Accounting Journal Reconstruction with Variational Autoencoders and Long Short-term Memory Architecture

Mario Zupan[1], Svjetlana Letinic[1], and Verica Budimir[1]

Polytechnic in Pozega, Vukovarska 17, Croatia
mzupan@vup.hr

Abstract. Our tries to learn machines how to reconstruct journal entries with the aim of finding anomalies lead us to deep learning (DL) technologies. Nowadays Variational autoencoder and Long short- term memory architectures as well as other deep learning architectures solves wide range of problems, yet they are not enough implemented in a field of accounting information systems (AIS). Inside AIS, accounting data follows accounting logic and makes specific datasets constructed by different type of columns - categorical and continuous variables. Our aim is reconstruction of these variables. Development of the model capable for precise reconstruction is not an easy task. This paper describes our research for anomaly detection model architecture which will be capable to reconstruct dataset with categorical features mixed with continuous monetary value feature. We developed basic models trained on accounting journals from 2007 to 2018 and then tested in the fiscal year 2019. Still, lots of hyperparameters need to be checked if we want to improve accuracy. Deep learning research is an engineering task leaded by experience so there is no linearity in the model improvement. Consequently, this paper is our contribution to collection of experience in developing accurate, useful and intelligent accounting control system.

Keywords: general ledger · journal entry · bookkeeping · accounting · deep learning · variational autoencoder · long short-term memory · anomaly detection · accounting control system.

1 Introduction

As in every information system, human efforts, as well as interaction between modules, can cause errors. Anomalies in accounting books occur on a daily basis, and unintentional human errors, attempted fraud, and continuous legislative changes are some of the critical causes. Anomalies occur despite the fact that the most existing controls integrated into accounting modules of modern ERP systems are created in compliance with bookkeeping rules. Because small and medium enterprises (SMEs) do not have audit obligations regulated by law, manual tax inspections are the only mechanism of their accounting and tax control. In general, detection of errors, made intentionally or not, consumes a large portion of a bookkeeper's or tax inspector's time, and correction of errors is not an easy part of their job, particularly owing to the architecture and functioning of the accounting software modules. Namely, most of today's ERP systems have specialized documents (digital forms) for an every specific business event. An every digital form is connected with one or more journal entry schemes created by senior accountants. Junior accountants or non-accountant employees do not have to be familiar with journal schemes because they communicate only through forms. As long as the modern accounting modules inside ERP systems are functioning based on the described principle, a single error in the only one journal entry scheme can cause an incorrect accounting entry for the whole set of connected digital forms.

When statistic methodology had become the part of a financial audit process, life of employees involved in auditing became a lot easier. Now it is the time for improving and make more easier auditing, accounting and tax inspection processes by utilization of deep learning algorithms.

2 Related works

When the idea for this research began to form, we found a previous study dealing with the same challenge and methodology written by Schreyer, M et. al. [14], which processed two datasets extracted from an SAP ERP. The first dataset represents the *accounting document header* (e.g., document id, type, time, and currency), and the second contains *journal entry details* (e.g., general ledger account, debit, credit, and amount). Because the majority of attributes correspond to categorical variables, the authors preprocessed the journal entry attributes to obtain a one-hot encoded representation of each attribute. They obtained 401 encoded dimensions for dataset A and 576 encoded dimensions for dataset B. Each journal entry was labeled as either a synthetic global anomaly, synthetic local anomaly, or non-synthetic regular entry. Dataset A contains a total of 307,457 journal entry line items comprised of 6 categorical attributes. In total 95 (0.03%) synthetic anomalous journal entries have been injected into dataset. These entries encompass 55 (0.016%) global anomalies and 40 (0.015%) local anomalies. Dataset B contains a total of 172,990 journal entry line items comprised of 10 categorical attributes. In total 100 (0.06%) synthetic anomalous journal entries have been injected into the dataset. These entries encompass 50 (0.03%) global anomalies and 50 (0.03%) local anomalies. The described datasets became inputs in nine distinct autoencoder architectures. The training was conducted via standard back-propagation until convergence (max. 2,000 training epochs). Anomaly threshold $Beta = 0{:}01$ implying that a journal entry is labeled *anomalous* if one of its attributes was not reconstructed correctly or occurs very rarely. The best performing results are selected based on parameterizations that (1) result in a recall of 100% of the synthetic anomalies and correspond to (2) the highest area under the ROC curve (ROC-AUC). On the same accounting datasets, Schreyer, M et. al. also trained the Adversarial autoencoder to learn a semantic meaningful representation of journal entries recorded in real-world ERP system [16]. Schreyer, M. et. al. [15] also showed an adversarial attack against Computer-aided audit tools (CAAT) using deep neural networks. They first introduce a real-world thread model designed to camouflage accounting anomalies such as fraudulent journal entries. Second, they showed that adversarial autoencoder neural networks are capable of learning a human interpretable model of journal entries that disentangles the entries latent generative factors. They used AAE architecture which extends the concepts of Autoencoder Neural Networks.

