Applied Feature-oriented Project Life Cycle Classification

Oliver Böhme and Tobias Meisen
Chair for Technologies and Management of Digital Transformation, Bergische Universität Wuppertal, Rainer-Gruenter-Str. 21, Wuppertal, Germany

Keywords: Machine Learning, Classification, Prediction, Deep Neural Networks, MLP, LSTM, Multivariate, Automotive, R&D, Projects Progressions, Project Life Cycle, Comparative Analysis.

Abstract: The increasing complexity in automotive product development is forcing traditional manufacturers to fundamentally rethink. As a result, many companies are already investing in the development of methods to increase the controllability of their development processes. The use of data-driven approaches is a promising way to provide an early prediction of potential problems in the course of a project by learning from the past. In vehicle development, projects can be divided into two basic categories: new vehicle launches and model enhancement projects. The course of projects according to the above-mentioned categories can be based on different influencing factors. To verify this hypothesis and to determine the extent of the differences in the data, we carry out a data-driven classification of the project category. In contrast to the recognition of other time-dependent data (e.g., univariate sensor data courses), we use multivariate project information from the automotive industry. With this paper, which is of an application nature, we prove that a multivariate classification of automotive projects can be realized based on the underlying project's progression.

1 INTRODUCTION

The automotive industry is facing unprecedented challenges. There are new, fresh competitors entering the world markets, technological advancements call for further developments and increasing customer requirements force the classic manufacturers to transform. At the same time, growing volumes of information processing, an increase in the number of interconnected features and electronic control units (ECU) in the vehicle as well as a simultaneous, steadily increasing focus on data-based business models are creating ever greater complexity in development, production and sales. In addition to classic purchasing criteria such as comfort, design and engine performance, high-quality software is becoming increasingly important (Simonazzi et al., 2020). It has become a success factor for the reliability of the automotive product and its ability to succeed on the market. Product quality has an influence on the automotive company's reputation. For this reason, it is becoming increasingly important for automotive companies to develop methods that ensure the successful management of their product development.

As described in Boehme and Meisen (2021) we therefore strive to develop a data-driven approach that focuses on a quantitative evaluation and prediction of the progress of vehicle development projects by using machine learning methods (Boehme and Meisen, 2021). A system like that will help to predict risks of milestone shifts at an early stage of the project in order to develop measures to steer the project back on track.

In the research area, vehicle development projects are mainly divided into two categories: New vehicle launches with regard to the start of production (SOP) and model enhancement project, that are managed and continuously improved along their life cycle (LC). When a vehicle manufacturer intends to develop a new vehicle or a new derivative of an existing vehicle, then it is an SOP project. In order to continuously improve the product, integrate new features or react to current technical changes in the vehicle environment, the vehicles are advanced throughout their product life cycle. Within the framework of further development, deadlines are also defined by which the hardware-software-compound is to be released. Projects whose product life cycle falls into this category are assigned to LC projects.

Böhme, O. and Meisen, T.
Applied Feature-oriented Project Life Cycle Classification.
DOI: 10.5220/0010578402850291
In *Proceedings of the 10th International Conference on Data Science, Technology and Applications (DATA 2021)*, pages 285-291
ISBN: 978-989-758-521-0

Based on the knowledge of the domain's experts, which were interviewed by the authors, there is evidence that in many cases the course of SOP and LC projects appears to be differentiated. Hence, the aim of this paper is to validate, that the differences in the project's courses can be recognized by a data driven classification approach. For this purpose, we apply machine learning methods to automatically recognize the project type and derive evidence for the hypothesis based on the model's performance. Therefore, a dataset is used that contains real-world data from the electrical/ electronics development department of an automobile manufacturer. In addition, common methods for the classification of time series are implemented. Finally, the results are compared and discussed.

The remainder of this paper is structured as follows; Section 2 provides an overview of the current state of the art in time series classification methods. Section 3 describes the experimental setup. It introduces the dataset used and describes the development and optimization of the classification model. The results as well as the comparison of these results with those of common classification methods are presented in section 4. A summary of the goals, methodology and results are presented in section 5. Additionally, a recommendation for further work is given. Our vision is to predict the progress of vehicle development projects. For this paper, we aim to present an approach that is capable of classifying vehicle development projects based on various influencing factors.

2 RELATED WORK

As far as our own literature review showed, there are no approaches for the classification of project progressions. Considering this, we examine existing approaches from other industrial application areas and evaluate their adaptability.

