Revista Brasileira de Computação Aplicada, July, 2023

DOI: 10.5335/rbca.v15i2.14154
Vol. 15, Nº 2, pp. 51–59
Homepage: seer.upf.br/index.php/rbca/index

ORIGINAL PAPER

Transfer learning applied to bivariate forecasting on product warranty data

Joel Machado Pires[1,3], William Torelli[2], Luciana Escobar[1,3]

[1,3]IEL/BA - Instituto Euvaldo Lodi, [1]UFBA - Universidade Federal Da Bahia

*joelpires70@gmail.com; wttorelli@gmail.com; luescobar8@gmail.com

Received: 2022-11-29. **Revised:** 2023-06-14. **Accepted:** 2023-07-11.

Abstract

The reliability and resource management of products for warranty is important. Furthermore, the number of failures of a product over time of use and level of expenditure can assume different distributions. Approaches with parametric models bring good results when there is a normal distribution, and the application of Deep Learning (DL) is very promising. We show a new methodology for the application of DL models with transfer learning to bivariate forecasts of repair rates in products that are under warranty. The solution was applied to data from an American company, recorded from 2015 to 2022, of 12 different types of parts from 69 different types of cars. An evaluation of the absolute error of the forecasts was performed for each combination of part, car and model year. Tests showed that the model performed well in predicting data for 70 months in service and 70,000 miles, using data from cars with at least 15 months in service and 1,000 miles as input. It was also concluded that the solution is robust for cases of incomplete data and distributions far from the normal distribution.

Keywords: Forecast of repair rates; machine learning; reliability; warranty data analysis.

Resumo

A confiabilidade e o gerenciamento de recursos de produtos com garantia são importantes. Além disso, o número de falhas de um produto ao longo do tempo de uso e nível de gastos pode assumir diferentes distribuições. Abordagens com modelos paramétricos trazem bons resultados quando há uma distribuição normal, e a aplicação de Aprendizado Profundo (Deep Learning - DL) é muito promissora. Apresentamos uma nova metodologia para a aplicação de modelos de DL com aprendizado transferido para previsões bivariadas das taxas de reparo em produtos sob garantia. A solução foi aplicada a dados de uma empresa americana, registrados de 2015 a 2022, referentes a 12 tipos diferentes de peças de 69 tipos diferentes de carros. Uma avaliação do erro absoluto das previsões foi realizada para cada combinação de peça, carro e ano do modelo. Os testes mostraram que o modelo teve um bom desempenho na previsão de dados para 70 meses de serviço e 70.000 milhas, usando dados de carros com pelo menos 15 meses de serviço e 1.000 milhas como entrada. Concluiu-se também que a solução é robusta para casos de dados incompletos e distribuições distantes da distribuição normal.

Palavras-Chave: Análise de dados de garantia; aprendizado de máquina; confiabilidade; previsão de taxas de reparo.

1 Introduction

Customers value a good product warranty. However, they become dissatisfied and lose confidence when they have a problem, even in the face of efficient repairs or exchanges. Many product manufacturing companies need technological solutions for predicting the amount of products that will have some type of failure within a time interval before the event. This is because warranty and reliability costs are important when measuring a corporation's performance. They need active action even in production time, as the early identification of a failure trend in a set of products can make big differences in the budget (Lee et al., 2021; Wang et al., 2017; Wang and Xie,

2018). This active action can be preventive maintenance or even the release of a fix (He et al., 2018; Khoshkangini et al., 2020; Rai and Singh*, 2005; Wu, 2012).

In particular, there are technological solutions proposed in the literature for the prediction of warranty claims. Among them, we can mention solutions for forecasting based on time series models, such as the Markov, Box-Jenkins Model and Multi Layer Perceptron, which are promising in the theme. Other technical engagement work focused on predicting a specific failure for each product already sold and in use. In this case, the objective is to obtain a portion of the products that will present a specific failure, with a determined period of relatively short time in advance. With this, it is possible to act proactively to improve the user experience and assist in decision making. This technique can be used in the context of confidence assurance, but also has limitations with respect to the prediction time in advance (Xu et al., 2003). Assuming that the failures are related to the age and level of usage of the products, the exploration of solutions for bivariate forecasting becomes very important (Chehade et al., 2022; Gupta et al., 2014, 2017).

Among the problems to be faced in the construction of such solutions, there are the non-maturity of the data, failure rate depending on time in service and level of use, and high level of complexity for the ideal parameterization of each chosen model. Time-in-service failure rate curves change according to the emergence of new failures or new products being produced/sold over time. The phenomenon of non-maturity occurs when the curves do not represent the real failure rate for time in service and current usage level. In this context, although there are many promising techniques, some improvements can be made, such as applying them together with a simplified methodology.

On the other hand, DL techniques have great potential and are already being used in this type of application. In particular, neural networks are great generic approximants of functions, which can be the substitute of other parametrics models (Xu et al., 2003; Zainuddin and Pauline, 2008).In addition, new techniques were introduced to improve the performance of this tool when applied to problems where there are few examples observed, such as oversampling, data ensemble and transfer learning (Feng et al., 2019).

