\ a_1x_{41}+a_2x_{42}+\ldots + a_m x_{4m} \\ \vdots \\ a_1x_{n1}+a_2x_{n2}+\ldots + a_m x_{nm} \end{pmatrix}$$

$$Y = a_1\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} + a_2\begin{pmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{pmatrix} + \ldots + a_m\begin{pmatrix} x_{1m} \\ x_{2m} \\ \vdots \\ x_{nm} \end{pmatrix} = a\cdot X_1 + a_2\cdot X_2 + \ldots + a_m\cdot X_m \tag{20}$$

Here, a_j are the discriminant coefficients to be estimated and X is a (n x m)-matrix with $n = n_A + n_B$, which includes the matrices X_A and X_B, i.e. $X = (X_A \,;\, X_B)$. The discriminant coefficients vector a points to the origin of the coordinate system with $a^T = (a_1, a_2, ..., a_m)$. By multiplying the matrix X with the vector a, Y is a vector, containing discriminant values. The previous explanation for two features x_1 and $x_2$are visualised in Figure 11.

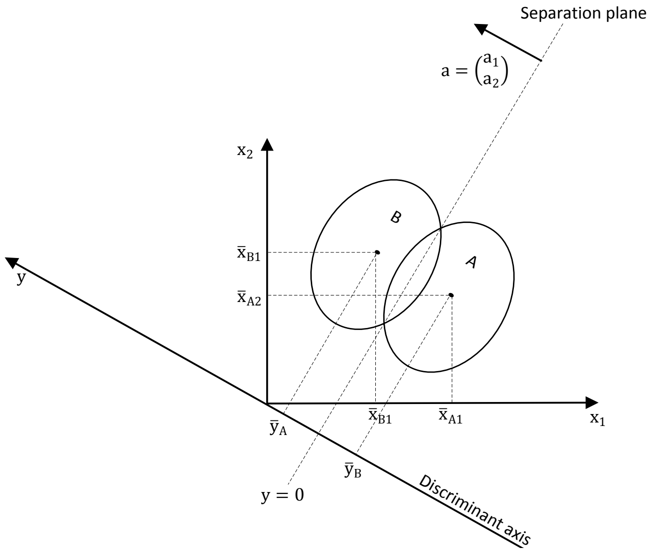


Figure 11. Optimal separation of the groups A and B by the discriminant function

Since the LDA is already well-established in today's technology and mathematical derivations are not a priority in this paper, we skip derivation steps for generating an equation for TS for two groups A and B at this point. The most important derivation steps are shown in (21) and (22) as extract from the total derivation procedure and an intermediate result of TS as a function of discriminant coefficient vector a is shown in (23).

$$Y = \begin{bmatrix} Y_A \\ Y_B \end{bmatrix} \; ; \; \overline{Y} = \frac{n_A \cdot \overline{Y}_A + n_B \cdot \overline{Y}_B}{n_A + n_B} \qquad (21)$$

$$\mathrm{var}(Y) = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \overline{y})^2 = \mathrm{var}(X \cdot a) = a^T \cdot \mathrm{cov}(X) \cdot a \qquad (22)$$

$$TS(a) = \frac{(\overline{y}_A - \overline{y}_B)^2 \left(\frac{n_A \cdot n_B}{n_A + n_B}\right)}{\frac{1}{n_A + n_B - 2}\left((n_A - 1) \cdot a^T \cdot S_A \cdot a + (n_B - 1) \cdot a^T \cdot S_B \cdot a\right)} \qquad (23)$$

In (22) "cov" stands for mathematical expression covariance. S_A and S_B are the covariance matrices of X_A and X_B which can be calculated with (24).

$$S_A = \frac{1}{m-1} \cdot (X_A - \overline{X}_A)^T \cdot (X_A - \overline{X}_A) \quad \text{and} \quad S_B = \frac{1}{m-1} \cdot (X_B - \overline{X}_B)^T \cdot (X_B - \overline{X}_B) \qquad (24)$$

The next step is to simplify (23) by using an abbreviation. Let S be an auxiliary variable called total covariance matrix (25). This simplifies function TS(a) according to (26).

$$S = \frac{S_A(n_A - 1) + S_B(n_B - 1)}{n_A + n_B - 2} \qquad (25)$$

$$TS(a) = \frac{(\overline{y}_A - \overline{y}_B)^2}{a^T \cdot S \cdot a} \left(\frac{n_A \cdot n_B}{n_A + n_B}\right) \qquad (26)$$

Since S basically consists of the sum of two covariance matrices S_A and S_B, S is symmetric and square and can be calculated as a multiplication of $S^{1/2}$ with $S^{1/2}$. Moreover, the constant $n_A \cdot n_B/(n_A + n_B)$ can be omitted since it plays no role in maximising the target TS (a). In addition, the numerator in (26) is replaced by (27) to obtain the expression for TS (a) in (28). The product $S^{1/2}$ $S^{-1/2}$ corresponds to the unit matrix and can thus be inserted in (28) without changing it. It is inserted in the numerator twice at left and right and we obtain (29).

