Journal of
Risk and Financial Management

MDPI

Article

A Comparative Study of Traditional, Ensemble and Neural Network-Based Natural Language Processing Algorithms †

Achraf Chikhi [1], Seyed Sahand Mohammadi Ziabari [1,*] and Jan-Willem van Essen [2]

[1] Faculty of Science, Mathematics and Computer Science, University of Amsterdam, 1098 XH Amsterdam, The Netherlands; achraf.chikhi@student.uva.nl
[2] Department of IT Advisory, Baker Tilly, 1114 AA Amsterdam, The Netherlands; j.vanessen@bakertilly.nl
* Correspondence: s.s.mohammadiziabari@uva.nl
† Classification of Ledgers in Reference Classification System of Financial Information.

Abstract: Accurate data analysis is an important part of data-driven financial audits. Given the increased data availability and various systems from which audit files are generated, RCSFI provides a way for standardization on behalf of analysis. This research attempted to automate this hierarchical text classification task in order to save financial auditors time and avoid errors. Several studies have shown that ensemble-based models and neural-network-based natural language processing (NLP) techniques achieved encouraging results for classification problems in various domains. However, there has been limited empirical research comparing the performance of both of the aforementioned techniques in a hierarchical multi-class classification setting. Moreover, neural-network- based NLP techniques have commonly been applied to English datasets and not to Dutch financial datasets. Additionally, this research took the implementation of hierarchical approaches into account for the traditional and ensemble-based models and found that the performance did not increase when implementing the included hierarchical approaches. DistilBERT achieved the highest scores on level 1-2-3-4 and outperformed the traditional and ensemble-based models. The model obtained a F1 of 94.50% for level 1-2-3-4. DistilBERT also outperformed BERTje at level 1-2-3-4 despite BERTje being specifically pre-trained on Dutch datasets.

Keywords: audit; BERT; BERTje; DistilBERT; classification; financial; hierarchical; LightGBM; RCSFI; XGBoost

Citation: Chikhi, Achraf, Seyed Sahand Mohammadi Ziabari, and Jan-Willem van Essen. 2023. A Comparative Study of Traditional, Ensemble and Neural Network-Based Natural Language Processing Algorithms. *Journal of Risk and Financial Management* 16: 327. https://doi.org/10.3390/jrfm16070327

Academic Editor: Thanasis Stengos

Received: 22 June 2023
Revised: 1 July 2023
Accepted: 3 July 2023
Published: 11 July 2023

1. Introduction

The way financial audit services are performed has fundamentally changed. Audit firms are expected to use data analysis in their approach and leverage available datasets to ultimately provide financial audit stakeholders with greater assurance that these statements do not contain material misstatements (Tang and Karim 2017). In order to efficiently use the available data, standardization of the data from various accounting systems is needed. Several standards are available to serve this purpose, including the 'Reference Classification System of Financial Information' (RCSFI) (van Buuren and Wijma 2022). The standard was established through collaboration between accounting firms, software developers and Dutch government institutions. Working with fixed codes from RCSFI helps to improve financial analysis through comparable data. However, RCSFI is not supported in every accounting software and auditors have to manually map the client's way of accounting to RCSFI. This makes it cumbersome to take advantage of standardized data analysis.

RCSFI has a hierarchical class structure with five nested levels; the higher the level, the greater the number of classes. The problem addressed in this comparative study is, therefore, hierarchical multi-class text classification (HTC). This HTC problem poses particular challenges as RCSFI has a large number of closely related categories. Traditional classification algorithms are likely to perform well in problems with a small number of

well- separated classes. However, accurate classification over large sets of closely related classes is inherently difficult (Stein et al. 2019). Chen and Guestrin (2016) proposed eXtreme Gradient Boosting (XGBoost) as an effective tree-boosting ensemble machine learning method that is used to achieve state-of-the-art results for machine learning challenges, including classification. González-Carvajal and Garrido-Merchan (2020), on the other hand, mentioned Bidirectional Encoder Representations from Transformers (BERT) as a state-of-the-art deep learning model that is able to cope with tasks such as supervised text classification. Based on empirical tests, González-Carvajal and Garrido-Merchan (2020) claimed that BERT is superior to other approaches and stated that BERT should be used as a default technique in (NLP) problems such as classification. Furthermore, the authors also described with the hotel reviews experiment that BERT outperforms the classical NLP approach regardless of the language. However, these experiments were mainly conducted on binary classification tasks, so the question arises to what extent the performance will be when addressing a hierarchical multi-class classification problem with a Dutch financial dataset. Additionally, XGBoost was also not considered in the experiments conducted. Simultaneously, the research of (Stein et al. 2019) did not include BERT as a classifier. This indicates the existing lack of empirical comparison between ensemble-based techniques, such as XGBoost, and transformer based techniques, such as BERT.

In the literature, most of the proposed machine learning algorithms for classification have applied the flat approach. Zimek et al. (2008) experimented with classification using a hierarchical approach and concluded improved performance on the artificial dataset considered. However, the hierarchical approach did not provide a clear benefit for the real-world protein classification datasets. Based on their observations, the recommendation was made to use strong multi-class models as baseline methods when investigating the benefit of expert-defined hierarchies (Zimek et al. 2008). Stein et al. (2019), on the other hand, obtained results showing that classification models using a hierarchical strategy (LCPN) outperform the flat approach in all of the experiments carried out. Because of these different outcomes, this study also examined whether the adoption of hierarchical strategies contributes to the classification performance of the algorithms included on a Dutch financial dataset.

Because of the promising performance of ensemble-based and pre-trained-transformer-based techniques and the lack of comparison, this study attempts to compare these techniques in a hierarchical classification setting within the financial audit domain and with a Dutch financial dataset. By doing so, it also attempts to automate RCSFI classification to save financial auditors time and make it more convenient to perform analyses. To justify the need for these sophisticated models, traditional algorithms were also included in this comparative study by using the well-known Naive Bayes and Logistic Regression algorithms.

The main research question within this study was defined as: To what extent can neural- network-based natural language processing techniques outperform traditional and ensemble-based algorithms in classifying general ledgers into 'Reference Classification System of Financial Information' compliant categories?

This paper is organized as follows: Related work is presented in Section 2. The methodology is described in Section 3, followed by the results in Section 4. The discussion is presented in Section 5, and the conclusion is provided in Section 6.

2. Related Work

The classification of financial ledgers has not yet received sufficient attention within the available research. However, the emergence of ensemble algorithms and neural-network-based natural language processing techniques has made it possible to solve classification tasks in various domains such as healthcare (Abdurrahman and Sintawati 2020; Martínez-Castaño et al. 2021; Paleczek et al. 2021), e-commerce (Bilal and Almazroi 2022; Chen and Wan 2023), and finance (Arslan et al. 2021; Hajek et al. 2022; Lei et al. 2020). A brief summary of the research and techniques in the area of classification is provided in the following Sections.