A bibliographic review was carried out on forecasting expenses with product warranty in general and applications of computing techniques in this problem, using the Google Scholar search as the main tool. The principal scope is in the study of an integrated solution that can be automated. Therefore, solutions with machine learning models received more attention than those with parametric models. The main contributions were separated, without discarding the solutions based on parametric models, so that a more integrated solution can be made, but with a reduced scope. These steps were taken to provide the necessary tools for modeling, testing and analyzing the results.

The article is divided as follows: Section 2 provides an overview of the problem in the current context, along with a discussion of previous work. Section 3 presents related works that have also contributed to the formulation of the proposed solution. Section 4 describes the modeling of the proposed solution, the dataset, applications, and tests conducted. In Section 5, the results are discussed. Finally, conclusions are drawn, and future steps for improving the work are outlined in Section 6.

2 Background

Some of the main ways of predicting auto warranty claims are forecasting based on past warranty claim data, and classifying the instantaneous state of each car being used by customers. Khoshkangini et al. (2020) made a comparison between these two methodologies, for this, they built machine learning models, one for linear regression and the other for classification, respectively. Linear regression provides the expected warranty claim rate value per time in service for a specific component of cars sold. However, this proposed solution has focus on exploring the applicability of data obtained from cars always connected and sending data to the cloud in real time.

Thinking of proposing a solution for three-dimensional analysis of warranty data, Gupta et al. (2014) determined the usage rate as the division between the usage level (m - mileage) and the time in service (t), as two independent variables, that is, with low correlation. Considering only the guarantee interval, the probability of failure in this interval is calculated. Where the observed fault density is given by Eq. (1).

$$p(t <= t_0, m <= m_0) = \frac{q}{N} \tag{1}$$

In Eq. (1), q is the cumulative number of failures and N is the number of products sold, for $t <= t_0$ and $m <= m_0$. Where t_0 is the limit for the warranty in the direction of time in service and m_0 is the limit for the warranty in the direction of the mileage axis, which mileage represents the level of use of the cars. Similarly, Xie et al. (2017) provided an approach for bivariate modeling of the number of warranty claims as a non-homogeneous Poisson process (NHPP) distribution. They considered that the distribution of warranty claims is related to both the level of use and the age of the products under study.

He et al. (2018) proposed a reliability model considering the engineering learning effect and use effect. The engineering learning effect concerns the possible inverse relationship of component failure rate with the new version or model year released. The hypothesis is that the more recent the component version, the lower the failure rate, as observed in some datasets. The use effect concerns the possible relationship between the level of use of the cars and the failure rates. In this modeling, they assumed that the failure rate of the studied components follows a Weibull or Log-Normal distribution.

The phenomenon of data immaturity can be a big problem in most analysis for forecasting. To address this, Chehade et al. (2022) proposed a model based on the Conditional Gaussian Mixture Model (CGMM) that describes this distribution as a function of the time in use of a product. The idea is that there are components that have similar warranty claim distributions, despite many

of these components being distinct from one another. So, they proposed a clustering, which groups the historical warranty data. Thus, through an inference using CGMM, it is possible to obtain the behavior most similar to the distribution to be analyzed. However, it is necessary to ensure that the normalized behavior of the number of warranty claims is close to a normal distribution, through an observation on a data set.

On the other hand, Artificial Neural Networks (ANN) can be used in the context of forecasting as approximators of functions that describe the failure rate (Schmidt-Hieber, 2020). Zainuddin and Pauline (2008) compared different types of ANN models to approximate different curve shapes. The study was conducted based on the basic structure of an ANN with a hidden layer, varying the activation of this layer with the Mexican Hat, Gaussian Wavelet (WNN), Morlet and Gaussian (RBFN) functions. It is concluded that the activation function in the hidden layers of an ANN has a high weight in the behavior in the result (Yang et al., 2013).

In a comparison between different types of models for forecasting the reliability of automobile components, Lee et al. (2021) showed that, in the tests, ANNs performed better when compared to parametric models. In addition, it was possible to perceive that there are many distributions that are very distant from a normal distribution and that modeling manually may not bring good results and be costly.

These works allowed us to notice that the application of parametric models has promising results but requires a priori certification that the behavior of warranty claims follows a known distribution. In addition, it can be concluded that deep ANN has potential for application to forecast warranty claims, especially when the distribution of these data is unknown.