$$\begin{aligned}(\overline{y}_A - \overline{y}_B)^2 &= (a_0 + \overline{X}_A \cdot a - a_0 - \overline{X}_B \cdot a)^2 = (\overline{X}_A \cdot a - \overline{X}_B \cdot a)^2 \\ &= (\overline{X}_A \cdot a - \overline{X}_B \cdot a)^T \cdot (\overline{X}_A \cdot a - \overline{X}_B \cdot a) \\ &= ((\overline{X}_A - \overline{X}_B) \cdot a)^T \cdot ((\overline{X}_A - \overline{X}_B) \cdot a) \\ &= a^T \cdot (\overline{X}_A - \overline{X}_B)^T (\overline{X}_A - \overline{X}_B) \cdot a \qquad (27)\end{aligned}$$

$$S(a) = \frac{a^T \cdot (\overline{X}_A - \overline{X}_B)^T \cdot (\overline{X}_A - \overline{X}_B) \cdot a}{a^T \cdot S^{1/2} \cdot S^{1/2} \cdot a} \qquad (28)$$

$$TS(a) = \frac{a^T \cdot S^{1/2} \cdot S^{-1/2} \cdot (\overline{X}_A - \overline{X}_B)^T \cdot (\overline{X}_A - \overline{X}_B) \cdot S^{-1/2} \cdot S^{1/2}}{a^T \cdot S^{1/2} \cdot S^{1/2} \cdot a} \qquad (29)$$

In the following step, a further simplification is carried out by the auxiliary vector d = $S^{1/2} \cdot a$. Since $S^{1/2}$ is a symmetric matrix, $(S^{1/2})^T = S^{1/2}$, $d^T = a^T \cdot S^{1/2}$ follow. This leads to (30) for TS as a function of d. The point is to maximise TS(d) by choosing a suitable vector h.

$$TS(d) = \frac{d^T \cdot S^{-1/2} \cdot (\overline{X}_A - \overline{X}_B)^T \cdot (\overline{X}_A - \overline{X}_B) \cdot S^{-1/2} \cdot d}{d^T \cdot d} \qquad (30)$$

The expression $d^T S^{-1/2} (\overline{X}_A - \overline{X}_B)^T$ is a scalar, whose square is decisive in TS(d). If this number and thus its square are to become maximal, the vector d must be selected parallel to $S^{-1/2} (\overline{X}_A - \overline{X}_B)$, which follows in (31). By multiplying with $S^{-1/2}$ twice we obtain the desired discriminant coefficient vector and thus the desired discriminant function (32) which separates the two groups.

$$d = S^{-1/2}\cdot(\overline{X}_A - \overline{X}_B)^T = S^{1/2}\cdot a \tag{31}$$

$$a = S^{-1}\cdot(\overline{X}_A - \overline{X}_B)^T \text{ and } Y = X\cdot a \tag{32}$$

The discriminant coefficients a_j can also be obtained, though more complex, directly by partial differentiation of TS with respect to vector a and through (33). "N" stands for numerator, "D" for denominator and the symbol "’" for the operation of derivation. By maximizing TS(a), through equating the derivation in (32) with zero, one obtains the determination equations, which are needed for calculation of the discriminant function. By substituting the values for X_A and X_B into these equations, the linear equation can be determined.

$$TS(a) = \frac{TS_N}{TS_D} \rightarrow MAX \Rightarrow \frac{dTS(a)}{da} = \frac{TS_N' \cdot TS_D - TS_D' \cdot TS_N}{TS_D^2} = 0 \tag{33}$$

However, the method discussed in this paper is, same as for PCA, based on a simplification using matrices and thus easier to derive and implement in a program code.

3.3.2. Flow chart of the LDA

In this chapter, the mathematical derivations in 0are summarised in a flow chart (Figure 12) in order to enable a program code for LDA algorithm as did for PCA. From the input of a database D ((n x m)-matrix) to the discriminant function y, the LDA proceeds, after the desired grouping into independent groups, in five steps: (Step 1) Calculation of the covariance matrices S_A and S_B of the groups A and B, (step 2) calculation of the self-defined total covariance matrix S, (step 3) calculation of the discriminant coefficient vector a and determination of the constant term a_0 and (step 4) subsequent representation of the discriminant values vector Y. Analogously to the PCA, the LDA is to be represented in the following section by means of an example and the results analysed. For this purpose, the same database and the same example as in the PCA are used.

3.4. Assignment of a test vector

This section demonstrates how an object or a test vector V, containing the coordinates of an object, can be assigned to a group with the aid of LDA. Let V be a test vector with $V^T = (v_1, v_2, ..., v_m)$ containing new m-measured values i.e. features, V is assigned to the group A or B, to which the distance from

$$\underbrace{(\overline{X}_A - \overline{X}_B)\cdot S^{-1}}_{a^T} \cdot V \text{ to } \underbrace{(\overline{X}_A - \overline{X}_B)\cdot S^{-1}}_{a^T} \cdot \overline{X}_A \tag{34}$$

or the distance from

$$\underbrace{(\overline{X}_A - \overline{X}_B)\cdot S^{-1}}_{a^T} \cdot V \text{ to } \underbrace{(\overline{X}_A - \overline{X}_B)\cdot S^{-1}\cdot}_{a^T} \overline{X}_B \tag{35}$$

is smaller. Figure 13 illustrates this principle. (36) and (37) apply for the distances d_A and d_B in Figure 13 with $d_A, d_B > 0$.

