Using SMOTE techniques for industrial defects diagnosis to feed Machine Learning models

Yadini Pérez López *. Abrãao Jonas Caldas.*
Victor Freire de M. G. Pereira .*
Micila Sumaria M. Pereira. **
Laura Michaella B. Ribeiro. **

**Pólo de Inovação IFAM - INOVA/CTHM/IFAM, Manaus, AM 69075-351 BR*
(e-mail: yadini.lopez@ifam.edu.br, abraaocaldas@gmail.com, victor.freire@ifam.edu.br.)
***Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Manaus, AM 69075-351 BR (e-mail: micila.pereira@ifam.edu.br, laurarb@ifam.edu.br.)*

Abstract: Industrial process automation has become a priority for most modern enterprises over the past years. Product inspection for defect detection is a commonly automated process since is a repetitive pattern seeking activity. Regarding this, Machine Learning based solutions has been applied for industrial defect detection. However, most of the available solutions approach defect diagnosis using Computer Vision based solutions, which implies the installation of infrastructure for capturing production line images. This is not always a viable option for small young business because of budget limitations. In this work we approach industrial defect diagnosis using tabular history defect diagnosis records. We propose using Synthetic Minority Oversampling Technique for augmenting insufficient original data to feed Machine Learning algorithms for defect detection. We present the results of training with original and SMOTE oversampled data, reaching better results with SMOTE, approximately 69% of accuracy.

Resumo: A automação de processos industriais tornou-se uma prioridade para a maioria das empresas modernas nos últimos anos. A inspeção do produto para detecção de defeitos é um processo comumente automatizado, pois é uma atividade repetitiva de busca de padrões de erros. Nesse sentido, soluções baseadas em Aprendizado de Máquina têm sido aplicadas para detecção de defeitos industriais. No entanto, a maioria das soluções disponíveis abordam o diagnóstico de defeitos utilizando soluções baseadas em Visão Computacional, o que implica a instalação de infraestrutura para captura de imagens da linha de produção, mas nem sempre é uma opção viável. Neste trabalho é abordado o diagnóstico de defeitos industriais usando logs de registros tabulares de diagnóstico de defeitos, assim é aplicada a técnica de sobreposição de amostra minoritária sintética, do inglês SMOTE, com o intuito de aumentar dados originais insuficientes provendo *inputs* para algoritmos de aprendizado de máquina utilizados na detecção de defeitos. Os resultados a partir do uso da técnica SMOTE, permitiram o alcance de cerca de 69% de precisão a partir de dados tabulares.

Keywords: defect diagnosis, industrial quality automation, machine learning, data balancing, SMOTE.

Palavras-chaves: diagnóstico de defeitos, automação de qualidade industrial, aprendizado de máquinas, balanceamento de dados, SMOTE.

1. INTRODUCTION

Over the past decades, the industry has become increasingly digitalized. Most modern enterprises invest in process automation for productivity improvement and cost reduction. Quality verification is a process commonly automated since product inspection searching for specific defects is a repetitive and mechanical task that can be efficiently performed through Machine Learning (ML) based techniques.

Nowadays, several research approaches for industrial defect detection and diagnosis using Artificial Intelligence based techniques. Most available works are focused on surface anomaly detection using Computer Vision algorithms (Qifan Jin, 2022) (Alireza Saberironaghi, 2023) (SIEMENS, 2023). Defect detection models are trained to examine items that pass through the production line visually, recognize surface anomalies, and spot inconsistencies in dimensions, shape, and color (Akhremenko, 2022). Although defect recognition using image-based information is highly efficient, for many small companies, this approach is not a viable option due to the limited budget that prevents the necessary infrastructure to capture images or videos of production stages.

To avoid working with images for automated defect diagnosis, a usual approach is to record human-reported defect-diagnosis data for training ML models for automatically predicting a diagnosis given a defect. However, implementing this approach using ML techniques requires a

large amount of data, commonly a problem for small young businesses that lack historical defect detection records (Angeloupoulos et al., 2019). Moreover, the data available is frequently not well structured or clean, since oftentimes it is collected from different sources or departments not following recording rules, which leads to having duplicated, incomplete, or meaningfulness information.

In this study, we approach the automation of industrial defect diagnosis from a small, unbalanced, and unclean dataset using different ML learning algorithms to predict a diagnosis given a defect. This paper is organized as follows. Section 2 presents a dataset description and the techniques used to process the data for standardizing, cleaning, and balancing. Section 3 shows the algorithms implemented for defect diagnosis and gives a summary of the obtained results. Finally, in Section 4 the conclusions are presented.