In 2006, Yang & Wu identified time series classification as one of the ten most difficult problems in data mining research (Yang and Wu, 2006). Since then, it has been studied for several years (Esling and Agon, 2012). Research interest has grown with the increasing availability of existing time series datasets (Silva et al., 2018). Since 2015, hundreds of time series classification algorithms have been published (Bagnall et al., 2017). One of the most traditional and widely used approaches is Nearest Neighbor (Bagnall et al., 2017) (Lines and Bagnall, 2014). Recent contributions have therefore focused on methods that can go beyond k-NN (in conjunction with Dynamic Time Warping [DTW] as a distance metric).

Baydogan et al. focused their research on the application of random forests (Baydogan et al., 2013). From 2015 onwards, different types of discriminative classifiers such as Support Vector Machines (SVM), became more focused on by the research community (Bagnall et al., 2016) (Bostrom and Bagnall, 2015) (Schäfer, 2015) (Kate, 2016). Most of the approaches developed have the common property of a data transformation phase, in which the time series are transformed into a new feature space (Bostrom and Bagnall, 2015) (Kate, 2016).

Motivated by these considerations, an ensemble of 35 classifiers called Collective of Transformation based Ensembles (COTE) was created (Bagnall et al., 2016). COTE was further developed by Lines et al. by adding a hierarchical system component to HIVE-COTE by using a new hierarchical structure with probabilistic adjustment and by adding two additional classifiers to the ensemble (Lines et al., 2016) (Lines et al., 2018). In 2017 the authors stated that the method is considered state-of-the-art for time series classification (Bagnall et al., 2017). However, the method is not practicable in many areas of application because the calculation, optimisation and cross-validation of 37 classifiers is computationally intensive (Bagnall et al., 2017) (Lucas et al., 2018).

Due to the system limitations shown, some attempts have been made recently to apply deep learning approaches to time series classification problems. After the success of deep neural networks in the field of computer vision, a number of researchers have proposed different architectures for deep neural networks to solve time series classification tasks (NLP, machine translation, learning word embedding or document classification) (Sutskever et al., 2014) (Bahdanau et al., 2015) (Mikolov et al., 2013) (Mikolov et al., 2013) (Le and Mikolov, 2014) (Goldberg, 2016).

In 2015 Ordonez and Roggen used Deep Convolutional and Recurrent Neural Networks for Human Activity Recognition (Ordonez and Roggen, 2016). Similar research was carried out by Atzori et al. and successfully applied in the field of motion detection of prosthetic hands (Atzori et al., 2016).

Cui et al. presented a multi-scale convolutional neural network (MCNN) in 2016 that could achieve state-of-the-art performance (Cui et al., 2016). A year later, Wang et al. evaluated the performance of eleven different classifiers on 44 UCR datasets. With significant improvements compared to NN-DTW and COTE, a fully convolutional network was able to establish itself as the most powerful classifier (Wang et al., 2017). Bai et al. presented a generic convolutional network with dilations and residual connections in 2018 that showed more effective results than the common RNNs (such as LSTM) (Bai et al., 2018). At the same time, Karim et al. tested an enhancement of an FCN with LSTM submodules on 85 UCR datasets, which provided the best accuracy on average (Karim et al., 2019).

As it can be derived from the related work there is a wide range of classification methods for different fields of application. Furthermore, the literature research showed that no implementation in the automotive sector and the development of vehicle projects has been undertaken so far. Given this, we want to prove the applicability of the identified classification methods on the complexity of vehicle development projects in the next step.

3 RESEARCH METHOD

In this section the design and execution of the experiment is described. In addition, the experimental setup is explained.

3.1 Dataset Description and Statistical Analysis

The dataset used in this contribution was built by the authors and contains real-world data from the electrical/ electronics development department of an automotive manufacturer.

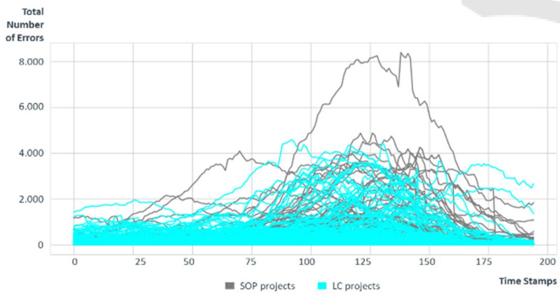

Figure 1: Two-dimensional Visualization of the Data.