Shultz, M. et. al. [17] use three-layer autoencoder, *relu* as an activation function in a step when the encoder *e(x)* transforms the input x to a hidden representation h. The encoder (x) transforms the input x to a hidden representation h. In this step they use *leaky-relu* as an activation function. From the review of related work subsection, it can be concluded that the application of deep learning techniques in auditing is a promising research field with several open questions to be addressed. However, in the current audit practice, the full potential of such techniques is not yet realized. Mainly, less complex techniques like static rules are applied that check only a few journal entry attributes at a time.

Another study that inspired our research [3] described the use of a decision tree classification algorithm for financial journal entry fraud. This web-based application labels journal entries as either fraudulent or non-fraudulent. Classification algorithms are also tested through the Sherlock system development for identification of accounting irregularities within unlabeled accounting data extracted from general ledger [4]. Authors used Positive Nave Bayes (PNB) algorithm but they experimented with many different classification algorithms.

Anomaly detection accounting systems are developed by applying different technologies as well as inputs. Instead of accounting data, the authors of the study [10] experimented with two data mining algorithms applied on SAP R/3 security audit log data. Their results depends on different transaction threshold values.

Other available papers considered did not exploit a deep learning technology in the accountancy when solving a specific real-world problem. Still they were extremely helpful to us because they contained general thoughts regarding the application of artificial intelligence in the field of accounting [9], [5], [13]. The authors described accounting and auditing problems (bookkeeping routines, fraud detection, revenue prediction, an analysis of unstructured data, financial reporting, etc.) that might be potentially solved using machine learning technology, and described the strengths and limits of machine learning. As a common theme for related studies, *machine learning has highlighted promising results but still does not outperform the existing implementation, which is simple and deterministic* [2].

3 Methodology

Inspired by Schreyer, M. et. al. [14], [15], [16] and other related works, we conducted research into deep learning capabilities in our previous paper [18]. We trained and tested a variational autoencoder (VAE) model on a 3.731 row x 57 column dataset using journal entries for four fiscal years, namely, 2014 to 2017. The dataset was divided into a training part and a testing part at a 1:9 ratio, with a 374 x 57 testing shape and 3.359 x 57 training shape. The aim of the research was the reconstruction of all journal entries of the test dataset. The journals were audited and approved by accountants and a tax department. Our model incorrectly reconstructed 4 out of 374 journal entries of the test dataset. The entire test dataset has 183 unique rows; however, they are repeated through the years except for four years marked by the model. The model cannot reconstruct the journal entries if it sees an entry for the first time in a test part of the dataset. Consequently, the precision of the model reconstruction was 99.9893%.

Keras deep learning library has provided a significant contribution to our understanding of neural networks and artificial intelligence(AI). This high-level neural network application programming interface (API) is capable of running Google's TensorFlow, Microsoft's Cognitive Toolkit (CNTK), or the Theano deep learning library. Deep learning libraries can be used for supervised learning, in which a network is trained on labeled datasets. A supervised model can be well optimized but may be useless at the same time when new data are applied. However, neural networks can also be used for semi-supervised learning, through autoencoders and with unlabeled datasets. Semi-supervised learning is more suitable for anomaly detection problems and the nature of our particular data, namely, unlabeled and without errors or fraud entries, according to an audit by an experienced accountant.

