

# PGNAA Spectral Classification of Aluminium and Copper Alloys with Machine Learning

Henrik Folz\*, Joshua Henjes\*, Annika Heuer\*, Joscha Lahl\*, Philipp Olfert\*, Bjarne Seen\*, Sebastian Stabenau\*, Kai Krycki, Markus Lange-Hegermann, Helmand Shayan

**Abstract**—In this paper, we explore the optimization of metal recycling with a focus on real-time differentiation between alloys of copper and aluminium. Spectral data, obtained through Prompt Gamma Neutron Activation Analysis (PGNAA), is utilized for classification. The study compares data from two detectors, cerium bromide ( $\text{CeBr}_3$ ) and high purity germanium (HPGe), considering their energy resolution and sensitivity. We test various data generation, preprocessing, and classification methods, with Maximum Likelihood Classifier (MLC) and Conditional Variational Autoencoder (CVAE) yielding the best results. The study also highlights the impact of different detector types on classification accuracy, with  $\text{CeBr}_3$  excelling in short measurement times and HPGe performing better in longer durations. The findings suggest the importance of selecting the appropriate detector and methodology based on specific application requirements.

**Index Terms**—PGNAA spectral classification, maximum likelihood classifier, conditional variational autoencoder, cerium bromide detector, high purity germanium detector

## I. INTRODUCTION

IN 2019, humankind mined 62.9 million tonnes of aluminium and 20.7 million tonnes of copper in the world [1]. The worldwide recycling rate of End of Life products from copper is 45% [2], while the rate for aluminium is 70% [3]. With recycling of aluminium 95% of energy and 90% of  $\text{CO}_2$  can be saved compared with mining of aluminium [3] [4]. This shows the increasing importance of developing efficient recycling processes.

For recycling of scrap metal into *high-quality* alloys, one needs to know the atomic composition of the scrap in detail. Currently, there are mainly sensor methods that can classify scrap metal using traditional chemical processes or Laser Induced Breakdown Spectroscopy (LIBS) [5]. Both methods only consider the analysis of the scrap metal surface instead of integrally measuring the scrap metal composition and chemical methods even destroy the material.

Prompt Gamma Neutron Activation Analysis (PGNAA) examines the integral metal composition without destruction. It excites atoms by a neutron field, such that the de-excitation leads to measurable gamma radiation, which is characteristic

All first-contributors are with the Department of Electrical Engineering and Computer Science, OWL University of Applied Sciences and Arts, Lemgo, 32657 Germany.

Kai Krycki is with the Department of Aerospace Engineering, FH Aachen - University of Applied Sciences, Aachen, 52064 Germany.

Helmand Shayan and Markus Lange-Hegermann are with the inIT - Institute Industrial IT, OWL University of Applied Sciences and Arts, Lemgo, 32657 Germany.

\* equal contribution, alphabetical order

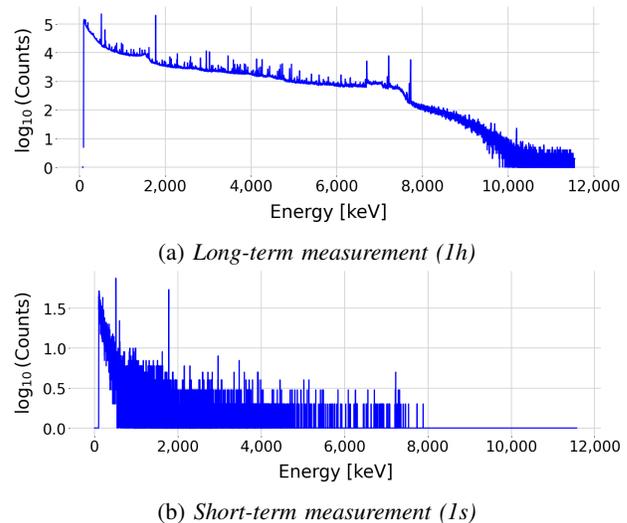


Fig. 1: Example prompt gamma spectra of the gamma quanta of an aluminium alloy measured with a HPGe detector as basis for classification. The long-term measurement of 1h in (a) can easily be classified into a specific aluminium alloy by considering the clearly recognizable characteristic peaks. The short-term measurement of 1s in (b) is very noisy and statistical methods cannot recognize the characteristic peaks. Classifying alloys by such spectra is the goal of this paper.

for the specific element [6]. As you can see in Fig. 1, we use this spectral data for each alloy. For each measured energy level, we can see the measured amount of gamma radiation as "Counts". The different elements of which the alloys consist cause peaks in the PGNAA spectra at different energy levels, by which it is possible to classify the different alloys using a long-term measurement. However, short-term measurements result in very noisy spectra without clearly detectable peaks, as shown in Fig. 1b.