$$d_A = -a_0 - a^T\cdot\overline{X}_A^T-(- a_0 - a^T\cdot V) = a^T\cdot V - a^T\cdot\overline{X}_A^T \tag{36}$$

$$d_B = a_0 + a^T\cdot\overline{X}_B^T+(-(a_0 + a^T\cdot V)) = a^T\cdot\overline{X}_B^T - a^T\cdot V \tag{37}$$

In general, for a vector V the multiplication $a^T\cdot V < 0$, if V is in the direction of a, otherwise $a^T\cdot V > 0$. The distances d_A and d_B are always assumed positive. According to Figure 13, it can be said that the test vector V is closer to group A than group B in the ratio d_B / d_A. Thus, the probabilities P_A and P_B in (38) can be calculated as the affiliation of the test vector to group A and group B.

$$P_A = \frac{d_B}{d_A + d_B} \qquad P_B = \frac{d_A}{d_A + d_B} \tag{38}$$

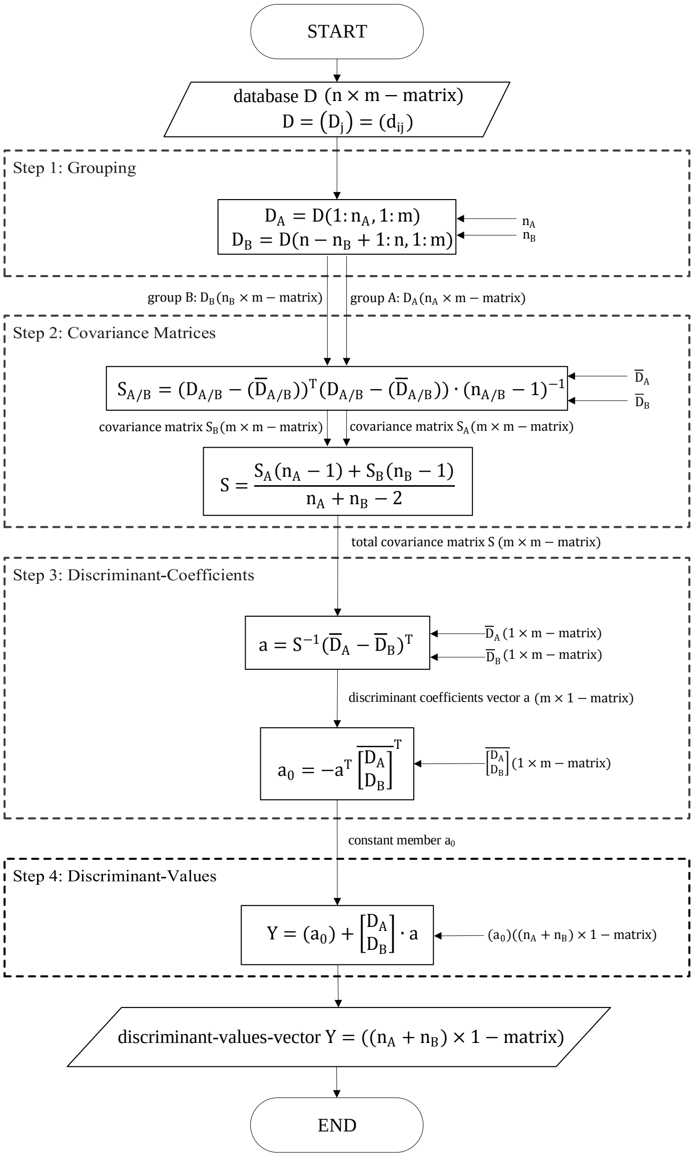

START
database D $(n \times m - \text{matrix})$
$D = (D_j) = (d_{ij})$
Step 1: Grouping
$D_A = D(1{:}\,n_A, 1{:}\,m)$
$D_B = D(n - n_B + 1{:}\,n, 1{:}\,m)$
n_A
n_B
group B: $D_B(n_B \times m - \text{matrix})$
group A: $D_A(n_A \times m - \text{matrix})$
Step 2: Covariance Matrices
$S_{A/B} = (D_{A/B} - (\overline{D}_{A/B}))^T (D_{A/B} - (\overline{D}_{A/B})) \cdot (n_{A/B} - 1)^{-1}$
$\overline{D}_A$
$\overline{D}_B$
covariance matrix $S_B(m \times m - \text{matrix})$
covariance matrix $S_A(m \times m - \text{matrix})$
$S = \frac{S_A(n_A - 1) + S_B(n_B - 1)}{n_A + n_B - 2}$
total covariance matrix S $(m \times m - \text{matrix})$
Step 3: Discriminant-Coefficients
$a = S^{-1}(\overline{D}_A - \overline{D}_B)^T$
$\overline{D}_A\,(1 \times m - \text{matrix})$
$\overline{D}_B\,(1 \times m - \text{matrix})$
discriminant coefficients vector a $(m \times 1 - \text{matrix})$
$a_0 = -a^T \overline{\begin{bmatrix} D_A \\ D_B \end{bmatrix}}^T$
$\overline{\begin{bmatrix} D_A \\ D_B \end{bmatrix}}\,(1 \times m - \text{matrix})$
constant member a_0
Step 4: Discriminant-Values
$Y = (a_0) + \begin{bmatrix} D_A \\ D_B \end{bmatrix} \cdot a$
$(a_0)((n_A + n_B) \times 1 - \text{matrix})$
discriminant-values-vector $Y = ((n_A + n_B) \times 1 - \text{matrix})$
END