3.1 Variational autoencoder (VAE) architecture

Semi-supervised learning can be realized through an autoencoder architecture. Autoencoders reconstruct an input given the same input, and are usually trained and tested on separate data. If they are trained and tested on the same dataset, the generalization of the model is sacrificed.

Based on the friendliness of Keras and the power of TensorFlow, a neural network model was built as a result of this research. More precisely, instead of a *classical* autoencoder, the model is a *variational* autoencoder (VAE), which was simultaneously discovered by Kingma and Welling in December 2013, and Rezende, Mohamed, and Wierstra in January 2014 [7]. According to Kingma and Welling [11], in variational auto-encoders, neural networks are used as probabilistic encoders and decoders. There are many possible choices of encoders and decoders, depending on the type of data and model. In their example they used relatively simple neural networks, multi-layered perceptrons (MLPs). For the encoder they used a MLP with Gaussian output, while for the decoder they used MLPs with either Gaussian, in case of real-valued data, or Bernoulli outputs (in case of binary data). Kingma and Welling [11] trained generative models of images from the MNIST and Frey Face datasets and compared learning algorithms in terms of the variational lower bound, and the estimated marginal likelihood. Rezende, Mohamed, and Wierstra demonstrated the ability of the model to impute missing data on MNIST, Frey Face and SVHN images datasets. They have also projected the MNIST dataset to a 2-dimensional latent space and use 2D embedding as a visualisation of the data.

In contrast to a *classical* autoencoder (AE), a VAE is capable of sampling from a latent space to create an entirely new output, which is possible because it turns the input into the parameters of a statistical distribution, namely, the mean and variance, instead of compressing its input into a fixed code in the latent space, which is how a AE operates. Our decision to use VAE was influenced by An, J. and Cho, S. [1]. They explained VAE differences from an AE based anomaly detection. First, probabilistic encoder extends the expressive power of the VAE compared to the AE in that even though normal data and anomaly data might share the same mean value, the variability can differ. Second, VAE has selective sensitivity to reconstruction according to variable variance. This is also a feature that the AE lacks in due to its deterministic nature. Third, autoencoder based anomaly detection uses reconstruction errors as anomaly scores, which are difficult to calculate if the input variables are heterogeneous. Also a 1% probability is always a 1% for any data. Thus deciding the threshold of the reconstruction error is much more objective, reasonable, and easy to understand than that of the reconstruction error.

So the main advantage of the variational autoencoder is its probabilistic manner for describing an observation in latent space. VAEs can be used to develop latent spaces of sound, music, or even text; however, in practice, the most interesting results have been obtained with pictures. Accordingly, a VAE is potentially capable of generating new journal entries from a latent space as well as reconstructing existing entries.

The VAE model (figure 2a). was trained through 1,000 epochs (figure 1a.), and our algorithm saved the best model according to the binary_cross entropy loss function. We tried to set the number of epochs as high as it is possible with the available hardware. The *model checkpoint* technique allowed us to set high number of epochs but only the best model is saved according to the binary_cross entropy metrics. Binary cross entropy measures how far away from the true value (y, which is either 0 or 1) the prediction $\hat{y}$ is for each of the features and then averages these class-wise errors to obtain the final loss according to equation 1.

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{1}$$

We used *relu* activation in all layers except the last one in decoder where we used *sigmoid* activation. Our model is compiled with *rmsprop* optimizer. According to Chollet ([7]), cross-entropy is a quantity from the field of information theory that measures the distance between the probability distributions, or in our case, between the ground-truth distribution and our reconstructed distribution.

3.2 Long short-term memory (LSTM) architecture

In addition to an autoencoder, this research exploited the capabilities of a *long short-term memory* (LSTM) architecture. LSTM is applied to a deep learning model with LSTM layers. According to Chollet [7], LSTM is a layer that saves information across numerous time steps. Whereas autoencoders are used in credit card fraud detection models, LSTM is used for price stock prediction, temperature-forecasting problems, and other time-series predictions. Owing to the fact that our VAE model inputs did not cover the monetary amount applied in our previous study[18], and was inspired by Arvaniti, V. [2], which used unsupervised data mining algorithms, *unusual financial transactions in this thesis are defined as account pairs inside journal entries containing an unusual amount of money compared to their frequent behavior*, and we expanded our research in this study by adding an LSTM model and a monetary value column to our prototype web application. The monetary values were normalized using a MinMaxScaler (default range of 0 to 1).

