XAI in the Audit Domain - Explaining an Autoencoder Model for Anomaly Detection

Nico Gnoss
University of Applied Sciences Hamburg, Germany, nicolai.gnoss@haw-hamburg.de

Martin Schultz
University of Applied Sciences Hamburg, Germany, martin.schultz@haw-hamburg.de

Marina Tropmann-Frick
University of Applied Sciences Hamburg, Germany, marina.tropmann-frick@haw-hamburg.de

XAI in the Audit Domain - Explaining an Autoencoder Model for Anomaly Detection

Nicolai Gnoss, Martin Schultz, and Marina Tropmann-Frick

HAW Hamburg, Department of Computer Science, Hamburg, Germany
{nicolai.gnoss,martin.schultz,marina.tropmann-frick}@haw-hamburg.de

Abstract. Detecting erroneous or fraudulent business transactions and corresponding journal entries imposes a significant challenge for auditors during annual audits. One possible solution to cope with these problems is the use of machine learning methods, such as an autoencoder, to identify unusual journal entries within individual financial accounts. There are several methods for the interpretation of such black-box models, summarized under the term eXplainable Artificial Intelligence (XAI), but these are not suitable for autoencoders. This paper proposes an approach for interpreting autoencoders, which consists of labeling the journal entries first using a previously trained autoencoder and then training models suitable for applying XAI methods using these labels. The results obtained are evaluated with the help of human auditors, showing that an autoencoder model is not only able to capture relevant features of the domain but also provides additional valuable insights for identifying anomalous journal entries.

Keywords: XAI, Autoencoder, Anomaly Detection, Auditing, Journal Entries

1 Introduction

Many companies are required by law to have their financial statements audited annually by external auditors. The purpose of this annual audit is to assure prospective stakeholders that the published financial information of the entity complies with applicable accounting standards and is free from misstatements (e.g., due to fraud or errors). Nowadays, companies try to standardize and automate their business processes as much as possible, relying mainly on information systems (IS). For example, enterprise resource planning systems (ERP) process many business transactions on a daily basis, the financial impact of which is recorded as journal entries in the financial accounts. Consequently, this leads to an increasing amount of electronically available data that is relevant for auditing.

In this context, the auditing standards require an analysis of the accounting data at a detailed level, namely at journal entry level [1]. With computer-assisted audit techniques [2], auditors can examine the entirety of journal entries. In most cases, static rules are applied, which only check a few attributes of a journal entry at a time (e.g., postings with a high amount or postings close to fiscal year-end), resulting in a high false positive rate. To cope with this problem and the large amount of data, the application of machine learning methods has been discussed in auditing research recently [3, 4]. Several research studies have shown that for anomaly detection tasks in auditing machine learning methods can indeed achieve valuable results [5, 6].

17th International Conference on Wirtschaftsinformatik
February 2022, Nürnberg, Germany

These high-performing but complex models generated by machine learning methods are called black-box models as their internal logic is not transparent, and their results are not self-explanatory. To foster practical application, especially in the audit domain, approaches to make them explainable are required. By verifying that learned models adequately capture relevant aspects of the domain, auditors will gain confidence in these black-box models and benefit from the new insights these models provide. The research area eXplainable Artificial Intelligence (XAI) deals with this topic.

This paper proposes an approach for explaining an autoencoder model, implemented and trained with a real-world dataset in a previous work [6], using XAI methods. To explore whether such methods add value in audit practice, the obtained results are evaluated with auditors. The remainder of the paper is structured as follows: The next section covers related research regarding anomaly detection in the context of external auditing and XAI. Section 3 presents three different XAI methods and their respective results. In section 4, these results are evaluated together with auditors. The paper concludes in section 5 with a summary followed by a conclusion and implications for future research work.

2 Related Work

2.1 Anomaly detection in the context of auditing

In recent decades, it is often stated that the audit domain has lagged behind technological progress as a large number of regulations and audit standards causing a delaying effect [3]. However, various research studies have shown that the use of artificial intelligence can have a great impact on auditing and change it significantly [4, 7]. *Bay et al.* presented a method for identifying irregularities in a company's general ledger. Their method is based on the development of features that capture irregularities in the data and applying a classifier afterwards to find suspicious financial accounts [8]. *McGlohon et al.* applied a link analysis to flag suspicious accounts not only based on irregularities in a single account, but on accounts that are linked together through shared transactions [9].