2.1. Multi-Class Classification with Imbalanced Data

According to Lorena et al. one way to address multi-class machine learning problems is to change them to multiple binary classification tasks with decomposition strategies, such as One-against-all (OAA) and One-against-one (OAO) (Lorena et al. 2008). Based on their survey, they stated that OAA generally showed good accuracy rates, but also mentioned that an imbalance in the data could be a problem. Since the induction of each binary predictor only uses data from one class against that of all other classes, the negative effect of the imbalance can be emphasized, hindering the proper classification of examples from minority classes. In addition, given a problem with k classes, k binary classifiers have to be generated in the case of OAA. This could be problematic in this research, as the classification problem being addressed in this study has imbalanced data with many different classes, as further explained in the Methodology section. One-against-one (OAO), on the other hand, requires a different number of binary classifiers. Given k classes $k \times (k-1)/2$ binary classifiers are generated, which is still a large number of classifiers in the case of RCSFI.

To address the class imbalances, Chawla et al. (2002) proposed an approach in which the minority classes are oversampled by creating synthetic examples using k-nearest neighbors to define the neighborhood of samples. This neighborhood is used to generate the synthetic samples with a pre-processing technique called SMOTE-N (for nominal features). Their research showed that the combination of SMOTE and undersampling performs better than undersampling alone. The reason for this is that, with the replication of the minority samples, the decision region that leads to a classification decision becomes smaller and more specific. The synthetic oversampling method of (Chawla et al. 2002) causes the classifier to build larger decision regions by providing more related minority class samples to learn from. This leads to more coverage of the minority classes.

2.2. Ensemble Machine Learning

A state-of-the-art approach for multi-class classification problems includes using gradient boosting algorithms. Boosting is an ensemble machine learning technique that combines several lower-accuracy models in order to create a high-accuracy model (Rahman et al. 2020). Chen and Guestrin (2016) proposed eXtreme Gradient Boosting (XGBoost) as an effective tree-boosting machine learning method used to achieve state-of-the-art results for machine learning challenges, including classification.

Ali et al. (2023) used XGBoost to develop a Financial Statement Fraud (FSF) detection model and also addressed the problem of class imbalance with SMOTE. The model obtained an accuracy of 96.05% in the detection of FSF. Bentéjac et al. (2020) performed a comparative analysis by applying different gradient boosting algorithms, including CatBoost, Random Forest, LightGBM and XGBoost for multi-class classification on multiple datasets. The authors stated that CatBoost obtained the best results in generalization accuracy and area under the curve (AUC) in the studied datasets. However, the differences were small when compared to XGBoost, and the performance of CatBoost when training the final model was much slower than the other methods. LightGBM was the fastest of all methods, but it was not the most accurate. XGBoost placed second in both accuracy and training speed and has consistently placed among the top ten nominees in various Kaggle competitions. Bentéjac et al. (2020) presented the performance of the default and tuned models in terms of their average accuracy on different datasets. Table 1 shows the results of the tuned models included in their study. The accuracy of the tuned XGBoost models over the datasets ranges from 67.00% to 99.64%. The accuracy of the tuned Random Forest models ranges from 61.83% to 99.13% and the tuned LightGBM models have accuracies from 64.46% to 99.71%. There was no model that performed best on all datasets.

Table 1. Average accuracy of the tuned ensemble models on several datasets (XGBoost, Ran dom Forest, LightGBM (Goss) (Bentéjac et al. 2020)).

Dataset	T. XGB	T. RF	T. LGBM
Australian	87.53%	86.08%	86.81%
Banknote	99.64%	99.13%	99.71%
Breast	95.99%	96.85%	96.57%
Cleveland	83.16%	82.46%	84.50%
Dermatology	92.27%	97.30%	96.78%
Diabetes	76.56%	76.69%	75.79%
Echo	98.75%	97.32%	97.14%
Ecoli	89.05%	89.11%	86.14%
German	77.40%	75.80%	77.20%
Heart	84.07%	84.44%	84.07%
Hepatitis	67.00%	61.83%	64.46%
Ionosphere	92.59%	93.16%	93.21%
Iris	94.00%	92.67%	94.67%
Liver	68.11%	67.58%	69.24%
Magic04	88.63%	88.18%	88.42%
Newthyroid	95.80%	96.28%	94.83%
Parkinsons	90.14%	90.70%	92.64%
Phishing	91.13%	89.51%	90.98%
Segment	98.70%	98.18%	98.53%
Sonar	86.97%	85.59%	86.97%
Soybean	95.22%	94.65%	93.78%

2.3. Neural Network Based Natural Language Processing

González-Carvajal and Garrido-Merchan (2020) presented Bidirectional Encoder Representations from Transformers (BERT) as a state-of-the-art deep-learning model that is able to cope with NLP tasks, including supervised text classification. BERT is based on transfer learning in which a neural network model is pre-trained and fine-tuned. Pre-training is carried out on an unlabeled large corpus and, during fine-tuning, the parameters are updated using a labeled dataset for specific tasks. BERT differs from other transfer models in its ability to extract information from sentences from left to right and vice versa (Devlin et al. 2018). Based on empirical tests (González-Carvajal and Garrido-Merchan 2020) claimed that BERT is superior to traditional machine learning algorithms on average NLP problems and stated that BERT should be used as a default technique in NLP problems. One of the reasons stated is the independence of BERT regarding features of the NLP problem, such as language. However, the experiments were mainly conducted using binary classification tasks. The accuracy of BERT for the different experiments ranges from 83.61% for the RealOrNot experiment to 93.87% for the experiment on the IMDB dataset.

A concern when using pre-trained models, such as BERT, is that it could be challenging to run these models with limited computing power due to the large number of parameters (110 million in BERT Base). Therefore, Sanh et al. (2020) proposed DistilBERT, in which a 40% smaller, 60% faster general-purpose language model is pre-trained that retains 97% of the language-understanding capabilities by applying knowledge distillation. Knowledge distillation is a compression technique in which the compact model is trained to reproduce the performance of a larger model or an ensemble of models. Arslan et al. (2021) compared several pre-trained models in a financial multi-class text classification study and the results showed that DistilBERT outperformed BERT for some datasets.

de Vries et al. (2019) used the same BERT architecture and parameters and developed a monolingual Dutch BERT model named BERTje. Whereas the multilingual BERT includes Dutch but only based on Wikipedia text, BERTje is based on a diverse dataset of 2.4 billion tokens. According to (de Vries et al. 2019), BERTje outperforms the multilingual BERT model on the following NLP tasks: part-of-speech tagging, named-entity recognition, semantic role labeling, and sentiment analysis. Multilingual BERT obtained an accuracy of 89.1% on the Dutch Book Reviews Dataset with two classes, whereas BERTje obtained an

accuracy of 93.0%. Since the dataset in this paper is Dutch, it can be assumed that BERTje also outperforms BERT in this research. To ascertain this in the context of this research, the models mentioned in this section were all included.

