Erschienen in: IUI '20 : Proceedings of the 25th International Conference on Intelligent User Interfaces / Paternò, Fabio; Oliver, Nuria (Hrsg.). - New York, NY : ACM, 2020. - S. 202-206. - ISBN 978-1-4503-7118-6
https://dx.doi.org/10.1145/3377325.3377528

Towards Visual Debugging for Multi-Target Time Series Classification

Udo Schlegel,[1] Eren Cakmak,[1] Hiba Arnout,[2,3]
Mennatallah El-Assady,[1] Daniela Oelke,[2] and Daniel A. Keim[1]
[1] University of Konstanz, [2] Siemens CT Munich, [3] Technical University Munich

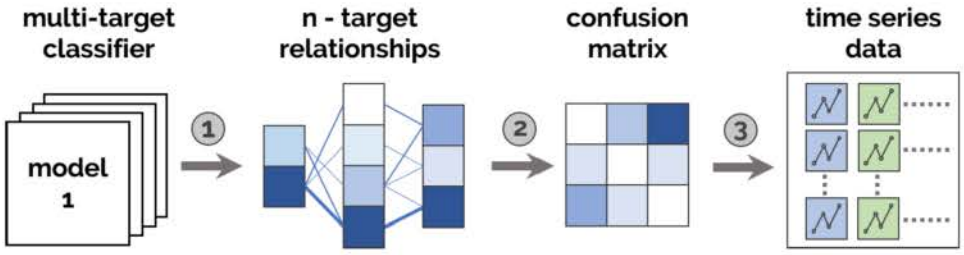

Figure 1: Multi-Target Visual Debugging Workflow. (1) A user selects a model from a set of multi-target classifiers to inspect its performance. (2) Through filtering of partially correct results and the visual investigation between the relationships of the n-target, the user can focus on a certain single-target. (3) By inspecting the filtered single-target confusion of multi-classes, a drill down onto the underlying time series classification data is possible that aims to give meaningful insights into the class confusions.

ABSTRACT

Multi-target classification of multivariate time series data poses a challenge in many real-world applications (e.g., predictive maintenance). Machine learning methods, such as random forests and neural networks, support training these classifiers. However, the debugging and analysis of possible misclassifications remain challenging due to the often complex relations between targets, classes, and the multivariate time series data. We propose a model-agnostic visual debugging workflow for multi-target time series classification that enables the examination of relations between targets, partially correct predictions, potential confusions, and the classified time series data. The workflow, as well as the prototype, aims to foster an in-depth analysis of multi-target classification results to identify potential causes of mispredictions visually. We demonstrate the usefulness of the workflow in the field of predictive maintenance in a usage scenario to show how users can iteratively explore and identify critical classes, as well as, relationships between targets.

1 INTRODUCTION

The field of multivariate time series classification is a well-established research discipline with many real-world applications such as stock market forecasting [14] or heartbeat anomaly detection [6]. In many such applications a multivariate sample is not only assigned to a single target t with x classes but multiple targets $t \in T$ with each x classes. A related field is multi-label classification [23] which assigns multiple classes of a target to an instance. For instance, in predictive maintenance, a common task is to increase the productivity of sophisticated industry machines by predicting multiple targets (e.g., component failures) by the usage of historic multivariate (time series) sensor data [15, 16]. The main task of such a multi-target classification is to build accurate models that learn to predict all targets with their relations (dependencies and correlations), for example, possible future states (e.g., working or broken) of each of the interconnected components. However, the training of such accurate multi-target models is challenging due to the time series data being multivariate, consisting of potentially incorrect classes, and further requiring complex classification algorithms. Visual analytics supports the building and analyzing of these multi-target models by involving the human through interactive visualizations [20].

Related approaches from machine learning and visual analytics tackle the visual evaluation and debugging of classifiers based on the model output and input data. Primarily, confusion matrices [17] and their shading [24], color [13], and representation [1] enhancements [11] facilitate to visualize classifier performances. Other approaches explore the performance through histogram visualizations [19] or support the model building and performance exploration [2]. However, most of the already existing approaches do not investigate partially correct results in which some, but not all, targets are classified correctly. Investigating such partially correct predictions may help to filter the misclassifications down to individual failures to highlight possible unknown problems.