The paper [7] shows that a classification of scrap metal with PGNAA is possible in short measurement times, based on the assumption of 50,000 counts per second for sampling. There, the spectrum is viewed as a categorical probability distribution of the energy levels and this probability distribution can be estimated reliably by a long-term measurement. Using this distribution, one can sample more spectral data to train machine learning algorithms and use the maximum likelihood method for real-time scrap metal classifications.

It remains open which gamma detectors are best suited for PGNAA-based scrap metal detection in real time and

how machine learning methods can be improved for even quicker classification results. Hence, this paper focuses on the optimization of metal recycling using new sensor solutions in combination with machine learning algorithms, with a specific focus on the differentiation of copper and aluminum alloys in real time.

We compare two different types of gamma detectors, one is a semiconductor detector equipped with a germanium crystal, called high purity germanium (HPGe) detector, the other a scintillator detector based on cerium bromide ( $\text{CeBr}_3$ ). Selecting the appropriate detector requires careful consideration of cost, measurement time, and accuracy. While the HPGe detector is very expensive, the  $\text{CeBr}_3$  detector is more affordable, but might have a worse performance. The detectors measure a different amount of counts per second and have a different energy resolution. In addition to the sampling methods in [7], we test further methods for data generation (Section III-A) such as Conditional Variational Autoencoder (CVAE). Furthermore, we evaluate various preprocessing methods like reducing and aggregating energy levels or weighting of different energy levels and classification methods like a Maximum Likelihood Classifier (MLC), XGBoost (XGB), Logistic Regression (LR) and a Linear Support Vector Machine (Linear SVM).

We achieved the best classification results with the MLC together with CVAE as data generation. When working with short measurement times, the  $\text{CeBr}_3$  detector obtained better results than the HPGe detector. At high measurement times, this phenomenon occurred in reverse.

## II. DATA AND DETECTORS

We analyse different metal alloys, which differ mainly in their content of aluminium or copper. Two different detectors were used to identify the materials. On the one hand the high purity germanium detector (HPGe), where a very high energy resolution can be achieved when detecting gamma rays [8] and on the other hand the cerium bromide detector ( $\text{CeBr}_3$ ), which has a greater sensitivity and can be used with room-temperature [9]. The HPGe detector has a resolution of 16,384 energy levels, while the  $\text{CeBr}_3$  detector has a resolution of 2,048. The different data available to us can be seen in Table I. For every material, we have spectra for five different alloys.

TABLE I: Overview of the recorded data for copper and aluminium in different states and with different types of detectors.

material	state	detector	counts per second
copper	block	HPGe	30,000
aluminium	block	HPGe	19,000
aluminium	chips	HPGe	7,000
aluminium	chips	$\text{CeBr}_3$	11,000

Measurements were sometimes taken using metal in block form and other times using metal chips. Metal blocks are solid pieces of metal formed into specific shapes or sizes, often used as raw materials or components in manufacturing processes. Metal chips refer to small pieces or shavings of metal generated during machining or cutting operations, typically considered waste material. The chips have a lower density as they were measured in a cup with air between the different pieces. As the block only consist of the metal, the density

and the weight of the metal are higher. This phenomenon can be seen in Table I, as the counts per second for block are significantly higher than for the chips.

## III. METHODS

In the following, we describe our approach to determine the best machine learning method for classifying the composition of metal scrap using different PGNAA methodologies. For most of the classification methods, we use the scikit-learn library [10], except for the neural network, where we use PyTorch [11]. For data generation with a Conditional Variational Autoencoder (CVAE) and data preprocessing with a Denoising Autoencoder (DAE), we use TensorFlow [12].

### A. Data Generation

We were able to achieve our results by using only one given long-term measured spectrum  $S$  ( $\geq 1h$ ) per alloy. We used these to generate short-term spectra  $s$  ( $< 10s$ ) for the training of our machine learning methods. This is an advantage, since real measurement data is expensive to produce and therefore only a limited number of measurements are available. For this, we have two different approaches which are explained below.

#### 1) Categorical Sampling

For this data generation method, we create a categorical probability distribution from the long-term measurement  $S$ . To obtain this distribution, we divide the counts  $S = (c_1, \dots, c_n)$  per energy level  $(1, \dots, n)$  by the sum of all counts ( $u = c_1 + \dots + c_n$ ). As a result of this, we get the relative frequencies  $(\frac{c_1}{u}, \dots, \frac{c_n}{u})$ , which serve as an estimator for the categorical probability distribution  $p(x|S)$  of a single gamma quantum of energy  $x$ . We use direct independent sampling from this distribution to generate test data, which closely replicates real-world conditions. To ensure training and test data diversity, the long-term measurements are divided into six shorter measurements by using dependent sampling. With the probability distributions generated as described above from these six measurements the short-term training spectra are generated. This enhances model robustness and mitigates overfitting risks. A longer simulated measurement time leads to more frequent sampling from the long-term spectrum, which results in a higher number of simulated counts.