Figure 12. Flow chart of the LDA

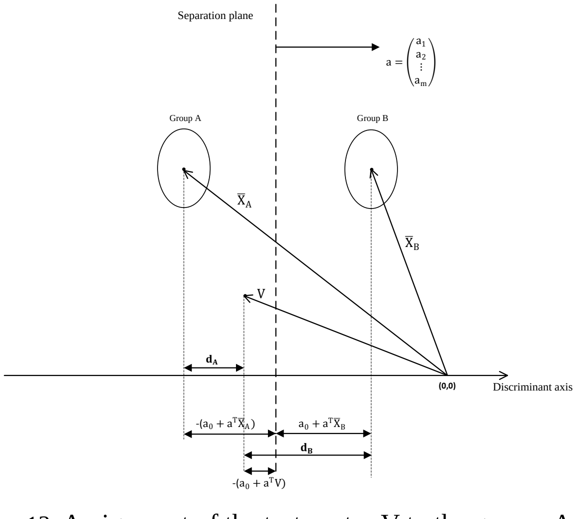

Separation plane
$a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$
Group A
Group B
$\overline{X}_A$
$\overline{X}_B$
V
d_A
(0,0)
Discriminant axis
$-(a_0 + a^T\overline{X}_A)$
$a_0 + a^T\overline{X}_B$
d_B
$-(a_0 + a^T V)$

Figure 13. Assignment of the test vector V to the groups A or B

3.5. Application example

As in the PCA (Figure 2), we consider the same database with about 2300 samples (objects) and 68 features to be examined through LDA. After performing the LDA by means of a MATLAB code using the five steps from the flow chart in Figure 12, m = 68 discriminant coefficients and a single constant term for 68 features are obtained. Now the LDA is tested for two randomly chosen features out of 68 for the 900 objects. These objects consist of 450 samples of group A (red) and 450 samples of group B (green) and are shown in Figure 14

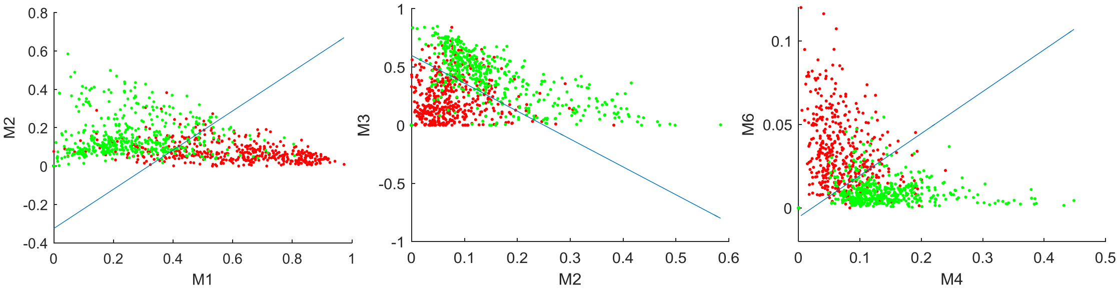


Figure 14. Separation of early and late failed for two random features

The obtained discriminant coefficients and the constant term a_0 were used with (14) to obtain the separation lines. These lines are drawn in blue in the individual plots in Figure 14, thus enabling separation of the objects. For each of the three plots shown below, the separation lines appear to effect an optimal separation of the objects. Despite some overlapping objects, the distribution of the red and green samples with respect to the separation lines is unambiguous. Considering three instead of two features, the separation line appears now as separation plane as shown in Figure 15. The display format of this representation in Figure 15 is not suitable for the exact visualisation of the separation plane. By selecting a suitable viewing direction by rotating the graph, the position of the separation plane becomes clear (Figure 16).

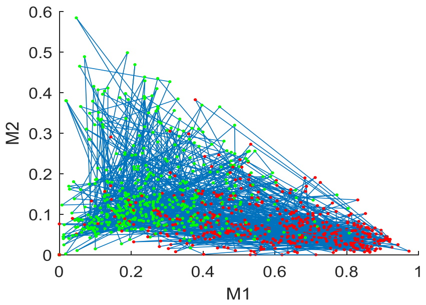


Figure 15. Separation of early and late failed for three random features

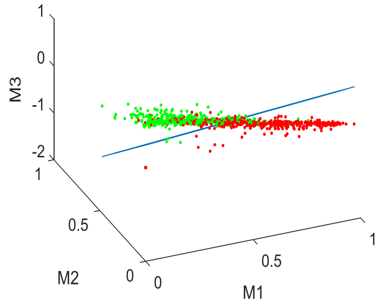


Figure 16. Rotation of Figure 15 (features M1, M2 and M3)

If we now consider all 68 features, 68 discriminant coefficients are obtained. If these coefficients are multiplied with each data set (object) i.e. with each row of the database, a vector with $n = n_A + n_B$ values is obtained. The resulting vector Y is, in spite of one column, a multidimensional variable, that means the information of 68 features were combined into one vector. A representation of the values of Y on a y-axis of a graph leads to 68 plots. The x-axis describes each individual feature and thus each point describes an object. From that, the following plots in Figure 17, Figure 18 and Figure 19 arise for $n_A = n_B = 450$. Figure 17 consists of 23 plots (M1 to M23), Figure 18 for features M24 to M46 and Figure 19 for the remaining features. Taking a closer look on e.g. the plots of M1, M6, M7, M8, M9 and M25, we can notice that much more red dots occur with higher feature values, compared to green dots, until almost no green dots are present. Otherwise, only green dots appear with higher feature values in e.g. the plots of M4, M56, M57, M59, M65 and M66, while approximately no red dots are found. The features listed here as examples also belonged to the relevant features of this database in the results of the PCA (Figure 7 and Figure 8).The following Figure 20 shows interesting plots that arise when the database is reduced on the basis of the relevant features, through prior PCA execution, and subsequently analysed by LDA.