It contains 302 examples, each of which is described with 20 attributes. One of the attributes indicates whether the vehicle project under consideration is an SOP or an LC project. Each example is a time series representing the progress of a vehicle development project over 195 time stamps.

Figure 1 shows a two-dimensional visualisation of the data, where the Y-axis represents the total number of errors in the project at the time of the respective time stamp. By calling errors, we understand hardware and software errors appeared over the development of each project. Each time stamp represents a weekly snapshot of the project status, in terms of the current total number of errors, considering project meta- and environmental factors.

Table 1 shows an overview, description and statistical evaluation of the different features. The mean, median, standard deviation, maximum and minimum values are shown. The two classes are represented in our data in a ratio of 65% (LC) to 35% (SOP).[1]

3.2 Experimental Setup

The experiments are carried out using Python version 3.7. In order for our dataset to become a suitable input for our learning models, the original structure had to be pre-processed. Before splitting the dataset into a train- and test set, the input data was normalised. With normalisation, we strive to transform the values in our columns to a common scale. The split percentage was chosen at 30/70, since that has been suggested in several literatures in the field of machine learning (Khosla, 2015). Finally, train and test data were transformed into numpy arrays for efficiency reasons.

The classifiers used are the SKLEARN implementations of AdaBoost, Decision Tree classifier, Discriminant Analysis, Gaussian Process classifier, MLP classifier, Support Vector Machine (SVC, Linear SVC, SGD classifier), Random Forest classifier and K-NN. In addition, by using Tensorflow and Keras we implemented a baseline LSTM-classifier. Except for the LSTM, each classifier was optimised based on its individual (hyper-)parameters using GridSearch.

The LSTM was designed with a first embedded layer that uses vectors of the length of the trainings data shape to represent each vehicle project. The next four layers are bidirectional LSTM layers with 100 memory units. Since it is a classification problem, we use a dense output layer with a single neuron and a sigmoid activation function to make 0 or 1 predictions for the two classes (SOP and LC) in the classification problem. Due to the fact that it is a binary classification problem, the logarithmic loss is used as

[1] For reasons of confidentiality, the dataset cannot be released at the moment. This will be done by the authors at the appropriate time.

Table 1: Features of the Automotive Dataset.

Feature ID	Feature Name	Feature Description	Mean	Median	Standard Deviation	Maximum	Minimum
X1	Project Priority	Classification of the Projects Importance: most important (1) - least important (3)	2.18	2	0.60	3	1
X2	Launching Type	Whether (1) or not (0) the project is the first to launch on a new platform	0.12	0	0.32	1	0
X3	Type of Driving	Classification of the type of driving: combustor (1), PHEV (2), BEV (3)	1.25	1	0.59	3	1
X4	Carry Over Indicator	Degree of carry over of existing architecture parts or ECUs: maximum carry over (0), architectural modification (1), all new E/E architecture (2)	0.88	1	0.44	2	0
X5	Market Launch EU/RoW	Whether (1) or not (0) the project is launched on the market EU/RoW	0.48	0	0.50	1	0
X6	Market Launch ME/LA	Whether (1) or not (0) the project is launched on the market ME/LA	0.17	0	0.37	1	0
X7	Market Launch NAR	Whether (1) or not (0) the project is launched on the market NAR	0.05	0	0.21	1	0
X8	Market Launch China	Whether (1) or not (0) the project is launched on the market China	0.30	0	0.46	1	0
X9	Top Management Support	Whether (1) or not (0) the project has given top management level support	0.11	0	0.31	1	0
X10	Parallel Projects	Number of parallel developed projects	157.22	166	53.76	227	1
X11	Employees	Number of employees	1222.36	1200	175.63	1482	704
X12	In-house performance ratio	Ratio of internally completed activities to outsourced activities	18.70	21	5.49	30	9
X13	Average Employees	Approximate number of employees per project	10	7.55	15.36	1016	5.56
X14	Employee Satisfaction Index	Level of overall employee satisfaction	72.83	72.56	1.44	77.12	70.70
X15	Satisfaction within the Teams	Level of employee satisfaction within the organizational units	69.81	69.50	1.37	73.91	68
X16	Collaborative Satisfaction	Level of employee satisfaction regarding collaboration	77.09	76.97	1.71	81.76	74.5
X17	Personal Satisfaction	Level of employee satisfaction with respect to self-fulfilment and appreciation	73.73	73.67	1.45	77.94	71.30
X18	Organizational Satisfaction	Level of employee satisfaction in terms of the company's reputation	69.94	69.96	1.43	73.99	67.4
X19	Total Errors	Total number of unfixed errors	321.40	98	616.19	8396	0
y	General project type	Classification of whether the project is categorized as SOP (1) or LCM (0)	0.35	0	0.48	1	0

the loss function. In addition, the efficient ADAM optimisation algorithm is used. The model is fit for 50 epochs. The batch size of 32 reviews is used to distribute the weight updates.