3 Related Works

Gupta et al. (2014) presented a model based on Weibull and Exponential Distribution, to describe the probability of failure as a function of time and level of use of products, algorithm from the nlm package of the R language to optimize it. The fitted model approximates the observed probability. Gupta et al. (2017) also presented some problems about bivariate forecasts, in a literature review. Dai et al. (2019) also showed a model for forecasting warranty claims based on the non-homogeneous Poisson process. They used synthetic data and data obtained from a Chinese car manufacturer to fit and validate the model. Chehade et al. (2022) optimized a model for clustering product repair distributions (CGMM) with the Expectation and Optimization algorithm to apply in a parametric model for forecasting. As validation, it was applied to estimate car warranty claims using data provided by Ford. This data is composed of information of 15,000 parts of 10 million vehicles, from model years from 2010 to 2013. Rai and Singh* (2005) explored the application of the type of ANN called radial basis function (RBFN) to forecast vehicle warranty costs. Lee et al. (2021) compared the application of different types of reliability forecast models for car components. The different types were Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Sequence To Sequence Model, Support Vector Machine, Decision Tree Ensamble, ARIMA and Weibull Distribution. RNN had the best performance while the Weibull parametric model had the worst, in tests with a database built with information from 147 car parts from 9 different model years. Mitra et al. (n.d.) presented a model similar to an ensemble for retail forecasting. It is a linear regression model that is fitted with data from the combination of inference from the Random Forest and XGBoost models. They used data from a company selling various products to compare the proposed solution with existing architectures, such as AdaBoost, XGBoost and Random Forest and RN, with the metrics Mean Squared Error, Mean Absolute Error and R2. The Table 1 shows a summary of the closest works found.

4 Proposed Solution

The proposed solution integrates some results found in the literature. This solution is a machine learning model that takes the learning obtained from historical data and merges it with the new observed data to make a bivariate forecast. For this, a model of Deep ANN is built with the different existing distributions in the database, called base model. These distributions are the data from past repairs, which occurred over the years up to a cut-off point; this cut-off point could be years or months behind the analysis date. From this base model, another model is adjusted, but retrieving the parameters of the base model (learning transfer), with the data available for adjustment of the distribution (current data) that one wants to make the prediction. Considering that all data that were used to adjust the base model are already matured, it is understood that this model describes the shape of past distributions without the phenomenon of immaturity. From there, the distribution of the base model closest to the current data is obtained and adjusted. The result of this adjustment is a sub-model that describes the behavior of the extrapolated current distribution.

4.1 Modeling

Consider the scenario in which you have a set of historical repair data for several components of a company with information about these components and the date of occurrence. We want to obtain a model that describes the repair rate (z) of a specific component as a function of time in service (t), in months, and level of use (m), in miles. The values of these two variables, t and m, represent the level of stress that such components or products suffered until the moment of repair. It is assumed that, a priori, the distribution of z is not known, and there exists a relationship between z and t and m. It is also assumed that a deep ANN model can be approximated to follow the z-distribution behavior.

The Deep ANN models can be used for regression without defining the behavior form beforehand. Thinking about the normalized distributions of products as a function of time of use and expenditure, it is assumed that there is at least one DNN model, for each one, that

Table 1: Related works

Authors	Objective	Tools	Bivariate	Consider non maturity	With transfer learning
Gupta et al. (2014)	Bivariate forecast of failure probability as a function of time and level of product use	Weibull	Yes	Yes	No
Gupta et al. (2017)	Bivariate forecast of product failure rate	Weibull	Yes	Yes	No
Dai et al. (2019)	Bivariate forecast of warranty claims	non-homogeneous Poisson process	Yes	Yes	No
Chehade et al. (2022)	Bivariate forecast of product repair	Normal Distribution, Gaussian Mixture Model	Yes	Yes	No
Rai and Singh* (2005)	Forecast of warranty costs over time	radial basis function	No	Yes	No
Lee et al. (2021)	Comparison of techniques for forecasting	several	No	Yes	No
Mitra et al. (n.d.)	Forecast of product retail	Linear Regression, Xg Boost, Random Forest	No	No	No

describes them in an interval. Consider $F_k(X)$, the model that describes the behavior of any distribution, d_k.

Consider a DNN, $F(X)$, which represents the sum of all $F_k's$, so it behaves like an F_k according to the input. In other words, it is considered that it is possible to find all models of DNN's that describe all distributions of the dataset and that there is another larger DNN that is the composition of all of them, observe the abstraction of this modeling in the Fig. 1 and Fig. 2.

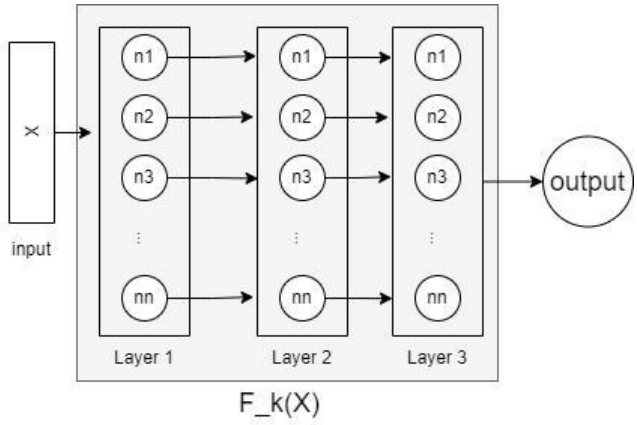

Figure 1: Abstract model for F_k and F for k DNN's of 3 layers and n neurons. F_k is a DNN. The arrows represent the data transition, where the base of the arrows represents the origin and the tip represents the destination