#### 2) Conditional Variational Autoencoder (CVAE)

As an alternative for data generation, we use a CVAE, which is a modification of the Variational Autoencoder (VAE) [13]. A VAE consists of a probabilistic encoder  $q_\phi(z|s)$  and a probabilistic decoder  $p_\theta(s|z)$ , where  $s$  is a (short-term) spectrum and  $z$  is the latent representation. The probability distributions of the encoder and decoder are dependent on the parameters  $\phi$  and  $\theta$ , which are learned during training. The decoder can also be referred to as a generative model. To generate artificial data, we sample latent variables from the prior  $p(z)$  and feed them into the trained decoder  $p_\theta(s|z)$ , where we sample again. With a CVAE [14], we additionally feed a label for the long-term spectrum  $S$  to the decoder  $p_\theta(s|z, S)$  in order to generate samples for a certain class. For the encoder and decoder, we chose fully-connected neural networks with a single hidden layer, as in [13]. For the training of the CVAE we use 10,000

sampled spectra. When calculating the loss, we use a  $\beta$ -parameter [15] and set  $\beta = \frac{N}{M}$  with  $N$  being the size of the input  $s$  and  $M$  being the size of the latent space  $z$ .

## B. Preprocessing

### 1) Scaling

Initial tests indicated that using the standard scaler is beneficial for the models XGBoost and Neural Network. For the other models, we use unscaled data.

### 2) Subsetting

Subsetting involves the reduction and aggregation of energy levels by using just a subset of the spectra. An application could be the deletion of higher energy levels from the spectra. This technique was applied on spectra obtained from the HPGe detector.

### 3) Denoising Autoencoder (DAE)

A DAE [16] is a modification of the traditional Autoencoder where the input is corrupted in order to achieve robustness to small changes in input. We use a DAE to produce long-term measurements out of short-term measurements. A classifier, which was trained using long-term measurements, should now be able to classify the output of the DAE. In [17], a similar approach is used for angle-resolved photoemission spectroscopy data. We use the fully-connected neural network architecture from [18] and test different numbers of neurons in each layer.

### 4) Escape and Double Escape Peaks (EPs and DEPs)

EPs are characteristic features in gamma spectra that occur when measuring gamma rays. They arise due to interactions between the gamma rays and the detector material [19]. These are detection events where not all of the photon's energy is converted in the detector, but a certain, consistently sized portion of the energy is detected. If the peak at energy  $E$  is above 1022 keV, an Escape Peak at  $E - 511$  keV can occur in the spectrum [20]. This results in an additional peak in the spectrum at a correspondingly lower energy. Similarly, a DEP can arise from an EP at  $E - 1022$  keV. As the location of the EPs and DEPs depend on the material, a different weighting of them can help classifying the different materials. We only use this method with Maximum Likelihood Classifier (MLC), because here the reference spectrum can be weighted dependent on the EPs and DEPs of the alloy.

### 5) Unique Peaks

A possibility to improve the classification is the weighting of peaks, which only appear at one alloy. Therefore, we detected those peaks at the long-term measurements and accordingly weighted the different training and test spectra.

## C. Classification

We did each test five times and then reported the average to counteract randomness. Unless otherwise stated, we used 10,000 spectra for training. In order to achieve sufficient representativeness, we worked with 5,000 test spectra each.

### 1) Maximum Likelihood Classifier (MLC)

With MLC [21] we compare the categorical probability distribution from the short-term measurement to the different long-term measurements to find the best fitting distribution.

To classify the short-term measurement  $s$ , take the number of short-term photons measured  $s = (c'_1, \dots, c'_n)$  per energy level and multiply it by the appropriate log relative probability of the long-term measurement  $S$  to obtain the log-likelihood:

$$\log(p(s|S)) = \sum_{i=1}^n c'_i \cdot \log\left(\frac{c_i + 1}{\sum_{i=1}^n (c_i + 1)}\right).$$

Since there were difficulties in the case when  $c_i = 0$ , we add the value 1 to each number of photons of the long-term measurements.

Instead of just using the distribution of the long-term measurement, we use measurements generated by the Categorical Sampling and vary time and amount of reference distributions. In case of multiple distributions per alloy, the mean of the log-likelihoods is taken and compared to classify the short-term measurement  $s$ . The best results were achieved using 500 reference spectra per alloy, each corresponding to a measurement time of 1,800 seconds.

### 2) XGBoost (XGB)

For its capacity to handle large datasets and model complex patterns, we selected XGBoost, a robust ensemble classifier using gradient boosting [22]. Initial tests favor its linear-based [23] variant over tree-based methods in our context, hence adopting it exclusively for our purpose.

### 3) Logistic Regression (LR)

It is a statistical approach for modeling binary outcomes based on predictor variables. It extends linear regression to the logistic function, making it suitable for categorical predictions. The training algorithm uses the one-vs-rest (OvR) scheme for a multiclass case.