For comparable results we also used 5-fold cross validation for each algorithm. To determine the best model, we used the F1-Score, since it is the harmonic mean of respective recall and precision values (Tatbul, 2018). For further details on the implementation and for the purpose of further use we published our code.[2]

4 RESULTS

In Table 2 we show a comparative evaluation of all methods applied to the dataset presented. The MLP classifier showed the lowest performance at 79.1%. The second neural network approach, LSTM, also performed only 0.4 percentage points higher. The Discriminant Analysis (79.5%) and the Gaussian Process classifier (79.7%) also ranked on approximately the same level. With 80.9%, K-NN is in the midfield of the comparative evaluation. In contrast, AdaBoost, Decision Tree and Support Vector Machine showed a slightly better performance, achieving a F1-Score between 82.4% and 82.5%. The Random Forest classifier showed the best performance with a F1-Score of 85.7%. Compared directly with the neural network approaches, it achieved over 6 percentage points better.

To further analyse the performance of the models, we divided the time series into three equal periods. The following is a simplified explanation of the choice of the three periods. As can be seen in Figure 1, the differentiation of the time series based on the total number of errors in the first period cannot yet be measured consistently. This is due to the fact that this phase is usually used for function build-up and therefore only minor testing can be carried out over this period. In the second period, testing the functions at vehicle level is one of the main tasks in terms of integration and ensuring product quality. In this phase, it is crucial to find all the errors possible. Finally, the third phase describes the reduction of errors. In this phase, it is decided whether the quality and deadline targets can be kept. Afterwards, again a comparative evaluation is carried out for each period. Figure 2 shows the performance of the algorithms measured by the F1-Score for the shapelets and the whole series.

Once again, the Ada Boost, Decision Trees and Random Forest were able to confirm their performance. In relation to the first third, Decision Tree showed the best performance with an F1 score of 85%. This demonstrates the algorithm's robustness to errors and shows that it can handle categorical and continuous data well at the same time.

[2] https://github.com/tmdt-buw/carclass

Table 2: Comparative Evaluation on the implemented Classifiers.

Algorithm	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]
Ada Boost Classifier	76.9	81.7	83.1	82.4
Decision Tree Classifier	75.8	86.7	78.8	82.5
Discriminant Analysis	65.9	100	65.9	79.5
Gaussian Process Classifier	67	98.3	67	79.7
K-Nearest Neighbor	71.4	91.7	72.4	80.9
Linear SVC	70.3	95	70.4	80.9
LSTM Classifier	65,9	100	65,9	79,5
MLP Classifier	69.2	88.3	71.6	79.1
Random Forest Classifier	78	100	75	85.7
SGD SVC	70,4	100	67.4	80.5
Support Vector Classifier	74.7	90	76.1	82.4

Ada Boost was able to classify the vehicle projects with respect to the period in the middle with an almost perfect F1 score (98%). Due to the small number of training samples, this demonstrates on the one hand the adaptability to the complex time series in the vehicle development environment. The high accuracy (96.8%) in this phase also confirms this result. The good performance with so few training samples could be an indicator that the factors influencing project performance depend on the type of project. This supports the hypothesis that for a holistic multivariate prediction of project progress, differentiation in terms of predictors may be useful to maximise the performance of the prediction model.
However, the measurement results of the algorithm for the third period point to the known disadvantages such as its sensitivity to noisy data and outliers. This means that it is always more difficult to achieve good performance with Ada Boost without overfitting when the data cannot be easily assigned to a particular separation plane. With only 58.8% F1 score, AdaBoost represents the worst result in the method comparison here. Among other things, this can be attributed to the fact that the AdaBoost reacts very sensitively to noise and outliers, both of which can be triggered by the different project ends. Despite the high heterogeneity of the data in the last period, the Random Forest was still able to achieve an F1 score of 77.8%. This confirms the algorithm's robustness against outliers and its good handling of non-linear data.