2. DATA PREPARATION FOR DEEP LEARNING

ML algorithms demand a large amount of data to learn from it. Besides quantity the data must reflect the correct patterns that the model needs to memorize for doing accurate predictions (Johnson, 2019). In classification tasks it is crucial that classes are balanced since building a classification model with an imbalanced dataset will cause the underrepresented class to be overlooked or even ignored (N. Poolsawad, 2014).

In this work, we approach defects from data collected from a real industrial scenario, where some defects are more commonly reported while others rarely occur, leading to a natural imbalance of defect and diagnosis classes on data. To validate this problem two main techniques can be applied, oversampling and undersampling. Both change the ratios of the classes present and represent a re-sampling of available data. However, when the cross-classes dataset density is small regarding the classification algorithm, is a common practice just to oversample the underpopulated classes, for not losing samples of overpopulated classes (Mahendiran, 2019).

2.1 Data description

In this research we work with an industrial dataset that contains records of automated defect diagnosis with description about a test scope and diagnosis. This information is not expressed in natural language, instead, it is basically identifiers and product model codes.

Specifically, the data is the defect id, diagnosis id, product model, test id and test phase id, presented in Table 1, all expressed as Strings. From this data was defined as relevant information for defect diagnosis task, the defect, model, and diagnosis identifiers. Since the data is expressed as String type, we used data encoding for feeding the ML algorithms. The results of data transformation are shown in Table 2.

Although the available dataset has a not particularly low number of samples 128.562, the classes are not balanced. Some classes have more than 10.000 samples while other classes do not exceed 1 sample. In total there are 76 diagnosis classes. Figure 1 shows the number of samples per diagnosis class, omitting classes that don't have at least one sample.

Table 1. Data before encoding data for SMOTE.

Original Data		
defectId	diagnosisId	modelName
4501.0	3757	RTV9015VW
4501.0	3757	RTV9015VW
4501.0	3757	RTV9015VW
4502.0	3773	RTV9015VW
4502.0	3773	RTV9015VW

Table 2. Data after encoding for SMOTE.

Encoded Data		
defectId	diagnosisId	modelName
68	6	10
68	6	10
68	6	10
69	22	10
69	22	10

2.2 Oversampling with SMOTE

Data sampling is a collection of techniques that transform the training dataset to balance the class distribution. Once balanced, standard ML algorithms can be trained directly on the transformed dataset without any modification. There are different dataset balancing techniques such as Random Undersampling, Random Oversampling, Autoencoders based augmentation and Synthetic Minority Oversampling Technique (SMOTE) variants.

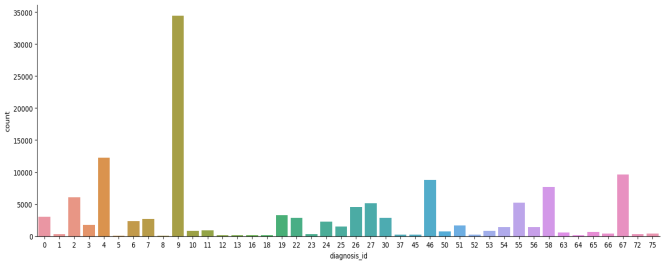


Fig. 1 Number of Samples per diagnosis class (original data).

SMOTE is the most frequently used technique for generating augmenting underpopulated samples since it has the advantage of not creating duplicate data points, but rather synthetic data points that differ slightly from the original data points (Mahendiran, 2019).

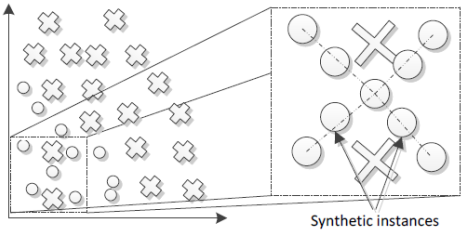


Fig. 2 SMOTE synthetic instances (N. Poolsawad, 2014).

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line (N. V. Chawla, 2002). Using the over-sampling approach, the minority class is over-sampled by creating artificial examples of k nearest class neighbours as shown in Figure 2. Samples are created through the following steps: identify the feature vector and its nearest neighbour, compute the distance between the two sample points, multiply the distance with a random number between 0 and 1, identify a new point on the line segment at the computed distance, and repeat the process for identified feature vectors.

Table 3 Parameter configuration of models trained for defect diagnosis.