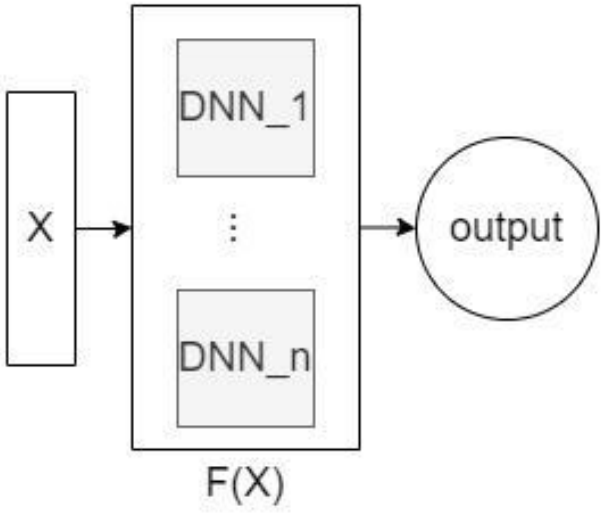

Figure 2: Abstraction of $F(x)$.

Similar to a model for classifying X, F(X) obtains the F_k that follows the closest distribution of X. Take X as a vector of the form

$X = [[m_1, t_1, g_1], \ldots, [ml, t_l, g_l]]$, to m_i and $t_i \in \mathbb{R}$

Where g is a value, in binary, that represents the identification of the distribution to which m and t belong. It is found by comparing the current distribution X with each distribution used to build F, through the Frobenius norm. The current distribution is the data available to forecast the product to be analyzed, which is a pair X, z.

Assuming that the historical database has d distributions, g is described as

$g = a_1, a_2, ..., a_j$, to $j = \frac{log(d)}{log(2)} = length(g), d \in \mathbb{N}$

Thus, $X = [m, t, a_1, a_2, ..., a_j]$. The macro view of the architecture of the proposed solution can be represented by Fig. 3.

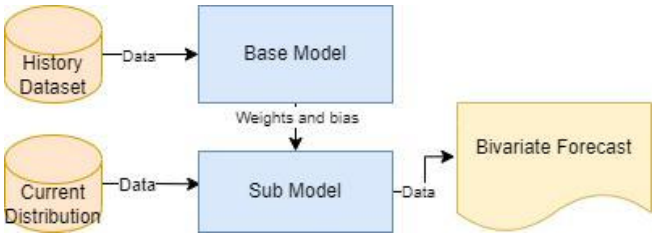

Figure 3: Macro architecture of the proposed system. The DNN's are in blue. Base Model is the representation of F

Therefore, the base model (F) has the function of describing the different forms of distributions and the submodel receives the transfer learning from F to describe the current distribution.

4.2 Dataset

The dataset consists of a historical distribution of car parts exchange records from model years 2015 to 2021, from an American car manufacturer. There were 69 different types of cars distributed among these model years. The database was filtered to only have records of 12 different types of car components (part number), manually selected by a person specializing in the company's warranty. Data from the model years from 2015 to 2017 were separated for the adjustment of the base model (History Data - D1), while the other remaining data were left for the adjustment and validation of the sub models (Current Data - D2).

The database obtained consists of two matrices, one with information on part number exchanges (W_{clm}) and another on the quantity produced (W_{prod}). W_{clm} has the following columns: vin, model_year, vehicle_line, part_number, tis, mileage. Whereas W_{prod} has the following columns: vin, model_year, vehicle_line. A unique combination of model_year, vehicle_line and part_number identifies a distribution, called a filter.

Thinking about the application of the modeling proposed in the previous section, D1 and D2 are sets of fault density distributions, for each filter, built with a Python algorithm using Equation 3. Also, a normalization was applied to the datasets so that each distribution has values between 0 and 1. 199 distributions were obtained for D1 and 266 for D2, thus, $D_1 = D_{f1}, D_{f2}, \ldots, D_{f199}$ and $D_2 = D_{f200}, D_{f201}, \ldots, D_{f465}$.

Each D_{fi} has a maximum of 4900 elements, because the warranty limit is defined for tis <= 70 and mileage <= 70,000 and z is defined for all elements of the cartesian product between *tis* and *mileage*.

4.3 Application and Tests

The proposed solution works in two stages, the first consists of building and testing the base model, and the second consists of adjusting and validating the sub-models. The base model architecture is shown in Fig. 4. This model was thought to be robust to non-linear distributions, to have good generalization power and to be able to learn the 199 different distributions present in the historical database.

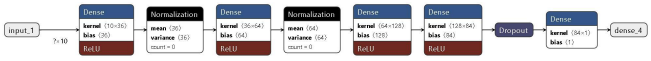

Figure 4: Base model architecture.

In this Fig. 4, Dense is an artificial neural network layer with 36 neurons fully connected to the input vector with 10 elements, with output activated by the ReLU (Rectified Linear Unit) function. The Normalization layer is responsible for dragging and normalizing the result of the previous layer by mean and variance (Ba et al., 2016). Dropout is a regularization layer to decrease overfitting and increase the model's generalization power (Srivastava et al., 2014). The last layer is activated by a linear function, $\phi(x) = x$.