A major obstacle in identifying suspicious journal entries is the absence of labels used for classifying entries. To overcome this problem, unsupervised learning methods such as an autoencoder can be used. *Schreyer et al.* use a deep autoencoder network to detect fraudulent journal entries in a large set of financial data extracted from ERP systems. For their quantitative evaluation, they injected a small fraction (0.03-0.06%) of synthetic anomalies. In comparison with other unsupervised methods, the autoencoder approach performed better [5]. *Schultz et al.* prove that anomaly detection using an autoencoder is also possible on smaller datasets by training the autoencoder not on all journal entries of a complete fiscal year, but only on those of individual accounts. Moreover, their approach does not use any synthetic anomalous journal entries [6].

2.2 Explainable AI

In general, there is a trade-off between the performance of a machine learning model and its ability to produce explainable and interpretable models [10]. The high-performing, but complex models are often referred to as so called black-box models, which means that information about their inner logic is missing [11]. In some domains this lack of transparency is acceptable, but in other domains, for example healthcare and also the audit domain, it is a huge drawback. To overcome this drawback and to foster the adoption of such models in practice, the field of eXplainable Artificial Intelligence (XAI) has risen a lot of attention in the last couple of years.

The focus of XAI lies on interpretability and explainability, both terms are used interchangeably by researchers. *Miller et al.* describes interpretability as the extent to which a human can understand the cause of a decision [12]. In general, XAI can be specified as a collection of machine learning techniques that enable human users to understand, trust, and effectively handle the emerging generation of AI systems [13].

Linardatos et al. classifies XAI methods based on various criteria. One possible criterion is the distinction between transparent models and post-hoc explainability methods. Transparent models are inherently interpretable, which means they are interpretable without further application of an additional method or algorithm. Post-hoc-explainability describes all methods that are applied to an already trained black-box model, for example SHapley Additive exPlanations (SHAP) [14]. Furthermore, XAI methods can be distinguished by their locality. Methods that only explain a specific instance are called local, whereas methods that explain the whole model are called global [15]. *Samek et al.* mention the verification, improvement, and learning from systems as goals of XAI [16]. Another essential objective is compliance with legislation. For example, the new General Data Protection Regulation (GDPR) anchors the right to an explanation of automated individual decisions [17].

A large part of the literature are comprehensive surveys, which include topics such as terminology, objectives, and summary of methods in the field of XAI [13, 15, 18]. This paper aims to address the lack of papers about real-world applications by applying three XAI methods on models trained on real data for external auditing. Our goal is to open black-box models, in this case an autoencoder trained for anomaly detection on a set of journal entries and explore whether the model can not only capture relevant aspects of the domain, but additionally provide new information for domain experts. To accomplish this, the results obtained from three different XAI methods are discussed with auditors to evaluate the applicability and usefulness of XAI methods in the audit domain.

3 Analysis of the prediction of different ML models

The autoencoder model is explained by determining which features of a journal entry have the greatest influence on the decision whether it is a regular or a suspicious one. To avoid an one-sided explanation, the results of three different XAI methods are compared.

3.1 Applicability of XAI methods for autoencoder

An autoencoder neural network is an unsupervised machine learning technique, which consists of two components, an encoder and a decoder. The encoder *e(x)* maps an input of $x \in R^{dim}$ to a hidden compressed representation *h*, and the decoder *d(h)* tries to reconstruct the original input from this compressed representation, such that $d(e(x)) \approx x$.

The main concept is that both encoder and decoder are trained together to minimize the difference between the original input and its reconstruction. This difference is called reconstruction error and can be used as an indicator for anomalies (e.g., journal entries that differ significantly from other journal entries).

In general, post-hoc-explainability methods are used to explain the predictions of a black-box model like an autoencoder [15]. However, there are two problems with autoencoder models that prevent a direct application of post-hoc-explainability methods.

