Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 130 (2024) 1801–1808

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Explanation of the Acoustic Features for Detecting a Cut Interruption in the Laser Cutting Process

Kathrin Leiner*[a,b], Tobias Bosse[d], Luca Keck[a], Marco F. Huber[b,c]

[a] *TRUMPF SE + Co. KG, Ditzingen, Germany*
[b] *Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Stuttgart, Germany*
[c] *Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Stuttgart, Germany*
[d] *Karlsruhe Institute of Technology, Karlsruhe, Germany*

* Corresponding author. *E-mail address:* Kathrin.Leiner@Trumpf.com

Abstract

The machine learning (ML) algorithm RandOm Convolutional KErnel Transform (ROCKET) is used to recognise a cut interruption during laser cutting to avoid rejects. For this purpose, an audio signal recorded at the laser cutting machine is categorised with help of ROCKET into three classes. These are: Good cut, transition region and cut interruption. However, when using ML algorithms, the user is not given any insight into the algorithm's decisions. It is not possible to develop an understanding of the process, i.e. information about what indicates a cut interruption in the audio signal. For this reason, ML algorithms are often referred to as black box models. In this paper, reROCKET is introduced to make the ROCKET models more transparent. The randomly generated ROCKET kernels are analysed by reverse engineering. The effectiveness of the kernels is determined by calculating the quality of class separation by applying a kernel. The best kernels are then identified and provide information about the features in the audio signal. The performance of a ROCKET model with 500 kernels and a conventional kernel approach is compared. With a single kernel transformation, results close to the prediction of a 500-kernel ROCKET model are obtained. This is particularly advantageous in real-time applications, as the computing time of a convolution is significantly less than that of the 500 kernels.

Keywords: Machine Learning Application; Laser Cutting; Time Series Classification; Explainable AI

1. Introduction

In practice, machine learning (ML) models are often viewed with scepticism. This is mainly due to the fact that the decision-making process is not comprehensible and transparent for the end user. In this context, ML models are often referred to as black box models [1]. In this paper, a new method for explaining the RandOm Convolutional KErnel Transform (ROCKET) algorithm is presented [2]. This explanatory method is called *reROCKET* (**re**verse **e**ngineering ROCKET). In order to establish a link to practice, the cut interruption detection in laser cutting is considered as an example application. Recognising cutting interruptions during laser cutting is important for the automation of the laser cutting process. Incorrectly cut parts that can no longer be removed from the sheet lead to rejects. In laser cutting, there is also a risk that entire batches will be rejected if such a cut breakage is not recognised. To avoid this unnecessary waste, monitoring systems are required, for which visual inspection systems have been developed in the past. These in turn can be categorised according to the underlying methodology. There are monitoring systems based on conventional image processing [3, 4] and on ML algorithms [5, 6, 7].

An alternative to visual cut interruption detection is acoustic cut interruption detection, as according to the findings of laser cutting experts, bad cuts can even be recognised with the human ear. For the purpose a microphone is installed into the cutting machine. On this acustic signal a ROCKET algorithm is trained. Then, this algorithm is able to distinguish between good cuts and cutting errors [8]. However, the obtained solution is a black box and thus, is not providing deeper insights and understanding of the audio signal and the underlying problem. For this purpose, the reROCKET algorithm is applied to the acustic cut interruption detection.

2212-8271
10.1016/j.procir.2024.10.319

The following paper begins with a analysis of related, relevant work. In Sec. 3 the problem statement is defined. Then, in Sec. 4, the labelling procedure, the ROCKET algorithm and its extension reROCKET are explained. Both methods are applied and compared in Sec. 5. The paper ends with a conclusion as well as the follow-up work in Sec. 6.

2. Related Work

In this section, the relevant state of the art will be discussed. The chapter will be divided into two topics addressed in this paper: the ROCKET algorithm and the topic of xAI.

2.1. ROCKET

ROCKET randomly initializes a large number of convolution kernels for feature extraction without training them. It is a fast and efficient approach for time series classification, utilizing a bank of random convolution kernels.[2] MiniROCKET is based on the same idea as ROCKET, but is designed for more efficient computation on resource-constrained devices. It aims to reduce the computational complexity, by aggregating the PPVs from multiple convolutional kernels.[9] A further development of ROCKET that selects less effective kernels from RCOKET is S-ROCKET. By iteratively optimizing the subset of active kernels based on their contribution to classification performance, S-ROCKET efficiently searches for the most effective kernels while discarding less important ones, thereby reducing computational complexity without sacrificing accuracy [10]. The final kernels used are not investigated further, there is no explainability of these.

2.2. Explainable AI

As mentioned in the literature [1], machine learning models are often referred to as black box models. To make these models more interpretable, the field of explainable AI (xAI) was established. [11] evaluates xAI methods that specifically use ROCKET for the classification of time series, using state-of-the-art techniques such as SHapley Additive exPlanations (SHAP) and dimension-wise Class Activation Map (dCAM). The paper focuses on multivariate time series. Further references for explanations of ROCKET are not known, but there are publications for the explanation of time series models such as [12], which presents dCAM for the explanation of multivariate time series classification. dCAM improves interpretability in CNNs by identifying discriminant temporal and dimensional data features. [13] presents a framework for explaining black-box time series classifiers using saliency maps, instance-based and rule-based explanations. [14] Introduces the timeXplain framework for explainability of time series classifiers. The author uses model-agnostic explainers such as SHAP for interpretable predictions in time series classifiers.