### 4) Linear Support Vector Machine (Linear SVM)

The SVM obtained the best results with a linear kernel. Therefore, we used the linear SVM to improve the runtime [24].

### 5) K-nearest Neighbors (KNN)

The KNN algorithm is based on the nearest neighbor decision rule proposed by Cover and Hart [25]. It operates on the principle of identifying the nearest neighbor(s) of an example and assigning the example to the same class as its nearest neighbor(s).

### 6) Radius Neighbors Classifier (RNC)

The RNC, an extension of the nearest neighbor decision rule [25], classifies examples by considering all data points within a specified radius. By assessing the proximity of points within the defined radius, the algorithm assigns a class label based on local patterns.

### 7) Neural Network (NN)

We used the PyTorch library [11] for the implementation of a fully connected, 4-layer network. The exact structure is described in more detail in the appendix A.

### 8) Random Forest Classifier (RFC)

The RFC [26] is an ensemble learning method that combines multiple decision trees to make predictions.

### 9) Extra Trees Classifier (ETC)

The ETC [27] builds an ensemble of decision trees with an extra level of randomness in comparison to the RFC. This randomness is used to reduce over-fitting and speeding up training [27].

### 10) Kuiper test (Kui)

It is a statistical method used to classify and compare probability distributions. Kui takes into account the largest positive and negative differences between the cumulative distribution functions of two samples. This was implemented using basic Pandas operations. One short-term measurement is compared with all long-term measurements. We determine the smallest absolute maximum difference between the distributions of long-term and short-term measurements respective energy levels in order to assign the metal alloy. The smallest difference is identified as the most probable assignment for the short-term measurement.

## IV. RESULTS

### A. Classification

We tested the classifiers from Section III-C with various parameters using Categorical Sampling as data generation and the HPGe detector with block data. As you can see in Table II, the best results could be achieved with MLC, XGB, LR and Linear SVM.

TABLE II: Accuracy of all examined models on aluminium and copper block measurements with Categorical Sampling as data generation, HPGe detector and MLC achieving the best results.

classifier	aluminium measurement time			copper measurement time		
	0.5s	1.0s	2.0s	0.5s	1.0s	2.0s
MLC	<b>96.32</b>	<b>99.03</b>	<b>99.87</b>	<b>91.87</b>	<b>95.18</b>	<b>97.96</b>
XGB	93.93	98.19	99.65	89.42	93.48	97.09
LR	93.28	97.69	99.50	88.82	93.00	96.51
Linear SVM	91.94	97.28	99.55	88.21	92.43	96.24
NN	90.80	95.62	98.45	87.51	91.85	95.22
KNN	89.18	95.41	98.63	86.57	91.44	95.06
RFC	88.58	95.34	98.66	85.21	90.01	94.20
ETC	89.94	95.34	98.63	85.90	90.48	94.66
RNC	89.03	95.26	98.66	86.46	91.51	95.06
Kui	77.07	84.32	91.53	83.68	89.56	94.29

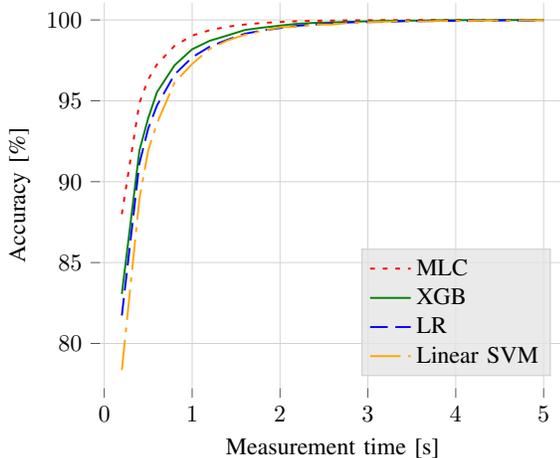


Fig. 2: Accuracy of the best models on aluminium measurements (Table II).

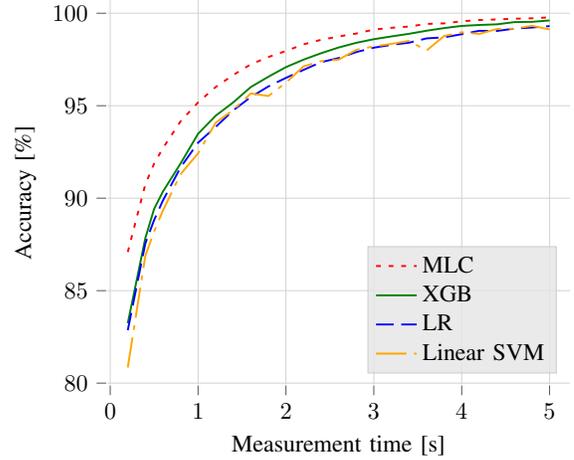


Fig. 3: Accuracy of the best models on copper measurements (Table II).