- The prediction of an autoencoder is the best possible reconstruction of the input. Therefore, it does not give a direct answer to the question in which we are interested, for example whether an instance is an anomaly or not. This answer is usually derived indirectly from the reconstruction error. Most post-hoc-explainability methods try to explain the prediction of a model but regarding autoencoders we are not interested in their prediction, which is why their application is problematic.
- In case of the autoencoder, the prediction is not just a single number, such as in a classification where a probability between 0 and 1 is calculated, but the prediction has the same dimension as the input. This multidimensionality complicates the use of post-hoc-explainability methods.

Regarding these problems, Antwarg et al. proposes one approach to explain anomalies detected by autoencoders using a post-hoc method (SHAP), which consists of producing one explanation per high error feature for the prediction of a single instance, making this approach local. In contrast, our global approach consists of the following steps:

1. Label the journal entries using the autoencoder.
2. Apply either an inherently interpretable model or a black-box model, which can be directly interpreted by a post-hoc-explainability method, and train it with the labeled journal entries.
   (a) *Inherently interpretable model*: Explain the autoencoder model directly depending on the chosen model type.
   (b) *Black-box model*: Explain the autoencoder model using a post-hoc method.

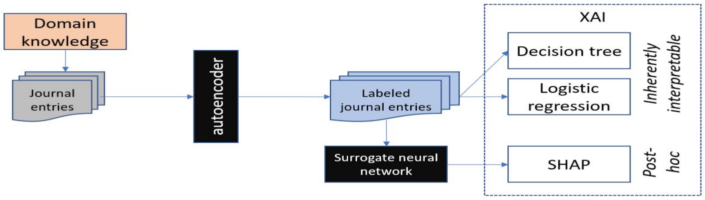


Figure 1. Illustration of the process for explaining the predictions of an autoencoder

The idea behind this approach is as follows. While we cannot directly explain the autoencoder with post-hoc-explainability methods, we instead train models that can be explained easier. In approximating the decision behavior, we assume that the models assign high relevance to the same features as the autoencoder, which can eventually be revealed by the application of XAI methods. In our research two inherently interpretable models (decision tree and logistic regression) and one black-box model (surrogate neural network) in combination with SHAP are used to explain the autoencoders predictions.

3.2 Dataset and data preparation

The real-world dataset for our analysis has been extracted from an SAP ERP system of a food production and trading company. In total, the dataset consists of 72.917 journal entries, but in this paper we are only focusing on the domestic revenue account with 6.643 journal entries. All attributes are categorical except *AmountLocalCurrency*. Due to the strict data privacy regulations in the audit domain, all categorical journal entry attribute values of the original dataset have been anonymized using a one-way hash function. Moreover, three additional boolean attributes are computed to incorporate relevant aspects of the audit domain (domain knowledge): 1) *DatesEqual* - indicating whether all three date fields (posting date, document date, creation date) are equal or not, 2) *OneDateOutsideAccountYear* - indicating whether one date field lies outside the date range of the fiscal year under review and 3) *postingCloseToFiscalYearEnd* - indicating whether the posting date is close to the end of the fiscal year. To take the structure of the posting document into account in the analysis, sorted lists of all account numbers on the debit and credit side of the respective accounting document are supplemented for each journal entry line item (*credit_accountno_list* / *debit_accountno_list*). The same approach is also used for the attributes for the list of creditor/debitor identifiers *CredDebNumberList* and the list of creditor/debitor countries *DebCredCountryList*.

In the preprocessing, all categorical attributes are one-hot-encoded. We consider a feature as a concatenation of the attribute name and its value, for example tcode_4cc8, where tcode is the attribute name and 4cc8 its actual value. The only numerical attribute *AmountLocalCurrency* is normalized, resulting in a mean value of zero and a variance of one. Each journal entry has 25 attributes before and 357 attributes after preprocessing.

