Journal on Data Semantics (2021) 10:41–56
https://doi.org/10.1007/s13740-021-00124-z

ORIGINAL ARTICLE

Toward Automated Support of Complaint Handling Processes: An Application in the Medical Technology Industry

Philip Hake[1] · **Jana-Rebecca Rehse**[2] · **Peter Fettke**[1]

Received: 3 February 2020 / Revised: 1 March 2021 / Accepted: 29 March 2021 / Published online: 9 June 2021

Abstract
Complaints about finished products are a major challenge for companies in the medical technology industry, where product quality is directly related to public health and therefore strictly regulated. In this paper, we examine how available data can be used to provide automated support to the complaint handling processes in the medical technology companies. We identify the automation potentials in the 8D reference process for complaint management and discuss their organizational and technical challenges. Using data from a large manufacturer of medical products, we show how partial process automation can be achieved in practice by designing, implementing, and evaluating a deep learning-based prototype for automatically suggesting a likely error code for future complaints, given their textual description. Our approach is able to assign the correct error code for more than 75% of all cases and outperforms the conventional classification approaches used as a baseline comparison. Our results show that partial automation of a complaint management process by means of deep learning can be achieved in practice.

Keywords Complaint management · Quality management · Process prediction · Machine learning · Deep learning

1 Introduction

For manufacturing companies, complaints about finished products are a major challenge [8]. They not only reduce profits, but also generate additional costs for, e.g., repairing products or handling returns. Complaint management requires time and personnel resources, both for designing and implementing the processes and for executing them. In addition, regular complaints about quality defects have a lasting negative effect on the perceived process quality, which may lead to a reduced customer loyalty and therefore damage a company's reputation among its customers and in the public eye.

Defective products can be particularly damaging for manufacturers of medical technology, which must meet special quality requirements in the regulated environment [26]. The new EU regulation on medical devices has further increased the requirements on quality and safety of medical technology [12]. In this industry, lawmakers consider product quality as directly related to public health. Quality defects therefore pose a risk to the company's success in two ways. They could lead to a decline in sales, but also could be the cause for official interventions that may lead to a forced closure of entire manufacturing plants in the worst case. To avoid that and catch potential health threats at an early stage, complaint processes in the medical technology industry are subject to specific legal requirements. Manufacturers must establish a prompt and consistent approach to the acceptance, assessment, and investigation of complaints and the decision on follow-up measures.

At the same time that requirements toward complaint handling processes are becoming stricter, there is a strong shift

This work was conducted within a project sponsored by the German Ministry for Education and Research (BMBF), project name "Reklamation 4.0", support code "01IS17088B". We also gratefully acknowledge the support of NVIDIA for the donation of a GPU used for this research.

✉ Jana-Rebecca Rehse
rehse@uni-mannheim.de

Philip Hake
philip.hake@dfki.de

Peter Fettke
peter.fettke@dfki.de

[1] Institute for Information Systems, German Research Center for Artificial Intelligence and Saarland University, Saarbrücken, Germany

[2] University of Mannheim, Mannheim, Germany

toward digitizing and automating processes, for example, with technologies like Robotic Process Automation [1]. This shift is in part caused by the increasing maturity of technology like machine learning and process mining and the increasing availability of data. In the context of the ongoing digitization of manufacturing companies ("Industry 4.0"), more and more process and production data are recorded and stored [23]. The considerable amount of real-time sensor, machine, and process data from product lifecycle (PLC), manufacturing execution (MES), and enterprise resource planning (ERP) systems can be further enriched with data from the systems used for complaint and error handling processes as well as customer-related data. These data hold great potential for improved complaint management [15].

In this paper, we investigate the potentials of using these data for automating activities in complaint handling processes in the medical technology industry. This is particularly interesting due to the nature of complaint handling processes. On the one side, they are highly standardized, typically following a reference process that is established in multiple industries. On the other side, they are highly individualized, as complaints vary greatly with regard to the product and customer. By definition, complaints are caused by erroneous products, which are most likely the cause of erroneous and therefore exceptional production processes, requiring each complaint to be handled separately and individually. Therefore, our goal is to further examine the potentials for automation or at least automated support of complaint processes.

One key element in the strive toward process automation is machine learning (ML). Particularly, approaches of supervised ML (SML) have the potential to automate process steps that were previously conducted by human experts, because they can independently learn how to conduct those tasks from observing data collected during manual execution [28]. Classification of natural texts is an example for such a task. Traditionally, a human domain expert would have to read any message that a company received and, based on the available domain knowledge, decide on the next steps. Nowadays, ML-based text mining approaches can recognize the relevant phrases in a message and automatically assign or execute the appropriate next steps. Such an approach has already been successfully applied to complaint messages in the telecommunications industry [41] or to patient messages in medical portals [38].

In this paper, we illustrate the automation potentials of using ML in complaint handling by automatically classifying textual complaint descriptions by means of a deep neural network. The first version of this approach was published at the 2019 AI4BPM workshop [16], located at the international conference on Business Process Management (BPM) in Vienna. This paper focuses more on the general perspective of automated support of complaint handling processes. In addition, it describes a redesigned evaluation of our approach with focus on practical adoption.

For this purpose, the paper is organized as follows. In Sect. 2, we report on the foundations of medical technology quality management to explain the organizational context of complaint processes. Moreover, we analyze the automation potentials of complaint handling according to the 8D reference process. Section 3 describes a prototype for supporting a complaint process in a medical technology company that we implemented in a research project. The prototype's realization and evaluation are described in Sect. 4. Section 5 addresses the challenges of ML-based automation services in complaint handling processes. Section 6 contains related work on automated process support in complaint handling, before we conclude the paper in Sect. 7.

2 Toward Automated Support of Complaint Handling Processes

2.1 Quality Management in the Medical Technology Industry

Although process automation is relevant for all business and organizations, complaint processes in the medical technology have some unique characteristics that make automation and the application of ML both interesting and challenging. In this subsection, we give some background on those characteristics that help understand the automation potentials that we describe in the following subsection.

The medical technology industry is part of the so-called regulated environment, i.e., companies that are specifically monitored and controlled by public authorities due to their direct influence on public health. Both processes and products in the regulated environment are subject to high quality requirements, summarized by the term GMP (Good Manufacturing Practice). These binding quality requirements result from national and international regulations (such as laws and standards) and must be considered during production [11]. GMP regulations affect central sectors of the economy, such as the pharmaceutical industry, biotechnology, medical technology, chemical industry, and food industry. Compliance with GMP regulations is of fundamental importance to companies in these industries, as they influence their manufacturing authorization. Core processes of GMP compliance and quality management include process management and document management, improvement management, corrective and preventive actions and controls, risk management, change management, deviations, employee training, as well as internal and external audits for GMP-relevant processes.

A central part of quality management in the regulated environment is complaint management. This is partially regulated by law. For medical device manufacturers, ISO standard 13485 (which largely conforms with ISO 9001) prescribes the use of a quality management system designed to demonstrate consistent compliance with quality standards [2]. Typically, the systems follow the 8D problem-solving process (see Fig. 1) [4]. Originally developed by automotive companies and used across many different industries, this process describes a structured approach to the identification and long-term elimination of problems and their causes and is therefore an integral part of complaint management.

The 8D is a major factor of why process automation by means of machine learning is particularly interesting for the medical technology industry. First, it is officially mandated, such that many companies use it. Second, it prescribes the collection of large amounts of complaint-specific data. Because of those two factors, there are standardized IT systems that support it and that enable the collection of large amounts of process-specific data. For the companies, these data are an important source for internal complaint handling and identifying errors in the production process. However, they can also be used for automating steps in the 8D process itself.

2.2 Automation Potentials of the 8D Process

As explained, the execution of the 8D process is supported by specific systems, which companies use as a part of their quality management system. Those systems typically support the complete 8D process, with one process instance for each single complaint. However, they are usually restricted to collecting data, which the employees have to enter manually. Automation of at least some process steps would therefore relieve employees from the repetitive and error-prone task of manual data entry and save time by reducing the processes' dependency on manual process execution (which can be crucial in the regulated environment). In order to realistically estimate those automation potentials in the 8D process, we are inspecting each stage individually, considering the data available to train potential support services.