To see the differences of the four best classifiers, you can see more detailed plots in Fig. 2 and Fig. 3 for aluminium and copper. In the following experiments, we will further examine the results of these classifiers using different data generation and preprocessing methods, as well as different types of detectors.

### B. Data Generation

Table III shows the results of the four best classifiers with CVAE data generation as well as the increase or decrease in accuracy compared to Categorical Sampling data generation. CVAE led to a slight increase in the accuracy of MLC while the accuracy of the other classifiers decreased. Thus, in the following experiments, we use MLC in combination with CVAE and the other classifiers in combination with Categorical Sampling.

TABLE III: Accuracy of best models on aluminium and copper block measurements with CVAE data generation and HPGe detector. Only MLC improves using the CVAE data generation.

classifier	measurement time			
	aluminium	0.5s	1.0s	2.0s
MLC	<b>96.52</b>	<b>99.12</b>	<b>99.91</b>	(+0.04)
LR	86.37	94.09	98.32	(-1.18)
Linear SVM	84.38	93.32	98.10	(-1.45)
XGB	51.61	62.67	76.14	(-23.51)
copper				
MLC	<b>92.09</b>	<b>95.27</b>	<b>98.07</b>	(+0.11)
LR	81.59	89.19	94.30	(-2.21)
Linear SVM	77.04	87.12	93.62	(-2.62)
XGB	51.21	60.92	73.78	(-23.31)

### C. Preprocessing

In Table IV, a subset of the entire spectra was selected as described in Section III-B2. For each, training and testing, a dataset of 1,000 spectra per material with a simulated measurement time of one second were created via Categorical Sampling and CVAE. The number of energy levels describe

TABLE IV: Accuracy decrease of the best models with decreased amount of energy levels for aluminium and copper blocks with HPGe detector and a simulated measurement time of one second.

classifier	aluminium			copper		
	number of energy levels 4000	8000	16384	number of energy levels 4000	8000	16384
MLC	<b>98.29</b>	<b>98.69</b>	<b>99.04</b>	<b>93.83</b>	<b>94.23</b>	<b>95.32</b>
LR	96.65	97.66	97.93	91.40	91.74	92.69
Linear SVM	96.04	96.98	97.38	91.24	92.37	92.87
XGB	97.03	97.47	98.08	92.73	92.52	93.35

the upper boundary from which on features were deleted. A reduction in the number of features diminishes the overall performance of the respective model. However, this reduction does not exhibit a linear relationship with the number of features. This observation suggests that there is not a substantial information gain in the last energy levels of the spectra. It is the case because the first energy levels have a considerably higher count-value. In the higher energy levels, oftentimes no count is measured, resulting in no information gain. The initial energy levels are considerably more crucial, proving sufficient for achieving a relatively high level of accuracy. However, because using all the features leads to the overall best results, and the prediction runtime is very quick in any case, we decided to keep the original energy level count for our further experiments.

Using a DAE to convert the short term measurements to 60 seconds measurements and classifying those led to a decrease in accuracy. For aluminium, the accuracy decreased by approx. 6% and for copper by approx. 7% on average for the best four classifiers respectively.

The preprocessing with EPs and DEPs was only tested with MLC and mostly brings small improvements for the accuracy. The results can be seen in Table V. For these experiments, we used 1,000 spectra for training and five long-term measurements as reference spectra.

The preprocessing with Unique Peaks brings slight positive and negative variations in the accuracy. In total, it is difficult to say whether the variations in the accuracy are due to the randomness of the data or due to the preprocessing with Unique and Escape Peaks.

TABLE V: Accuracy of MLC with CVAE as data generation, HPGe detector, block data and EPs and DEPs as preprocessing.

material	measurement time		
	0.5s	1.0s	2.0s
aluminium	96.67 (+0.15)	99.15 (+0.03)	99.92 (+0.01)
copper	91.78 (-0.31)	95.39 (+0.12)	98.28 (+0.21)

#### D. Detectors

In our analysis so far, we have evaluated the accuracy of four top classifiers - MLC, XGB, Linear SVM and LR - using aluminium and copper data in block form, with the MLC

achieving the best results. In this section of the investigation, we turn to aluminium chips data, as only data in this form is available for the CeBr<sub>3</sub> detector. In order to examine the results, we will restrict ourselves to the MLC in the following, as it shows better results compared to other methods. With a measurement time of 0.5 seconds, the CeBr<sub>3</sub> detector has a maximum accuracy of 62.58%, which is significantly higher than HPGe detector with 50.40% using Categorical Sampling to generate the data. When using CVAE, the accuracy barely increases, 62.88% for CeBr<sub>3</sub> detector and 50.69% for HPGe detector.