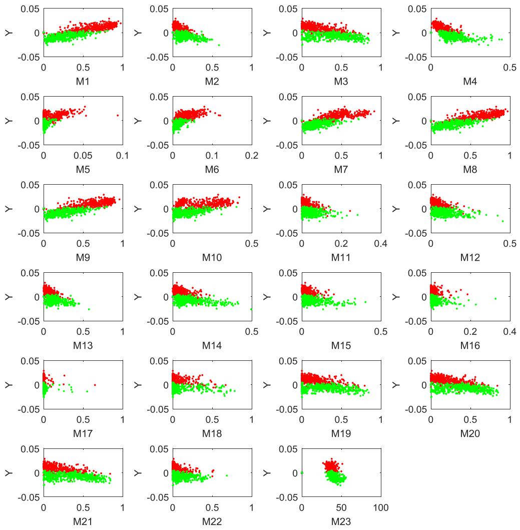


Figure 17. LDA for the first 23 features of the database (M1 to M23)

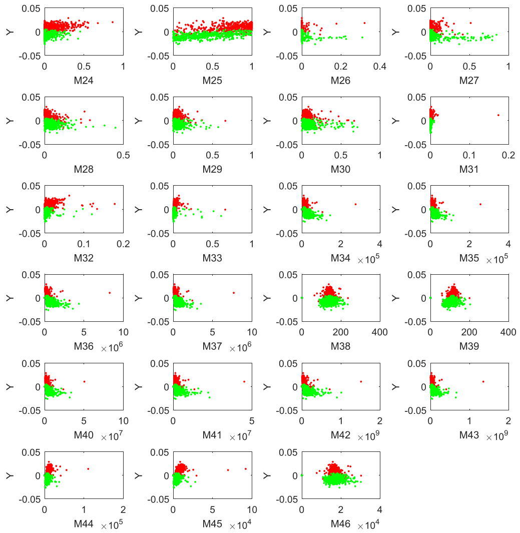


Figure 18. LDA for 23 features of the database (M24 to M46)

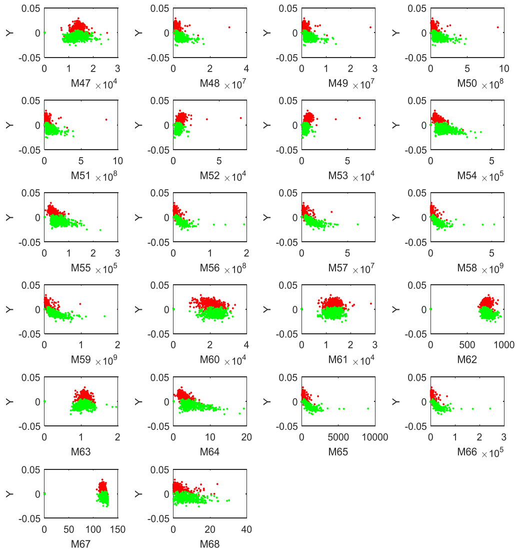


Figure 19. LDA for the remaining 22 features of the database (M47 to M68)

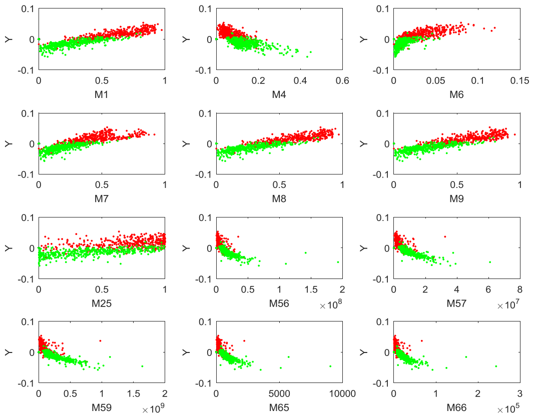
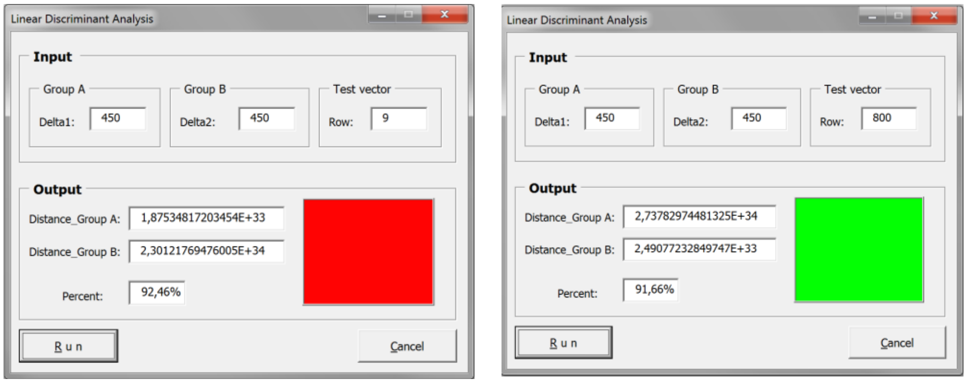