The first stage of the 8D process is team assembly, where quality managers put together a team of experts that will investigate the complaint at hand. In this stage, automation would be targeted to externalize the knowledge about the expertise of individual employees. A potential support service could, for example, analyze past assignments and the quality of their identified solutions to suggest employees for the team, who have successfully worked together in the past and are therefore more likely to handle a complaint quickly and efficiently. Also during this initialization stage of the 8D process, an ML-based classification trained on attributes of past complaints can assist to determine the complaint's severity and therefore the priority and speed with which it must be addressed.

The second stage of the 8D process (problem description) potentially has the most potential for automation, because it involves many classifications of the complaint. For instance, based on its textual description, a complaint can be forwarded to an employee in the team, who is more experienced in dealing with this type of mistakes. Another interesting attribute is the criticality assessment, i.e., whether or not the complaint describes a problem that is so important that it has to be reported to the authorities. This is a more challenging classification problem, as critical incidents are typically very rare, such that only very little training data for machine learning approaches exist. The assignment of a pre-defined error code, which we describe in detail in Sect. 3, also falls into this stage. Techniques for generating more (artificial) training data or hybrid approaches, which combine ML techniques with a priori domain knowledge, could be applied to achieve automation here.

After the complaint has been described in the second stage, automated support services could identify those containment actions that were successfully implemented in similar cases in the past and suggest them as input for the third stage. These actions could theoretically be implemented in an automated fashion if they only required software-related changes. However, such changes would require a high degree of interdependence between different systems and could only be executed after being confirmed by an employee. Since containment actions happen only rarely, automating them is probably not worth the cost and effort.

The fourth stage (identify root causes) may be the most interesting with regard to automated support. On the one side, finding the source of an error might require analyzing large

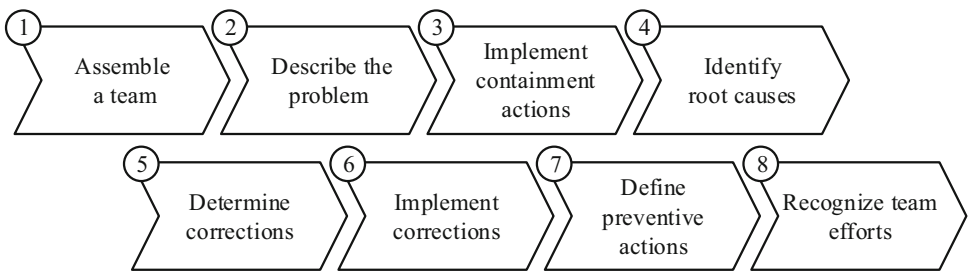

Fig. 1 8D reference process for complaint management

amounts of process, sensor, or machine execution data, for which methods from machine learning or data mining can be very helpful. On the other side, those methods are only able to identify correlations, but not causation. This means that they are able to find recurring patterns in the data that coincide with a higher probability for an erroneous product, but they cannot determine the cause of the pattern or the error. Implementing a notion of causality to ML remains a big challenge in AI research today [33]. To address this problem, causality-related methods like causal forests or IV regressions could be employed to identify potential causes for the error. Another option would be a hybrid approach, where an ML approach finds correlations in the data and provides a human employee with an interesting starting point for determining the root cause of the given complaint.

In the remaining stages of the 8D process, there is less potential for automation. For stages 5 (determine corrections), 6 (implement corrections) and 7 (define preventive actions), it is very difficult to assess, as corrections and preventive actions are first and foremost dependent on the identified error cause from the previous stage. Furthermore, we cannot expect to have much training data available in those stages, as known errors either do not reoccur or, if they do, previous corrections apparently were not as effective as expected, so they should not be repeated. However, depending on the organizational environment, it might make sense to support certain individual tasks within the stages. On the contrary, the last stage (recognize team efforts) has a foremost social function, so introducing automation here might be counterproductive.

3 A Prototype for the Automated Support of Complaint Handling Processes

3.1 Background

Against the backdrop of the automation potentials in complaint handling processes, an ongoing need to optimize those processes, and increased data availability, the research project "Reklamation 4.0" was set out to find new approaches to use these data in order to improve the complaint handling process in medical technology companies. Following the lead of two application partners, a large- and a medium-sized company from the medical technology industry, we examined which complaint-related data are currently available in companies and how we could use it to gain additional insights, with the help of machine learning and data mining.

For the larger application partner, we took a first step toward automating the complaint process. The goal was to train a machine learning approach that provides automated support for repetitive, but time-critical process steps. One of these steps is the error code assignment. Medical technology companies usually receive textual descriptions of customers' complaints, which the employees then categorize according to an internal catalog of potential error codes. In addition to structuring the externally generated complaints, this is a necessary prerequisite for the ensuing analysis of the erroneous product.

The application partner provided us with a dataset containing around 15,000 textual descriptions and assigned error sources for past complaints. We use this dataset to design, implement, and evaluate a novel approach for automatically suggesting a likely error source for future complaints based on the customer-provided textual description. The approach makes use of a deep learning technology, which has already been used for natural language processing (NLP) in other application domains.

In this and the following section, we present a revised and extended version of the previous workshop paper [16], where we improved the selection of evaluation data, the performance measurements, and the evaluated models. The approaches are evaluated using a test dataset allowing us to estimate the performance in a practical adoption. Moreover, we took steps to further prevent our models from overfitting by introducing early stopping and a variation in different regularizing parameters. While our first evaluation only included a downsampled dataset, we now evaluate our approach on the initial imbalanced dataset as well. Besides a naïve classifier which we used as a baseline comparison, we compare our approach to a random forest classifier.

3.2 Challenge and Solution Design

Employees usually file 8D reports after they receive either an internal or external complaint about product quality. First and foremost, filing such a report entails recording a lot of potentially relevant data, but in a second step, the employee also has to assess the claim in terms of its criticality and the potential error source. Both determine the actions to be taken next. The criticality denotes the risk of another customer's health. If, for example, the bacterial load is too high on a previously sterilized product, it must be reported immediately to the responsible authorities in order to avoid public health risks. The potential error source is an internal assessment and the first step toward identifying and fixing the production problem that has caused the quality complaint. Companies usually have an internal set of predefined error codes, which represent potential error sources. These codes vary in terms of specificity, going from a generic (e.g., "packaging error") to a more precise (e.g., "lack of maintenance on machine 5") classification, depending on the information available at the time.

Correctly assessing each filed incident is a difficult and time-consuming task, especially for less experienced employees, who might not have the necessary knowledge.

Using a machine learning approach, which is able to automatically analyze all past complaints in order to assist employees in correctly assessing their incidents, may not only reduce the number of wrong assessments, but also accelerate the process, such that the issue can be fixed more quickly. For this purpose, we develop a new approach based on a deep neural network to automatically assign a likely error code to a complaint. As input, the network receives free text as recorded in the 8D report and the error code as assigned by an employee. During training, the network learns which complaint characteristics are decisive for the classification. The trained network can then automatically submit proposals for an assessment of a newly arriving complaint to the responsible employee.

3.3 Solution Architecture

In order to classify textual descriptions of complaints according to their likely error source, we use a recurrent neural network (RNN) with long short-term memory (LSTM) cells [18]. RNN layer cells feed information back into themselves, evolving their state by "forgetting" or "remembering" previous inputs. Our network consists of one input layer, one or more hidden LSTM layers, and one output layer. The input layer is responsible for generating a numerical representation of the input text, a so-called embedding. We use a pretrained embedding layer of English words [29] and allow the architecture to adapt the word embeddings to the specific context during training. For the hidden layers, we use LSTM cells, because they have been found to be particularly well-suited to handle data with long-term dependencies, such as the natural language in our textual descriptions. The output layer is a fully connected dense layer with a softmax activation, which transforms the activations of the last LSTM layer to the number of potential classes to obtain the probability distribution $\widehat{y}$ over the classes.