The base model was fitted with dataset D_1. Noise was added to the D_1 distributions, with 2.5% of the maximum value, following a Gaussian distribution. Since the maximum value of all distributions belonging to D_1 is 1, the noise distribution has the configuration $\mu = 0.025$ and $\sigma = 0.025$, with $z_{noise} = z + noise$. The z vector was concatenated with the z_{noise}. This noise also assumes the role of regularization in the model.

The vector g for each distribution is defined by enumerating the database, from 0 to 198, therefore, 8 positions are needed to represent it. The model was adjusted using the Adam optimizer (Kingma and Ba, 2014), Mean Absolute Error (MAE) as an error computation and metric, with an early stop when reaching 20 epochs without error reduction and with a tolerance of up to 20 thousand epochs. The size of the data packet per epoch was 128 elements, randomized. To arrive at the architecture of Fig. 4, we started by adjusting the model with only 3 layers of 16 neurons. As the error was high, the number of neurons was increased and the adjustment process was performed again. This process of checking the error and adjusting the model architecture was repeated until a satisfactory level of error was reached. Thus, the adjusted model was obtained to describe curve shapes with an error close to 0.004.

The storage of this model is done together with the maximum value of each normalized distribution, the maximum value for *tis* and *mileage*, and the dataset D_1 cutted for *mileage* <= x_k and *tis* <= y_k, this last database is called M_{db}. This process was repeated to $x_k = 25,000$ and $y_k = 25$, and to $x_k = 15,000$ and $y_k = 15$. Datasets D_1 and M_{db} are illustrated in Fig. 5.

The second step is very similar to the solutions available in the literature. If, on the one hand, there are parametric models, which describe a specific behavior and take this

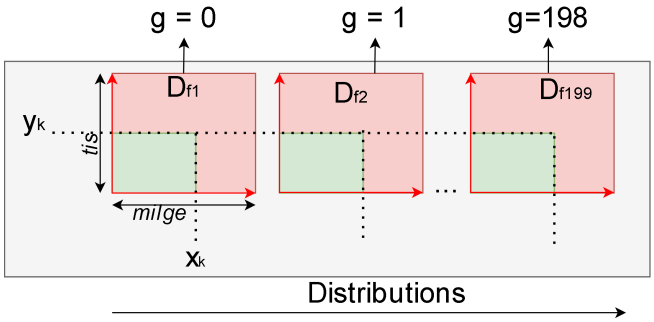

Figure 5: Description of dataset D_1 with 199 distributions. The part in green represents the data enumerated and copied to M_{db}.

behavior and adjust it to the experimental data, on the other hand, the base model provides different behaviors, such behaviors can also be adjusted to the experimental data.

For data in D2, there is the problem of non-maturity, which is when the data is not yet complete. For example, component data for 2021 model year cars that were sold in January are only complete for tis <= 2 when captured in March 2021, because all of these cars have only completed 2 months in service at that time. Knowing this, only the complete part of a sub model is taken for adjustment. Noise is also added to D_2 proportionally to the maximum value for each distribution. The value of g is defined by comparing M_{db} (green part in D_1) with the equivalent part in D_1, using the Frobenius norm. This step to find the value of g for distributions other than D_1 can be replaced by a clustering model such as K Means, Gaussian Mixture Model and DNN.

Each filter has a submodel that describes it. This submodel is obtained by adjusting the base model with a separate dataset for fitting. For each filter, D_1 was divided into training data, where *mileage* is less than or equal to x_c, and *tis* is less than or equal to y_c, and validation data, where *mileage* is greater than x_c and *tis* is greater than y_c. The fitting process was performed for a maximum of 1000 epochs, with a stopping condition of no reduction in absolute error for 30 consecutive epochs. The model with the lowest error was stored during this process. The batch size for training was 16 randomly selected elements. After this fitting process, a comparison was made between the model's inference and the observed distribution for each distribution.

5 Results and Discussions

Two variations were made for the adjustments and tests of the models that describe a forecast distribution. One with the dataset clipped at x_c = 20000 and y_c = 20, and the other at x_c = 1000 and y_c = 15. Table 2 shows a summary of the mean absolute error of the models in inferring each distribution separated for testing.

In this Table 2, the values for the models with cutoff at x_c = 20000 and y_c = 20 were slightly better than with the second cutoff because the first provides more data for the model to be fitted. Note that the errors were low, since the adopted metric is the sum of the absolute differences divided by the size of the vector, where each vector has between 1500 and 4900 elements. The "25%" column refers to the error value that up to 25% of the models reached, and so on for the "50%" and "75%" columns. Figs. 6 and 7 shows an example of an observed versus predicted distribution.