Figure 20. LDA after prior reduction of the database to relevant features by PCA

$$y_i = a_0 + a_1 \cdot M_{i1} + a_2 \cdot M_{i4} + a_3 \cdot M_{i6} + a_4 \cdot M_{i7} + a_5 \cdot M_{i8} + a_6 \cdot M_{i9} + a_7 \cdot M_{i25} + a_8 \cdot M_{i56} + a_9 \cdot M_{i57} + a_{10} \cdot M_{i59} + a_{11} \cdot M_{i65} + a_{12} \cdot M_{i66} \quad (39)$$

The projections of the data sets on the discriminant axis describe the multidimensional variables of the discriminant function y, which in this case is defined by (39) with i = 1, 2, ..., 900 (450 + 450). The discriminant value vector Y consists of m-y_i values, i.e. Y = (y_i), which are having a_k for k = 1, 2, …, 12 as weighting for M_j with j = 1, 4, 6, 7, 8, 9, 25, 56, 57, 59, 65, 66. The greater the magnitude of a_k, the more meaningful $M_{j.}$ It becomes clear that the green and red areas accumulate in specific areas of the plots. For example, the first plot to M1 in Figure 20 shows that it would be more reasonable to select the lowest possible value of feature M1 since one is here in the green cloud area, thus an early failure of the device might be avoidable. The same conclusion follows from the plots for M6, M7, M8, M9 and M25. In the plots of M4, M56, M57, M59, M65 and M66, the range of early failures lies within lower feature values. A closer look on M25 shows, that this feature is not as important as e.g. the features M1, M6, M7, M8 or M9 because there is no clear trend from the green to the red object cloud. A review of Figure 7 certifies that, where we pointed out that M25 does not correlate as strong as the other features with the red object cloud and therefore does not play an equally strong role in predicting early and late failure. These plots provide essential information about how each feature is associated with early and late failure. The here 12 features pointed out in (39) must be considered at the same time as well as certain limit values of the individual features have to be determined. Nevertheless, it must be assured that the required functionality of the product must remain unrestricted. The actual goal of the LDA is not just to separate two or more groups by a separation function but also to make predictions for a new test vector (new object). One wants to determine to which of the two groups the test vector belongs. This has already been discussed in 0. In addition, it is appealing to implement the LDA in *Visual Basic for Applications* (VBA, Microsoft). Analogously to MATLAB programming, all five steps of the LDA from Figure 12 were implemented into a VBA code and the discriminant coefficients were calculated. In order to generate more accurate results, the reduced database with the relevant features (M1, M4, M6, M7, M8, M9, M25, M56, M57, M59, M65 and M66) are considered. In addition, VBA allows a practical use of user forms for manual input. The implementation was automated as follows: The number of samples n_A for group A (early failure) was defined as input "Delta1" and nB for group B (late failure) as "Delta2". Finally, the test vector, as a row number of the database representing in our example a device number, is entered. Of course, new devices or respectively new data can be entered as new lines to the database. As an example, the values 450 for group A and group B are used. Line 9 is used as the test vector and the execution is confirmed with "Run". Figure 21 shows the output of the LDA through the VBA code. As a result, the distance values of the test vector to group A and group B calculated by (36) and (37), are obtained. If the distance of the test vector to group A is smaller than to group B, a viewport is colored in red. Otherwise, a green color appears. In addition, the assignment of the test vector to the groups is quantitatively evaluated with a percentage value, using (38) a. In Figure 21 it can be seen that line 9 belongs to group A with 92,46% regarding the distance behaviour because the data are sorted by their lifetime. As a second test example, we choose row 800 of 900 sorted data sets. Again, the result is logical, since sample number 800 belongs to group B with 91,66%.It is therefore useful to firstly reduce a database to the most relevant features through PCA and to obtain the most suitable separation or discriminant function through a subsequent LDA procedure. The more precisely the most relevant features can be filtered out, the better the separation of the objects by the discriminant coefficients, of which the number is also, as much as the relevant features, getting reduced. This allows new objects to be assigned very accurately to their group and thus the operating behaviour of the products can be predicted.



Group A Delta1: 450
Group B Delta2: 450
Test vector Row: 9
Distance_Group A: 1,87534817203454E+33
Distance_Group B: 2,30121769476005E+34
Percent: 92,46%

Group A Delta1: 450
Group B Delta2: 450
Test vector Row: 800
Distance_Group A: 2,73782974481325E+34
Distance_Group B: 2,49077232849747E+33
Percent: 91,66%

Figure 21. Result of sample sizes 450/450 and line numbers 9 and 800 as test vector

4. SUMMERY, DISCUSSION AND SUGGESTIONS

The PCA is a linear, multidimensional data analysis with an orthogonal transformation of the original variables into a new set of variables (PCs) in the m-dimensional space. These variables are then analysed in detail in order to further reduce the amount of data and to interpret the test result. The PCA is a simple and good method to gain insight into the structure of large amounts of data. In particular, valuable information about the correlation structure of variables is obtained.The results of the PCA are dependent on axis scaling and are subject to a subjective interpretation. Another problem with the PCA is that the PCs are difficult to interpret. That is why a further computational and automated analysis after the PCA is necessary, in order to read out the valuable information from the reduced database.