Overall, our network architecture is a standard one for text classification problems. Our loss function L (Eq. 1), which we use for computing the gradient during training, is given by the categorical cross-entropy for the expected output y and the predicted output $\widehat{y}$ as well as an additional regularization loss. Given I the number of layers, C_i the number of cells in layer i and A_c the activation of cell c, the regularization loss $L1$ is defined as the sum over all activations A_c of the hidden layers (Eq. 2). By regularizing the layer activations, we intend to prevent our model from overfitting. Furthermore, we use a dropout probability for the activations of each hidden layer to approach the problem of overfitting [37].

$$L(y, \widehat{y}) = -\sum_{i=0}^{c} y_i * \log(\widehat{y_i}) + \lambda * L1 \tag{1}$$

$$L1 = \sum_{i=1}^{I-1} \sum_{c=1}^{C_i} |A_c| \tag{2}$$

3.4 Data Characteristics and Data Preparation

To realize our solution design and train the neural network, we use the complaint management data of a globally operating medical technology company. It contains 15,817 customer complaints about products, including both mass products and products manufactured according to the customer's requirements. The individual complaints contain sensitive information about the business processes and products of the manufacturer. Therefore, we cannot make the dataset publicly available. Resolving this issue would require semantically altering the data, resulting in an artificial dataset, which would counteract our goal to provide insights about the performance of machine learning in a real word scenario.

Each complaint in our dataset consists of a textual description and an error code, which is manually set by the employee handling the complaint. The error code is a numerical representation of the assessment result. In contrast with our earlier approach [16], we filter complaints that contain multiple complaint texts and multiple error codes. In these cases, we are not able to establish a mapping from text to error code. Cleaning the initial dataset leaves us with 14,634 complaints and respective error codes.

The dataset exhibits 186 different error codes. Table 1 compares the characteristics of the codes that occur in at least 500 cases with the remaining codes. The overview reveals that less than 5% of the codes account for more than 50% (7,371) of the cases. Because a customer may either file a complaint by phone or by letter, the responsible employee

Table 1 Dataset characteristics

Class		Cases	Distinct words	Distinct words compared to other classes
	All codes	14,634	11,748	
Class 1	Code 1	2286	6631	2152
Class 2	Code 2	965	3756	849
Class 3	Code 3	717	4682	1380
Class 4	Code 4	623	3336	632
Class 5	Code 5	590	4463	1009
Class 6	Code 6	581	4273	1202
Class 7	Code 7	566	3162	621
Class 8	Code 8	527	3731	760
Class 9	Code 9	516	3563	620
Class 0	Code 10–186	7263	26,671	17, 713

summarizes the complaint and submits it to the information system handling the complaint process. The textual description of a complaint exhibits an average length of 102 words with a standard deviation of 110. The following description is an example for a complaint: "customer bought the product on 27 May 2019, he claims that the Velcro does not adhere anymore, he also claims that the problem did not occur in previous orders." The dataset contains 1,498,330 tokens, of which 36,598 distinct words.

Table 1 depicts the number of distinct words that are contained in the textual description of the cases labeled with the same code. In addition, we provide insights on the number of distinct words that occur in cases exhibiting the same code but are not contained in any other case. Since machine learning-based classification approaches require sufficient data per class to perform well, we require a class to contain at least 500 samples to be considered for evaluation. Thus, we obtain nine classes that can be directly mapped to error codes and one additional class (class 0) containing the samples of the remaining classes. Cases classified with code 1 to 9 are mapped to the respective classes 1 to 9, while the remaining cases exhibiting the codes 10 to 186 are mapped to class 0. Based on this selection, we subtract 5% of the cases (732) for testing the model performance retaining the initial distribution of the selection. The resulting data which are used for training and validation contain 13,902 cases. Since the distribution of the selection shows a class imbalance, we generate another balanced dataset based on these cases to examine the effect of data imbalance. We derive the balanced dataset using a downsampling strategy. We randomly sample x complaints of each class where x is the number of complaints contained in the minority class (class 9). Since we removed 5% of the samples for testing, the minority class remains with 490 samples. Thus, the other classes are also represented by 490 samples resulting in a balanced dataset with 4900 samples for training and validation. In our evaluation, we assess the performance of our approach on the imbalanced and the balanced dataset.

Table 2 Dataset Overview

Dataset	Train	Validation	Test
Imbalanced	12,512	1390	732
Balanced	4410	490	732

Table 3 Hyperparameter search space

Hyperparameter	Search Space
LSTM Layers	1, 2, 4
Hidden Units	16, 32, 64
L1 regularization (λ)	0.005, 0.05, 0.5
Sequence Padding	200
Training Epochs	200
Dropout	0.1, 0.3, 0.5
Batch Size	32, 64, 128, 256
Learning Rate	0.01
Early Stopping	0, 20

Table 4 Additional learning rate search space

Hyperparameter	Search Space
LSTM Layers	2, 4
Hidden Units	64
L1 regularization (λ)	0.005, 0.05, 0.5
Sequence Padding	200
Training Epochs	3000
Dropout	0.1, 0.3, 0.5
Batch Size	256
Learning Rate	0.001, 0.0001
Early Stopping	0, 100

4 Evaluation

4.1 Evaluation Setup

To evaluate the robustness of our model, we perform a stratified 10-fold cross-validation on both datasets. The stratified cross-validation preserves the percentage of samples for each class in the training and validation split. Table 2 depicts the number of samples for each dataset and each split. Table 3 shows the hyperparameters of our initial model described in Sect. 3.3 and the respective search space, whose permutations yield 648 models to evaluate. Moreover, we evaluate different learning rates in an additional search space presented in 4 yielding in a total of 720 models. We use the training splits to optimize the loss function of the models. The model optimization is conducted using a stochastic strategy called Adam [19].

We perform the incremental optimization on training batches of different sizes. Beside the $L1$ regularization, we introduce early stopping as a hyperparameter. Early stopping is a common approach to reduce the generalization error and has been proved to have similar effects as weight decay regularization [5]. If the validation loss stops decreasing within a predefined window of epochs, we stop the training and use the model exhibiting the lowest validation loss across all epochs for further evaluation. The size of this window can be freely chosen; in our case, we set its value to 20. This means that after each training epoch j, the losses from the following ten epochs $j + k, k = 1, \ldots, 10$ are inspected. If the validation loss in epoch $j + k$ is higher than the validation loss in j, we continue in epoch $j + k$. If none of the losses exceeds

the validation loss in j, we select the model with the lowest validation loss over all evaluated epochs to become the final one.

Each model is trained separately on all 10-fold of the training split and evaluated by measuring the performance on the respective validation split. We measure the performance of a validation split using accuracy. For further investigation on the test set, we use precision, recall, and f-measure for the individual classes. The used evaluation measures are defined in the following Eqs. 3 to 9. The number of classes is denoted by n and for each class $i = 0, \ldots, n-1$, TP_i is the number of samples that are correctly assigned to class i, FP_i is the number of samples that are incorrectly assigned to class i, and FN_i is the number of samples that should be assigned to i, but are assigned to another class. Moreover, k denotes the number of samples used in an evaluation set while k_i denotes the number of samples of class i in the respective evaluation set.

$$accuracy = \frac{\sum_i TP_i}{\sum_i TP_i + FP_i} \tag{3}$$

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{5}$$

$$f\text{-}measure_i = \frac{2 * precision_i * recall_i}{precision_i + recall_i} \tag{6}$$

$$precision_{avg} = \frac{1}{k} * \sum_i k_i * precision_i \tag{7}$$

$$recall_{avg} = \frac{1}{k} * \sum_i k_i * recall_i \tag{8}$$

$$f\text{-}measure_{avg} = \frac{1}{k} * \sum_i k_i * f\text{-}measure_i \tag{9}$$

Finally, we measure the model using the average validation performances across the 10-fold. We select the model with the best validation performance for evaluation on the previously unseen test split. Furthermore, we compare our LSTM classifier (LSTMC) to a naïve classifier (NC) and a random forest classifier (RFC). The naïve classifier is considered a baseline. It is built up on the assumption that a complaint's class can be identified by certain words that are unique to this class. Therefore, when classifying complaints, it attaches more importance to words that appear in this class and none other. The NC uses a bag of words approach and the Jaccard similarity coefficient (Eq. 10) to map a sample input s to a class $i \in 0, \ldots, 9$. Given a training set, the classifier generates a bag of words b_i for each class based on the words contained in the training samples labeled with class i. A sample s is assigned a class according to the maximum similarity coefficient between b_i and the Bag of Words s_{bow} derived from s (Eq. 11). Outperforming the naïve classifier justifies using an ML-based approach, because the task at hand cannot be solved with an easier method, i.e., keyword matching.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

$$NC(s) = min(i | J(b_i, s_{bow}) = max \bigcup_{i=0}^{10} \{J(b_i, s_{bow})\}) \tag{11}$$

We use the random forest classifier implemented in scikit-learn version 0.22.1 [34] and the default parametrization. After removing the stop words from the complaint texts, we vectorize the texts using the term frequency-inverse document frequency of the words and bigrams contained in the training data. The balanced dataset contains 87,968 features, and the imbalanced dataset contains 210,586 features.