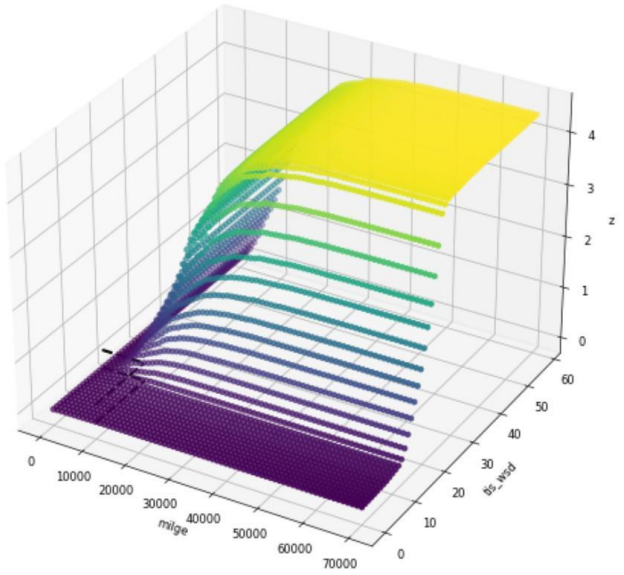

Figure 6: Observed distribution to a filter.

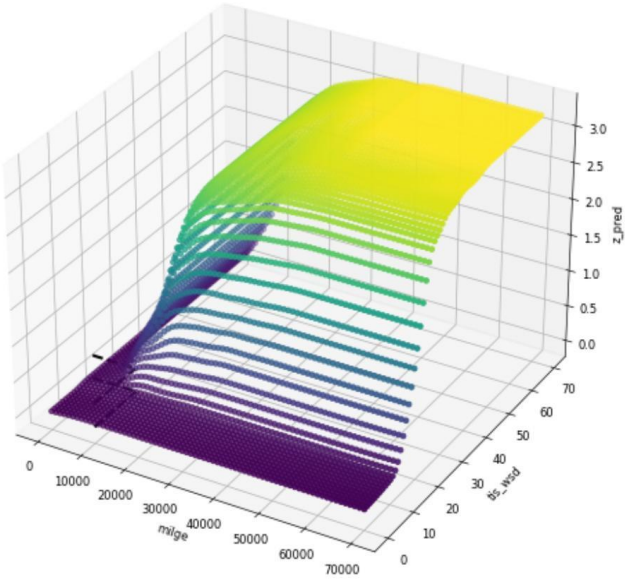

Figure 7: Description of dataset D_1 with 199 distributions. The part in green represents the data enumerated and copied to M_{db}.

Table 2: MAE for models fitted with data clipped in mileage <= xc and tis <= yc

	min	mean	25%	50%	75 %	max
x_c = 20, 000 and y_c = 20	0.034891	0.448511	0.127836	0.186426	0.376528	8.494953
x_c = 1000 and y_c = 15	0.070525	0.702727	0.266114	0.425506	0.688175	8.831922

To the both figures, the part number has been omitted for data confidentiality reasons. The dashed lines describe the portion of data clipped by x_c = 1000 and y_c = 15. It shows that the model follows a surface shape with values very close to the observed values, with MAE = 1.519368. Regarding axes, mileage = milge (mileages) and tis = tis_wsd (months), z-axis is in percentage, $z = (q/N) * 100$. Another example is shown in Figs. 8 and 9, this time providing more data for the model.

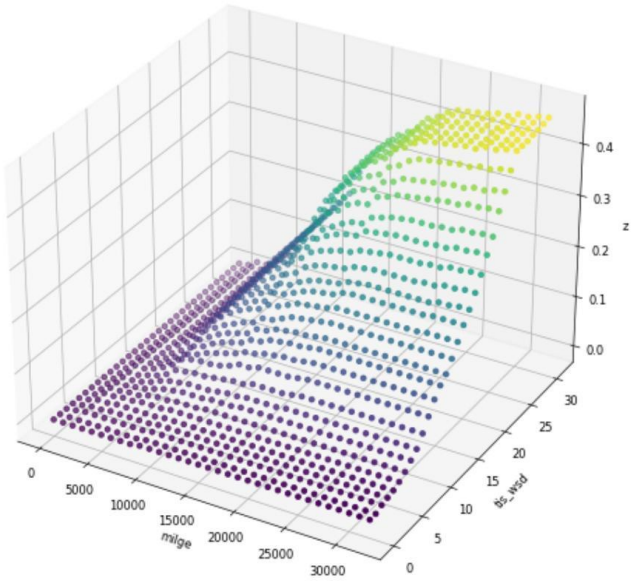

Figure 8: Description of dataset D_1 with 199 distributions. The part in green represents the data enumerated and copied to M_{db}.

A deeper analysis of the inner workings of the proposed model can be done later to understand why the model continued to follow the same distribution for mileage > xc and tis > yc. However, it was observed that the model returned a smoother surface as the regularization increased. On the other hand, with more regularization, we had greater error during the adjustment of the base model, so the model was increased so that it describes all the shapes present in the historical base but returns the smoothest possible surfaces. Thus, it is understood that an adjustment in the model's architecture may be necessary when new components with matured data appear to be included.