3.3 Decision Trees

Way of Explanation Decision trees are particularly well-suited for explaining individual predictions. After selecting an instance, starting from the root node, it can be comprehended split by split why this instance was assigned to a certain class (in case of classification). A huge advantage of decision trees is that they provide a graphical way of interpretation that people can easily understand, on the premise that the depth of the tree is manageable. It encourages to do what-if-analyses. Taking a different path at a particular split node, one can observe in which class a particular instance falls. In addition, decision trees offer an alternative way for explaining predictions at a global level. Some specific implementations measure the importance of a feature by looking at how much the tree node that use that feature reduce impurity on average compared to the respective parent node. These values are usually calculated automatically after the training, cf. Table 1.

Table 1. Decision Tree - Feature Importance Values

Feature	Feature Importance Value
DebitCreditIndicator_D	0.4153
OneDateOutsideAccountYear_T	0.1307
DatesEqual_F	0.1302
tcode_4d15	0.0867
AmountLocalCurrency	0.0790
taxcode_2bf0	0.0340
debitornameList_3d0d906868f4e76770f1683a6	0.0303
taxcode_2b29	0.0274
reversalDocNoTrueFalse_F	0.0162
postingCloseToFiscalYearEnd_T	0.0135
credit_accountno_list_177c45aedc|830747d392|9753091b94	0.0069
userid_9a0f55d0b547	0.0068
CredDebNumberList_f6ac3520c905	0.0051
userid_b312cc8653ff	0.0047
CredDebNumberList_b1bd739ac7f0	0.0032
debitornameList_a7b8ac8b91bc79ed83cc4baed	0.0028
CredDebNumberList_8de76941ee6a	0.0025
postingCloseToFiscalYearEnd_F	0.0015
reversalDocNoTrueFalse_T	0.0015
debitornameList_ace2a37b29d4e0111a2ad4f45	0.0008

Implementation The only numerical feature *AmountLocalCurrency* has been converted back to its original state, because decision trees do not benefit from scaled data and the original feature range is much more interpretable. For our implementation we choose the decision tree classifier of the python scikit-learn library with default parameter settings. The max_depth is set to 10, since a higher depth not results in significantly more splits.

Results Table 1 shows the top twenty features ordered by their feature importance value. For reasons of clarity and comparability to the other two methods, the table shows the 20 most important features.

3.4 Logistic Regression

Way of Explanation Logistic Regression predicts the probabilities for classification problems with two possible outcomes. The following equation can be derived from the original logistic regression equation in order to explain predictions, where x_j is the value of the j-th feature of entry x and β_j is the trained weight of the j-th feature [19]:

$$\frac{odds_{x_j+1}}{odds} = exp(\beta_j(x_j+1) - \beta_j x_j) = exp(\beta_j) \tag{1}$$

The term *odds* defines the probability of an event divided by the probability of no event. For example, a logistic regression model predicts a probability of 0.8 that an instance is an outlier. Then the complement probability is 0.2. In this case, the odds would take the value 4, meaning that this instance is 4 times more likely to be an outlier than a normal instance. Thus, a change in a feature by one unit changes the corresponding odd ratio by a factor of $exp(\beta_j)$.

Implementation Our implementation is done in python using the logistic regression classifier of the scikit-learn library with default values for all parameters. The classifier is trained on the preprocessed and labeled dataset. Finally, with the trained model and its weights, the odd ratio value for each feature is computed, cf. Table 2.

Results Table 2 shows the top twenty odd ratio values rounded to two decimal places with their corresponding feature.

Table 2. Logistic Regression - Odd Ratio Values

Feature	Odd Ratio Value
AmountLocalCurrency	46.75
OneDateOutsideAccountYear_T	19.86
postingkey_36b8	12.33
DebitCreditIndicator_D	12.33
reversalDocNoTrueFalse_T	10.97
DatesEqual_T	10.30
postingCloseToFiscalYearEnd_T	9.16
taxcode_2b29	4.72
tcode_4c7d	4.26
debitornameList_41b32f3b3f784a4d631f05114	3.19
CredDebNumberList_40741ca51bfa	3.19
userid_b312cc8653ff	2.93
userid_c90a918b859b	2.37
userid_9a0f55d0b547	2.29
credit_accountno_list_830747d392	2.20
tcode_4db7	2.17
tcode_4cc8	2.12
debit_accountno_list_1404a3eacc|8700084883	2.02
credit_accountno_list_830747d392|9753091b94	1.97
debitornameList_3d0d906868f4e76770f1683a6	1.80