In the following, we report the mean training, mean validation, and mean test accuracy, as well as the standard deviation for the top LSTMC across the 10-fold. Moreover, we compare it to performance achieved by the naïve classifier and the random forest classifier. Furthermore, we present precision, recall, and f-measure for each individual class. Finally, we report a normalized confusion matrix of the top LSTMC and the RFC.

4.2 Implementation and Results

The presented classifier is implemented in Python 3.7.6 and is online available on GitHub[1]. The LSTM model was implemented, trained, and evaluated using TensorFlow[2] version 2.3 and the integrated Keras API. The training of the models was conducted on a machine with a Intel Xeon W-2175 CPU 2,50 GHz (28 threads), 128 GB RAM, and an Nvidia GeForce GTX Titan GPU.

4.2.1 Training Evaluation

Table 5 shows the hyperparameter configurations of the top 4 LSTMC regarding validation performance on the balanced dataset. The best model achieved a mean training accuracy of 0.99 with a standard deviation of 0.005 across the 10-fold and a mean validation accuracy of 0.67 with a standard deviation of 0.02. The RFC model yields a mean training accuracy of 0.99 with a standard deviation of 0.0001 and a validation accuracy of 0.65 with a standard deviation of 0.02. Training the naïve classifier achieved a mean training accuracy of 0.45

[1] https://github.com/phakeai/aicomplaint.

[2] https://www.tensorflow.org.

Table 5 Hyperparameter configuration for the top 4 models regarding validation accuracy (balanced dataset)

Hyperparameter	1	2	3	4
Validation accuracy	0.674	0.673	0.672	0.67
LSTM layers	1	2	1	1
Hidden units	64	64	32	64
L1 regularization (λ)	0.005	0.005	0.005	0.005
Sequence padding	200	200	200	200
Training epochs	100	100	100	100
Dropout	0.2	0.3	0.3	0.1
Batch size	128	128	128	128
Learning rate	0.01	0.01	0.01	0.01
Early stopping	20	20	20	20

and a standard deviation of 0.027 on the training folds. Contrary to the other models, the naïve classifier was not able to fit the dataset. Its performance disproves the assumption that certain keywords are responsible for a complaint to belong to a certain class.

Table 6 shows the hyperparameter configuration of the LSTM model that achieved the best validation accuracy on the imbalanced dataset. This model exhibits a mean training accuracy of 0.97 with a standard deviation of 0.002 across the 10-fold and a mean validation accuracy of 0.78 with a standard deviation of 0.009. The RFC model yields a mean training accuracy of 0.99 with a standard deviation of 0.0001 and a validation accuracy of 0.69 with a standard deviation of 0.007. Training the naïve classifier achieved a mean training accuracy of 0.60 with a standard deviation of 0.04 on the training folds. Again, NC underfits the training data.

Figure 3 shows the impact that different hyperparameter values have on the validation accuracy. For the hyperparameters layers, dropout, units, and batch size, we selected all models with a certain value (e.g., one, two, or four layers) and plotted the distribution of their validation accuracy. This allows us some insights about the impact of these hyperparameters on the overall model quality. For example, we can see that the accuracy drops significantly for models that have four layers. An increase in the dropout ratio improves the validation accuracy, both in terms of a slightly higher median value and in terms of a higher quantile values. An increase in the unit size, however, does not generally lead to an increased accuracy; 32 units appear to be the best choice. For 64 units, we can see a number of low-accuracy outliers, which can be attributed to an interconnection between the unit size, a high number of layers, and a low dropout value.

Table 6 Hyperparameter configuration for the top 4 models regarding validation accuracy (imbalanced dataset)

Hyperparameter	1	2	3	4
Validation accuracy	0.775	0.773	0.771	0.77
LSTM layers	1	1	1	1
Hidden units	32	64	64	32
L1 regularization (λ)	0.005	0.005	0.005	0.005
Sequence padding	200	200	200	200
Training epochs	100	100	100	100
Dropout	0.5	0.3	0.5	0.5
Batch size	128	256	128	256
Learning rate	0.01	0.01	0.01	0.01
Early stopping	20	20	20	20

The low standard deviations show the robustness of LSTMC and RFC regarding our datasets. Figure 2 depicts the training history of LSTMC on the balanced and imbalanced dataset. The left curves show the mean training and validation accuracy, while the curves on the right side present the mean training and validation loss. While the training loss continuously decreases, the validation loss starts increasing from epoch 15 on. Moreover, we observe that although the validation loss increases, the validation accuracy only slightly decreases. This effect is caused by different methods of measurement in combination with overfitting. While the loss is computed using probabilities, accuracy only relies on the highest probability observed for a probability distribution over the classes. Thus, the overall certainty of the model concerning the validation split decreases (loss increases), but the validation accuracy only changes slightly.

The steep training loss curve shows the capacity of the model to fit the training data. Although we apply means of regularization, the validation loss increases over time (epochs). Increasing the degree of regularization by adding $L1$ activity regularization and increasing the dropout probability resulted in a decreased training and validation accuracy on both datasets, while the training loss and validation loss keep diverging. Without the application of early stopping, the model yields a slightly lower training accuracy of 0.66 on the balanced and 0.74 on the imbalanced dataset. Although the LSTMC with early stopping exhibits a validation loss that is closer to the training loss, the validation performances differ only slightly from the standard LSTMCs. Finally, the variation in the learning rate did not result in an increased validation performance. The best model on the balanced dataset achieved a validation accuracy of 0.63. The best model on the imbalanced dataset achieved a validation accuracy of 0.70.

4.2.2 Test Evaluation

The evaluation of the naïve classifier yields a mean test f-measure of 0.19 on the balanced and 0.12 on the imbalanced dataset across the 10-fold. Although there are unique words within each class, the naïve bag of words approach appears to be unsuitable for the classification task on our evalua-