2.4. Flat and Hierarchical Classification

In the literature, most of the proposed machine learning algorithms for classification follow the flat approach. With the flat approach, a single multi-class classifier is trained to only predict the leaf nodes and the previous levels of the hierarchy are ignored (Miranda et al. 2023). Zimek et al. (2008) experimented with classification using a hierarchical approach and concluded improved performance on the artificial data set considered. However, the hierarchical approach did not provide a clear benefit for the real-world protein classification datasets (Zimek et al. 2008). Based on their observations, the recommendation was made to use strong multi-class machine learning algorithms as baseline methods when investigating the benefit of expert-defined hierarchies. Stein et al. (2019), on the other hand, obtained results on hierarchical text classification, and showed that classification models using a hierarchical strategy (LCPN) outperformed the flat approach in all of the experiments conducted. Silla and Freitas (2011) wrote a survey on the task of hierarchical classification. Throughout this survey, the authors described the following hierarchical approaches used in HTC:

- Local Classifier Per Node Approach;

This approach consists of training one binary classifier for each node of the class hierarchy except the root node (Silla and Freitas 2011). This approach was used by (Stein et al. 2019).

- Local Classifier Per Parent Node Approach;

In this (top-down) approach, for each parent node in the class hierarchy, a multi-class classifier is trained to distinguish between its child nodes (Silla and Freitas 2011).

- Local Classifier Per Level Approach;

This classifier approach is the least used in the literature. The local classifier per level approach consists of training one multi-class classifier for each level of the class hierarchy. The major drawback of this class-prediction approach is being prone to class-membership inconsistency (Silla and Freitas 2011).

Additionally, Silla and Freitas (2011) also highlighted the use of hierarchical precision (hP), hierarchical recall (hR) and hierarchical f-measure (hF) as evaluation metrics of the hierarchical classifiers. These are extended versions of the well-known metrics of precision, recall and f-measure and are customized to the hierarchical classification setting.

3. Methodology

This research concerns supervised learning and a hierarchical multiclass text classification task was performed. The addressed task in this comparative study was "classifying general ledgers into RCSFI compliant categories". This task was completed by applying traditional algorithms (Naive Bayes, Logistic Regression) and state-of-the-art techniques in the form of gradient-boosting tree ensemble methods (Random Forest, LightGBM, XGBoost) and deep learning transfer methods (BERT, BERTje, DistilBERT). Naive Bayes was included in this research as a baseline because the Naive Bayes method is a well-known method for text classification due to its effective assumptions and explainability (Abbas et al. 2019). However, according to (Ng and Jordan 2001) as the number of training examples increases, Naive Bayes is expected to initially perform better; however, Logistic Regression is expected to catch up and overtake the performance of Naive Bayes. Nevertheless, this is not always the case and depends on the number of training examples. By including Naive Bayes and Logistic Regression, the potential added value in terms of classification performance of the more advanced models could be highlighted. A visual representation of the methodology applied in this study is shown in Figure 1. The components within this visualization are explained in the following Sections.

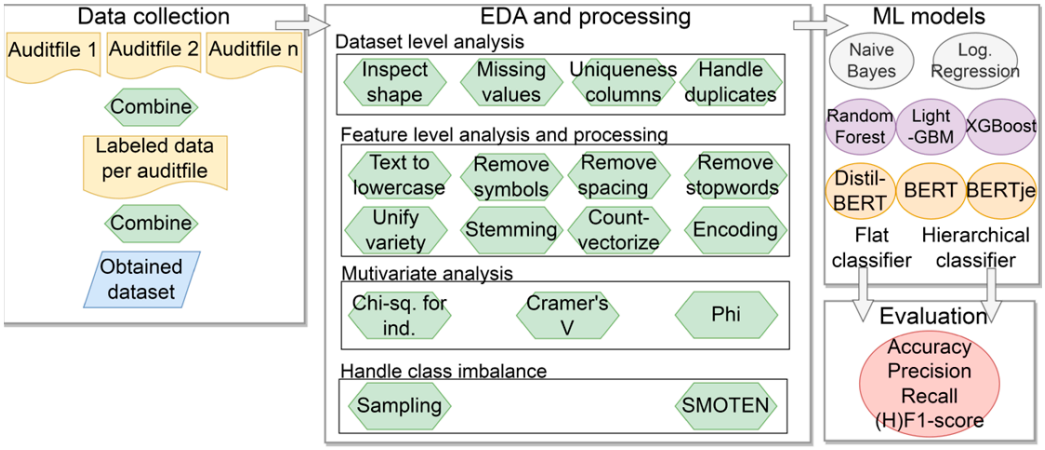


Figure 1. Overview methodology.

3.1. Data Collection

The first data source used in this research is twenty audit files in XAF format provided by the external research organization. Additionally, the corresponding files for each audit file with the labelled RCSFI categories on the ledger level are used as well. The general ledger is a record of payments and receipts within a given category used in summarizing financial data (Nurhayati and Muda 2022).

- XAF audit files;

An audit file contains data regarding ledger accounts, journals, transactions, VAT, customers and suppliers. These audit files can be generated from accounting software such as Exact, Unit4 and AFAS. The audit file consists of different elements with nested fields. An overview of these elements and fields is given in Appendix A. XAF is a form of XML and the structure of an audit file is illustrated in Figure 2.

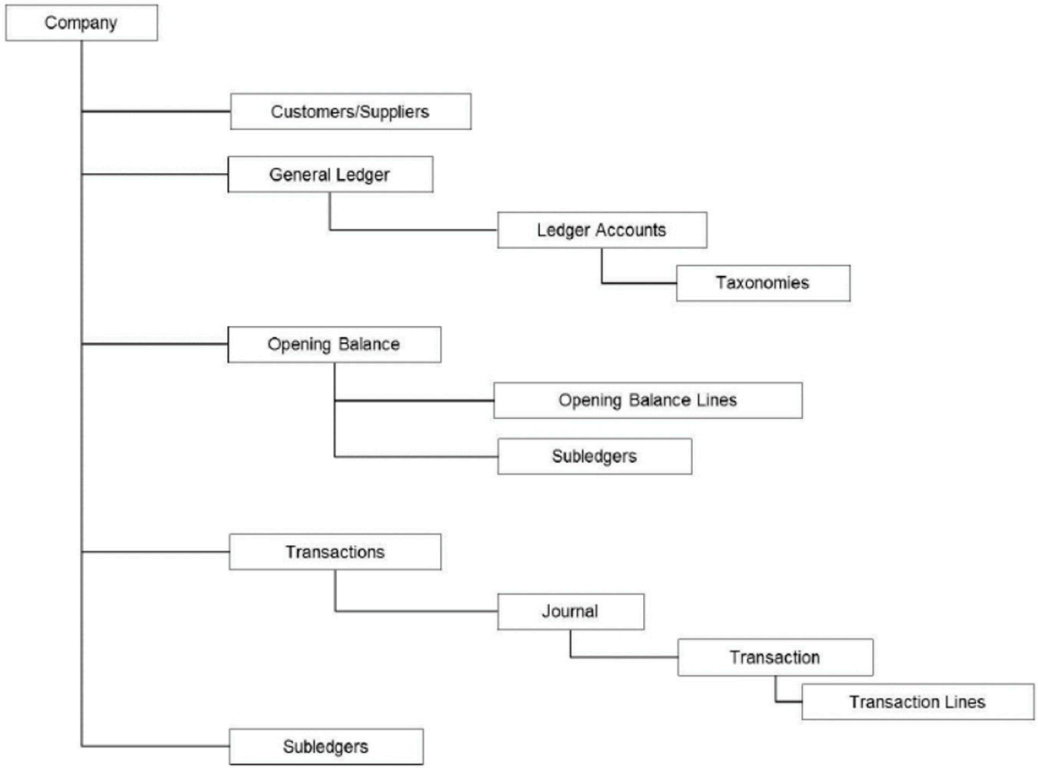


Figure 2. Structure of XAF audit files (adopted from Auditfiles.nl).