In Fig. 4 we have determined the accuracy as a function of time from 0.2 to 10 seconds measurement time in steps of 0.2 seconds. While the CeBr<sub>3</sub> delivers the best results for measurement times up to 1.3 seconds, the HPGe achieves better accuracy for longer measurement times. Both detectors converge to 100% over time, with HPGe converging much faster. The analysis of the results shows that the HPGe detector only needs 10 seconds to reliably distinguish all aluminium alloys, while the CeBr<sub>3</sub> detector takes significantly longer for the same task. As shown in the corresponding figure, the curve for the HPGe detector ends at 10 seconds, at which point a classification accuracy of 100% is achieved. In contrast, the CeBr<sub>3</sub> detector achieves full classification accuracy of 100% only after about 80 seconds.

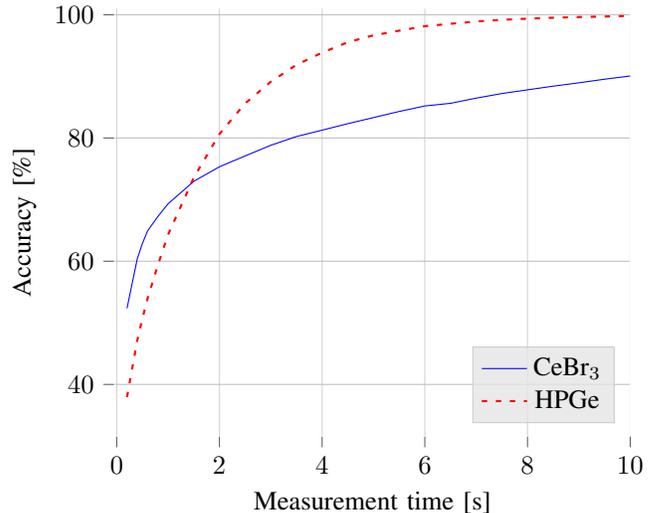


Fig. 4: Comparison of the accuracy over time of the two detectors with aluminium chips data using CVAE and MLC.

## V. DISCUSSION

As the Linear SVM and Logistic Regression achieve good results on our data, it seems that the data is linear separable in a high dimension. In addition to the linear separability of the spectra, the regularization techniques provided by XGBoost in the form of L1 and L2 are the reason why the classifier generalizes well and thus achieves very good accuracy.

The good results of MLC could be attributed to its ability to model the data distribution more accurately. In scenarios where the underlying data distribution is complex and doesn't

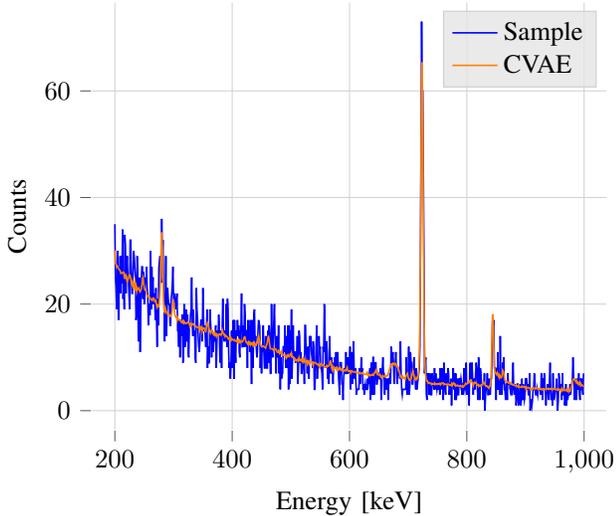


Fig. 5: Section of an example spectrum of an aluminium alloy generated with different methods.

conform strictly to linear relationships, like our spectral data, MLC can capture the nuances better than the other classifiers.

Using CVAE as data generation, we see that the results of MLC can be improved while for the other classifiers, the accuracy decreases. A reason could be that the CVAE generated spectra contain less noise and thus are more suitable as a probability distribution for MLC. A comparison of the spectra generated by the different methods is shown in Fig. 5. Thanks to this characteristic, MLC requires less CVAE spectra than spectra generated by Categorical Sampling to achieve good results, because for the latter it needs more data to build a mean representing the data adequately. This reduces the prediction runtime and thus allows a faster classification. For the other classifiers, the accuracy decreases using CVAE as data generation. This could be due to the fact that the CVAE data is not as similar to the test data in its variance as the Categorical Sampling data. Thus, the Categorical Sampling data is more suitable for training.

The  $\text{CeBr}_3$  data results are better than the HPGe data results, because the higher number of counts at short measurement times has a greater influence on the determination due to a higher probability of detector response. This is important when measuring very low levels of gamma radiation.

The high resolution of the HPGe detector leads to fewer counts per channel when the same number of counts is distributed across more channels, increasing relative uncertainty of each count rate. This makes it more challenging to identify significant patterns between the aluminium alloys.

The results from the chip data are worse than the block data because the mass of the sample is significantly reduced and therefore fewer nuclei are irradiated and fewer gamma rays are counted.