The linear SVC shows the worst result in period 1 with only 69.8 %. In particular, with reference to the class distribution (ratio 1:2), this performance is only significantly above a random classification. In terms of ranking, again we see the same pattern for period 2. However, the lowest F1 score achieved is 87.3%, which is still a very good result considering the number of training samples.

Furthermore, the LSTM has to be seen as one of the weaker classifiers. The deep neural network performs only slightly better than the weakest classifier in each case. It should be emphasised that the LSTM is only available in its baseline variant and has therefore not yet been fully optimized. In addition, it is to be expected that the performance will improve considerably with an increasing amount of data.

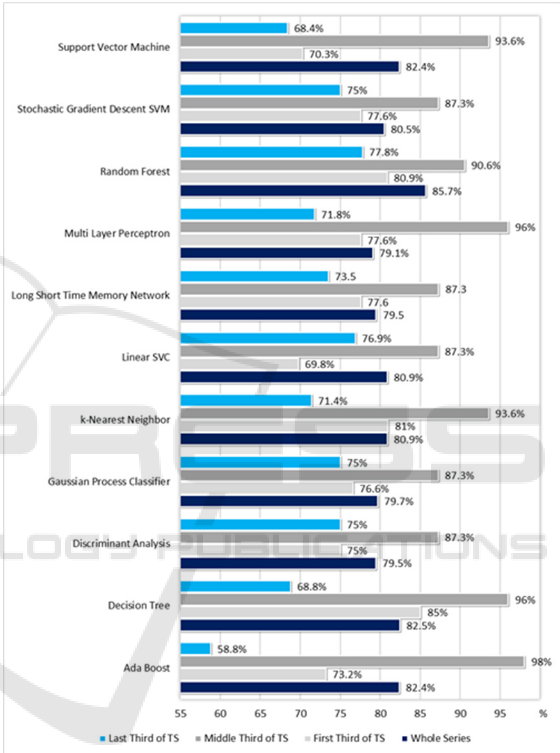

Figure 2: Segment-wise Comparison regarding F1-Score.

5 CONCLUSIONS

Our aim was to investigate, whether the common methods for multivariate classification can be applied to the recognition of the project type of vehicle development projects. We aimed to prove the hypothesis that the project types "SOP" and "LC" are subject to different influencing factors. For this purpose, we implemented current state-of-the-art methods from the field of machine learning as well as deep neural networks (baseline classifiers) and compared them with respect to their performance.

Our results have shown that multivariate time series classification of vehicle development projects is feasible. Even with a small number of training samples and a comparatively high number of features, an F1 score of 85.7% (at 78% accuracy) could be achieved. Considering the class distribution, this is a promising result. By dividing the time series into three periods, these results could be considerably increased again with an F1 score of 98% (at 96.8% accuracy).

Ensemble methods such as Ada Boost and Random Forest stood out in particular. Along with decision trees, these two methods not only showed very good applicability for the given problem, but also outperformed the neural networks (likely due to a lack of training data). In addition, these white box models offer the advantage of transparency, interpretability and lower computing time. Due to this definite assignability of the project type, we see our hypothesis confirmed.

For similar problems, we therefore recommend the use of ensemble methods, considering the classification results, the implementation effort and the computing time. However, it can be assumed that the performance of the neural networks will increase with an increasing number of training samples.

Further work will therefore consist in adding additional training samples to the dataset. Furthermore, for having a complete picture, besides considering the approach presented in this paper, a comparative evaluation of the results with other classification methods focusing on optimised neural networks (e.g. FCN, CNN, LSTM) and ensemble methods (e.g. HIVE-COTE) should be performed. We will also consider different fold sets in our training and testing.

In our future work, we will also conduct detailed considerations for a better understanding of feature importance. In order to address the curse of dimensionality, the relevance of the individual features will be determined, compared and evaluated depending on the respective project phases. Finally, the implementation of prediction models is planned, enabling the prediction of the progression from any point in time within the project.

ACKNOWLEDGEMENTS

We would like to thank all reviewers for their valuable comments.