The LSTM model was trained through 500 epochs (figure 1b.), across 3 time steps. In a wide pallet of loss functions packed in Keras, we chose the mean_squared_error metrics. According to equation 2, the best LSTM model was saved and used to predict the amounts in the test datasets.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i)^2 \tag{2}$$

Suppose we want to predict the fourth value in this sequence:

$$\left[10\ 20\ 30\ 40\ 50\ 60\ 70\ 80\ 90\right]$$

Then, x_test and y_test appear as follows:

$$\left[10\ 20\ 30\right]\left[40\right];\left[20\ 30\ 40\right]\left[50\right];\left[30\ 40\ 50\right]\left[60\right];$$

$$\left[40\ 50\ 60\right]\left[70\right];\left[50\ 60\ 70\right]\left[80\right];\left[60\ 70\ 80\right]\left[90\right]$$

Suppose we have MSE = 0.02. According to equation 2, the error of the LSTM model is $\sqrt{0.02}$=0.14. Looking at our toy example target values ranging from 40 to 90, we can say that, on average, the error of the model was 0.14. This means that, when predicting a value of 40, we guessed 39.86, and when predicting a value of 50, we may have guessed 50.14, and when predicting a value of 60, we may have estimated 59.86, and so on. However, these are average values. Each individual error could have been negative or positive, making their sum zero but not their squared sum. Thus, this is simply the average of the squared difference between the predicted and actual data points.

3.3 Chart of accounts

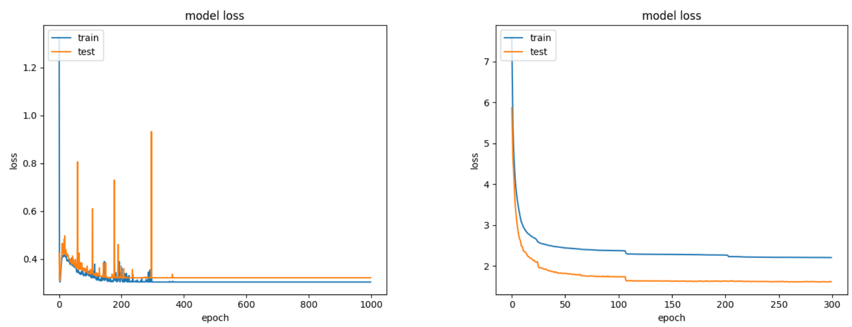
Transactions are classified through a pre-defined chart of accounts based on a prescribed structure of the balance sheet, the income statement, and off-balance sheet items. Accounts are divided into ten classes according to the specific needs of the entrepreneur. Most enterprises use a chart of accounts from some of the most popular Croatian accounting magazine publishers and adjusted to their particular needs because the classes are not provided by law. Because the number of account digits is also not regulated by law, but based on the needs of the entrepreneur, in our research, we decided to use the first three digits.

3.4 t-Distributed Stochastic Neighbor Embedding (t-SNE) technique

In the year 2008, Laurens van der Maaten and Geoffrey Hinton presented technique t-Distributed Stochastic Neighbor Embedding (t-SNE) that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map [12]. In opposite to Principal Components Analysis (PCA), t-SNE handles non-linear features. In the first step, t-SNE calculates density of all points in Gaussian distribution which represents probabilities. In the second step it calculates second set of probabilities but for the Student t-distribution with one degree of freedom. In the last step t-SNE measures the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL) and by using gradient descent algorithm tries to minimize KL cost function.