The visual analysis of partially correct results, as well as problematic samples, furthermore, helps to identify relationships between correct and misclassified targets. Such insight helps to improve the multi-target classifier by, for instance, either pre-processing the multivariate time series data or selecting more suitable algorithms. In general, a multi-target problem can be simplified or interpreted as a multi-class problem by building the cross-product [23], as a chain-of-classifiers problem [18] or as multi-target classification. Prominent examples of multi-target classifiers are Random Forests [4] and Neural Networks [8], which can be directly applied to solve the classification problems. However, still, only limited work explores and analyzes partially correct results [18] and target relationships [23, 25]. In contrast to the proposed approaches, the goal of this work is to allow a visual inspection of partially results to get insights by examining hard to predict classes.

We contribute a visual debugging workflow to examine the performance of multi-target models, which takes partially correct results into account and enables to investigate the causes of mispredictions visually. The workflow fosters an in-depth analysis of classification results, including the investigation of different multi-target classifiers and facilitates an iterative refinement and visual debugging of the model and data. Domain experts can use the workflow to generate new insights by visually analyzing the causes of possible mispredictions and compare those to correctly predicted samples. We show the applicability of the workflow through a prototypical implementation using a real-world use case on predictive maintenance with anonymized data from an industry partner. Based on the data, a usage scenario presents the workflow applied to data from an expert's view. Our primary contributions are as follows:

- **a conceptualization of a visual debugging workflow** for multi-target models to investigate causes of mispredictions;
- **a prototypical implementation** of the workflow that allows visually examining mispredictions of multi-target classifiers;
- **a method for visual analysis and inspection** of multi-target classifiers and partially correct results.

2 VISUAL DEBUGGING WORKFLOW

The goal of the proposed visual debugging workflow is to provide different levels of abstractions to enable a top-down visual exploration based on the Visual Information Seeking Mantra [21]. We incorporate partially correct results to identify and refine hard-to-predict relationships, between different targets and potential causes of mispredictions. Our proposed workflow allows us to investigate and visually debug possible errors in multi-target classifiers through various methods and techniques.

Design goals – From ongoing collaborations with domain experts in the field of predictive maintenance, we collected four design goals for our visual debugging workflow. *First*, the workflow should apply to any multi-target multivariate time series classifier. As a result, we did not include any visualizations that depict only the internal processes of one particular algorithm. *Second*, the workflow should support the visual analysis of relationships between multiple targets and classes. *Third*, it should visualize classification results for one target and descriptive representations of corresponding classes to enable a high-level comparison of correct and misclassified samples. *Fourth*, it should support the investigation and comparison of the raw multivariate time series data used for classification.

The interactive visual debugging workflow – We propose several visual methods to analyze the classification results of multi-target models interactively. The interactive analysis starts via the selection of a classification model from multiple trained classifiers (see Figure 1 (1)) and tackles our first design goal. To provide a first overview of the selected model, we depict the relationships between the multiple targets and their underlying classes to support the examination of links between the different targets (second design goal). The display of the target relationships also presents partially correct results and critical class confusions that need further exploration. The selection of a combination of classes in the overview allows to filter down to a particular target and inspect the confusion matrix of the single not selected target (see Figure 1 (2)). The drill-down to the confusion matrix enables, furthermore, to focus on specific critical classes and comprehensive analysis of descriptive representations of predictions (third design goal). The selection of a confusion matrix prediction visualizes the multivariate time series data in another view to enable the comparison of each feature to the correct and wrongly predicted class representatives (see Figure 1 (3)) and addresses the last design goal. The detailed analysis of such misclassified samples enables to test and verify previous hypotheses.