- RCSFI files;

RCSFI is hierarchically nested and has five different levels in its current version. Table 2 shows how RCSFI is structured, with an example of the RCSFI label and description in the last two columns (RCSFI). Predicting the first level is binary, as the general ledger could be classified as a profit and loss statement or a balance sheet item. The second level has 42 different classes, of which 14 correspond to balance sheet items and 28 to the profit and loss statement, and it becomes a multi-class classification problem. At the third level, there are 143 different classes related to balance sheet items and 147 to the profit and loss statement. In level 4, a distinction can be made between 613 different classes that deal with the balance sheet and 755 classes that fall under the profit and loss statement. In level 5, the deepest level, a distinction can be made between 2497 classes related to the balance sheet and 441 related to the profit and loss statement.

Table 2. Structure of RCSFI starting at level 1, where the distinction is made between a balance sheet item and profit and loss statement. RCSFI ends at level 5, in which details at the mutation level are specified (adopted from referentiegrootboekschema.nl/opbouw-rgs).

Level	Level Description	Reference Code	Description
1	Profit and loss account or balance sheet	B	Balans (Balance)
2	Main heading	BMva	Materiele vaste activa (Tangible fixed assets)
3	Headings	BMvaBeg	Bedrijfsgebouwen (Industrial buildings)
4	General ledger accounts	BMvaBegVvp	Verkrijgings- of vervaardigingsprijs bedrijfsgebouwen (Acquisition costs industrial buildings)
5	Mutations	BMvaBegVvpIna	Investeringen nieuw aangeschaft bedrijfsgebouwen (Investments of acquired industrial buildings)

3.2. Exploratory Data Analysis and Pre-Processing

The audit files were originally supplied in XAF format and were converted to xlsx format using a data analysis tool called Qlik Sense. Each audit file was then combined with the corresponding file that contained the supplied labels based on the ledger account number. Subsequently, by means of the supplied RCSFI label, a link was made to the file containing additional RCSFI information such as the code descriptions. Based on the length of the labeled RCSFI value, the corresponding hierarchical level is determined. If the length of the RCSFI code is greater than or equal to 1, then level 1 is determined. If the length is greater than or equal to 4, then level 2 is determined. Values greater than or equal to 7 make it possible to determine level 3 and so on, up to and including level 5 if the code has a length of 13 characters. Combining these datasets resulted in a dataset consisting of 915.414 rows and 57 columns. After the dataset was composed, Exploratory Data Analysis (EDA) was performed. The data provided was analyzed at the corpus level, feature level and multivariate level. Further data processing was also carried out during the EDA phase.

3.2.1. Corpus Level

At the corpus level, an overview of the entire dataset was outlined by determining the columns and the number of rows. A calculation was performed for each column to gain insight into the percentages of missing values. Based on these percentages, the conclusion

was drawn that a number of columns were not usable for further EDA and further research because these had more than 80% missing values. An additional obtained insight was that not all ledgers had a labeled target variable. These rows were not usable in the training phase of the models and were therefore removed as well. As shown in Figure 3, almost 70% of the labels are missing for level 5. This is because, in practice, the highest level is not widely used, nor is it always available as further specification for every level 4 class. For this reason, consultation with the external research organization led to the decision to proceed with levels 1 through 4 in this study. Furthermore, increased awareness regarding the presence of client-specific fields was obtained. These fields were unique for each client and did not provide useful input for the models. Some examples are 'company/companyIdent', 'company/taxRegIdent', 'phone', 'email', etc. It also became clear which columns had many unique values and which did not. For instance, 'header/curCode' had only one unique value, upon which it was decided to drop this column.

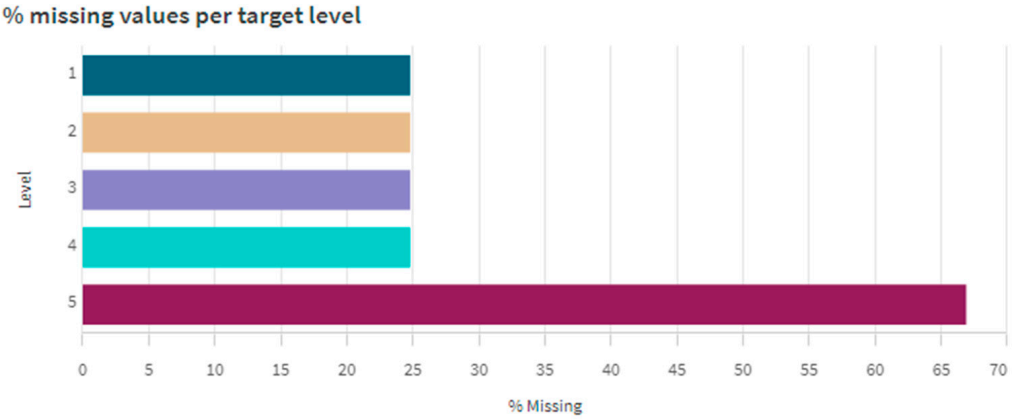


Figure 3. Missing values for the target levels.

3.2.2. Feature level

During EDA at the corpus level, it became clear that certain columns had a high number of unique values. For example, the TransactionLineDescription column had 68.488 unique values and the ledgeraccountdescription column had 1.822 unique values. A plot was made for the columns with the descriptions to see what the most frequent words were. This clarified that some of the words occurred multiple times including "Debiteuren" and "DEBITEUREN". In addition, some Dutch stopwords such as "te" and "en", became visible. Therefore, the following steps were applied to the description fields (Ledgeraccountdescription, Transactionlinedescription and Journaldescription):

- Convert text to lowercase
- Remove punctuation
- Remove abundant spacing
- Remove digits from descriptions
- Remove Dutch stopwords such as "de", "een", "is", etc.
- Remove parts with less than two characters
- Apply stemming to reduce words to their root form (Porter 2001). (BERT inherently uses a word-piece algorithm so the columns before stemming were kept for later usage)

Furthermore, standardization was applied to categorical fields that contained multiple values for the same category. For example, the "ledgeraccount type" column contained "Balans" and "B", with both indicating the same. The categorical fields were also encoded using either one-hot encoding when the number of categories was two or binary encoding in case of more categories. Binary encoding is a combination of hash encoding and one-hot encoding. The features are first converted to numerical values and are then transformed to binary numbers, which results in the need for fewer additional columns. An alternative for binary encoding could be ordinal encoding which adds one new column to the dataset.