The LDA is a test method for analysing group differences using a classifier. Future observation can be classified by means of an assignment rule, i.e. the assignment to exactly one group. The predicted class of a new observation is determined by its location in space. The advantages of the LDA lie in its clear structure and interpretability. The discriminant coefficients can be determined very easily. Then only the determination of a linear separation function is necessary. Due to its flexibility and simplicity, LDA could be performed after an initial reduction of the database or removal of irrelevant features by a previous PCA, thus enabling a better separation. Interactions or even non-linear relationships of variables can, however, make separation of the groups more difficult. If the characteristics are overlapping, a clear separation through the discriminant function can be more difficult. In these cases, the LDA is not robust enough.

Nowadays, design and technology are increasingly reaching their limits, so the validation and verification methods, such as simulations, are not capable of processing all possible test scenarios [18]. Consequently, the cost of testing complex systems has increased for why test is becoming a dominant factor in overall production cost [19]. This is why it is useful to use a method as support to keep the enormous test costs as low as possible while still covering all scenarios during testing. A system usually consists of subsystems, which are usually created in different departments of a company. There is very little information about their dependency and correlation among one another, if the system should be testes as a whole. The methods PCA, and/ or LDA discussed here could be used to extract valuable information from the database of a system in order to describe the correlations of the subsystems more closely and finally to formulate a meaningful dependency model. The larger the database, the more valuable information could be contained. This is achieved by inserting information as e.g. developer's know-how, production information and all test results of e.g. simulations and others into the database. The most valuable are additional information about the user behavior (user variable), which are stored by chips integrated into the system and are available in the event of a system failure. The multidimensional database, known as Big Data, can be analysed by the methods presented here to gain new information about the system, to understand the system better and to allow a new angle of view into the functionality. By integrating the analysis procedures (PCA and LDA) into the system, the database is constantly updated and the results of the analysis procedures are adapted so that the system is trained (machine learning). Updating the database and adapting the analysis allows an adaptive, multidimensional analysis that is accurate enough to make clear statements about the fault diagnosis during testing and even to make accurate predictions about early failure. This approach of adaptive multidimensional analysis as well as machine learning [20] [21] [22] reduces test costs, enables us to understand the complex system even better and allows the realisation of a more reliable construction.

5. CONCLUSIONS AND OUTLOOK

In this paper, two test procedures were presented for the reduction, separation and classification of high-dimensional test data from electronic systems and thus for the early failures diagnosis of an electronic device. The PCA and LDA are multidimensional analysis methods and have the goal of classifying individual dimensions of the data sets according to their relevance, i.e. the deviation of the quantity in this dimension. We presented the test procedures individually and derived the important and necessary mathematical relationships. Flow diagrams of two algorithms were presented and used on an application example of a comprehensible test area. In this paper it was shown that the LDA works more effectively after a prior reduction of the database on relevant features by the PCA and consequently a more effective separation of the groups was enabled.

PCA and LDA are linear methods, so it is advantageous that the data points have a correlation and are in a linear relationship to each other and thus contain crucial information about the separation. But if there are no linear relationships in the database, it makes more sense to use nonlinear separation methods first, to completely separate the non-linearly separable data, and then PCA and LDA can be used. This improve the quality of the results more and more [23].

REFERENCES

[1] L. Zhao, Z. Chen, Y. Hu, G. Min and Z. Jiang, "Distributed Feature Selection for Efficient Economic Big Data Analysis," in IEEE Transactions on Big Data, vol. PP, no. 99, pp. 1-1, 2016.

[2] X. Bian, H. Krim, A. Bronstein and L. Dai, "Sparse null space basis pursuit and analysis dictionary learning for high-dimensional data analysis," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, 2015, pp. 3781-3785.

[3] K. Pearson, "On lines and planes of closest fit to systems of points in space," Philosophical Magazine, vol. 2, no. 6, pp. 559–572, 1901.

[4] H. Hotelling, "Analysis of a Complex of Statistical Variables Into Principal Components", Journal of Educational Psychology, vol. 24, pages 417-441 and 498-520, 1933.

[5] T. Zhang and B. Yang, "Big Data Dimension Reduction Using PCA," 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, 2016, pp. 152-157.

[6] Yuan Xie and Tao Zhang, "A fault diagnosis approach using SVM with data dimension reduction by PCA and LDA method," 2015 Chinese Automation Congress (CAC), Wuhan, 2015, pp. 869-874.

[7] D. p. Gao, Q. x. Zhu, X. j. Yang and C. r. Li, "A Novel Features Extracting Technique useing Location in Face Recognition," 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp. 585-588.

[8] A. W. Łuczyk, K. Neneman and W. A. Pleskacz, "Principal component analysis of accelerations in human dynamic movements: A sample set length effect study," 2017 MIXDES - 24th International Conference "Mixed Design of Integrated Circuits and Systems, Bydgoszcz, 2017, pp. 601-606.