2.3. Summary: Related Work and Contribution of the Paper

As shown in this section, only one relevant publication was found to explain ROCKET. This is based on state-of-the-art algorithms that are more frequently used in the context of images. An interesting approach is the extension of s-ROCKET, as this approach reduces the high number of kernels in ROCKET by iterativly searching for kernel combinations.

The novelty of this paper lies in the human-interpretable explainability of the ROCKET algorithm, applied to a real-world industrial use case. To achieve this, we introduce reROCKET. The algorithm identifies a single kernel from ROCKET that is also capable of solving the classification task of the original problem. This kernel can then be analyzed and explained. An additional benefit, although not the primary focus of this paper, is the single use of the discovered individual kernel, resulting in faster application time.

3. Problem Statement

Two problems are presented in this paper. In the first case, acoustic cut detection must be implemented using ROCKET.

The audio signal recorded during laser cutting is given as a sound pressure level in Pascal (Pa) and is classified into two classes: Good Cut (class label OK) and Cut Interruption/Cut Problem (class label CP). The available data is labeled based on these two classes, which yields a supervised learning problem. For the training data set, the audio signal, with the length of m, is divided into smaller, non-overlapping windows with length l, resulting in the number of training windows $n_{\text{train}} = \lceil m/l \rceil$, $n_{\text{train}} \in \mathbb{N}$. For the test data set, the windows overlap with the length $u = l - s$, where $s \in (0, l)$ is the step size. Resulting in the number of windows $n_{\text{test}} = \lceil m-l/s + 1 \rceil$, $n_{\text{test}} \in \mathbb{N}$. Section 4.2 explains the advantage of overlapping windows in the test data. The ML model $M(x) = \hat{y}$ is trained on a data set $D = \{x_i, y_i\}$, $i = 1 \dots n_{\text{train}}$, where $y \in \{0, 1\}$ is the label with $y = 0$ being the label CP and $y = 1$ the label OK. The input data x_i are the audio windows with length l and the output $\hat{y}$ is the prediction of the label. The model is then tested on an independent test data set $T = \{x_i, y_i\}$, $i = 1 \dots n_{\text{test}}$.

In the second step, the trained ROCKET model $M(x)$ is used to analyse the convolutions it contains with kernels K_j. For this purpose, the kernels K_j are tested with help of reROCKET for their ability to separate the classes. After the analysis with reROCKET, the influence of the change in focus position on the prediction of the models $M_i(x)$ and how the kernels K_j change in response is investigated.

4. Material and Methods

This chapter presents the procedure used to generate the data sets. In addition, the ROCKET algorithm is introduced, together with a description of the proposed reROCKET extension.

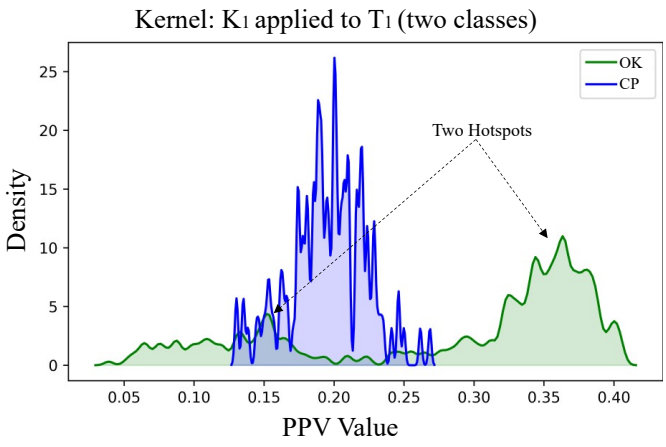

Abbildung 1: The distribution of ppv values for kernel K_1, for a two-class problem, applied to T_2 as an example. The second (local) maximum of the class OK (green) is striking; this indicates that there is another class in the data set.

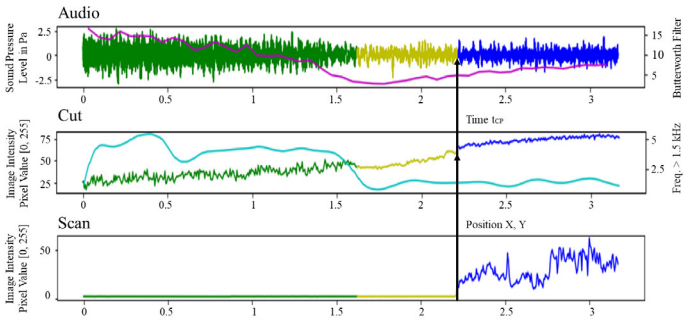

Abbildung 2: Illustration of the label procedure. Top: the labeled audio signal. Middle: the image intensity of the cutting images. Bottom: the image intensity of the scanning images.The data (images and audio) is labelled as a cut interruption (blue) according to the abrupt signal increase in the scan images. The transition area TN (yellow) can be labelled using the light blue and magenta coloured line. This is determined at the centre point of the lines.