3 VISUAL ANALYTICS WORKSPACE

In this section, we present a prototypical implementation of the proposed workflow using multiple model independent visualizations for the exploration and analysis of multi-target classifications of multivariate time series data. In particular, we target classifiers used for predicting two or more targets over time based on same-length segmented time series data with measurable features collected by multiple sensors. Section 4 introduces an application scenario to explore the forecast of machine states based on sensor data to prevent malfunctions by early maintenance.

The proposed visualizations support users to explore, inspect and identify errors of classifiers in a visual environment using given and engineered (e.g., weighted feature combinations, Fourier-Transformation [5] of time segments) features of the underlying time series data. Our approach consists of three interlinked views that are targeting the exploration of misclassifications; the *Multi-target Parallel Coordinate Plot* (**Figure 2 (B)**) for overview and class filtering, the *Enhanced Confusion Matrix visualization* (**Figure 2 (C)**) for critical class exploration, and the *Comparative Time Series View* (**Figure 2 (D/E)**) for detail investigation.

3.1 Multi-target Parallel Coordinate Plot

Figure 2-B depicts the *Multi-target Parallel Coordinate Plot* and enables entry-level exploration and navigation of classification errors by giving a first overview of the classes and targets based on parallel coordinates [10] and parallel sets [3]. Parallel coordinates [10] are the basis to show geometrical properties [9] of the target and class correlations and relations to support an exploration of the relationships by reordering capabilities [10]. Different class combinations can be selected and filtered to explore the classifier outcomes. The vertical axis or columns visualize the classes of a target. A plot with one column corresponds to a single-target or multi-class problem. Multiple columns with connections between them show a multi-target problem. The color of the cells depicts the

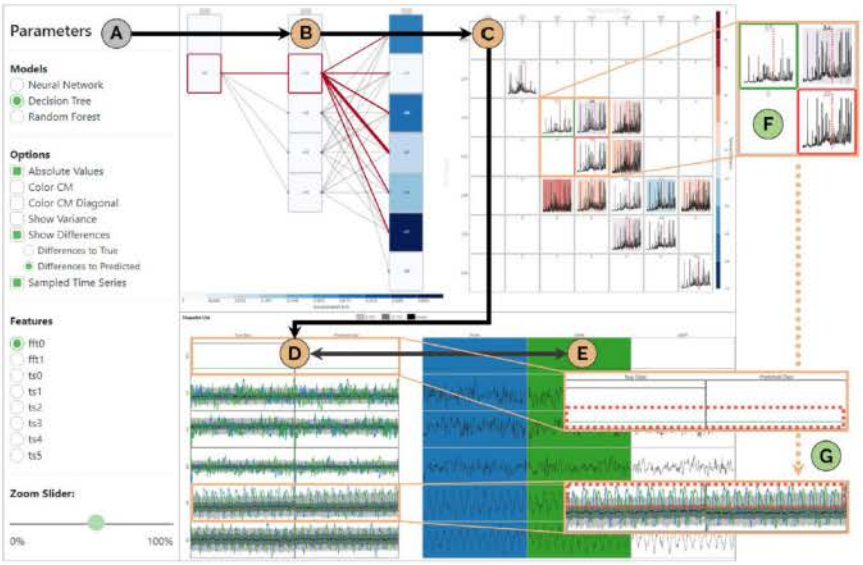

Figure 2: Demonstrator of the multi-target visual debugging workflow: (A) enables a domain expert to select a model. (B) depicts the Multi-target Parallel Coordinate Plot with (C) the Enhanced Confusion Matrix Visualization and further adjustable (A) settings panel. (D+E) shows the Comparative Time Series View with (D) class quantile graphs and the (E) underlying misclassified time series samples. (F+G) present a finding based on a domain expert's analysis. (F) shows zoomed sample representatives of a descriptive feature in which confused samples are more similar to a predicted class. (G) shows class quantile graphs with the samples of the previous class confusions plotted in.

misclassification error of the class of the target for the test data. A light color □□□ holds a near or ground truth prediction and a dark color ■■■ a significant misclassification error to visually highlight starting points for the analysis. The links between the axis present the relations between the classes to reveal partially correct results. The thickness of the links represents the misclassification of the relationship between the targets. A good classification would lead to thin lines to visualize only the existing relation. Solid and broad lines depict a high error and another analysis entry point. If there is no connection, the test data does not have such a combination.