3.5 SHAP (SHapley Additive exPlanations)

Way of Explanation SHAP can be considered as an add-on to Shapley Values, a method from coalitional game theory created by Llyod Shapley in 1953 [20]. Shapley Values is a method for assigning shares to players depending on their contribution to the outcome of a game. The game is the prediction task (e.g., classification) applied to each instance of the dataset. The players are the feature values of the instance. The outcome of the game is the prediction for the instance.

In theory, a model needs to be trained for any possible combination of features, leading to 2^n models, where n is the number of features. In practice only one model is trained with all features. The different combinations are realized by randomly sampling values for features that are not present in a combination from the corresponding value range. Afterwards all the marginal contributions of a specific feature (e.g., *amount* of the journal entry) are computed. It can be interpreted as the gap between the prediction for a combination which contains the feature *amount* and the prediction for the same combination without the feature *amount*. All possible marginal contributions are then aggregated through a weighted average formula to obtain the Shapley Value.

At its core, SHAP is based on Shapley Values, but it contains novel approaches, such as the kernel-based KernelExplainer for model-agnostic methods or model-specific ones such as the TreeExplainer for decision trees or DeepExplainer for deep learning models. Moreover, SHAP provides many global interpretation methods based on aggregations of Shapley values, making SHAP both a local and a global XAI method [14].

Implementation Our implementation is done in python using the tensorflow.keras API. The neural network being used consists of 5 dense layers with [32 (input layer), 16, 8, 4, 1 (output layer)]. Moreover, a dropout layer is added between each dense layer, except for the last two layers. As an activation function, we use *relu* for all dense layers except the output layer, where a *sigmoid* function is typically used for classification problems. In addition, a kernel regularizer was included in each dense layer. As the optimizer *adam* was chosen, as the loss function *binary_crossentropy*.

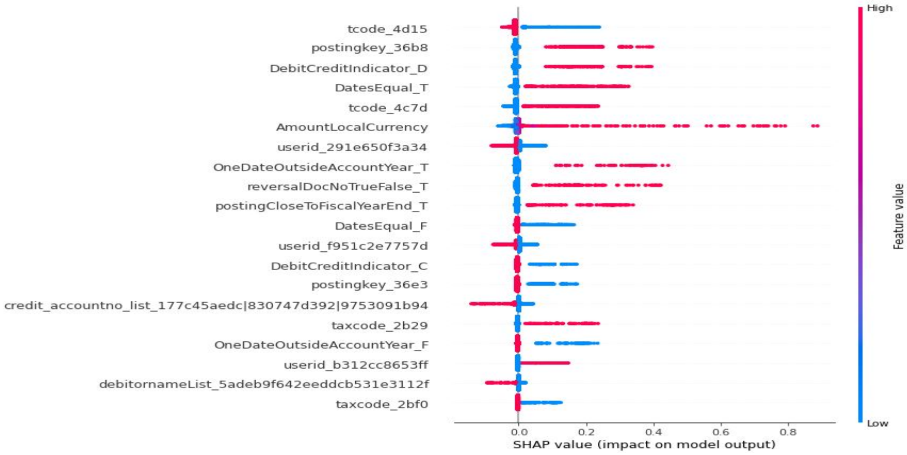
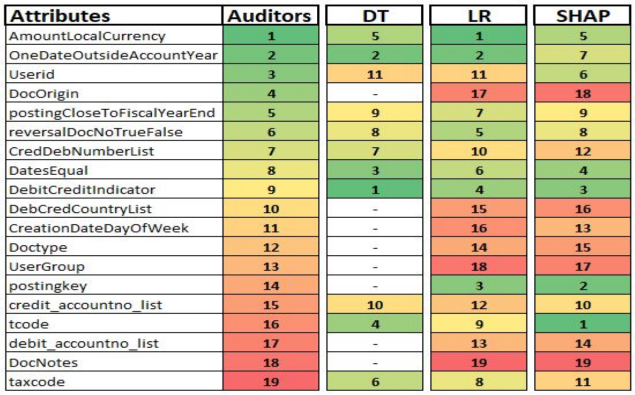