4.1. Introducing a Three Class Problem

With the help of reROCKET, a three class problem is uncovered. As can be seen in Fig. 1, the visualisation of reROCKET can uncover subclasses that are noticeable through a second hotspot, which means a local maximum, within a class. Thus, in Fig. 1, a second hotspot can be seen for the class OK, which represents the data of the new class TraNsition area (class label TN). The transition is an area where the cutting quality decreases, burr formation and edge roughness occur. The data is now labeled based on these three classes. For the ML model $M(x) = \hat{y}$, the labels $y \in \{0, 1\}$ are changed into $y \in \{0, 1, 2\}$, where $y = 2$ being the label TN.

4.2. Labeling

In order to obtain labeled data, cut interruptions must be provoked. The tests are performed on the TruLaser5030. This machine is a 2D laser machine from the company TRUMPF. The cutting head is equipped with an Active Speed Control (ASC) [3]. This is an assistance system comprising a camera. The camera observes the laser beam hitting the sheet to be cut through the nozzle. Images of the cutting gap can thus be acquired during the cutting process. The images capture the so-called process light, i.e., the optical emissions of a certain wavelength range that occur during cutting. Based on this technique, a method was developed to automatically label the audio signal. The labeling method is shown in Fig. 2. During cutting, the images and audio signals are recorded simultaneously. Both signals are time synchronized. After the cut, the cutting head is set back to the initial position X_0, Y_0. The cutting gap is scanned again with the same coordinates X, Y. In this so-called scan phase, the laser power is reduced to 200 W. If there is material in the gap, the image intensity rises sharply. Everything after this rise is labeled as a cut interruption. The X, Y position can be mapped to the cut signal, from there it can be mapped to the audio with the timestamp t_{CP}. The recorded images are not used for this work, but they are necessary for the time synchronisation of the audio signal.

The transition area is much more difficult to label, since it behaves dynamically, e.g., due to heating of the sheets. For this purpose, two new label characteristics are introduced, which are displayed in light blue in the cut plot and in magenta in the audio plot in Fig. 2. During the tests, it became apparent that the images and also the audio in the good cut range fluctuates more. The fluctuations are visualized and the transition area is defined at the inflection point of the two curves.

After the labeling, the audio signal is cut into windows in preparation for the training. For the training, window sizes of $l = 1,000$ are selected, which corresponds to a time duration of 62.5 ms. The windows follow each other and there is no overlap in the training. This is necessary to avoid overfitting during training. However, overlapping windows are chosen for the test data set. This has the advantage of having less time delay in the real application. The windows overlap with a step size of $s = 32$, which corresponds to a time duration of 2 ms.

4.3. ROCKET

ROCKET was introduced by [2] as time series classifier. The algorithm is based on the principle of convolutions, which might be familiar to ML experts from the field of image processing, especially when using convolutional neural networks (CNNs). With the original ROCKET, 10,000 kernels are generated randomly, which is in contrast to CNNs, where the kernels are learned during the training. The sizes, biases, dilations, paddings and weights of these kernels differ within a given range of values. Each individual kernel is then convolved with the time series, in our case a window. After each convolution, two values are calculated. The first value is the *max value*, i.e., the maximum value of the convolution result. The second one is the *proportion of positive value* (ppv), which is newly introduced in ROCKET. It indicates the percentage of the values from the convolution that are positive [15]. After the transformation, the resulting $20k$ scalar feature values are utilized to train a $20k$ dimensional ridge classifier [16]. For the implementation of the

presented ROCKET, the `pyts` pipeline [17] is used which is based on the original ROCKET paper [2]. The pipeline contains the actual ROCKET transformer, a standard scaler [17] and the ridge regression classifier [16]. In the ROCKET transformer, a validation set is created from the training data by default and is used in the training step. Thus, when mentioning a ROCKET model, this always means the combination of the feature space transformer ROCKET and the ridge classifier. To analyse the performance of the model, the accuracy [18] score

$$\text{accuracy} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} 1(\hat{y}_i = y_i) \,, \tag{1}$$

from the `scikit-learn` is calculated, where n_{test} is the number of test samples and $1(\hat{y}_i = y_i)$ is the indicator function, which is one if the model prediction matches the real label.