[9] F. Bingqing, H. Shaolin, L. Chuan and M. Yangfan, "Anomaly detection of spacecraft attitude control system based on principal component analysis," 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, 2017, pp. 1220-1225.

[10] F. R. On, R. Jailani, S. L. Hassan and N. M. Tahir, "Analysis of sparse PCA using high dimensional data," 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), Malacca City, 2016, pp. 340-345.

[11] X. Zhao, Y. Lin and J. Heikkilä, "Dynamic texture recognition using multiscale PCA-learned filters," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 4152-4156.

[12] R.A. Fisher, "The Use of Multiple Measures in Taxonomic Problems," Ann. Eugenics, vol. 7, pp. 179-188, 1936.

[13] K. Papachristou, A. Tefas, I. Pitas, "Facial image analysis based on two-dimensional linear discriminant analysis exploiting symmetry", Proc. IEEE Int. Conf. Image Process. (IEEE ICIP 2015), pp. 3185-3189, Sep. 2015.

[14] M.H. Siddiqi, R. Ali, A. M. Khan, P. Young-Tack, L. Sungyoung, "Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields", IEEE Trans. Image Process., vol. 24, pp. 1386-1398, Apr. 2015.

[15] A. Iosifidis, A. Tefas, I. Pitas, "Merging linear discriminant analysis with Bag of Words model for human action recognition", Proc. IEEE Int. Conf. Image Process, pp. 832-836, Sep. 2015.

[16] Bahrenberger, Giese, Nipper, "Statistical methods in geography - Statistische Methoden in der Geographie", vol. 2, Multivariate Statistik, Teubner, Stuttgart, 1992.

[17] N. di Pietro, G. Zémor and J. J. Boutros, "LDA Lattices Without Dithering Achieve Capacity on the Gaussian Channel," in IEEE Transactions on Information Theory, no. 3, pp. 1561-1594, 2018.

[18] Xu, Y.: Algorithms for Automatic Generation of Relative Timing Constraints. Ph.D. dissertation, Salt Lake City, UT, USA. (2011).

[19] F. Poehl, F. Demmerle, J. Alt and H. Obermeir, "Production test challenges for highly integrated mobile phone SOCs — A case study," 2010 15th IEEE European Test Symposium, Praha, 2010, pp. 17-22.

[20] M. J. Barragan and G. Leger, "Efficient selection of signatures for analog/RF alternate test," 2013 18th IEEE European Test Symposium (ETS), Avignon, 2013, pp. 1-6.

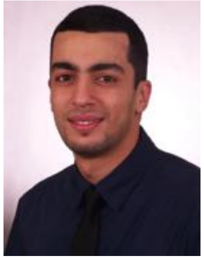
[21] E. Yilmaz, S. Ozev and K. M. Butler, "Adaptive multidimensional outlier analysis for analog and mixed signal circuits," 2011 IEEE International Test Conference, Anaheim, CA, 2011, pp. 1-8.

[22] H. G. Stratigopoulos, P. Drineas, M. Slamani and Y. Makris, "RF Specification Test Compaction Using Learning Machines," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 6, pp. 998-1002, June 2010.

[23] M. Denguir and S. M. Sattler, "A dimensionality-reduction method for test data," 2017 International Mixed Signals Testing Workshop (IMSTW), Thessaloniki, 2017, pp. 1-6.

Authors

Mohamed Denguir received his bachelor and master degree in Electrical engineering, Electronics and Information technology (EEI) from the University of Erlangen-Nuremberg (FAU) in 2013 and 2015. His specialization were microelectronic in his bachelor and automation technology (control engineering) in the master degree. He is a scientific assistant at the Institute for Reliable Circuits and Systems, University of Erlangen-Nuremberg since 2015. He has published five papers in the fields of: verification and testing of circuits and systems, structure-preserving modelling of circuits, fault diagnosis, test compaction and multidimensional analysis and separation of high-dimensional test results.



Naime Denguir received her master degree in Integrated Life Sciences-Biophysics, Biomathematics and Biology from the University of Erlangen-Nuremberg (FAU) in 2015 with a specialization in biological structures and processes and interactions of biological macromolecules. She is a research assistant at the Institute for Reliable Circuits and Systems, University of Erlangen-Nuremberg since 2016. Her fields of activities include multidimensional analysis, separation of high-dimensional test results and analysis procedures.

Sebastian Sattler received the Dipl.-Ing. and Dr.-Ing., both in Electrical engineering and Computer Science, from the Munich University of Technology in Germany, in 1989 and 1994, respectively. From 1996 to 2009, he was at Infineon Semiconductor AG (former Siemens Semi- conductor AG) as CAD/CAT Engineer, Manager for Analog Design for Test, and Manager for Acquisition of Public Funding in Testing. Since 2009 he is head of the Institute for Reliable Circuits and Systems at the University of Erlangen-Nuremberg, Germany.He has been engaged in research and development on Analog & Mixed- Signal Design for Test applications and techniques, integrated into SOC/SIP for communication and automotive systems. In these fields, he has published about 60 papers and formed more than 30 patents. In 2005, he received the German EDA Achievement Award.His fields of activities include verification, production testing and life time diagnosis for analog, mixed-signal and digital circuits and systems.