Figure 2. Neural network trained with autoencoder labels - SHAP Values with positive or negative impact on model prediction

Results To get an overview of which features are most important for a model we can plot the SHAP values of every feature for every sample. Figure 2 displays the features sorted by the sum of SHAP value magnitudes over all samples and uses SHAP values to show the distribution of the impacts each feature has on the model output. This type of presentation shows not only the general influence of a feature, but also the influence of specific values of this feature on the prediction. The color represents the feature value (red high, blue low). Due to the one-hot-encoding, all feature values except the amount are zero (blue) or one (red).

3.6 Discussion

For comparing the results of the three applied XAI methods, ranked lists of all journal entry attributes are calculated as depicted in figure 3. For reasons of clarity, we removed five attributes that had only one attribute value and merged the attribute *debitorenameList* into the attribute *CredDebNumerList*, which has the same informative value (the dataset only contain debitors), resulting in a total of 19 attributes to be ranked by the auditors. It should be mentioned that the explanations of the XAI methods are at instance-level because the associated models were trained with one-hot-encoded data. To overcome this mismatch in the comparison, the results of the XAI methods were mapped to attribute-level by determining the first occurrence of an attribute's instance as the relevance of the corresponding attribute. Regarding the decision tree, the relevance could not be assigned for some attributes as they are not used for a split, even at any depth. A hyphen bullet (-) indicates these attributes. The three XAI methods result in slightly different attribute rankings. For most of the attributes, the rank of all three methods is quite similar, which is indicated by lines of similar color (hyphen bullets can be interpreted as red). Nevertheless, each method considers a different attribute as the most important one. Especially for the attribute *tcode* the methods differ significantly from each other. The attribute *postingkey* is also remarkable, as this is ranked very highly by LR and SHAP, but is considered as irrelevant by DT. The observed results indicate that the combined use of several XAI methods is meaningful for explaining black-box models. To verify the attribute relevance and for the proof of the trustworthiness of machine learning methods in general, the rankings are evaluated by domain experts.



Attributes	Auditors	DT	LR	SHAP
AmountLocalCurrency	1	5	1	5
OneDateOutsideAccountYear	2	2	2	7
Userid	3	11	11	6
DocOrigin	4	-	17	18
postingCloseToFiscalYearEnd	5	9	7	9
reversalDocNoTrueFalse	6	8	5	8
CredDebNumberList	7	7	10	12
DatesEqual	8	3	6	4
DebitCreditIndicator	9	1	4	3
DebCredCountryList	10	-	15	16
CreationDateDayOfWeek	11	-	16	13
Doctype	12	-	14	15
UserGroup	13	-	18	17
postingkey	14	-	3	2
credit_accountno_list	15	10	12	10
tcode	16	4	9	1
debit_accountno_list	17	-	13	14
DocNotes	18	-	19	19
taxcode	19	6	8	11

Figure 3. Comparison of the results of auditors and the three XAI methods (DT = Decision Tree, LR = Logistic Regression) sorted by auditors relevance rating. The color indicates the respective relevance from high (1, green) to low (19, red).

4 Evaluation

The purpose of this evaluation is to find out to what extent auditors perceive machine learning models along with results of XAI methods as useful and comprehensible for audit purposes. The evaluation is conducted with nine auditors. Two of them have a work experience of over 7 years, another two between 3-7 years, and the remaining between 1-3 years. The evaluation was carried out using Microsoft Forms in a two-step process and is based on the results of the domestic-revenue account (see section 3.2).

First, the auditors are asked to rate the journal entry attributes on a Likert scale (1-7) in terms of relevance, where 1 means irrelevant and 7 means of utmost importance. The results served as a basis for the subsequent online meeting. In this meeting, the auditors are asked to rank the top 1-10 and top 11-19 attributes to get a more distinct ranking than with the average value per attribute from the first step. The previous ranking with the Likert scale was necessary to find out which attributes are among the top 10 (regardless of their order). In figure 3 the relevance ranking of the auditors is compared to the results of the three XAI methods.