4.4. Reverse Engineering ROCKET

Due to the large number of kernels and the resulting features, it is not possible to understand the classification result of ROCKET in terms of which audio features, such as frequencies or volume, are decisive for the cut interruption. However, customers and developers in particular want this transparency. This paper presents *reROCKET*, which visualizes the kernel transformation of a single kernel. reROCKET is further used to evaluate the extent to which a single kernel can distinguish between different classes. The audio signal can be analysed for the influence of this kernel. The reROCKET procedure is the following:

1. *Generate kernels:* j kernels are randomly created. The training of a ridge classifier does not take place.
2. *Convolution of the training data:* The training data D_i are convolved with each of the j ROCKET kernels created in step 1, with the goal of finding the kernel with the best behaviour. A kernel is considered "goodïf there is a significant difference in the distribution of features between classes. The idea of finding a function that best separates the features of the classes is described in [19] as 'quality measures for features'[1]. The selection of a suitable separating function is complex and in our case is done by reROCKET.
3. *Calculation of the feature areas:* The reROCKET calculation for the quality of a kernel is performed using a *kernel density estimate* (KDE) plot, as shown in Fig. 3. The KDE was invented by [20] and the KDE plot visualises the distribution of classes that occur. The *seaborn* library is used for the purpose. Two KDE plots are created per kernel j. One is the distribution of the resulting max values and the other is the distribution of the resulting ppv values.
   To simplify the evaluation, the KDE plot is smoothed. A Gaussian kernel is used to equalise the degree of smoothing and to avoid under- or over-smoothing, whereby the degree of smoothing is selected manually.
   Each KDE plot consists of c distribution curves, for each class one. The underlying area of each curve is defined as $A_{j,c}$. The areas $A_{j,c}$ can overlap, resulting in overlapping Areas $O_{j,v}$. The smaller the overlap $O_{j,v}$, the more effective the kernel j is in isolating the classes. The number of overlapping areas v of a kernel j depends on the number of classes. This leads to $v = \lceil c \cdot (c-1)/2 \rceil$, $v \in \mathbb{N}$ overlapping areas for each kernel j, where c is the number of classes.
4. *Calculation of the overlap areas:* After calculating the overlap areas $O_{j,v}$, the kernel with the least overlap areas is determined. For this purpose, the maximum values of $O_{j,v}$ and the mean value of $O_{j,v}$ are calculated.
5. *Determine the best kernel:* There are three options for selecting the best kernel: averaging the $max(O_{j,v})$ and $mean(O_{j,v})$ for the ppv feature space to obtain the best ppv kernel, or averaging the best $max(O_{j,v})$ and $mean(O_{j,v})$ for the max feature space to calculate the best max kernel. The third option, which was chosen in this paper, involves calculating the $max(O_{j,v})$ and $mean(O_{j,v})$ over both the ppv and max feature spaces, and then averaging them. Regardless of the chosen option, the best kernel is determined by selecting the kernel j with the lowest averaged value. Each kernel j has two corresponding KDE plots, which can be used to display the feature space in relation to either the max value or the ppv value.
6. *Visualize the best kernel:* To create an understanding of the process, various diagrams are created. A histogram diagram is created for the best kernel in the two feature spaces max and ppv. Furthermore, the kernel is plotted over a sweep signal, i.e., a signal over all frequencies..

If a kernel is to be used to predict input data, this is also done with the KDE plot of the desired kernel, exemplary shown in Fig. 3. The sample to be predicted is convolved with the selected kernel, resulting in a max value or a ppv value, depending on which characteristic the kernel was selected. The resulting value is plotted on the x-axis. Now the y-values of the class

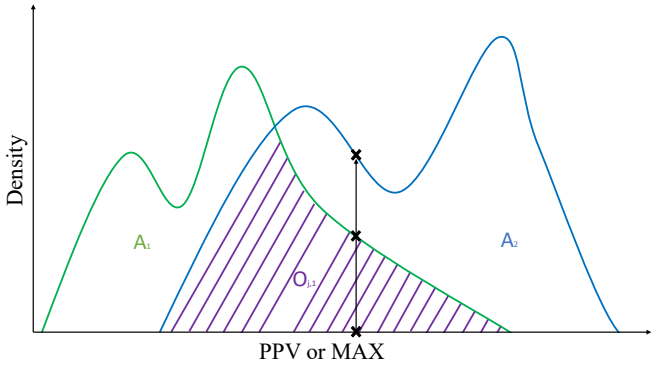

Abbildung 3: Basic illustration for calculating the overlap area A for one kernel and two classes. The overlapping area is referred to as O. The graphic can be created for the kernel in PPV or Max feature Space. To calculate a sample, the calculated PPV or Max value is plotted on the x-axis and the line of the higher y-value is taken.

[1] Translated from German 'Gütemaße für Merkmale'.

curves are compared for this x-value. The sample is assigned to the class with the larger y-value.

5. Experiments

This section describes the tests performed. Firstly, the various training and test data sets created are explained. The ROCKET models are trained on these, reROCKET is applied and the one-kernel is determined.

5.1. Data Sets

For the experiments, mild steel sheets with a thickness of 6 mm are cut. A total of 78 cuts are made, from which three training and test data sets D_{1-3} and T_{1-3}, respectively, are recorded, see Table 1. To generate as much data as possible, cutting interruptions were provoked. So-called feed ramps were cut. For this the laser cutting speed is increased linearly from 100% to 125%. Above a certain speed, the energy used is no longer sufficient to cut the sheet metal and the cut is interrupted. As the sheets become hot and the cuts are directly adjacent to each other, a cut interruption can occur at different feed speeds. In order to avoid overfitting and to represent a real laser cutting process in the data as far as possible, disturbance variables were introduced for $(D2, T2)$ and $(D3, T3)$. The focus position was changed in consultation with laser cutting experts. This can also occur in reality due to heating of the optics or incorrect adjustment of the machine. Adjusting the focus position has a major influence on the process. An adjusted focus results in poorer cutting edge quality and cut interruptions occur even at low speeds [21]. The present test setup can be summarised as follows:

- D_i represents the training test set, where $i = 1 \dots 4$.
- $D_1 \dots D_3$ are included analogue to Table 1 and $D_4 = \bigcup_{i=1}^{3} D_i$.
- M_i stands for the model that was trained on the training data set D_i
- T_p stands for the test set, where $p = 1 \dots 5$.
- T_4 is defined as $T_4 = \bigcup_{p=1}^{3} T_p$.
- The models $M_1 \dots M_4$ are also tested on a test set T_5 which consists of a set T_p, where the indices $p = i$ of the model M_i are omitted.

5.2. Training and Test Results

As mentioned in [8], the generation of random kernels has a large impact on the performance of a model. Random kernels can sometimes be generated better and sometimes worse for a model. To avoid randomly generating a bad model, the best model is selected from five training iterations. For our tests, four models $M_1, \dots, M_4$ each are trained on $D_1, \dots, D_4$, where D_4 is the mixture of all test datasets. The Table 2 shows the training results of the best model $M_1, \dots, M_4$. As expected, the performance of the models is best on their own associated test data T_i. The performance of the models tested on T_5, the test data set without the training setup, decreases. Model M_4 has the best overall performance on the mixed data set T_4. However, it is interesting to note in this experiment that the overall performance of model M_3 is similar to that of model M_4. The performance of M_3 on the test data sets T_1 and T_2, i.e. the disturbances not contained in its own training data set, is also the highest. The performance of M_1, M_2 and M_4 on T_3 is always the lowest. This leads to the assumption that the disturbance variable, the reduction of the focal point, has the greatest influence on the data. The good performance of M_3 can be explained by the fact that the training data has the highest variance and therefore the model is the most generalisable.

Tabelle 1: Overview of the generated data records, where the abbreviations FC stand for focus position change.

Name	Disturbance	# of Cuts	# of Windows
D_1	None	17	246
T_1	None	3	4,655
D_2	FC +1	18	303
T_2	FC +1	4	6,171
D_3	FC -0.5	19	510
T_3	FC -0.5	4	6,084

5.3. Application of reROCKET

After the training of all models $M_1, \dots, M_4$ is completed, the best kernel K_i is determined for each model M_i according to the procedure describe in Sec 4.4. Then the four resulting kernels $K_1, \dots, K_4$ are applied to the test data sets $T_1, \dots, T_5$, the results comparing to the performance of the models are shown in Table 2. Because ROCKET represents the data with the features max and ppv for each feature representation an accuracy is calculated, resulting in $maxK_i$ and $ppvK_i$.

Comparing the performance of $ppvK_1$ and $ppvK_4$, it can be seen that a single kernel achieves a similar good performance as the corresponding models M_1 and M_4, respectively, trained with 500 kernels and the ridge classifier. Since kernel $ppvK_1$ has the best overall performance, we consider it in Fig. 4 and Fig. 5. In Fig. 4 we can see the distribution of the classes of T_1 convoluted with K_1, here the kernel $ppvK_1$ performs with an accuracy of 90.72 %. We can see that the kernel separates these classes well. However, there is still some overlap between classes. This could be due to measurement errors, or more specifically, labelling errors. It cannot be ruled out that individual data are assigned to the wrong class, as the transition area in particular is difficult to label. Another reason may be that not every cut interruption behaves the same. A cut interruption may be stronger or weaker. This will affect the recorded audio and the recognisable features and may also lead to misclassification. Fig. 5 shows the distribution of the T_3 data convoluted with $ppvK_1$. It can be seen that there is an overlap between TN and CP, which explains the drop in performance to 80.67 %.

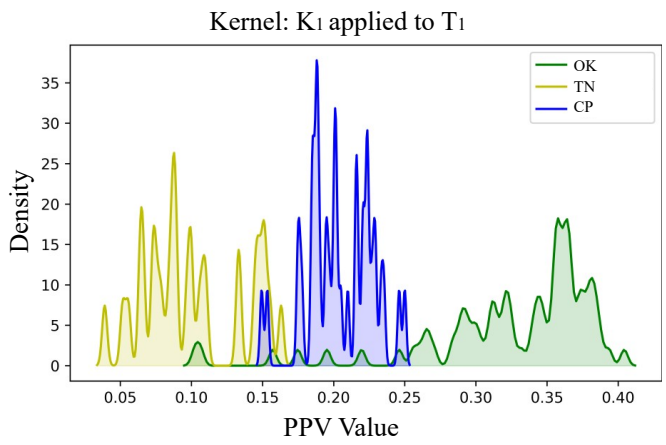

Abbildung 4: The distribution of ppv values for the kernel K_1, applied to T_1 as an example. There is a good separation in the feature space of the three classes, as the areas overlap slightly.