The HPGe detector shows its strengths at longer measurement times, which could be due to its very high resolution to precisely detect even fine energy differences. Peaks carrying relevant information regarding classification are thus

emphasised more clearly. Otherwise, the number of counts in the  $\text{CeBr}_3$  is not sufficient at longer measurement times for the compensation that would be necessary to distinguish the aluminium alloys.

## VI. CONCLUSION

In conclusion, the research demonstrates the potential of applying advanced classification methods and data generation techniques to improve metal alloy differentiation in recycling processes based on PGNA measurements. The MLC was found to be the most effective classification method combined with CVAE data generation technique.

The study shows that the  $\text{CeBr}_3$  detector excels in scenarios requiring very short measurement times, but overall the HPGe detector generally remains the preferable choice to deliver accurate results. The choice of detector depends on the specific needs of the application.

Because real short-term measurement data is lacking, it is necessary to perform and study short-term measurements in detail in the future to verify the applicability of our findings in the real world.

## APPENDIX

### HYPERPARAMETERS

#### A. Data Generation

Conditional Variational Autoencoder	Value
library	TensorFlow
scaling	MinMaxScaler <sup>1</sup>
hidden_units	100
latent_variables	10
loss	-ELBO <sup>2</sup>
optimizer	Adam <sup>3</sup>
learning_rate	0.001
batch_size	32
epochs	100

#### B. Preprocessing

Denoising Autoencoder	Value
library	TensorFlow
scaling	MinMaxScaler <sup>1</sup>
hidden_units	10
latent_variables	100
loss	MeanSquaredError
optimizer	Adam <sup>3</sup>
learning_rate	0.001
batch_size	32
epochs	100

<sup>1</sup>implementation of the scikit-learn [10] library

<sup>2</sup>Negative ELBO as in [13] using MSE, a  $\beta$ -parameter and KL divergence

<sup>3</sup>Adam optimization algorithm [28]

### C. Classification

<b>XGBoost</b>	<b>Value</b>
booster	gblinear
device	cpu
verbosity	1
lambda	0.25
alpha	0
updater	shotgun
feature_selector	cyclic
learning_rate	0.45

<b>Logistic Regression</b>	<b>Value</b>
penalty	L2
dual	False
tol	0.0001
C	1.0
fit_intercept	True
intercept_scaling	1
class_weight	None
solver	lbfgs
max_iter	150
multi_class	auto
verbose	0

<b>Linear Support Vector Machine</b>	<b>Value</b>
penalty	L2
loss	squared_hinge
dual	True
tol	1
C	3
multi_class	ovr
fit_intercept	True
intercept_scaling	1
class_weight	None
verbose	0
random_state	None
max_iter	100

<b>K-nearest Neighbors</b>	<b>Value</b>
n_neighbors	8000
weights	distance
metric	euclidean
algorithm	brute
leaf_size	30
p	2

<b>Radius Neighbors Classifier</b>	<b>Value</b>
radius	500
weights	distance
metric	euclidean
algorithm	brute
leaf_size	30
p	2
outlier_label	most_frequent

<b>Neural Network</b>	<b>Value Blocks</b>	<b>Value Chips</b>
library		PyTorch
n_layers		4
layer		fully connected
activation_function		relu
criterion		CrossEntropyLoss
optimizer		Adam <sup>3</sup>
num_epochs	2	5
n_batch_size	500	1000
learning_rate		0.0001

<b>Random Forest Classifier</b>	<b>Value</b>
n_estimators	4000
min_samples_split	50
criterion	gini
max_depth	None
min_samples_leaf	1
min_weight_fraction_leaf	0.0
max_features	sqrt
max_leaf_nodes	None
min_impurity_decrease	0.0
bootstrap	True
oob_score	False
max_samples	None

<b>Extra Trees Classifier</b>	<b>Value</b>
n_estimators	4000
min_samples_split	50
criterion	entropy
max_depth	None
min_samples_leaf	1
max_features	sqrt
min_weight_fraction_leaf	0.0
max_leaf_nodes	None
min_impurity_decrease	0.0
bootstrap	False
verbose	0
max_samples	None

### ACKNOWLEDGMENT

Research was conducted as part of the MetalClass project, funded by the German Federal Ministry of Education and Research with the funding reference number 01IS20082A/B. The authors express their sincere gratitude to our project partner AiNT GmbH for their support and enriching discussions throughout the project's duration. Some of the research was conducted while Kai Krycki was with AiNT GmbH. Special acknowledgment is also extended to Wieland-Werke AG for graciously providing the copper alloys essential to our investigation.