By selecting two of the three targets (or generally n-1 for n multi-targets) in the Multi-Target Parallel Coordinate Plot, the interface transitions to the Enhanced Confusion Matrix Visualization, as depicted in Figure 2-C. The confusion matrix corresponds to the not selected single-target and its multi-class confusions. The selected classes furthermore filter the shown data in the confusion matrix to incorporate the correct predicted classes from the partially correct results and thus follows the proposed workflow by first giving the overview and then drill down to details.

3.2 Enhanced Confusion Matrix Visualization

Figure 2-C shows the *Enhanced Confusion Matrix Visualization*, which supports a more focused analysis of the class confusions of a classifier due to the comparison of prediction and ground truth. A confusion matrix is an established classification performance measure for machine learning experts [22] and thus a common starting point for further classifier analysis. Further, the cells are enhanced with time series cell representatives encoding more information into the confusion matrix. The cell representatives are either the time series data itself or engineered features of the data like the frequency, the median, and other descriptive features. As there is only space for a single representative, one representation for a class or a misclassification has to be selected. In the case of general features, a median builds a first and robust solution to combine these without losing much information through an aggregation. Due to the temporal aspect of multivariate time series, a median over the whole time interval is not robust to shifting effects introduced by various sensor time measurement errors. A rolling median mitigates some of these shifting effects and aims for a more robust class representation. By hovering over a misclassified cell, the true class and the predicted class are highlighted as green and red, Figure 2-F, to facilitate the comparison of the class representations.

The background color of a cell either corresponds to the misclassification error or the pair- and point-wise difference of the misclassified samples to the correct or predicted class samples depending on the user's selection. The differences calculated for the features inside of the cells support the feature distance relations exploration to the classes to analyze which features could be decisive. Notably, the differences allow hypothesis generation by providing an additional visual measure to compare the cells.

To verify or dismiss such a hypothesis, users can select one cell in the confusion matrix and go to the individual data point level to transition from the filtered overview of the classification to the underlying data and the *Comparative Time Series View*.

3.3 Comparative Time Series View

The *Comparative Time Series View*, Figure 2-D+E, enables to further inspect raw time series data in detail by comparing quantile graphs of the correct and predicted classes to the selected samples to verify

or dismiss hypothesis about the heterogeneity. Quantile graphs for time series consist of univariate time-point-wise quantiles for a whole segment and are visualized using Themerivers [7] to present changes in the time series data. Further, quantile graphs from the true and the predicted class enable the comparison and inspection of time series features and data to explore time spans in which the classifier could have problems. A quantile graph shows selected quantiles of the distribution of segmented time series data in a color schema to enable an easier comparison to other data. The quantile graphs of the visualization show the quantile 50 (median) as a line and the quantiles from 5 to 95 and from 25 to 75 colorized as a stream, see Figure 2-D and G in the background.

The colorized quantile graphs enable to present class representatives and to show a general homogeneity or heterogeneity of the data as the graphs show the flowing distribution of the majority of the time series. Samples which are heavily shifted and outside of the quantile graph of the correct class in one feature are suitable starting cases for an in-depth analysis on, e.g., the target correctness. Plotting misclassified samples, Figure 2-E, into the quantile graphs improves the comparison between actual and predicted class by showing both in one plot. The comparison and exploration between the wrongly predicted samples and the quantile graphs facilitate the identification of heterogeneous or corrupt data for the inspection of areas in which a classifier might have problems.