However, this approach implies an order to the categorical values that does not exist (Potdar et al. 2017). An alternative for binary encoding could be ordinal encoding, which adds one new column to the dataset.

In order to deal with the description fields that still had a high number of unique values after the steps mentioned above, count vectorizing was applied. Count vectorizing provides a way to collect text, build the vocabulary of known distinctive words and to encode text using that vocabulary (Sajjad et al. 2022). The word frequency was determined for each description field to gain insight into the 200 most common words. Of these, the words that were not customer specific were then selected. Then, some of these words were standardized, as multiple terms with the same meaning were found that ultimately led to 165 additional features. As the included traditional and ensemble-based models only accept numerical features, one-hot encoding could have been applied. However, due to the high number of unique values, one-hot encoded features would lead to numerous columns, which negatively impact the computational performance of the model. For example, if the Ledgeraccountdescription column was transformed with one-hot encoding after the steps mentioned above, an additional 1.752 columns would be added to the dataframe.

3.2.3. Multivariate Level Analysis and Feature Selection

Overfitting and the curse of dimensionality are two problems that might have occurred during the development of the models. Therefore, feature selection was applied to help avoid these problems by reducing the number of input variables. This can lead to lower computational costs, better performance and an easier-to-interpret model. Regarding the multivariate analysis, a closer look was taken at the relationship between the features. As most of the features were categorical, the relationship was measured using the chi-squared test of independence. In order to attain additional insight into the strength of the relationship, the Cramer's V or Cramer's phi value was calculated depending on the number of columns and rows. The chi-squared test score was calculated as stated in Equation (1).

$$\tilde{x}^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{1}$$

where O stands for the observed frequency and E stands for the expected frequency (Zibran 2007). Based on the chi-squared test statistic, the Cramer's V or Cramer's Phi was determined. In terms of measuring the association of the two binary variables, the Cramer's Phi was calculated, otherwise the Cramer's V was computed. The Cramer's V was calculated as presented in Equation (2).

$$\sqrt{\frac{x^2/n}{\min(k-1,\ r-1)}} \tag{2}$$

where x^2 is the calculated Pearson's chi-square value, n is the number of observations, k is the number of columns and r is the number of rows. The Cramer's Phi, on the other hand, was computed as stated in Equation (3).

$$\sqrt{\frac{x^2}{n}} \tag{3}$$

A Cramer's V or Phi between 0 and 0.05 indicates no or a very weak relationship. A value between 0.05 and 0.10 represents a weak relationship. Values between 0.10 and 0.15 give an indication of a moderate relationship, whereas values between 0.15 and 0.25 indicate strong relationships. Very strong relationships are indicated with values bigger than 0.25 (Akoglu 2018). The results of the calculations for some features are presented in Table 3.

Table 3. Results of Cramer's V/Phi computations.

Feature 1	Feature 2	Cramer's V/Phi
amntTp	Level_1	0.169
custSupTp	Level_1	0.095
Ledgeraccountnr	Level_1	0.981
periodNumber	Level_1	0.062
AmountGrouped	amntTp	0.939
sourceID	jrnTp	0.774

Columns with a weak or moderate relationship with the target variable (level 1) were removed; i.e., periodNumber. Furthermore, a number of features with a high relationship between each other were also removed. However, for some of the features, the assumptions of the chi-squared test of independence were not met, which may have resulted in unreliable test results (McHugh 2013). These columns were kept to experiment with to determine whether the performance of the models increased.

3.2.4. Imbalanced Data

As shown in Figure 4, certain classes occurred more frequently than others. Without handling this problem, the models could be biased in favor of the most occurring classes. Because of this problem, a combination of undersampling the majority classes and oversampling the minority classes was applied using SMOTE-N, as proposed by (Chawla et al. 2002). The following settings were used for SMOTE-N:

- sampling_strategy = auto
- random_state = 0
- k_neighbors = 5
- n_jobs = 4

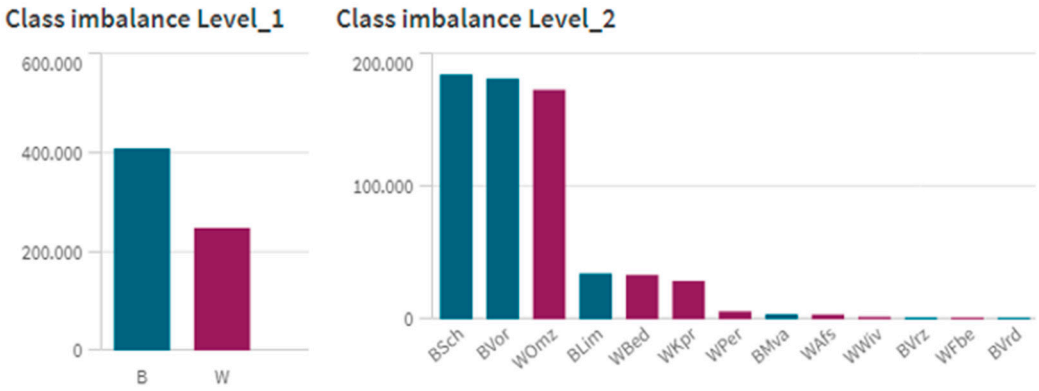


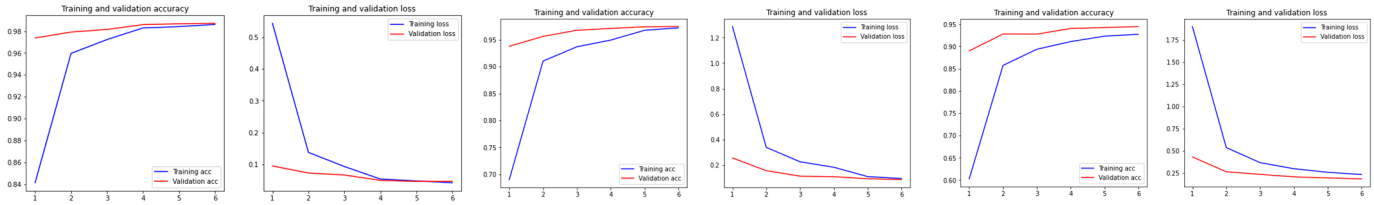
Figure 4. Imbalanced classes.