Model	Parameter	Parameter Description
Neural Network (NN)	Epochs = 2000	It defines the number of times the entire data set must be worked through the learning algorithm
	Activation functions = 'relu', 'soft relu', 'tanh'	The Activation function ensures the non-linearity of the model. The 'relu' function interprets the positive part of the arguments. The 'soft relu' function or SoftPlus is a smoother version of the 'relu' and can be used to constrain a machine's output always to be positive. The difference between ReLu and softplus is near 0, where the softplus is enticingly smooth and differentiable. The biggest advantage of the 'tanh' function is that it produces a zero-centered output, thereby supporting the backpropagation process.
Light Gradient Boosting Method (Light GBM)	num_boost_round=200	Number of boosting rounds (controls training time of GBM models)
	default = 36 lower = 33 upper = 66	Number of leaves in trees
Weighted Ensemble	Autogluon Default	Weighted average or weighted sum ensemble is an ensemble machine learning approach that combines the predictions from multiple models, where the contribution of each model is weighted proportionally to its capability or skill.
Hyperparameter		**Description**
time_limit = 15 * 60 * 60		Train various models for ~15 hours
num_trials = 20		Try at most 20 different hyperparameter configurations for each type of model
search_strategy = 'auto'		To tune hyperparameters using Bayesian optimization routine with a local scheduler

2.3 Augmenting defect-diagnosis dataset with SMOTE

To implement the SMOTE technique in the scope of the dataset project aiming to learn how to suggest diagnosis given a product and defect identifiers was used the python library smote-variants (Kovács, 2019). This library already implements the SMOTE approach.

SMOTE-variant library works with vectorial representation of data. However, the dataset is in string format. To make the dataset compatible with the expected input, all data was converted to label encoding format using the label_encoder (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html) function of sklearn library preprocessing package. Table 1 shows the original data in string format and the encoded version for feeding the SMOTE algorithm. Then the encoded data was converted to a numpy array (NumPy, 2022) which is the input format that smote-variant expects.

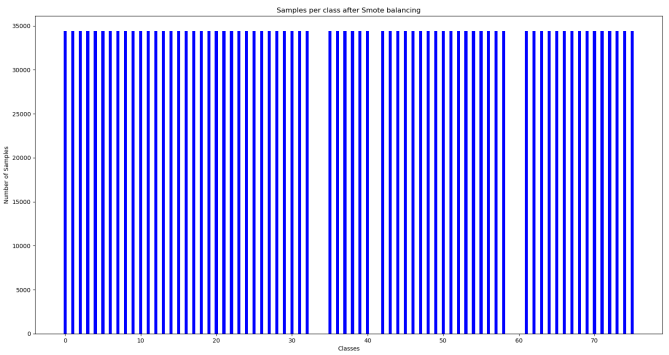
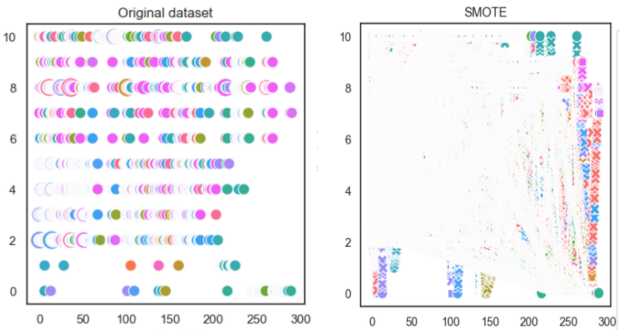


Fig. 3 Samples per class after SMOTE application.



(a) (b)

Fig. 4 Data density before (a) and after (b) applying SMOTE. The new samples (synthetic) were represented white "x", original class samples are represented with coloured circles.

Figure 3 shows the result of applying SMOTE balancing on the dataset. Most of the classes went up to 35.000 samples, matching the class that had the most samples. The bar graph presents some gaps, since some classes have almost no representation (1 sample), in those cases the balancing technique did not perform well. Figure 4 shows the result of applying SMOTE to original data. The new samples (synthetic) were represented with white "x".

3. MODEL TRAINING FOR DIAGNOSIS PREDICTION

After processing the data with smote-variants library, the information is more balanced and in a vectorial format which is expected to be the input for training most Machine

Learning algorithms. In this work were trained different models using the Autogluon (Autogluon, 2023) tool, which automates machine learning tasks enabling to easily achieve strong predictive performance for different applications, image, text, time series, and tabular data.

Dataset was split into three data sets, one for training data consisting of 60% of total data, 20% for testing data during training and 20% for final validation data.

Table 3 shows the architectures trained and the hyperparameter setup. Due to the lack of computational resources, the algorithms chosen were shallow Neural Network architecture, Light GBM and Weighted Ensemble, which have few hyperparameters and proved to perform accurately with unbalanced data. Also, Light GBM was selected since it is based on Decision Trees algorithms, which perform better when the dataset has a crucial "Feature" to decide. In this work, the diagnosis is highly impacted by the defect identifier.

3.1 Model training results

Both datasets were used as input to autogluon with a full set of algorithms as follows: LightGBM, Neural Net FAST AI, XGBoost, CatBoost, we selected top 5 results to be presented.