Second, the auditors are asked to rate statements on a Likert scale (1-7), where 1 means does not apply at all true, 4 means neutral, and 7 means fully applies. In a previous work [6], the suspicious journal entries detected by the autoencoder model were reconciled with the auditors. The following statements are proposed:

1. The relevance rating of the attributes by the machine learning methods matches my experience/expectation.
2. The relevance rating of the attributes by the machine learning methods gives me new insights/perspectives for the detection of suspicious journal entries.
3. The machine learning methods would better detect suspicious journal entries if I could provide my domain knowledge as an input.
4. The use of machine learning methods (black-box) can improve the quality of certain audit tasks.
5. The use of machine learning methods (black-box) allows me to complete particular audit task more efficiently.
6. The use of machine learning methods (black-box) in auditing is useful.
7. For the application of machine learning methods (black box) in auditing, accompanying explanations of their results are necessary (e.g., relevance ratings for attributes).
8. I would actually use insights gained through instance-level relevance ratings (e.g., review journal entries, that were generated by a particular transaction code or user).
9. Relevance ratings at instance-level (e.g., a specific transaction code or a specific user) is more useful to me in the context of auditing than attribute-level ratings.

The statements can be divided into three clusters. Cluster A (evaluation of results) includes statements 1-3, cluster B (general usefulness of ML methods) includes statements 4-7 and cluster C (relevance ratings at instance-level) includes statements 8 and 9.

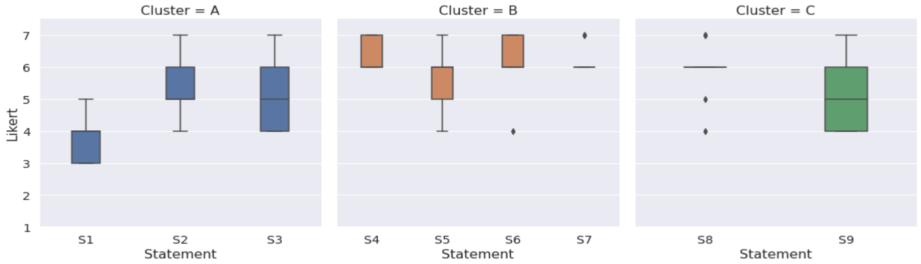


Figure 4. Boxplot - evaluation of the results

The evaluation depicted in figure 4 reveals that the auditors consider the use of machine learning to be useful, but the results of the XAI methods do not fully meet their expectations (average value 3). In this regard, the auditors state that they would like to prioritize particular attributes on their own (human-in-the-loop approach). At the same time, the auditors state that they gain new insights from the XAI results. For example, the high relevance of attributes such as *postingKey* and *tcode* is mentioned. They explain the large discrepancy to their relevance ranking by the fact that these are all technical or system-specific attributes where it is difficult for humans to recognize a pattern. Moreover, they emphasize that the explainability of the decision of ML models is very important for their application in practice. With regard to explanations at instance-level, the auditors noticed that not always rare values, but also the most frequent values were marked as especially suspicious, which they would not have expected.

5 Conclusion and future work

The increasing amount of electronically available account data poses a major challenge for auditors. Recently, it has become apparent that the use of machine learning is promising to tackle this challenge. In particular, so-called black-box models, such as an autoencoder, achieve very good results. However, for these models to be used in practice, their decisions must be explainable. This is the research area of XAI, but their methods are difficult to apply to autoencoders.

This paper presented an approach for explaining an autoencoder model, which consist of labeling the data first using the autoencoder and then training models suitable for the application of XAI methods using these labels. In our research we used three XAI methods for the explanation of the prediction results of an autoencoder. The autoencoder was developed in our previous work [6] for anomaly detection of the journal entries. The two inherently interpretable models (decision tree and logistic regression) are applied directly on the labeled values without any further preprocessing. They represent the simplest and intuitive explanation methods, but the simplicity of those methods doesn't allow to explain the direct behavior of more complex models like the autoencoder. For this purpose, we apply the SHAP method. Due to the specific learning functionality, SHAP cannot be applied directly to the autoencoder. Therefore, we use a surrogate neural network as a more suitable representation of the autoencoder learning functionality. Afterwards, an evaluation was conducted with auditors using the results of these methods, showing that XAI is able to capture domain knowledge, to gain new insights and is necessary to get machine learning into practice in this domain.