4 Results

Both models were compiled with *rmsprop* optimizer. Layers of the both models were used *relu* and *sigmoid* activation. The model VAE was trained and saved when the loss function reached the minimum (Figure 1.). During the training, we used 10% of the dataset for testing which is presented by the orange line in the figure. It is evident that the loss function curve has some spikes. High learning-rate, batch size (number of samples processed before the model is updated) and the number of epochs (the number of complete passes through the training dataset) are the hyperparameters which demand fine tuning. Loss spikes are often encountered when training with high learning rates, high order loss functions or small batch sizes, according to the authors [8] who developed Adaptive Learning Rate Clipping Stabilizes Learning (ALRC) to stabilize the training of artificial neural networks by limiting backpropagated losses. We also knew that the hyperparameters need to be tuned because the ideal settings for one dataset will not be the same across all datasets. When tuning the hyperparameters of an estimator, Scikit Learn offers the Grid Search and Random Search functions to simplify and automate the process. We have tried both functions and convinced that both are extremely costly in computing power and time, and still depend on chosen dataset. So, between grid search and manual search we choose the second one. When we increased the batch size from 50 to 100 with the *rmsprop* optimizer (default learning rate = 0.001) and with the same number of epochs the number of spikes decreased to only one. After we changed the optimizer to *adam* (learning rate=0.001) the spikes were completely disappeared. Although we knew how to avoid the spikes, we also new that they depend on dataset what we proved by experimenting with another datasets. Bergstra, J. & Bengio, Y. [6] discussed our presumption that different hyperparameters matter on different datasets which lead us from our aim of developing accounting anomaly detection model independent from data nature. Collected knowledge about hyperparameters will be used in future researches.



(a) VAE model

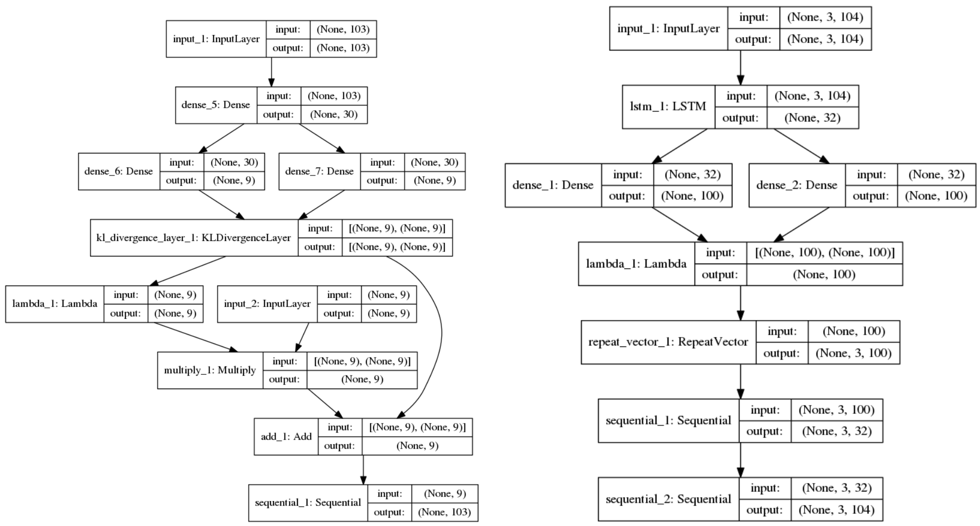
(b) LSTM-VAE model

Fig. 1: Models training: loss through the epochs

The figure 2a. shows types and connections between layers of the VAE model. First and last layer must have the same number of inputs (103) which is the same as the number of dataset columns. The model needs to learn 13,405 learnable parameters, such as weights and biases.

The models were trained on journals from 2007 to 2018 using 84,759 rows and tested on the fiscal year 2019 with 3,864 rows. The test dataset is 4.5% of the training size. Considering the document type, the account number from the chart of accounts and the debit/credit side of account, the VAE model incorrectly reconstructed 33 (table 1.) single journals entries and 7 journal entries pairs (table 2.) of 3.864 journal entries of the test dataset, that is 0.8% of the journal entries including 25 entries of the test dataset that have never happened in the trained dataset.