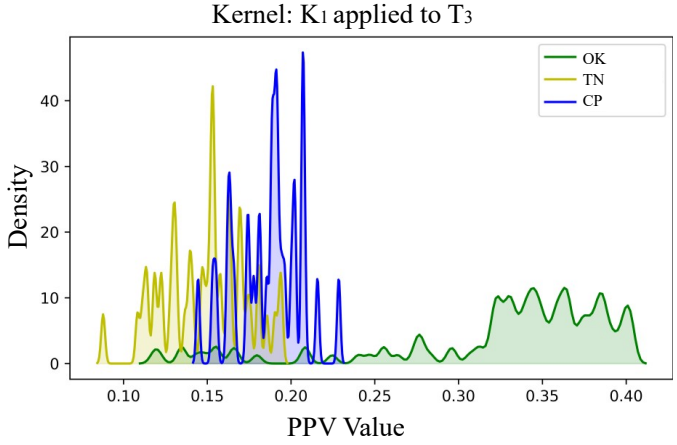

Abbildung 5: The kernel K_1 is applied to the test data set T_3. Compared to Fig 4, the separation appears weaker. There is an overlap between the CP and TN areas. The OK range extends over all PPV values.

Tabelle 2: Accuracy of the trained models with 500 kernels M_1, M_2, M_3, M_4 and the one kernel K_1, K_2, K_3, K_4 accuracy without a ridge classifier training.

M/D	T_1	T_2	T_3	$T_4 = \bigcup_{i=1}^{3} T_p$	$T_5 = \bigcup_{p=1}^{3} T_p \setminus T_p$ where $p = i$ of M_i
M_1	0,9094	0,7510	0,7531	0,7950	0,7521
$MaxK_1$	0,8975	0,7776	0,7263	0,8005	0,7520
$PPVK_1$	0,9072	0,8590	0,8067	0,8576	0,8329
M_2	0,8269	0,9133	0,7119	0,8165	0,7694
$MaxK_2$	0,6195	0,9090	0,5202	0,6829	0,5699
$PPVK_2$	0,6614	0,8837	0,5488	0,6980	0,6051
M_3	0,8934	0,8420	0,8558	0,8610	0,8677
$MaxK_3$	0,7233	0,6940	0,7715	0,7296	0,7087
$PPVK_3$	0,7536	0,7623	0,8096	0,7752	0,7579
M_4	0,9029	0,9034	0,8308	0,8769	None
$MaxK_4$	0,8442	0,7491	0,7922	0,7952	None
$PPVK_4$	0,8795	0,8527	0,8061	0,8461	None

5.4. Analysis of the Best Kernels

Now that the main kernels $K_1, \ldots, K_4$ have been identified, the actual functionality of the kernels is analysed in this section. A sweep signal is used for this purpose. This is a rising logarithmic sine wave from 10 Hz to 8 kHz. The effect on the corresponding frequency can be derived by comparing the signal before convolution with the kernel and after convolution with $20 \cdot \log\left(\frac{|\mathrm{FFT}_{\mathrm{out}}(f)|}{|\mathrm{FFT}_{\mathrm{in}}(f)|}\right)$ for each corresponding frequency in the Fourier transformed signals. The greater the gain of a frequency in this kernel, the more significant it becomes in distinguishing these classes.

Fig. 6–9 show the convolution of the kernels $K_1, \ldots, K_4$ with the sweep signal. All kernels show the behaviour of a high-pass filter. It is noticeable that K_1, K_3 and K_4 have a zero crossing at 1,000 Hz, K_2 only at about 4,000 Hz. This explains the poor performance of K_2 on T_1 and T_3. It can now be assumed that the critical features for distinguishing the classes of the given audio signal lie over a frequency range of 1,000 Hz. In the following, we relate the results of Figures 6-9 with the results shown in Table 2. The performance of the kernels T_3 and T_4, reveal that $ppvK_1$ and $ppvK_4$ work with an accuracy of over 83 %, while $ppvK_2$ and $ppvK_3$ have an accuracy of less than 76 %. If we now look at the damping curves of K_1, K_4 and K_2, K_3, it is noticeable that the kernels with an damping below 1,000 Hz and a gain above this perform better. K_4, which performs best according to the measured accuracy, also has an damping of the frequencies ~1.9 kHz and 4.1 kHz.

Another advantage of the single-kernel approach is the improved computing time for real-time applications. Experiments were carried out on the industrial-PC of the laser cutting machine. These showed that a ROCKET model requires up to 85 ms inference time to analyse an audio window. In contrast, the calculation with one kernel K takes less than 0.2 ms.