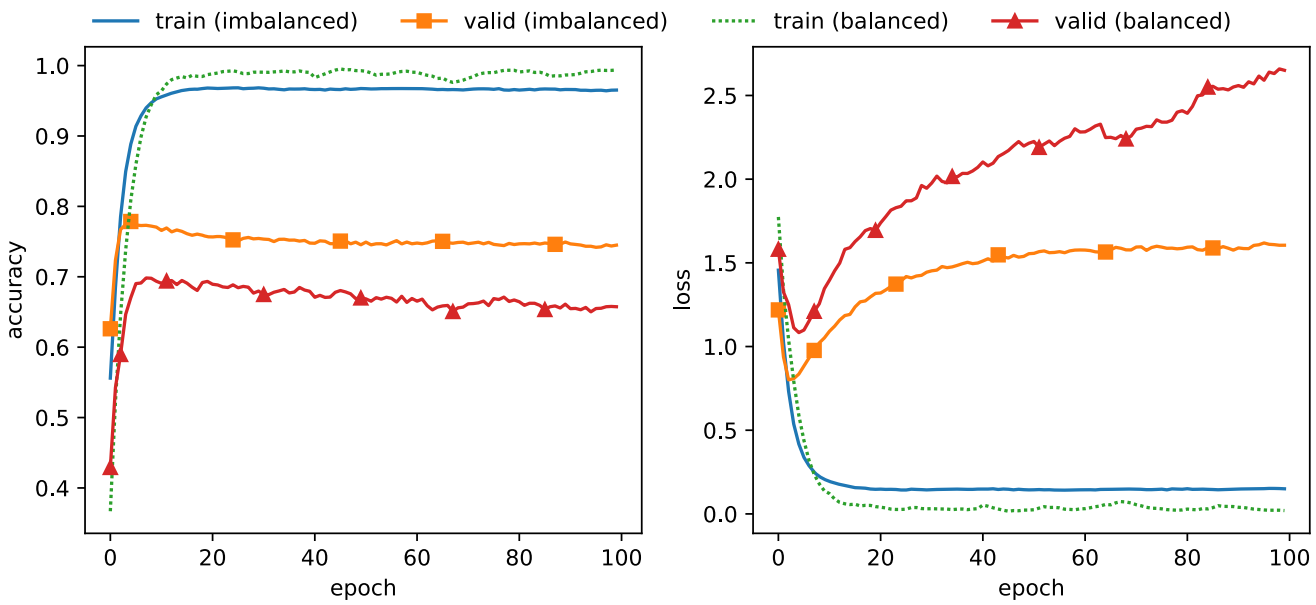

Fig. 2 Accuracy and loss for training and validation on both datasets

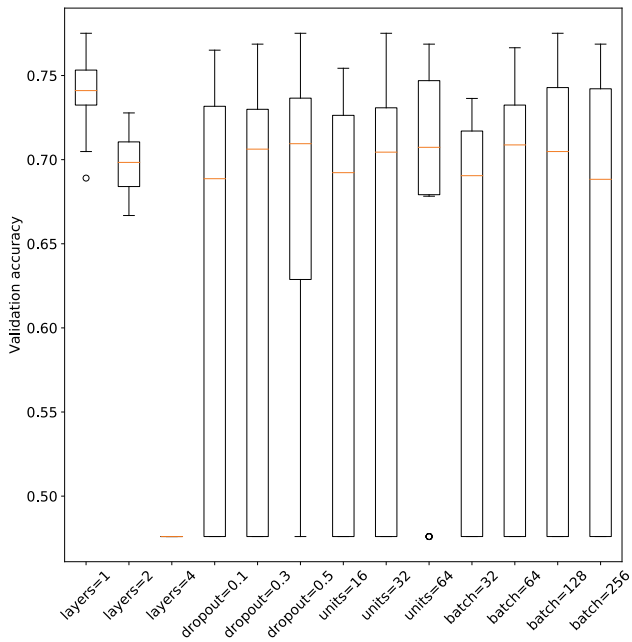

Fig. 3 Impact of different hyperparameter configurations (x-axis) on validation accuracy (y-axis)

tion dataset. One explanation could be the wide range of possible words that can be used to describe a certain error code. Another is the fact that a certain error can have many different manifestations. In addition, critical but frequent or non-specific words, such as negations, are not included in the set of distinct words for a class, altering its semantics. Therefore, in the following, we focus our analysis on the random forest as well as the two LSTMCs, where LSTMC denotes the standard training and LSTMCes the model generated using early stopping.

Tables 7 and 8 report on the performance of those three models, measuring the precision, recall, and f-measure for each of the ten classes as well as the weighted average (Eqs. 7 to 9) across all classes based on the test distribution. For the models trained on the balanced dataset (Table 7), we observe that all three models achieve an average f-measure of more than 0.44. The LSTMC with 0.56 and the LSTMCes with 0.55 slightly outperform the random forest with 0.44. While the LSTMC and LSTMCes only slightly differ regarding precision, RFC yields a lower recall. Furthermore, there are differences on how the models perform for different classes. Class 5 is a challenge for all models, resulting in f-measures lower than 0.25. In contrast with RFC, LSTMC and LSTMCes recognize class 0 fairly well. This is an unexpected behavior since class 0 is in fact the synthetic class containing several error codes.

On the imbalanced dataset, the RFC exhibits an average f-measure of 0.61. The f-measures of LSTMC and LSTMCes differ only slightly, but more than on the balanced dataset. LSTMCes outperforms RFC by 0.15. The performance gap between the LSTMCes and RFC in comparison with the balanced dataset only slightly increases by 0.03. The precision of class 0 decreases across all models, while the f-measure significantly increases. However, the f-measure of class 5 almost decreases to 0 for RFC. The metrics that are insensitive to class imbalance (Table 8) reveal that the LSTMCs fairly well handle the imbalanced dataset. In comparison with the balanced dataset, especially the performance of class 9 increased significantly, although the number of samples used for training remains the same for both datasets.

Table 7 Precision, recall, and f-measure of different models on the balanced dataset

	Class Samples	0 363	1 114	2 48	3 36	4 31	5 30	6 29	7 28	8 27	9 26	Avg
RFC	Precision	0.94	0.75	0.46	0.3	0.35	0.14	0.76	0.22	0.22	0.32	0.71
	Recall	0.22	0.67	0.64	0.76	0.48	0.31	0.89	0.75	0.7	0.8	0.44
	f-measure	0.35	0.71	0.54	0.43	0.41	0.2	0.82	0.34	0.33	0.46	0.44
LSTMC	Precision	0.84	0.81	0.6	0.39	0.3	0.18	0.61	0.51	0.27	0.28	0.68
	Recall	0.4	0.74	0.79	0.72	0.52	0.41	0.84	0.85	0.64	0.55	0.55
	f-measure	0.54	0.77	0.68	0.51	0.38	0.25	0.71	0.64	0.38	0.37	0.56
LSTMCes	Precision	0.86	0.83	0.71	0.4	0.31	0.17	0.71	0.53	0.27	0.27	0.71
	Recall	0.36	0.76	0.81	0.71	0.55	0.46	0.78	0.78	0.7	0.71	0.54
	f-measure	0.5	0.79	0.75	0.51	0.39	0.25	0.75	0.63	0.39	0.39	0.55

Table 8 Precision, recall, and f-measure of different models on the imbalanced dataset

	Class Samples	0 363	1 114	2 48	3 36	4 31	5 30	6 29	7 28	8 27	9 26	Avg
RFC	Precision	0.64	0.72	0.82	0.72	0.49	0.1	1.0	0.88	0.52	0.71	0.66
	Recall	0.95	0.78	0.25	0.21	0.05	0.0	0.47	0.4	0.04	0.45	0.67
	f-measure	0.77	0.75	0.38	0.33	0.09	0.01	0.64	0.55	0.07	0.55	0.61
LSTMC	Precision	0.78	0.87	0.73	0.61	0.41	0.33	0.87	0.81	0.43	0.49	0.73
	Recall	0.84	0.81	0.77	0.51	0.33	0.28	0.72	0.74	0.46	0.42	0.73
	f-measure	0.81	0.84	0.75	0.55	0.37	0.31	0.79	0.77	0.44	0.45	0.73
LSTMCes	Precision	0.78	0.86	0.88	0.72	0.49	0.42	0.92	0.85	0.48	0.67	0.76
	Recall	0.89	0.86	0.84	0.49	0.34	0.26	0.76	0.77	0.47	0.48	0.77
	f-measure	0.83	0.86	0.86	0.58	0.39	0.32	0.83	0.81	0.47	0.56	0.76

Figures 4 and 5 show the confusion matrices of the LSTMCes (on the left) and the RFC (in the center) as well as the differences between them (on the right) for both the balanced and the imbalanced dataset. These matrices provide more details on how the models performed with respect to the individual classes. We are able to see which classes the models most often confuse. Since LSTMC and LSTMCes performed equally well, we only show the LSTMCes here.

For the balanced dataset (in Fig. 4), classes 0, 4, and 5 appear to be challenging for LSTMCes and RFC alike. The most frequent misclassifications occur in class 0. The RFC displays more pronounced difficulties to decipher classes. Whereas it often misclassifies samples from class 5 as class 7 and 8, it also confuses classes 0 and 8. The LSTMCes more often misclassifies samples of class 1 to 9 as class 0 than the RFC.

The models trained on the imbalanced dataset (in Fig. 5) appear to be troubled by the correct classification of complaints in classes 4 and 5. We see that especially the RFC often wrongly assigns the code 0 to samples from other classes. The LSTMCes suffers from the same problem, but to a much lower extent. While RFC is barely able to correctly classify samples of class 4 (0.05), 5 (0.00) and 8 (0,04), LSTMCes correctly assignes 0.34 resp. 0.26 resp. 0.47 of the samples to those three classes.

We can conclude that the LSTMCes generalizes much better and distinguishes samples of classes 1–9 from class 0, giving it a clear advantage in supporting the employees in correctly assigning error code in an automated way.