### REFERENCES

- [1] "All the metals we mine each year, in one visualization," <https://www.weforum.org/agenda/2021/10/all-tonnes-metals-ores-mined-in-one-year/>, accessed: 2023-06-23.
- [2] M. Henckens and E. Worrell, "Reviewing the availability of copper and nickel for future generations. the balance between production growth, sustainability and recycling rates," *Journal of Cleaner Production*, vol. 264, p. 121460, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652620315079>
- [3] "The answer to the aluminium industry's emissions issue? aluminium's infinite recyclability," <https://www.weforum.org/agenda/2021/12/aluminium-emissions-recycling-circular-economy/>, accessed: 2023-06-23.
- [4] "Neue Methode des Aluminium-Recyclings. Raus aus der Wegwerfspirale." <https://www.tagesschau.de/wirtschaft/technologie/aluminium-recycling-hoppegarten-100.html>, accessed: 2023-06-23.
- [5] S. Park, J. Lee, E. Kwon, D. Kim, S. Shin, S. Jeong, and K. Park, "3d sensing system for laser-induced breakdown spectroscopy-based metal scrap identification," *International Journal of Precision Engineering and Manufacturing-Green Technology*, pp. 1–13, 2022.
- [6] R. L. Paul and R. M. Lindstrom, "Prompt gamma-ray activation analysis: Fundamentals and applications," *Journal of Radioanalytical and Nuclear Chemistry*, vol. 243, no. 1, pp. 181–189, 2000. [Online]. Available: <https://link.springer.com/article/10.1023/A:1006796003933>
- [7] H. Shayan, K. Krycki, M. Doemeland, and M. Lange-Hegermann, "Pgnaa spectral classification of metal with density estimations," *IEEE Transactions on Nuclear Science*, vol. 70, no. 6, pp. 1171–1177, 2023.
- [8] D. Demir, M. Eroglu, and A. Turşucu, "Studying of characteristics of the hpge detector for radioactivity measurements," *Journal of Instrumentation*, vol. 8, no. 10, p. P10027, oct 2013. [Online]. Available: <https://dx.doi.org/10.1088/1748-0221/8/10/P10027>

<sup>3</sup>Adam optimization algorithm [28]

- [9] P. Guss, M. Reed, D. Yuan, A. Reed, and S. Mukhopadhyay, "Cebra3 as a room-temperature, high-resolution gamma-ray detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 608, no. 2, pp. 297–304, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168900209013771>
- [10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [13] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." [Online]. Available: <https://arxiv.org/pdf/1312.6114>
- [14] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf)
- [15] I. Higgins, L. Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016. [Online]. Available: <https://www.semanticscholar.org/paper/beta-VAE%3A-Learning-Basic-Visual-Concepts-with-a-Higgins-Matthey/a90226c41b79f8b06007609f39f82757073641e2>
- [16] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML 2008 : proceedings of the twenty-fifth International Conference on Machine Learning*, A. K. McCallum, S. Roweis, and W. Cohen, Eds. [Place of publication not identified]: ACM, 2008, pp. 1096–1103.
- [17] Y. Kim, D. Oh, S. Huh, D. Song, S. Jeong, J. Kwon, M. Kim, D. Kim, H. Ryu, J. Jung, W. Kyung, B. Sohn, S. Lee, J. Hyun, Y. Lee, and Y. K. C. Kim, "Deep learning-based statistical noise reduction for multidimensional spectral data," 2021.
- [18] S. Dobilas, "Denoising autoencoders (dae) — how to use neural networks to clean up your data," *Towards Data Science*, 04.04.2022, accessed: 2023-06-23. [Online]. Available: <https://towardsdatascience.com/denoising-autoencoders-dae-how-to-use-neural-networks-to-clean-up-your-data-cd9c19bc6915>
- [19] R. Tanaka, K. Yuge, J. Kawai, and H. Alawadhi, "Artificial peaks in energy dispersive x-ray spectra: sum peaks, escape peaks, and diffraction peaks," *X-Ray Spectrometry*, vol. 46, no. 1, pp. 5–11, 2017. [Online]. Available: <https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/xrs.2697>
- [20] W. A. Metwally, C. W. Mayo, X. Han, and R. P. Gardner, "Coincidence counting for pgnaa applications: Is it the optimum method?" *Journal of Radioanalytical and Nuclear Chemistry*, vol. 265, no. 2, pp. 309–314, Jul 2005. [Online]. Available: <https://doi.org/10.1007/s10967-005-0826-2>
- [21] R. B. Millar, *Maximum likelihood estimation and inference: with examples in R, SAS and ADMB*. John Wiley & Sons, 2011.
- [22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [23] "Xgboost parameters," <https://xgboost.readthedocs.io/en/stable/parameter.html>, accessed: 2023-06-24.
- [24] S. Suthaharan, *Support Vector Machine*. Boston, MA: Springer US, 2016. [Online]. Available: <https://doi.org/10.1007/978-1-4899-7641-3>
- [25] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [26] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
- [27] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, pp. 3–42, 2006.
- [28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." [Online]. Available: <https://arxiv.org/pdf/1412.6980>