Nevertheless, the model may fail when we have a distribution that is very different from those learned by the base model. This can occur when the historical database is not robust or contains bias. Figs. 10 and 11 illustrates an

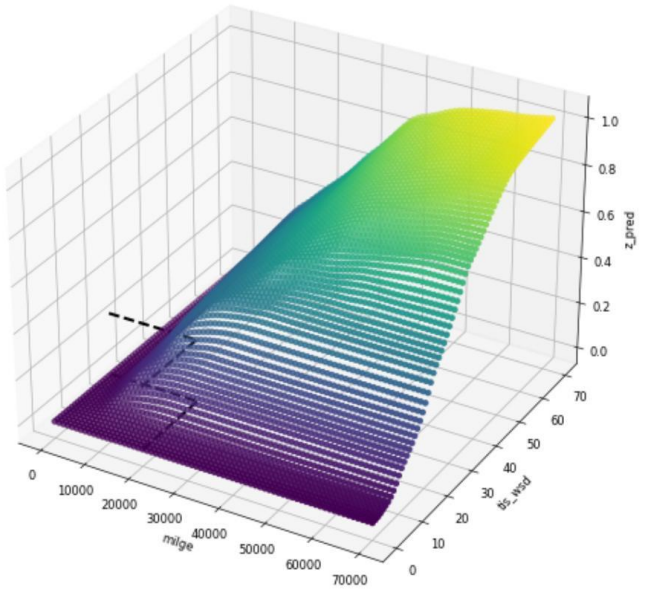

Figure 9: Description of dataset D_1 with 199 distributions. The part in green represents the data enumerated and copied to M_{db}.

unusual surface shape in the separate dataset for adjusting the base model, but which is very common in the separate dataset for forecasting, due to the large difference in years between the two bases. Ideally, the base model should be updated gradually over the years so that this does not occur, for study purposes, a large part of the most recent data was separated, so that validation could be carried out. That is, if the most recent dataset were used to fit the base model, there would be few copies to fit the submodels and perform validation.

It should be remembered that this forecast is based on past events, including the actions of the company's engineers when they encounter a problem with a part number or product model in general. Also, it is worth noting that the historical base used must be representative and its robustness can be increased with synthetic data inserted in it, such as the more common Weibull and Lognormal distributions.

6 Conclusion

An approach for bivariate forecasting using deep Artificial Neural Networks was shown, which can be easily automated. This methodology takes into account state-of-the-art techniques for assembling a model that forecasts failures of a part number, considering historical behaviors as a basis, through transfer learning. The base model

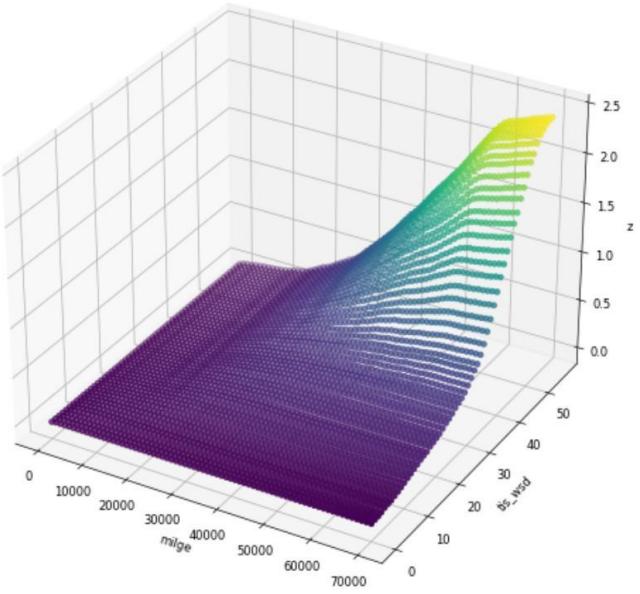

Figure 10: Distribution of a part number that takes longer to start to fail.