3.2.5. Hyperparameter Tuning

With the exception of the Naive Bayes model, hyperparameters were tweaked for better performance. Putatunda and Rama (2018) performed a comparative analysis on different techniques for the hyperparameter optimization of XGBoost. They compared Hyperopt, Random search and Grid search, and concluded that Bayesian optimization using Hyperopt achieved a higher mean Gini score, which indicates higher prediction accuracy. Therefore, Hyperopt was used during this research for hyperparameter tuning of the traditional and ensemble-based models following the study by (Bergstra et al. 2015) as a guideline. The same hyperparameters for the traditional and ensemble models are also used when the hierarchical approaches were implemented. The following settings were used for Hyperopt:

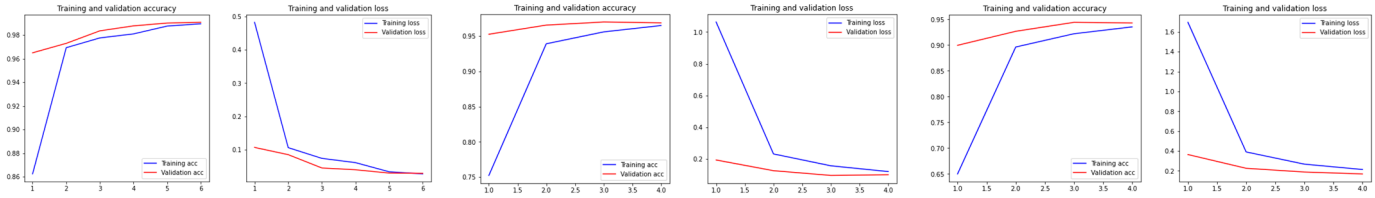
- Algorithm: tpe.suggest
- Number of evaluations: 25
- Scoring: f1-macro
- Number of cross-validations: 5

BERT, on the other hand, is a pre-trained model and requires less hyperparameter tuning. The authors of BERT prescribed the following procedure for hyperparameter fine-tuning: batch size (16 or 32), learning rate (5×10^{-5}, 3×10^{-5}, 2×10^{-5}) and number of epochs (2, 3 or 4) (Devlin et al. 2018). For these three models the Hugging Face Transformers framework was used to extract the pre-trained models in order to apply fine-tuning (Wolf et al. 2019). Figures 5–7 contain the graphs for BERT, BERTJE and DistilBERT, showing the validation loss and training loss per epoch. These demonstrate that the validation loss stagnates at epoch 4. After this point, the model starts to remember the data instead of learning from it, which leads to overfitting. The weights associated with the fourth epoch are used for the prediction of the RCSFI classes.



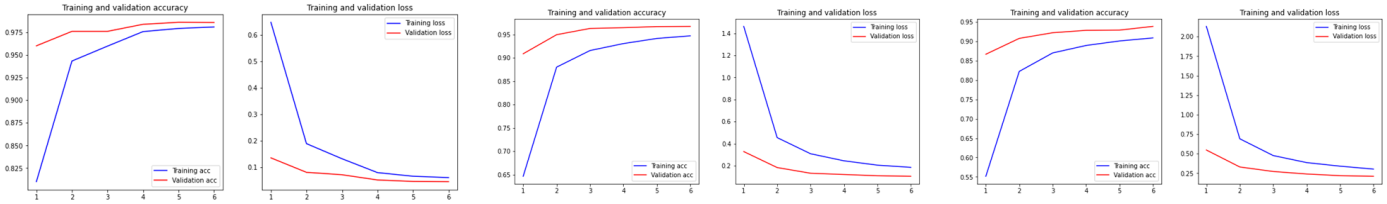
(**a**) BERT level 1-2 (**b**) BERT level 1-2-3 (c) BERT level 1-2-3-4

Figure 5. Training and validation accuracy and loss of BERT for the different RCSFI levels per epoch.



(**a**) DistilBERT level 1-2 (**b**) DistilBERT level 1-2-3 (**c**) DistilBERT level 1-2-3-4

Figure 6. Training and validation accuracy and loss of DistilBERT for the different RCSFI levels per epoch.



(**a**) BERTje level 1-2 (**b**) BERTje level 1-2-3 (**c**) BERTje level 1-2-3-4

Figure 7. Training and validation accuracy and loss of BERTje for the different RCSFI levels per epoch.

3.2.6. Evaluation

The classification performance of the different models is evaluated by the metrics found in the related literature. The hierarchical F1 for the hierarchical models is also presented as proposed by (Silla and Freitas 2011). Because of the readability of the results, the choice was made to use the HF1, since this score provides a balanced score between the precision and the recall.

- Precision;

  This metric indicates how well the model identifies the relevant data alone.
  Precision = $\frac{TP}{TP+FP}$

- Recall;

This metric indicates the number of true positive predication made out of all possible positive predictions.

Recall = $\frac{TP}{TP+FN}$

- F1;

This metric combines the previously mentioned metrics and gives insight into the accuracy of the models.

F1 = 2 $\times \frac{Precision \times Recall}{Precision + Recall}$

- Accuracy;

This metric summarizes the number of correct predictions divided by the total number of predictions.

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

For the evaluation and comparison of the flat models, the macro average scores were used. As the dataset was balanced with SMOTE this method is suitable because it treated all classes equally. The models were validated by running all models four times for more insight in the accuracy. In each run, the dataset was split again into a new training and test set to reduce the chance of accidentally having a perfect or flawed set. For the best-performing models, two more validation sets on which the models were not trained were also kept in order to test the generalizability of the models on unseen data. To facilitate the comparison of the flat models with the hierarchical models, the values of the RCSFI levels were listed in sequence as follows: level 1-2, level 1-2-3 and level 1-2-3-4.

4. Results

This section presents the results of the experiments conducted in this research.

4.1. Results of the Flat Classifiers

In Table 4, an overview of the classification performance of the different models using the flat approach is listed. Among the traditional algorithms, Logistic Regression has the best performance with an accuracy of 79.25% for level 1-2. However, the accuracy drops to 73.50% for level 1-2-3-4, although it is still 20% higher than Naive Bayes. Looking at the results of the ensemble-based algorithms, it is noticeable that XGBoost outperforms Random Forest and LightGBM by achieving an accuracy of 93% for level 1-2 and 79% for level 1-2-3-4.

Table 4. Classification results of the models in percentages when applying the flat approach.

	Accuracy			**Precision**			**Recall**			**F1**		
RCSFI level	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4
Naïve Bayes	67.25	59.75	53.50	74.25	61.75	60.50	52.25	60.00	49.00	58.00	56.00	50.50
Logistic Regression	79.25	75.00	7.50	81.25	80.50	75.50	75.00	76.50	73.75	77.25	67.75	71.50
Random Forest	93.00	83.75	78.75	94.00	88.00	80.25	92.25	85.50	79.00	93.00	87.75	77.50
LightGBM	92.25	83.00	78.00	93.50	87.75	80.25	91.75	84.75	78.00	92.50	85.50	77.00
XGBoost	93.00	84.00	79.00	94.00	88.75	81.00	92.25	86.00	79.00	93.00	86.00	78.00
BERT base multilingual	98.75	98.00	94.75	97.50	98.00	95.00	97.75	98.00	94.75	97.75	98.00	94.25
DistilBERT	98.75	97.50	95.00	98.00	97.75	95.25	97.75	97.00	95.00	97.75	97.00	94.50
BERTje	98.75	97.00	93.75	97.75	97.25	93.75	98.25	97.00	93.25	97.75	97.00	93.25

It can also be inferred from Table 4 that the neural-network-based NLP techniques BERT base multilingual, DistilBERT and BERTje obtained higher scores than the traditional

and ensemble-based algorithms. In this regard, DistilBERT achieved the highest accuracy for level 1-2-3-4.