Training was performed in a 12 Core Intel Xeon running at 2.2Ghz, 80GB of RAM, Nvidia A100 GPU with 40GB of VRAM. Test was performed for 15 hours for each dataset. Autogluon version was 0.8.2. Autogluon models were trained initially with the original dataset (non-balanced).

In this study, the optimal scores are accuracy, F1-weighted, and score on validation dataset. On Accuracy the value is calculated by the difference between the value returned by the model and the actual value on the test data. F1 Weighted is a measure used on multi-class classification, as on this dataset, the result is a probability of a certain defect, and model has many possible values for a diagnosis then, F1-weighted is a measure with weights equal to class probability. Score Validation is just the common accuracy on the validation dataset.The results are shown in Table 4.

For this training setup the models were not able to learn from the provided data. The score validation presents less than 50% denotes that the models remained in the randomicity. This is probably because the classes were unbalanced. Also, the tool did not return values of the CatBoost training result because they were too low.

Table 4 Autogluon results of training LightGBM, Neural Net FAST AI, and Weighted Ensemble on the original dataset.

No Smote Results			
Model	Accuracy	F1 Weighted	Score Validation
XGBoost_BAG_L2	0.4607	0.4187	0.4534
WeightedEnsemble_L3	0.4605	0.4132	0.4542
NeuralNetFastAI_BAG_L2	0.4598	0.4109	0.4521
WeightedEnsemble_L2	0.4526	0.3953	0.4480
LightGBM_BAG_L1	0.4504	0.3937	0.4460

To improve the results attained with the original data, the training process was executed on the augmented dataset after applying the SMOTE technique. The results are shown in Table 5, this time, the accuracy went up to 69.40% with the Weighted Ensemble model, LightGBM values were between 69.37% and 69.24%, and other algorithms were under the top 5 values.

Table 5 Autogluon results of training LightGBM and Weighted Ensemble on the augmented dataset with SMOTE.

Smote Results			
Model	Accuracy	F1 Weighted	Score Validation
WeightedEnsemble_L2	0.4021	0.4154	0.6940
LightGBM_BAG_L1T3	0.4528	0.3988	0.6937
LightGBM_BAG_L1T5	0.4404	0.3812	0.6931
LightGBM_BAG_L1T8	0.4436	0.4436	0.6928
LightGBM_BAG_L1T2	0.4311	0.3784	0.6924

To conclude, the score validation reached with SMOTE was capable of improving the defect predictions causing more effective diagnosis for product repair.

4. CONCLUSIONS

In this work we approach industrial defect diagnosis using an unbalanced dataset of tabular historical records. To validate the unbalanced nature of data we chose to apply SMOTE oversampling technique since it does not create duplicate data points, but rather synthetic data points that differ slightly from the original data points. For defect diagnosis classification we trained different ML algorithms on the original and synthetically balanced dataset. The models were not able to learn with the original dataset, probably due to the unbalance condition. In contrast with this, after the SMOTE balancing application models were able to learn and the best results for defect diagnosis were attained with the Weighted Ensemble model, accuracy of 69.40%.

REFERENCES

Akhremenko, V. (2022). *Defect Detection in Manufacturing With Unsupervised Learning*. Retrieved from https://mobidev.biz/blog/defect-detection-in-manufacturing-with-unsupervised-learning. Accessed on 05/22/2023.

Alireza Saberironaghi, J. R.-G. (2023). Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review. *Deep Learning Architecture and Applications.*

Autogluon (2023). Available at https://github.com/autogluon. Accessed on 05/22/2023.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. Journal of Big Data, 6(1), 1-54.

Kovács, G. (2019). Smote-variants: A python implementation of 85 minority oversampling techniques. Neurocomputing, 366, 352-354.

Mahendiran, A. (2019). Data Augmentation Techniques for Tabular Data. NEXT Labs. Vedanth Subramaniam, Intern, New York, USA.

Nitesh V. Chawla, K. W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research, Volume 16, pages 321-357*.

N. Poolsawad, C. K. (2014). Balancing Class for Performance of. *Proceedings of the World Congress on Engineering 2014 Vol I,*. London, UK.

Numpy (2022). Available at https://numpy.org/doc/stable/reference/generated/numpy.array.html. Accessed on 05/22/2023.

Qifan Jin, L. C. (2022). A survey of surface defect detection of industrial products based on a small number of labeled data. arXiv preprint arXiv:2203.05733.

SIEMENS. (2023). *Artificial intelligence in industry: intelligent production*. Retrieved from https://www.siemens.com/global/en/company/stories/industry/ai-in-industries.html.

Sklearn. Available at https://scikit-learn.org/stable/. Accessed on 02/01/2023.