6. Conclusion and Future Work

With the help of reROCKET, it was possible to gain a deeper understanding of the process. A third class, the transition region, was found by analysing the feature space. The influence of

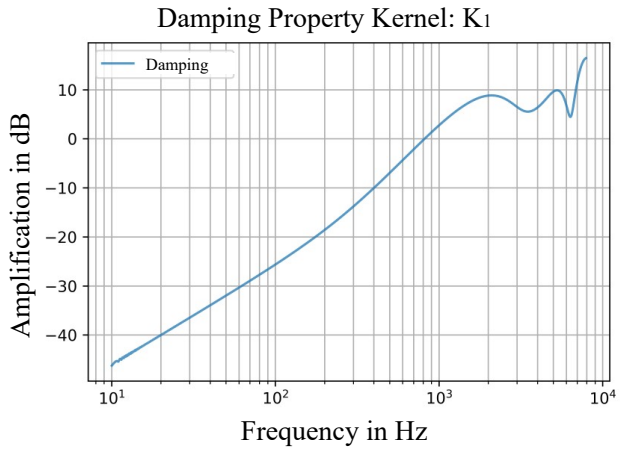

Abbildung 6: Damping characteristic of the most efficient kernel K_1 derived with reROCKET from the model M_1. A high-pass behaviour can be seen, as frequencies above 800 Hz are amplified. Frequencies below 800 Hz are attenuated

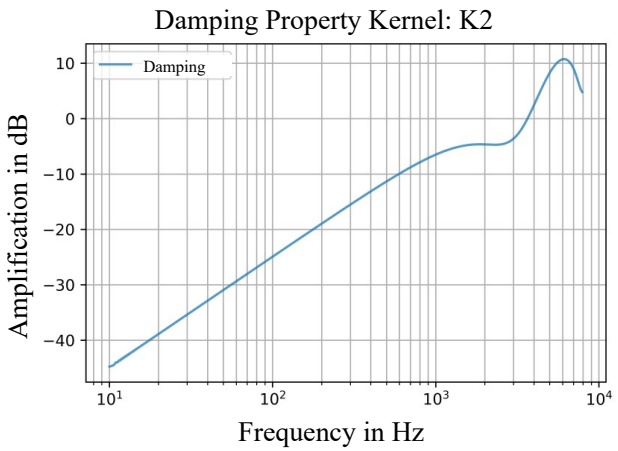

Abbildung 7: Damping property of the best kernel K_2 derived with reROCKET from the model M_2. A high-pass behaviour can be seen, as frequencies are amplified from around 2.5 kHz.

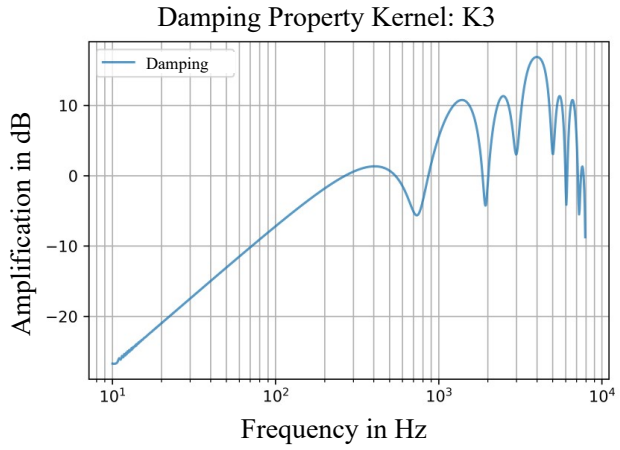

Abbildung 8: Damping property of the best kernel K_3 derived with reROCKET from the model M_3. No normal high-pass behaviour can be seen. Frequencies between 200 Hz and 500 Hz are amplified. Above 800 Hz there is an unsteady amplification and damping of individual frequencies.

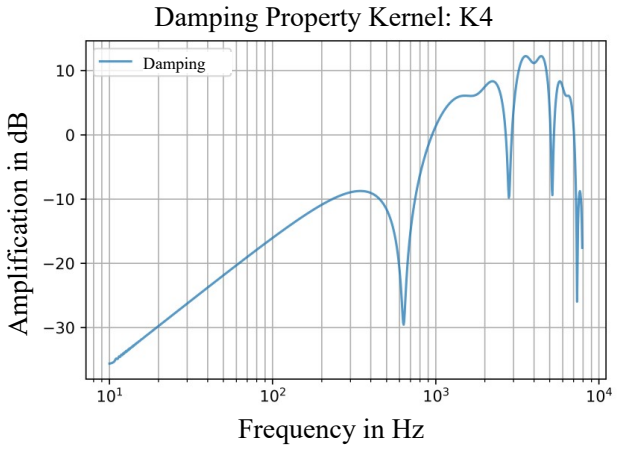

Abbildung 9: Damping properties of the most powerful kernel K_4 derived with reROCKET from the model M_4. Compared to the other Figs. 6-8, some frequencies (such as 650 Hz or 2 kHz or 4 kHz) are strongly damped. Other frequencies such as 1000-1.5 kHz and 2.5 kHz to 3.5 kHz are strongly amplified.

the focus change on the audio signal could be shown and explained with the help of the kernel analysis. With reROCKET, an alternative to the ROCKET models was found, the one-kernel approach. This offers the advantage of being able to analyse the kernels and considerably shortens the inference time on the machine.

As mentioned in the Sec. 5.3, it cannot be ruled out that the labeling process plays a role in the accuracy and evaluation of the model. In [22], the influence of independent labelers is discussed, and data consistency checks are conducted. This would also be an interesting experiment for the presented dataset in the future.

The one-kernel approach is to be used in future to implement a feed rate controller. This should adjust the laser cutting speed so that the model output is always in the OK range. With the results presented here, this would be an OK ppv target range between 0.25 and 0.4. The advantage would be that the influence of the disturbance variables is negligible, as the target OK range for the disturbance variables is constant. If the model output falls below the value 0.25, the machine is stopped.