For future work, it is advisable to conduct this evaluation on a larger scale, which means more participants, XAI methods and accounts. Another interesting area of research would be the granularity of the results of the XAI methods. It is common practice to convert the data into a one-hot-encoded representation for training machine learning methods, but explanations at instance-level can quickly become complex and require that the users are familiar with possible values of all attributes. Whereas explanations at attribute-level are more intuitive, therefore it would be important to find an effective way to overcome this mismatch. The evaluation also showed that although it is important for the auditors to understand the results of machine learning methods, they would still like to integrate their own knowledge into a model. For this reason, the integration of a human-in-the-loop approach, e.g., by providing an interactive interface between model and auditor, would be a promising future research direction for the audit domain.

References

1. International Federation of Accountants (IFAC), "International Standard on Auditing ISA 240. The Auditor’s Responsibility to Consider Fraud in an Audit of Financial Statements" (2018)
2. Braun, R.L., Davis, H.E.: Computer-assisted audit tools and techniques: analysis and perspectives. Managerial Auditing Journal 18(9), 725–731 (2003)
3. Issa, H., Sun, T., Vasarhelyi, M.A.: Research Ideas for Artificial Intelligence in Auditing: The Formalization of Audit and Workforce Supplementation. Journal of Emerging Technologies in Accounting 13(2), 1–20 (2017)
4. Kokina, J., Davenport, T.: The Emergence of Artificial Intelligence: How Automation is Changing Auditing. Journal of Emerging Technologies in Accounting 14 (2017)
5. Schreyer, M., Sattarov, T., Borth, D., Dengel, A.R., Reimer, B.: Detection of anomalies in large scale accounting data using deep autoencoder networks. ArXiv abs/1709.05254 (2017)
6. Schultz, M., Tropmann-Frick, M.: Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits. In: HICSS (2020)
7. Ting Sun, Vasarhelyi, M.A.: Deep Learning and the Future of Auditing: How an Evolving Technology Could Transform Analysis and Improve Judgment. CPA Journal 87(6), 24–29 (2017)
8. Bay, S., Kumaraswamy, K., Anderle, M., Kumar, R., Steier, D.: Large Scale Detection of Irregularities in Accounting Data. In: Sixth International Conference on Data Mining (ICDM’06). pp. 75–86. IEEE, Hong Kong, China (2006)
9. McGlohon, M., Bay, S., Anderle, M.G., Steier, D.M., Faloutsos, C.: SNARE: a link analytic system for graph labeling and risk detection. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’09. p. 1265. ACM Press, Paris, France (2009)
10. Gunning, D., Aha, D.: DARPA´s Explainable Artificial Intelligence (XAI) Program. AI Magazine 40(2), 44–58 (2019)
11. Bauer, K., Hinz, O., van der Aalst, W., Weinhardt, C.: Expl(AI)n It to Me - Explainable AI and Information Systems Research. Business & Information Systems Engineering (2021)
12. Miller, T.: Explanation in Artificial Intelligence: Insights from the Social Sciences. arXiv:1706.07269 [cs] (2018)
13. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115 (2020)
14. Lundberg, S., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs, stat] (2017)
15. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy 23(1), 18 (2020)
16. Samek, W., Wiegand, T., Müller, K.R.: Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. arXiv:1708.08296 [cs, stat] (2017)
17. Goodman, B., Flaxman, S.: European Union Regulations on Algorithmic Decision-Making and a Right to Explanation. AI Magazine 38(3), 50–57 (2017)
18. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access 6, 52138–52160 (2018)
19. Molnar, C.: Interpretable machine learning: a guide for making black box models explainable (2020), pp. 71-73
20. Shapley, L.S.: 17. a value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games, Volume II, pp. 307–318 (2016)