4 USAGE SCENARIO

We illustrate how a domain expert can use an instantiation of the proposed model-agnostic workflow to gain new insights into a classifier on an industry dataset provided by our partner. In the following usage scenario, see Figure 2, we are visually analyzing anonymized multi-target multivariate time series data from a real-world use case on predictive maintenance. The goal was to help domain experts to debug and build trust into existing models. The analyzed dataset consists of 20 million time points with six features representing sensor data and the status of a machine. The implementation starts by pre-processing the time series data and segmenting the dataset into user-defined segments (e.g., 512 milliseconds). Some existing models (e.g., a decision tree, a random forest, a neural network) from our partner get then fine-tuned onto the pre-processed data. After training, the domain expert can select a model in the selection panel. In our case, we start by selecting a baseline decision tree classifier see Figure 2-A.

Multi-target Parallel Coordinate Plot View – The first view aims to provide an overview of the classifier performance and the relations between multi-targets. In our case see Figure 2-B, the classifier result for two targets (left and middle columns) is quite good (low intensity), but has difficulties (high intensity) with some classes (dark blue) of one target (right column). The selection in Figure 2-B shows the two good predicted targets selected and depicts a relation with a high error to the last target and presents a partially correct result that requires a detailed investigation. Debugging the wrong predictions may steer the model to better performance as it improves, e.g., the quality of the data for a classifier [12].

After selecting a combination of good predicted classes, the data flow transitions to the *Enhanced Confusion Matrix View* Figure 2-C.

Enhanced Confusion Matrix View – In the selection panel, see Figure 2-A, a domain expert can select descriptive features as class representatives to enhance the confusion matrix and inspect class confusions. In our case, the selected feature is the mean of the Fourier-Transformation of every time series sample in the cell as it shows the homogeneity by either showing a clear peak (homogeneous) or a uniform distribution (heterogeneous). The pair- and point-wise differences in the cells allow comparing the representative of the misclassified samples to the predicted class representative. In our case, of Figure 2-F, the representative of the misclassified samples (top right) is more similar to the predicted class (red rectangle/bottom right) than to the correct one (green rectangle/top left) and generates the first hypothesis of a problematic feature (the dimension the Fourier-Transformation).

The selection of the misclassification cell in Figure 2-F transitions to the *Comparative Time Series View* and showing the underlying misclassified time series data, see Figure 2-D+E.

Comparative Time Series View – The view supports the user with the inspection of the raw time series data (6 dimensions) to verify a hypothesis. On the left, see Figure 2-D, the quantile graphs of the true and predicted classes with the selected misclassified samples, see Figure 2-E are plotted. Comparing the misclassified samples to the quantile graphs helps in this case to detect the features that are responsible for the misclassification; for instance, values vary from the usual distribution (Figure 2-G). In our case, see Figure 2-G, the first and the fourth feature fit rather the predicted class since the feature values are shifted downwards or better included in the quantile graphs. The not aligned features in Figure 2-G indicates a possible sensor error that is problematic for the classifier and needs to be analyzed to correct possible corrupt data.

In the shown use case, the first feature is constant for all samples and shows a sensor failure during this time interval. The fifth feature supports the claim as the feature overlaps with the predicted class. The claim was verified by our domain expert.

5 CONCLUSION

We presented a conceptual model-agnostic workflow for the visual debugging of multi-target multivariate time series classifiers. The top-down approach aims to support the visual debugging by depicting different levels of abstraction as well as the investigation of the propagation of information in the multi-target classifier. Our prototypical implementation with various views shows a possible realization of the workflow a visual investigation and comparison between possible causes of misclassifications. The usage scenario shows benefits of the workflow and implementation analyzing a real-world anonymized industry dataset. The current prototype implementation has some limitations that we plan to address in future work. We plan to investigate in a user study the computational and visual scalability of the implemented prototype by varying the number of targets, classes, as well as data records. The prototype needs an additional feedback loop to support the retraining and generation of new multi-target classifiers. More descriptive features, further, should be added to the enhanced confusion matrix view facilitating a more detailed comparison of classification errors.

ACKNOWLEDGMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892 and Siemens Corporate Technology.