Acknowledgment

The research and development project “Self-learning machine tools for highly efficient production (AutoLern)” is funded by the German Federal Ministry of Education and Research (BMBF) within the “Future of Value Creation - Research on Production, Services and Work” (funding numbers 02P20A020 to 02P20A026) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

In order to improve the writing style and provide the reader a better understanding of the text, DeepL Translator and DeepL Write were used for this work.

Literatur

[1] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, Jan. 2021. [Online]. Available: https://doi.org/10.1613/jair.1.12228

[2] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, Jul. 2020. [Online]. Available: https://doi.org/10.1007/s10618-020-00701-z

[3] TRUMPF. (2023) Autonomes laserschneiden. [Online]. Available: https://www.trumpf.com/en_SG/products/machines-systems/2d-laser-cutting-machines/webspecial-autonomes-laserschneiden/autonomous-laser-cutting-web-special/products-and-functions/

[4] M. Schleier, C. Esen, and R. Hellmann, "Evaluation of a cut interruption algorithm for laser cutting steel and aluminum with a high-speed camera," *Applied Sciences*, vol. 13, no. 7, p. 4557, Apr. 2023. [Online]. Available: https://doi.org/10.3390/app13074557

[5] B. Adelmann, M. Schleier, and R. Hellmann, "Laser cut interruption detection from small images by using convolutional neural network," *Sensors*, vol. 21, no. 2, p. 655, Jan. 2021. [Online]. Available: https://doi.org/10.3390/s21020655

[6] B. Adelmann and R. Hellmann, "Simultaneous burr and cut interruption detection during laser cutting with neural networks," *Sensors*, vol. 21, no. 17, p. 5831, Aug. 2021. [Online]. Available: https://doi.org/10.3390/s21175831

[7] N. Peghini, A. Zignoli, D. Gandolfi, P. Rota, and P. Bosetti, "Real-time cross-dataset quality production assessment in industrial laser cutting machines," in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds. Cham: Springer International Publishing, 2021, pp. 490–505.

[8] K. Leiner, F. P. Dollmann, M. F. Huber, M. Geiger, and S. Leinberger, "Cut interruption detection in the laser cutting process using ROCKET on audio signals," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. IEEE, Jul. 2023. [Online]. Available: https://doi.org/10.1109/indin51400.2023.10218267

[9] A. Dempster, D. F. Schmidt, and G. I. Webb, "Minirocket: A very fast (almost) deterministic transform for time series classification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, ser. KDD '21. ACM, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1145/3447548.3467231

[10] H. Salehinejad, Y. Wang, Y. Yu, T. Jin, and S. Valaee, "S-rocket: Selective random convolution kernels for time series classification," 2022. [Online]. Available: https://arxiv.org/abs/2203.03445

[11] D. I. Serramazza, T. T. Nguyen, T. L. Nguyen, and G. Ifrim, "Evaluating explanation methods for multivariate time series classification," 2023. [Online]. Available: https://arxiv.org/abs/2308.15223

[12] P. Boniol, M. Meftah, E. Remy, and T. Palpanas, "dcam: Dimension-wise class activation map for explaining multivariate data series classification," in *Proceedings of the 2022 International Conference on Management of Data*, ser. SIGMOD/PODS '22. ACM, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1145/3514221.3526183

[13] F. Spinnato, R. Guidotti, A. Monreale, M. Nanni, D. Pedreschi, and F. Giannotti, "Understanding any time series classifier with a subsequence-based explainer," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 2, p. 1–34, Nov. 2023. [Online]. Available: http://dx.doi.org/10.1145/3624480

[14] F. Mujkanovic, V. Doskoč, M. Schirneck, P. Schäfer, and T. Friedrich, "timexplain – a framework for explaining the predictions of time series classifiers," 2020. [Online]. Available: https://arxiv.org/abs/2007.07606

[15] B. Auffarth, *Machine Learning for Time Series with Python Forecast, predict, and detect anomalies with state of the art machine learning methods*. Packt Publishing Ltd, 2021.

[16] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, p. 80–86, Feb. 2000. [Online]. Available: http://dx.doi.org/10.1080/00401706.2000.10485983

[17] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, "Api design for machine learning software: experiences from the scikit-learn project," 2013. [Online]. Available: https://arxiv.org/abs/1309.0238

[18] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, p. 27–38, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2008.08.010

[19] H. Niemann, *Klassifikation von Mustern* -. Berlin Heidelberg New York: Springer-Verlag, 2013.

[20] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, p. 1065–1076, Sep. 1962. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177704472

[21] H. Hügel and T. Graf, *Materialbearbeitung mit Laser*. Wiesbaden: Springer Fachmedien Wiesbaden, 2022.

[22] B. Scheffler, P. Bründl, H. G. Nguyen, M. Stoidner, and J. Franke, "A dataset of electrical components for mesh segmentation and computational geometry research," *Scientific Data*, vol. 11, no. 1, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1038/s41597-024-03155-w