4.3 Limitations and Threats to Validity

The proposed LSTMCs achieve an average f-measure of 0.56 resp. 0.55 on the balanced and 0.73 and 0.76 on the imbalanced test dataset. Depending on the predicted classes, the f-measures range from 0.25 to 0.86. We observed that early stopping did not considerably influence the overall performance of the LSTMC, but the sampling strategy did. Using the complete and imbalanced dataset yielded a higher overall performance, but exhibited a tendency toward wrongly assigning samples to class 0. This can be explained by the size and the diversity of class 0, which represents 177 error codes (10 to 186). During the training stage, the network thus encounters many samples from this class. Since those samples actually come from different error codes, they do not have many similarities, for the network to recognize and generalize.

The selection of the error codes for prediction is based on the cutoff of 500 samples. A different selection of error codes or a clustering of error codes is likely to influence

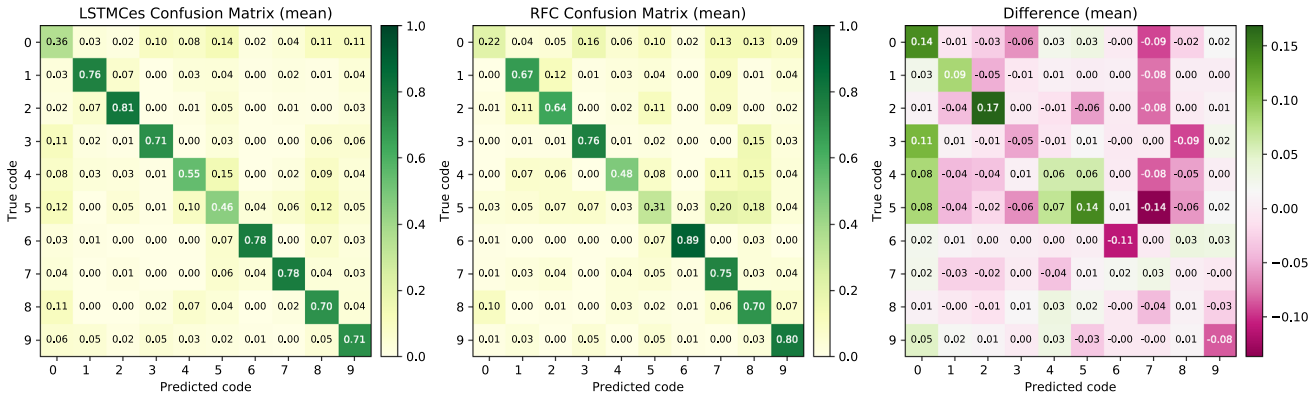

Fig. 4 Normalized confusion matrices of the LSTMCes (left) and the RFC (right) (balanced dataset)

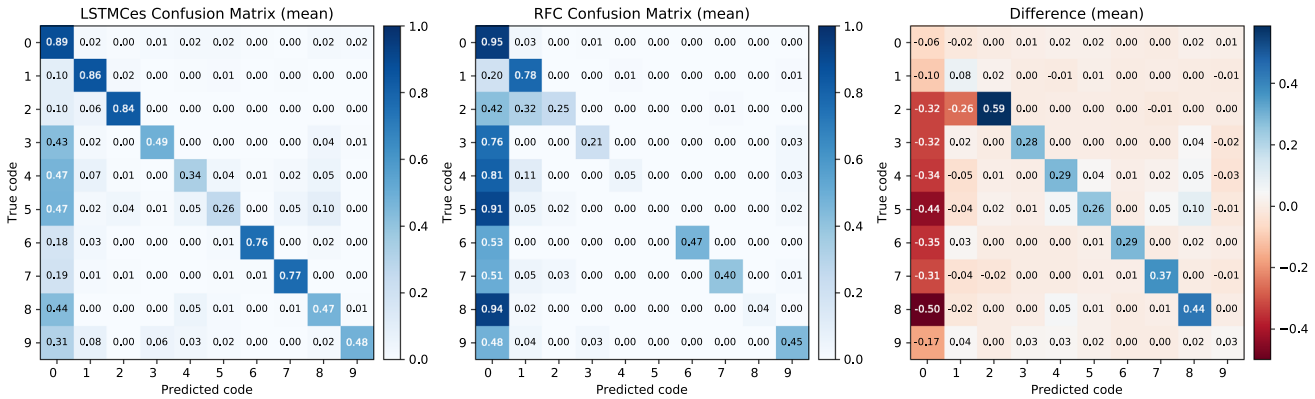

Fig. 5 Normalized confusion matrices of the LSTMCes (left) and the RFC (right) (imbalanced dataset)

the prediction performance. However, the optimal selection of clusters is not only determined by the achieved accuracy, but rather its usefulness in a concrete application scenarios. Thus, further constraints, e.g., misclassification costs, need to be considered.

The underlying problem here is that there are so many error codes and their distribution is skewed among the incoming complaints. The medical technology company, which provided the data, has already acknowledged this problem. Having many very rare error codes is not helpful for an efficient quality management process either. Therefore, one action point concluded from our research could be to analyze the available data to suggest new, more evenly distributed error codes.

As we see in the evaluation, the LSTMCs outperformed the RFC in correctly classifying complaint descriptions into error codes. However, this comparison needs to be put into context. We used a fairly standard random forest configuration and did not tune the hyperparameters to our problem. Besides that, a random forest will typically need much less training resources than an LSTM, while still producing satisfactory results. Depending on the objectives on the underlying business case (i.e., balancing resources and performances), a company might select the random forest classifier for supporting their own 8D process.

The provided dataset represents a snapshot covering incoming complaints of a predefined period of time. Since frequencies of error codes vary over time, the snapshot is not necessarily representative for the classification task. In our opinion, using all of the provided data was the best option to assess the practical applicability of automated support for the complaint handling process, because a company that wanted to realize such a support would face the same problem. In a real-life scenario, we would recommend a regular re-training of the models on more current data to account for a potential variation in problem causes and error codes. Moreover, we would suggest to consider time as a potential confounder variable.

Finally, we are aware that the work presented does not enable the reader to replicate the evaluation results, because the evaluation data are not publicly available. Nevertheless, by providing the concept implementation and a detailed description of the evaluation setup and experiments conducted, we provide the reader with the necessary information to reproduce the results in similar application scenarios.

4.4 Practical Adoption

Besides the scientific and technical limitations that we discuss in the previous section, there are also practical considerations to make before our prototype can be used in a productive environment. The first one concerns the balancing of the dataset, which we tested out above. As we found, balancing the dataset improved the classification performance for some classes, it decreased the performance for others. So, balancing the dataset does not produce a better prediction model than keeping the (imbalanced) dataset as is. Thus, choosing one model over another requires a business case quantifying the impact of a misclassification in a particular class.

We have seen that in its current state the best model still misclassifies about a fourth of all incoming complaints. It is unclear to us how much better the classification can become, even if we optimize the network architecture or have access to more training data. However, our approach was developed in close collaboration with the medical technology company, which provided the application case and the data. The responsible quality managers were closely involved in our conceptual design and see much potential in its realization. The complaint managers, who currently handle the incoming complaints, were also interested in our solution. They pointed out that even a small step like this one could save them some time, which they can use for completing the other steps of the 8D process. This is particularly relevant for known and less critical error sources, which should be handled quickly, such that the employees can focus on finding the causes for new and potentially more severe complaints.

After our initial results were published in the workshop paper, a first prototype was implemented, which was able to assign a likely error code to a given (previously unseen) complaint description. It was qualitatively evaluated in a workshop with the project partners. This prototype demonstrates the general feasibility of our approach and the pipeline to transport a pre-trained neural network into a software architecture. For this contribution, we retrained the network after cleaning the original dataset, removing duplicate assignments. We also tested an early stopping strategy and inspected the differences between balanced and imbalanced training data.

In the meantime, "Reklamation 4.0" project, in which we conducted the described research, was successfully completed. As part of its research dissemination strategy, our application partner intends to integrate the developed prototype into the process-supporting IT system, giving end users the opportunity to evaluate the approach directly in a productive environment.