The LSTM-VAE model needed to learn significantly larger number of learnable parameters (162,824) with the monetary column as the extra feature in contrast to the VAE model inputs. That is 104 inputs (LSTM-VAE) in opposite to 103 inputs (VAE). As it can be seen in the Figure 2b., encoder of the model was included LSTM layer as well as the decoder, but decoder's LSTM



(a) VAE (b) LSTM-VAE

Fig. 2: Graphs of the models structure

Document id	Account	Debit/Credit	Repeated
FINOPENINGX	1000, 1022, 150, 242, 2710, 2744, 27441, 298, 917, 946, 2608, 261, 275, 2752, 6630, 9301, 93011, 9940, 999	Debit Credit Credit Debit Debit Debit Debit Debit	16
PAYROLBOOK11	46161	Debit	5
ACCOUNTING	150	Credit	4
ACCOUNTING	789	Credit	2
URAEU	298	Debit	2
CASHWITHDRAW	1020	Credit	1
URAEU	6690	Debit	1
URAPE2	419	Debit	1
URAPE2	416	Debit	1

Table 1: Journal anomalies found in test dataset: single journal rows

Document id	Account	Debit/Credit	Appearing times
MATINPUT	224 351	Credit Debit	3
STOCKBALANCE	6600 7101	[Credit] [Debit]	1
ACCOUNTING	93011 946	Credit Debit	1
CASHRECEIPT	1009 1020	Credit Debit	1
URAPE3	14062 221 24062 4102	Debit Credit Credit Debit	1

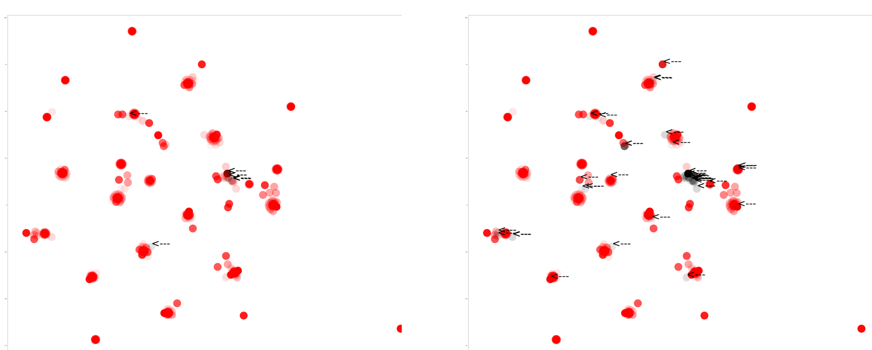
Table 2: Journal anomalies found in test dataset: journal entries pairs

layer is under Sequential layer. Although the model was trained on the same journals from 2007 to 2018 using 84,759 rows and tested on the same fiscal year 2019 with 3,864 rows, we do not have predictions for the first 4 rows due to the hyperparameter $timesteps = 3$. That gave us 3,860 predictions.

Both model needed to learn the document type, the journal account and the debit/credit side of the account. Additionally, LSTM-VAE model needed to reconstruct monetary values. We tried the model with different loss functions, number of layers and the best architecture had 138 (1.65% of the test dataset) reconstruction errors with monetary value included. It is important to mention that the LSTM-VAE model had difficulties with the very first 88 journal entries.

Despite the fact that the researched models had different inputs (monetary value included in the LSTM VAE model), decision has been led by the final aim of our research, that is development of the most accurate autoencoder. As a bonus we got the opportunity to analyze predicted versus real value.

Although the common characteristics of the journal entry clusters given by tSNE methodology and their relationship were not researched, we used tSNE to visualize dataset and the given results. tSNE reduced latent dimensions of the autoencoders to 2-dimensional space. Points in the figures 3a. and 3b. show journal entries of the tested dataset. Marked points represents entries that the VAE and the LSTM-VAE model reconstructed incorrectly. It can be seen that the both models have reconstruction errors in the same area of the 2-dimensional space. Further, the models did not have a problem to reconstruct the most obvious outliers though it was not capable to reconstruct outliers closed to the well defined clusters of the journal entries.



(a) VAE model

(b) LSTM-VAE model

Fig. 3: tSNE visualization of the models: Real journal entries with marked anomalies

5 Conclusion and future studies

With research described within this paper we are trying to collect experience in development of an artificial intelligence models by using real datasets generated in real world accounting books and with the final aim of developing intelligence module of accounting information system capable to assist to accountants, auditors and tax officers in finding anomalies. What we were considering for anomalies are errors, intentionally made or not. Precondition for intelligent error detection task is the model which is capable to learn the whole entries in a training process and then to reconstruct every single entry from the tested dataset except these that were seen for a very first time. To reached satisfied precision as well as to help us to understand the model functioning, we used the real-world dataset which is well known to us. It does not include any kind of errors or bookkeeping rules violation, but it includes entries that were happened in the test fiscal year for a very first time.