4.2. Results of the Hierarchical Classifiers

An overview of the classification performance of the hierarchical models is shown for each hierarchical strategy. The classification performance of the models using the Local Classifier Per Parent Node design pattern is shown in Table 5. The results show that XGBoost achieves the highest performance followed by LightGBM and Random Forest. The same order also applies for the Local Classifier Per Node and the Local Classifier per Level approaches. These results are shown in Tables 6 and 7. The obtained results show that the hierarchical approaches do not clearly increase the classification performance for the models on the dataset used. The hierarchical F1 suggests that performance has improved; however, the flat metrics show no improvement. The hierarchical F1 is also lower than the F1 obtained with the flat neural-network-based techniques.

Table 5. Classification results of the models when applying the hierarchical Local Classifier Per Parent Node Approach.

	Accuracy			**Precision**			**Recall**			**F1**			**Hierarchical F1**		
RCSFI level	1-2	1-2-3	1-2-3-4	1-2	1-2	1-2-3	1-2-3-4	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4
Naïve Bayes	64.08	46.92	39.95	64.56	52.88	38.68	36.39	46.71	40.61	49.01	37.66	40.06	52.88	38.68	36.39
Logistic Regression	75.35	63.34	59.01	75.04	65.24	60.67	71.68	61.60	58.89	72.61	61.07	56.55	82.26	75.86	72.07
Random Forest	89.25	79.79	74.59	93.40	85.81	75.20	87.12	79.80	74.47	89.72	80.76	73.07	93.36	88.91	85.30
LightGBM	91.25	81.01	75.62	92.43	86.85	77.87	90.33	82.64	75.64	91.06	83.61	74.32	95.29	90.47	86.83
XGBoost	92.68	83.26	78.99	93.76	87.85	81.45	92.11	85.18	79.03	92.76	85.76	77.97	96.15	91.91	89.09

Table 6. Classification results of the models when applying the hierarchical Local Classifier Per Node Approach.

	Accuracy			**Precision**			**Recall**			**F1**			**Hierarchical F1**		
RCSFI level	1-2	1-2-3	1-2-3-4	1-2	1-2	1-2-3	1-2-3-4	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4
Naïve Bayes	60.31	44.67	39.27	53.14	40.54	37.96	62.25	43.93	39.30	53.61	38.79	34.24	70.89	62.21	56.58
Logistic Regression	74.60	62.26	56.97	74.63	64.09	58.61	70.25	60.15	57.27	71.04	59.63	54.64	81.78	75.27	70.56
Random Forest	88.26	79.03	73.74	93.23	86.13	76.29	86.83	78.87	73.78	89.13	80.43	72.13	92.96	88.28	84.62
LightGBM	90.49	79.64	74.08	98.89	86.48	77.28	90.22	81.55	74.22	90.61	82.60	72.72	94.92	89.81	85.51
XGBoost	92.57	83.23	77.90	93.60	88.05	80.16	91.78	84.71	77.86	92.61	88.54	76.87	96.09	91.87	88.33

Table 7. Classification performance of the models when applying the hierarchical Local Classifier Per Level Approach.

	Accuracy			**Precision**			**Recall**			**F1**			**Hierarchical F1**		
RCSFI level	1-2	1-2-3	1-2-3-4	1-2	1-2	1-2-3	1-2-3-4	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4	1-2	1-2-3	1-2-3-4
Naïve Bayes	63.82	46.40	39.56	64.50	48.05	39.69	49.15	37.79	39.69	52.93	39.07	35.93	72.95	63.96	58.16
Logistic Regression	73.70	61.71	56.24	74.38	64.13	37.37	69.80	59.43	56.21	71.82	59.65	53.99	81.37	74.85	70.44
Random Forest	88.24	78.51	73.11	93.58	86.61	76.08	85.89	78.10	73.15	88.86	79.99	71.54	92.79	88.20	84.25
LightGBM	91.35	81.38	76.09	92.56	87.34	78.69	90.49	84.25	76.32	91.26	83.99	75.11	95.32	90.72	87.04
XGBoost	92.86	83.31	78.89	93.08	87.87	81.25	92.46	85.33	79.04	93.03	85.70	77.90	96.23	91.87	89.93

5. Discussion

Exploration of the type of features that can be used as predictors in the classification models, based on feature relation estimates, was conducted during the EDA phase. First, the client-specific fields were removed as these were different for each client and could not be used for unseen prediction. For the remaining features, the chi-square test of independence was performed. Additionally, in order to determine the strength between the features, the Cramer's V/Phi were calculated. This resulted in the following features being used in the traditional and ensemble algorithms: Ledgeraccountnumber (grouped), Ledgeraccountdescription, JournalDescription, TransactionLineDescription, JournalType and Amount (grouped). For the neural-network-based NLP techniques, the Ledgeraccountdescription, JournalDescription and TransactionLineDescription features were used as input.

After conducting a literature review, it was found that various hierarchical strategies could be employed to determine the design pattern for hierarchical classification, which can be utilized for both traditional and ensemble algorithms. Additionally, the literature review also highlighted the difference in classification performance between the flat and hierarchical approaches. The used design patterns were "Local Classifier Per Node," "Local Classifier Per Parent Node" and "Local Classifier Per Level Approach". The results from the previous section showed that the classification performance of the models did not clearly improve, this was in line with the findings of (Zimek et al. 2008) and their recommendation to use strong multi-class classifiers. The results were even less when compared based on the flat metrics, with some exceptions such as the precision and recall scores of XGBoost at level 1-2-3-4. Thus, a structural advantage could not be seen from the scores of the aforementioned design patterns. Stein et al. (2019), on the other hand, found an improved performance; this could be due to the fact that they used different text representation techniques and that the textual data were documents instead of shorter descriptions.

In order to comprehend the performance of neural-network-based natural language processing techniques in classifying different RCSFI levels of ledgers, the results in Table 4 demonstrate that BERT, DistilBERT and BERTje significantly outperformed the traditional and ensemble models. What is particularly noticeable here is that the decrease in the scores when adding a new level is more limited for these models than for the traditional and ensemble algorithms. It is also notable that BERTje did not always outperform BERT base multilingual, while BERTje was specifically trained on Dutch data. The reason for this could be the difference in the pre-training procedure between BERTje and BERT. Another reason could be the use cases on which the authors made the comparison. These were Named-Entity Recognition, Part-of-Speech tagging, Semantic Roles and Spatio-Temporal Relations and, finally, Sentiment Analysis and thus differed from the use case in this study. It should also be noted that the differences are small and that BERTje scored higher on level 1-2 precision and recall. DistilBERT outperforming the larger BERT model is also

noteworthy. This is in line with the study of (Arslan et al. 2021), as their results showed that DistilBERT outperformed BERT on some datasets.