5 Challenges to the Automated Support of Complaint Handling Processes

The previous two sections described our prototype for automating one small step within the 8D process as well as the challenges that we encountered. In addition to this rather practical discussion, we also need to consider the more large-scale conceptual challenges that come with applying ML-based automation in productive environments in the medical technology industry. These will be discussed in this section.

In comparison with other industries and application scenarios, a wrongly classified complaint in the medical industry could have severe legal and financial consequences. Therefore, it is probably not feasibly to completely automate the complaint handling process. Instead, we could consider the inputs of, e.g., complaint classification models as recommendations, which can support the employee handling the complaint. This could significantly increase the employee's performance. Depending on user preferences and the model's confidence on the predicted class, a certainty value or a ranking of the k top error codes could be provided to the employee. Both options would put the model's recommendations into perspective and foster their critical reflection.

On the other hand, it is not given that the employees that currently handle the complaints perceive the prediction as a helpful tool that supports their daily work. If recommendations are wrong, take too much time, or do not make sense, the employees will be less likely to use the prediction, making its potential benefits obsolete. Alleviating this risk takes multiple steps. First, we have to ensure that the approach produces high-quality recommendations sufficiently quickly. Second, we have to examine how the tool performs in a practical setting and how the employees integrate it into their own handling of the complaint processes. Third, if it turns out that employees do not use the tool, because they do not understand why it makes a certain recommendation, we might consider adding a tool that explains the decisions, using methods of Explainable AI [35]. Overall, a practical use of our developed prototype requires us to walk the fine line between helping an employee and ensuring that the employee herself makes all final decisions regarding the complaint handling.

Another set of challenges concerns the transferability of our prototype to other processes or other companies. The first one relates to data availability and data quality. Like all deep learning approaches, our model requires a comparatively large amount of data to deliver meaningful results. If a company does not have the required amount of data, because it sells fewer products and therefore has fewer complaints, it might not be able to realize the approach as we describe it here. This is especially true, if there are many different error codes, which become harder to classify if less data are available. These problems also occur in companies or processes

with limited data quality, such as incomplete descriptions or incorrectly classified complaints. In this case, our model will not be able to make reliable predictions, as it does not have access to a sufficiently large sample of correct examples to generalize from.

Furthermore, the handling of the complaint process in medical technology depends on the nature of the individual product. The same legal regulations apply to inexpensive commodity products and to complex medical devices, but their complaint handling differs considerably. Consider the differences between an erroneous adhesive patch and an ultrasound machine. In the first case, the customer might not even realize that there is a problem with the product itself or might not bother to file a complaint about it. If there is a complaint, the company can easily provide them with a new package of patches at a very low cost. On the other hand, if an ultrasound machine at a hospital breaks down, the employees there will almost always file a complaint with the manufacturer. Replacing or repairing such a machine can become very expensive very quickly, so if there is an error source in the production process, the manufacturing company needs to know about it in order to avoid the same error in other machines.

In this respect, automated support of complaint handling processes might not be applicable to every medical technology company. Each individual company needs to investigate whether partial automation is beneficial in their respective processes. This is particularly important, because after the initial training, the results of our model should regularly be supervised and re-trained with the appropriate current data, such that the quality of the result can be maintained or ideally improved and changes in company policy are reflected the model. Thus, the company requires an appropriate infrastructure consisting of computational power, hardware either on premise or as a service, as well as experts maintaining and developing the models. All of this requires a considerable investment of financial resources, which only makes sense economically if the number and/or the severity of complaints is sufficiently high. However, companies could also regard such an investment as a foray into artificial intelligence and machine learning, which have many other applications, therefore justifying higher research and development costs.

6 Related Work

6.1 Machine Learning for Quality Management

There are several approaches that apply other machine learning techniques in quality management. Coussement et al. introduce a binary classifier that is able to distinguish complaint from non-complaint emails [8]. The approach consists of a rule-based feature extraction and a boosting algorithm for binary classification. Ko et al. deal with the detection of anomalies in engine production [20]. Their approach combines data from production across supply chains with customer data and other quality data to classify the engines' quality. The approach of Weiss et al. is also concerned with the prediction of product quality along a supply chain, but considering microprocessors [40]. In this context, the main challenge is the lengthy production process and the availability of only little measurement data.

In addition, there are several approaches that develop models for quality forecasts across multiple production steps. Lieber et al. describe a case of application from the steel industry, in which the quality of interstage products is in focus [24]. Techniques of supervised and unsupervised machine learning, such as clustering or decision trees, are applied to data recorded during production (e.g., by sensors) to identify the most important factors influencing subsequent product quality. The approach of Arif et al., on the other hand, comes from the production of semiconductors, where decision trees are also used to develop a predictive model [3].

6.2 Deep Learning and NLP in BPM

Diverse applications for deep neural networks in BPM have recently been presented. Process prediction, i.e., forecasting the future behavior of running process instances, is arguably the most prominent. Our own approach to process prediction encodes the event log into a word embedding and uses this embedding to train a neural network that is able to predict the next steps in a process sequence [13,14]. Tax et al. present a similar approach, but they employ feature vectors and a one-hot encoding to represent the log [39]. Mehdiyev et al. refine those approaches with a more complex network architecture [27]. All of these methods focus on predicting the next process step, but there are also attempts to predict other process attributes, such as cost, runtime or process outcomes [25]. The paper by Di Francescomarino et al. provides an overview over the current state of the art in process prediction [10]. In very recent publications, neural networks are employed for simulating process logs[7] and supporting resource allocation in business processes [31].

While most of the deep learning-based process predictions rely on long short-term memory (LSTM) cells, there are also approaches that transform process event logs into spatial data to take advantage of convolutional neural networks [32].

Besides process prediction, another important application for machine learning in BPM is anomaly detection, i.e., the identification of process instances that deviate from the usual process behavior. Nolle et al. present an approach

based on autoencoders, which is able to consider process attributes in addition to sequences [30]. Lahann et al. build on this research to identify compliance violations in accounting data [22]. A previous approach by Böhmer and Rinderle-Ma uses likelihood graphs to a similar avail [6].

Regarding different types of representing process log data for neural networks, De Koninck et al. use representation learning to learn embeddings for activities, traces, logs, and processes [9]. Besides those uses focusing on instance log data and process monitoring, machine learning can also be used to support process modeling [17].

Similar to our approach, other researchers have used NLP for process automation. Shing et al. present an approach to extract workflow descriptions from written documents, specifically unstructured e-mails [36]. In an application case at an IT service provider, Koehler et al. automatically extract problem descriptions from a multi-language ticket system [21].

7 Conclusion and Outlook

In this contribution, we consider the potentials of automating complaint handling processes in the medical technology industry. After examining those potentials in the 8D reference process, we use data from a large manufacturer of medical products to design, implement, and evaluate a prototype for providing automated support to the second step of the 8D process (problem description). This prototype consists of a deep neural network, which is able to assign the correct error source to a textual complaint description in more than 75% of all cases. We evaluated numerous network configurations to identify the network with the highest performance. The performance of our approach was 3 times better than the keyword based naïve classifier which we used as a baseline comparison. It overall performed better than a random forest classifier and could better distinguish different complaint classes. In addition, we examined the organizational and technical challenges of automating complaint handling process.

Our results show the general potential of machine learning for process automation in medical technology. Compared to "classical" approaches for process automation, such as keyword search, ML has several advantages. First, it relies on only little domain knowledge (such as potential symptoms of patients or malfunctioning products). This knowledge is difficult to acquire and codify, requiring a high level of engagement by domain experts. Second, ML is able to deal with customer-specific vocabulary choices, considering semantic instead of syntactic matches with the given categories. Third, it may identify patterns and draw conclusions, which domain experts may have overlooked. So, ML has the potential to support employees instead of burdening them.

Partial automation of the complaint handling process supports employees in their work, leaving them with more time to identify and remove the causes of occurring complaints. This is particularly relevant for less experienced employees, since the necessary experience for quickly filing an 8D report can be at least partially replaced by a trained neural network. However, it is infeasible for the medical technology industry to fully automate its complaint handling process and remove the human employee from the decision process, at least for the foreseeable future. The evaluation of, e.g., the complaint criticality is an extensive decision, which can lead to official sanctions or expensive recalls, damaging a company's reputation and its revenue. In case of a critical error, the final assessment will not be automated, but be carried out by an employee, or rather a team of employees, instead.