In this research we tested two models based on autoencoder architecture. The first model is variational encoder (VAE) model and the second one is variational encoder with Long short memory layers (LSTM-VAE) model. The inputs in these models were accounting journals for 12 fiscal years of the small-sized enterprise. Journals did not include any kind of anomalies or errors.

To reach satisfied accuracy of the model, we included document types and journal accounts as unavoidable drivers of business events. Except the most necessary variables, common for every accounting system, we added the monetary value column as the input into the second model (LSTM-VAE). We will also try to add more variables as an input to our models in our future studies.

In general, we can conclude that semi-supervised methods, autoencoders, are promising technology for developing anomaly detection modules inside AIS. Our research also showed potentials of the journal entry anomalies control system development with the help of t-SNE architecture. Visualization helped us to better understand the nature of our data as well as the models functioning. We saw that both models have reconstruction errors of the journal entries in the same area of the 2-dimensional space. Errors of the LSTM-VAE model are the errors of the VAE model at same time. Maybe world of accounting could have benefits from visualization techniques as a supporting technology, so we will use t-SNE for latent dimension visualization in our future studies.

In this paper we did not analyze predicted monetary values generated with our LSTM-VAE model, because we tried to research autoencoders' learning capabilities of the bookkeeping rules and they are not dependable on monetary values. That will be researched in our future studies too.

Generally, VAE models have one more characteristic still unresearched in a field of accounting. They have capability of generating new data from the latent space. It will be interesting to see results of our model generator of fictive entries in our future researches.

Both models we developed with respect of deep learning rules are promising. Still, accuracy in an accounting, especially in anomaly detection problems is requested condition, so the hyperparameters of the models need to be tuned according to the collected knowledge. Lots of experimenting with architecture and hyperparameters are waiting for us to get the prototype for the real-world module of an account control system.

References

1. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**(1) (2015)
2. Arvaniti, V.: Data mining journal entries: Discovering unusual financial transactions
3. Awodele, O., Akinjobi, J., Akinsola, J.: A framework for web based detection of journal entries frauds using data mining algorithm
4. Bay, S., Kumaraswamy, K., Anderle, M.G., Kumar, R., Steier, D.M.: Large scale detection of irregularities in accounting data. In: ICDM (2006)
5. Bengtsson, H., Jansson, J.: Using classification algorithms for smart suggestions in accounting systems. unpublished thesis (2015)
6. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research **13**(Feb), 281–305 (2012)
7. Chollet, F.: Deep learning with python, vol. 1. Greenwich, CT: Manning Publications CO (2017)
8. Ede, J.M., Beanland, R.: Adaptive learning rate clipping stabilizes learning. Machine Learning: Science and Technology (2020)
9. ICAEW: Artificial intelligence and the future of accountancy. Tech. rep., Information Technology Faculty Chartered Accountants, Moorgate Place London EC2R 6EA, UK (09 2018)
10. Khan, R., Corney, M., Clark, A., Mohay, G.: Transaction mining for fraud detection in erp systems. Industrial engineering and management systems **9**(2), 141–156 (2010)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
12. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
13. Omoteso, K.: The application of artificial intelligence in auditing: Looking back to the future. Expert Systems with Applications **39**(9), 8490–8495 (2012)
14. Schreyer, M., Sattarov, T., Borth, D., Dengel, A., Reimer, B.: Detection of anomalies in large scale accounting data using deep autoencoder networks. arXiv preprint arXiv:1709.05254 (2017)
15. Schreyer, M., Sattarov, T., Reimer, B., Borth, D.: Adversarial learning of deepfakes in accounting. arXiv preprint arXiv:1910.03810 (2019)
16. Schreyer, M., Sattarov, T., Schulze, C., Reimer, B., Borth, D.: Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks (08 2019)
17. Schultz, M., Tropmann-Frick, M.: Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits. In: Proceedings of the 53rd Hawaii International Conference on System Sciences (2020)
18. Župan, M., Letinić, S., Budimir, V.: Journal entries with deep learning model. International Journal of Advance Computational Engineering and Networking (IJACEN) **6**(10), 55–58 (2018)