REFERENCES

Atzori, M., Cognolato, M. and Müller, H. 2016. Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands. *Frontiers in Neurorobotics, 10, 2016*

Bagnall, A., Lines, J., Bostrom, A., Large, J. and Keogh, E. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery 31(3), pp 606–660*

Bagnall, A., Lines, J., Hills, J. and Bostrom, A. 2016. Time-series classification with COTE: the collective of transformation-based ensembles. *In: International conference on data engineering, pp 1548–1549*

Bahdanau, D., Cho, K. and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *In: International conference on learning representations*

Bai, S., Kolter, J.Z. and Koltun, V. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, *arXiv, https://arxiv.org/pdf/1803.01271.pdf*

Baydogan, M.G., Runger, G. and Tuv, E. 2013. A bag-of-features framework to classify time series. *IEEE Trans Pattern Anal Mach Intell 35(11), pp 2796–2802*

Boehme, O. and Meisen, T. 2021. Predicting the Progress of Vehicle Development Projects – an Approach for the Identification of Input Features. *In: 13th International Conference on Agents and Artificial Intelligence (ICAART 2021)*

Bostrom, A. and Bagnall, A. 2015. Binary shapelet transform for multiclass time series classification. *In: Big data analytics and knowledge discovery, pp 257–269*

Cui, Z., Chen, W. and Chen, Y. 2016. Multi-Scale Convolutional Neural Networks for Time Series Classification, *arXiv*

Esling, P. and Agon, C. 2012. Time-series data mining. *ACM Comput Surv 45(1), pp 12:1–12:34*

Goldberg, Y. 2016. A primer on neural network models for natural language processing. *Artif Intell Res 57(1), pp 345–420*

Karim, F., Majumbar, S., Darabi, H. and Harford, S. 2019. Multivariate LSTM-FCNs for time series classification, *Neural Networks, 116, pp 237-245*

Kate, R.J. 2016. Using dynamic time warping distances as features for improved time series classification. *Data Min Knowl Discov 30(2), pp 283–312*

Khosla, R., Howlett, R.J. and Jain, L.C. 2005. Knowledge-Based Intelligent Information and Engineering Systems, *9th International Conference, KES 2005 Melbourne, Australia, September 2005 Proceedings, Part IV, p 3*

Le, Q. and Mikolov, T. 2014. Distributed representations of sentences and documents. *In: International conference on machine learning, vol 32, pp II–1188–II–1196*

Lines, J., Taylor, S. and Bagnall, A. 2016. HIVE-COTE: the hierarchical vote collective of transformation-based ensembles for time series classification. *In: IEEE international conference on data mining, pp 1041–1046*

Lines, J., Taylor, S. and Bagnall, A. 2018. Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles. *ACM Trans Knowl Discov Data 12(5), pp 52:1–52:35*

Lines, J. and Bagnall, A. 2014. Time series classification with ensembles of elastic distance measures. *Data Min Knowl Discov 29(3), pp 565–592*

Lucas, B., Shifaz, A., Pelletier, C., O'Neill, L., Zaidi, N., Goethals, B., Petitjean, F. and Webb, G.I. 2018. Proximity forest: an effective and scalable distance-based classifier for time series. *Data Min Knowl Discov 28, pp 851–881*

Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. Efficient estimation of word representations in vector space. *In: International conference on learning representations—workshop*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *In: Neural information processing systems, pp 3111–3119*

Ordonez, F.J. and Roggen, D. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Acitivity Recognition, *Sensors 16(1), p 115*

Schäfer, P. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Min Knowl Discov 29(6), pp 1505–1530*

Silva, D.F., Giusti, R., Keogh E. and Batista, G. 2018. Speeding up similarity search under dynamic time warping by pruning unpromising alignments. *Data Mining and Knowledge Discovery 32(4), pp 988–1016*

Simonazzi, A., Sanginés J.C. and Russo, M. 2020. The Future of the Automotive Industry: Dangerous Challenges or New Life for a Saturated Market, Institute for New Economic Thinking, 2020

Sutskever, I., Vinyals, O. and Le, Q.V. 2014. Sequence to sequence learning with neural networks. *In: Neural information processing systems, pp 3104–3112*

Tatbul, N., Lee, T.J., Zdonik, S., Alam, M. and Gottschlich, J. 2018. Precision and Recall for Time Series. *In: Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*

Wang, Z., Yan, W. and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. *International Joint Conference on Neural Networks (IJCNN), pp 1578–1585*

Yang, Q. and Wu, X. 2006.10 challenging problems in data mining research. *Inf Technol Decis Mak 05(04), pp 597-604.*