In addition, automation is also relevant for system validation. All production-relevant IT systems in a controlled production environment must be formally validated, documented, and tested, before the company is allowed to use them. From this point on, the validated computer system may only be changed within very strict boundaries. Inspectors from public health agencies, such as the American Food and Drug Administration (FDA), perform regular on-site audits of those systems and ban companies from selling in their respective markets if the systems are not properly validated. If a neural network made independent decisions within a process related to the manufacturing of medical products instead of just supporting the employees in their decisions, it would be regarded as a production-relevant system. This would be problematic for two reasons. First, the training stage of a neural network is a non-deterministic process, so a validated network could not be retrained on new data, even if that would improve its performance. Second, the auditor might require an explanation of the decision that the system makes, which is difficult to provide for black-box models like neural networks. On the other hand, the network's decision may be better than those of the human employee, so it would not make sense not to use them. This is a fundamental dilemma of the regulated environment, which the industry and public authorities have to address in order to find the best approach to promote public health in the future.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Aalst WM, Bichler M, Heinzl A (2018) Robotic process automation. Business Inf Syst Eng 60(4):269–272
2. Abuhav I (2018) ISO 13485: 2016: A complete guide to quality management in the medical device industry. CRC Press, CRC Press
3. Arif F, Suryana N, Hussin B (2013) A data mining approach for developing quality prediction model in multi-stage manufacturing. Int J Comput Appl 69(22):35–40
4. Behrens BA, Wilde I, Hoffmann M (2007) Complaint management using the extended 8D-method along the automotive supply chain. Prod Eng Res Devel 1(1):91–95
5. Bishop, C.: Regularization and complexity control in feed-forward networks. In: Proceedings international conference on artificial neural networks ICANN'95, vol. 1, pp. 141–148 (1995)
6. Böhmer K, Rinderle-Ma S (2017) Multi instance anomaly detection in business process executions. In: Carmona J, Engels G, Kumar A (eds) Business process management. Springer, Berlin, pp 77–93
7. Camargo M, Dumas M, González-Rojas O (2019) Learning accurate lstm models of business processes. In: Hildebrandt T, van Dongen B, Röglinger M, Mendling J (eds) Business process management. Springer, Berlin, pp 286–302
8. Coussement K, van den Poel D (2008) Improving customer complaint management by automatic email classification using linguistic style features as predictors. Decis Support Syst 44(4):870–882
9. De Koninck P, vanden Broucke S, De Weerdt J (2018) act2vec, trace2vec, log2vec, and model2vec: representation learning for business processes. In: Weske M, Montali M, Weber I, vom Brocke J (eds) Business process management. Springer, Berlin, pp 305–321
10. Di Francescomarino C, Ghidini C, Maggi FM, Milani F (2018) Predictive process monitoring methods: Which one suits me best? In: Weske M, Montali M, Weber I, vom Brocke J (eds) Business process management. Springer, Berlin, pp 462–479
11. European Commission: The rules governing medicinal products in the European Union - EU Guidelines to Good Manufacturing Practice
12. European Parliament, Council of the European Union: Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EE (2017)
13. Evermann J, Rehse JR, Fettke P (2017) A deep learning approach for predicting process behaviour at runtime. In: Dumas M, Fantinato M (eds) Business process management workshops. Springer, Berlin, pp 327–338
14. Evermann J, Rehse JR, Fettke P (2017) Predicting process behaviour using deep learning. Decis Support Syst 100:129–140
15. Foidl H, Felderer M (2015) Research challenges of industry 4.0 for quality management. In: International conference on enterprise resource planning systems, pp 121–137. Springer
16. Hake P, Rehse JR, Fettke P (2019) Supporting complaint management in the medical technology industry by means of deep learning. In: Di Francescomarino C, Dijkman R, Zdun U (eds) Business process management workshops. Springer, Berlin, pp 56–67
17. Hake P, Zapp M, Fettke P, Loos P (2017) Supporting Business Process Modeling Using RNNs for Label Classification. In: F Frasincar, A Ittoo, LM Nguyen, E Métais (eds.) Natural language processing and information systems: 22nd international conference on applications of natural language to information systems, NLDB 2017, Liège, Belgium, June 21-23, 2017, Proceedings, pp. 283–286. Springer
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. (2014). arXiv:1412.6980
20. Ko T, Lee JH, Cho H, Cho S, Lee W, Lee M (2017) Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data. Ind Manag Data Syst 117(5):927–945
21. Koehler J, Fux E, Herzog FA, Lötscher D, Waelti K, Imoberdorf R, Budke D (2018) Towards intelligent process support for customer service desks: Extracting problem descriptions from noisy and multi-lingual texts. In: Teniente E, Weidlich M (eds) Business process management workshops. Springer, Berlin, pp 36–52
22. Lahann J, Scheid M, Fettke P (2019) Utilizing machine learning techniques to reveal vat compliance violations in accounting data. In: 21th conference on business informatics. IEEE
23. Lasi H, Kemper HG, Fettke P, Feld T, Hoffmann M (2014) Industry 4.0. Business Inf Syst Eng 4(6):239–242
24. Lieber D, Stolpe M, Konrad B, Deuse J, Morik K (2013) Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. In: Conference on manufacturing systems, pp. 193–198. Procedia CIRP
25. Liu Q, Wu B (2018) Prediction of business process outcome based on historical log. In: Proceedings of the 10th international conference on computer modeling and simulation, ICCMS 2018, pp 118-122. Association for computing machinery, New York, NY, USA
26. Manz S (2019) Medical device quality management systems: strategy and techniques for improving efficiency and effectiveness. Elsevier, Amsterdam
27. Mehdiyev N, Evermann J, Fettke P (2020) A novel business process prediction model using a deep learning method. Business Inf Syst Eng 62(2):143–157
28. Mendling J, Decker G, Hull R, Reijers HA, Weber I (2018) How do machine learning, robotic process automation, and blockchains affect the human factor in business process management? Commun Assoc Inf Syst 43(1):297–320
29. Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In: Proceedings of the international conference on language resources and evaluation (LREC 2018)
30. Nolle T, Luettgen S, Seeliger A, Mühlhäuser M (2018) Analyzing business process anomalies using autoencoders. Mach Learn 107(11):1875–1893
31. Park G, Song M (2019) Prediction-based resource allocation using lstm and minimum cost and maximum flow algorithm. In: 2019 international conference on process mining (ICPM), pp. 121–128
32. Pasquadibisceglie V, Appice A, Castellano G, Malerba D (2019) Using convolutional neural networks for predictive process analytics. In: 2019 international conference on process mining (ICPM), pp. 129–136
33. Pearl J (2019) The seven tools of causal inference, with reflections on machine learning. Commun ACM 62(3):54–60
34. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E

(2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
35. Rehse JR, Mehdiyev N, Fettke P (2019) Towards explainable process predictions for industry 4.0 in the dfki-smart-lego-factory. KI - Künstliche Intelligenz 33(2):181–187
36. Shing L, Wollaber A, Chikkagoudar S, Yuen J, Alvino P, Chambers A, Allard T (2019) Extracting workflows from natural language documents: A first step. In: Daniel F, Sheng QZ, Motahari H (eds) Business process management workshops. Springer, Berlin, pp 294–300
37. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout?: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
38. Sulieman L, Gilmore D, French C, Cronin RM, Jackson GP, Russell M, Fabbri D (2017) Classifying patient portal messages using convolutional neural networks. J Biomed Inform 74:59–70
39. Tax N, Verenich I, La Rosa M, Dumas M (2017) Predictive business process monitoring with lstm neural networks. In: Dubois E, Pohl K (eds) Advanced information systems engineering. Springer, Berlin, pp 477–492
40. Weiss S, Dhurandhar A, Baseman R (2013) Improving quality control by early prediction of manufacturing outcomes. In: International conference on knowledge discovery and data mining, pp. 1258–1266. ACM
41. Yang Y, Xu DL, Yang JB, Chen YW (2018) An evidential reasoning-based decision support system for handling customer complaints in mobile telecommunications. Knowl-Based Syst 162:202–210