REFERENCES

[1] B Alsallakh, A Hanbury, H Hauser, S Miksch, and A Rauber. 2014. Visual Methods for Analyzing Probabilistic Classification Data. *IEEE Transactions of Visualization and Computer Graphics* 20, 12 (2014), 1703–1712. https://doi.org/10.1109/tvcg.2014.2346660

[2] Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. *CHI Conference on Human Factors in Computing Systems* (2015), 337–346. https://doi.org/10.1145/2702123.2702509

[3] Fabian Bendix, Robert Kosara, and Helwig Hauser. 2005. Parallel sets: Visual analysis of categorical data. *IEEE Symposium on Information Visualization* 1 (2005), 133–140. https://doi.org/10.1109/INFVIS.2005.1532139

[4] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[5] E Oran Brigham and E Oran Brigham. 1988. *The fast Fourier transform and its applications*. Vol. 448. prentice Hall Englewood Cliffs, NJ.

[6] Mooi Choo Chuah and Fen Fu. 2007. ECG anomaly detection via time series analysis. In *International Symposium on Parallel and Distributed Processing and Applications*. Springer, 123–135.

[7] S. Havre, B. Hetzler, and L. Nowell. 2002. ThemeRiver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization*. IEEE Comput. Soc, 115–123. https://doi.org/10.1109/INFVIS.2000.885098

[8] Simon Haykin. 1998. *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[9] Alfred Inselberg. 1985. The plane with parallel coordinates. *The Visual Computer* 1, 2 (1985), 69–91. https://doi.org/10.1007/BF01898350

[10] Alfred Inselberg. 2009. Parallel Coordinates. In *Encyclopedia of Database Systems*. 2018—-2024. https://doi.org/10.1007/978-0-387-39940-9_262

[11] Im Jean-Francois, Michael J Mcguffin, and Rock Leung. 2013. GPLOM : The Generalized Plot Matrix for Visualizing Multidimensional Multivariate Data. *IEEE Transactions of Visualization and Computer Graphics* 19, 12 (2013), 2606–2614.

[12] Liu Jiang, Shixia Liu, and Changjian Chen. 2018. Recent Research Advances on Interactive Machine Learning. *Journal of Visualization* (nov 2018), 1–17. arXiv:1811.04548 http://arxiv.org/abs/1811.04548

[13] W Kienreich and C Seifert. 2012. Visual Exploration of Feature-Class Matrices for Classification Problems. *EuroVis Workshop on Visual Analytics* (2012), 0–4.

[14] Kyoung-jae Kim. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55, 1-2 (2003), 307–319.

[15] Mark Last, Alla Sinaiski, and Halasya Siva Subramania. 2010. Predictive Maintenance with Multi-target Classification Models. In *Intelligent Information and Database Systems*. 368–377. https://doi.org/10.1007/978-3-642-12101-2_38

[16] R. Keith Mobley. 2002. *An Introduction to Predictive Maintenance* (second edi ed.). Butterworth-Heinemann, Burlington. https://doi.org/10.1016/B978-075067531-4/50000-2

[17] David M W Powers. 2007. *Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation*. Technical Report December.

[18] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (dec 2011), 333–359. https://doi.org/10.1007/s10994-011-5256-5

[19] D Ren, S Amershi, B Lee, J Suh, and Jason D. Williams. 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 61–70.

[20] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A Keim. 2014. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 12 (dec 2014), 1604–1613. https://doi.org/10.1109/TVCG.2014.2346481

[21] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages*. Elsevier, 336—-. https://linkinghub.elsevier.com/retrieve/pii/B9781558609150500469

[22] J T Townsend and West Lafayette. 1971. Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics* 9, 1 (1971), 40—-50.

[23] Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-Label Classification. *International Journal of Data Warehousing and Mining* 3, 3 (jul 2007), 1–13. https://doi.org/10.4018/jdwm.2007070101

[24] Jun Wang, Bei Yu, and Les Gasser. 2002. *Classification Visualization with Shaded Similarity Matrix*. Technical Report.

[25] Min Ling Zhang and Zhi Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 7 (2007), 2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019