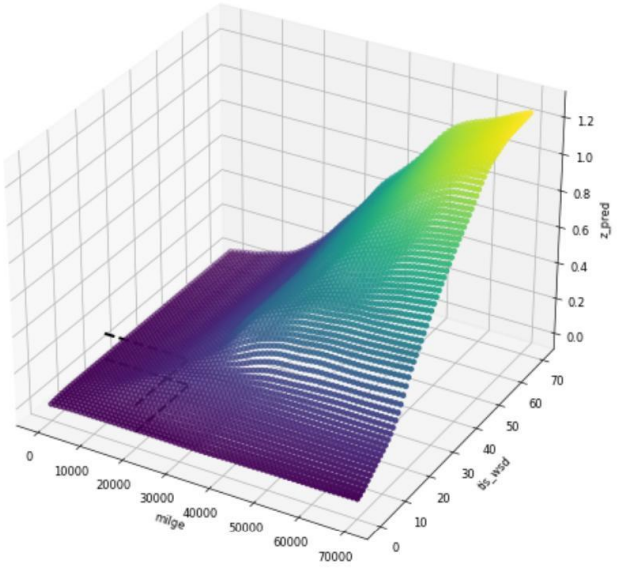

Figure 11: Prediction for the distribution described in Fig. 10

developed presented MAE close to 0.004 in the description of the 199 different dataset distributions. This model means the description of all failure behaviors that occurred with all part numbers considered. It can be gradually updated every time there is a new set of matured data of any part number. Was shown that it is possible that a DNN model can extrapolate surfaces keeping them smooth, through the tests done. As possible improvements, we can highlight the study of replacing the $M_d b$ base by a clustering model, which can be modeled separately or integrated as a layer in the base model itself, if it is a DNN model. Furthermore, the enumeration, g, can be done in an ordered way by, by some parameter, as the maximum value of the distributions, thus exploring the proximity relationship between them. A study focused on the inner workings of the DNN, applied to forecasting, can also be done to understand why it is possible that it continues to return values following the same distribution even outside the interval that was last adjusted. Therefore, this new approach demonstrates potential contribution in product reliability analysis and opens up a range of possibilities to be explored.

Acknowledgments

The opinions, hypotheses and conclusions or recommendations expressed in this material are the responsibility of the author(s) and do not necessarily reflect the vision of IEL/BA.

References

Ba, J. L., Kiros, J. R. and Hinton, G. E. (2016). Layer normalization. https://doi.org/10.48550/arXiv.1607.06450.

Chehade, A., Savargaonkar, M. and Krivtsov, V. (2022). Conditional gaussian mixture model for warranty claims forecasting, **218**: 108180. https://doi.org/10.1016/j.ress.2021.108180.

Dai, A., Zhang, Z., Hou, P., Yue, J., He, S. and He, Z. (2019). Warranty claims forecasting for new products sold with a two-dimensional warranty, **28**(6): 715–730. https://doi.org/10.1007/s11518-019-5434-8.

Feng, S., Zhou, H. and Dong, H. (2019). Using deep neural network with small dataset to predict material defects, **162**: 300–310. https://doi.org/10.1016/j.matdes.2018.11.060.

Gupta, S. K., De, S. and Chatterjee, A. (2014). Warranty forecasting from incomplete two-dimensional warranty data, **126**: 1–13. https://doi.org/10.1016/j.ress.2014.01.006.

Gupta, S. K., De, S. and Chatterjee, A. (2017). Some reliability issues for incomplete two-dimensional warranty claims data, **157**: 64–77. https://doi.org/10.1016/j.ress.2016.08.015.

He, S., Zhang, Z., Jiang, W. and Bian, D. (2018). Predicting field reliability based on two-dimensional warranty data with learning effects, **50**(2): 198–206. https://doi.org/10.1080/00224065.2018.1436831.

Khoshkangini, R., Sheikholharam Mashhadi, P., Berck, P., Gholami Shahbandi, S., Pashami, S., Nowaczyk, S. and Niklasson, T. (2020). Early prediction of quality issues in automotive modern industry, **11**(7): 354. https://doi.org/10.3390/info11070354.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980.

Lee, J.-G., Kim, T., Sung, K. W. and Han, S. W. (2021). Automobile parts reliability prediction based on claim data: The comparison of predictive effects with deep learning, **129**: 105657. https://doi.org/10.1016/j.engfailanal.2021.105657.

Mitra, A., Jain, A., Kishore, A. and Kumar, P. (n.d.). A comparative study of demand forecasting models for a multi-channel retail company: A novel hybrid machine learning approach, *Operations Research Forum*, Vol. 3, Springer, pp. 1–22. https://doi.org/10.1007/s43069-022-00166-4.

Rai, B. and Singh*, N. (2005). Forecasting warranty performance in the presence of the 'maturing data'phenomenon, **36**(7): 381–394. https://doi.org/10.1080/00207720500139930.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function, **48**(4): 1875–1897. https://doi.org/10.1214/19-AOS1875.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting, **15**(1): 1929–1958. "Available at https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf".

Wang, X. and Xie, W. (2018). Two-dimensional warranty: A literature review, **232**(3): 284–307. https://doi.org/10.1177/1748006X1774277.

Wang, Y., Liu, Y., Liu, Z. and Li, X. (2017). On reliability improvement program for second-hand products sold with a two-dimensional warranty, **167**: 452–463. https://doi.org/10.1016/j.ress.2017.06.029.

Wu, S. (2012). Warranty data analysis: A review, **28**(8): 795–805. https://doi.org/10.1002/qre.1282.

Xie, W., Shen, L. and Zhong, Y. (2017). Two-dimensional aggregate warranty demand forecasting under sales uncertainty, **49**(5): 553–565. https://doi.org/10.1080/24725854.2016.1263769.

Xu, K., Xie, M., Tang, L. C. and Ho, S. L. (2003). Application of neural networks in forecasting engine systems reliability, **2**(4): 255–268. https://doi.org/10.1016/S1568-4946(02)00059-5.

Yang, S., Ting, T., Man, K. L. and Guan, S.-U. (2013). Investigation of neural networks for function approximation, **17**: 586–594. https://doi.org/10.1016/j.procs.2013.05.076.

Zainuddin, Z. and Pauline, O. (2008). Function approximation using artificial neural networks, **7**(6): 333–338. https://dl.acm.org/doi/abs/10.5555/1376368.1376392.