The main research question was: To what extent can neural- network-based natural language processing techniques outperform traditional and ensemble-based algorithms in classifying general ledgers into 'Reference Classification System of Financial Information' compliant categories? The neural-network-based natural language processing techniques outperformed the traditional and ensemble-based algorithms in terms of accuracy, precision, recall and F1. The difference in accuracy between DistilBERT and the best performing traditional model (Logistic Regression) started at level 1-2, with 19.5%, and increased to 21.5%, at level 1-2-3-4. In terms of precision, the difference started at 16.75% at level 1-2 and increased to 19.75% at level 1-2-3-4. The difference in terms of accuracy between DistilBERT and XGBoost (the best-performing ensemble-based model) started at level 1-2 with 5.75% and increased to 16% at level 1-2-3-4. In terms of precision, the difference was 4% at level 1-2 and increased to 14.25% at level 1-2-3-4. These differences were also found to a similar extent among the recall and F1 metrics, as shown in Table 4.

One of the limitations of the data used is that no additional work was carried out to verify the validity of the labels provided. Since the labels were provided on behalf of different clients, certain choices in mapping might have conflicted with each other. This could have led to a decreased performance of the models. The same applied to the validity of the features extracted from the audit files. In addition, the selection of audit files did not take into account what type of companies these were or what the source system was. Another limitation on the data is that, due to the limited amount of data, not all RCSFI categories could be included in the models. This also negatively affected the generalizability of the models on new data, as these might include categories that were not known in the training data.

Additionally, a question that arises following the results of the neural-network- based natural language processing techniques is how generalizable the models are for unseen audit files. In order to ascertain their generalizability, the models were used for the predictions of two unseen datasets. The average macro accuracy dropped from 98.75% for level 1-2 to an average of 74.5% for DistilBERT. The precision dropped to 63.5% and the recall decreased to 61.5%. This indicates that the models are less generalizable to data that they have not been trained on. Although the decrease in performance for precision and recall was smaller when evaluating weighted averages instead of macro averages. The weighted average scores were 73% for precision and 74.5% for recall.

A limitation regarding the traditional and ensemble-based models is that, during the data pre-processing steps, count vectorizing was used, which is a simple word embedding technique. The performance of these models might be improved if a more advanced technique is implemented, such as in the research of (Stein et al. 2019). Finally, a limitation of this study is that it did not consider the neural-network-based natural language processing techniques in the hierarchical approaches, given their performance with the flat approach.

6. Conclusions

Increased data availability has changed the way audits of financial statements are conducted. This research aimed to assist in leveraging the data by the automatic classification of ledgers into the RCSFI-compliant categories. Several studies have shown that ensemble-based algorithms and neural-network-based natural language processing techniques achieve encouraging results in classification problems. However, there is a lack of empirical comparison between these two techniques, especially within the financial audit domain and on Dutch data. This research attempted to address to what extent the neural-network-based natural language processing techniques could outperform the ensemble-based and traditional algorithms in this context. This research also investigated whether the implementation of a hierarchical approach would improve the classification performance of the traditional and ensemble-based models in this classification problem.

As the results demonstrated, implementing the hierarchical approaches did not improve the classification performance of the traditional and ensemble-based models on the data used. Additionally, the results showed that for the RCSFI classification problem neural-network-based natural language processing techniques obtained the best results. These models outperformed both the flat and hierarchical traditional and ensemble-based models. The DistilBERT model achieved the highest scores on level 1-2-3-4. This model obtained an F1 of 97.75%, 97% and 94.50% for levels 1-2, 1-2-3 and 1-2-3-4, respectively. With these results, DistilBERT also outperformed BERTje at level 1-2-3-4 despite BERTje being pre-trained on multiple Dutch datasets. Similarly, BERTje did not outperform the BERT base multilingual model on all levels.

7. Future Work

As addressed in the Discussion section there are a number of points that could be taken into consideration for future work. As became clear, the performance of the DistilBERT model, among others, decreased with unseen data indicating a lesser degree of generalizability. It also became clear that not all categories of RCSFI were available in the datasets used. To improve this, the model could be trained with more data. Another area that could be looked into is to take the typology and the source system of the audit files into account. With this, clustered models could be created for data from similar organizations. It is then more likely that the words used in the description fields would be more similar and the generalizability of the model would increase for unseen data within the same cluster.

Furthermore, the application of hierarchical approaches for DistilBERT could be considered. This could then be used to determine if those approaches add value in terms of the performance and generalizability of this model.

Finally, Responsible AI is also a topic that could be looked into in greater depth. Rizinski et al. (2022) provided a paper in which finance ethics and machine learning ethics are mapped. Based on the outlined guidelines, this ML project could be further tested. For example, research could be carried out on the explainability of the predictions made by the DistilBERT model.

Author Contributions: Conceptualization, A.C., S.S.M.Z. and J.-W.v.E.; methodology, A.C., S.S.M.Z.; software, A.C.; validation, A.C., S.S.M.Z. and J.-W.v.E.; formal analysis, A.C.; investigation, A.C., J.-W.v.E.; resources, A.C., J.-W.v.E.; data curation, A.C.; writing—original draft preparation, A.C., S.S.M.Z.; writing—review and editing, S.S.M.Z.; visualization, A.C.; supervision, S.S.M.Z., J.-W.v.E.; project administration, J.-W.v.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: As the data provided by a private Dutch company, Baker Tilly, is unavailable due to privacy or ethical restrictions, the data cannot be accessed for this study.

Acknowledgments: We would like to express our gratitude to Baker Tilly for providing us with the data and computational servers necessary to deploy and experiment with our models.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. XAF AUDIT FILE (Adopted from Auditfiles.nl)

- Header; The header element contains the metadata about the audit file. The following fields are included (Table A1)

Table A1. The Header of audit file.

Nr.	Field	Description
1	auditfileVersion	XAF version of the auditfile
2	CompanyID	Company registration number
3	taxRegistrationNr	Tax number of the company
4	companyName	Name of the company
5	companyCity	City of the company
6	companyPostalCode	Postal code of the company
7	companyAddress	Address of the company
8	FiscalYear	Indication of the financial year
9	startDate	Fiscal year start date
10	endDate	Fiscal year end date
11	CurrencyCode	Local currency of the administration
12	dateCreated	Date on which the file was created
13	productID	Name of the accounting package
14	productVersion	Version of the accounting package
15	numberEntries	Number of mutations
16	TotalDebit	Total debit amount
17	TotalCredit	Total credit amount

- General Ledger; Within general ledger, the following information is stored (Table A2)

Table A2. The General Ledger of audit file.

Nr.	Field	Description
1	taxonomy	A code to the various ledger accounts
2	accountID	Ledger account code
3	accountDec	Ledger account name
4	accountType	Type of ledger
5	leadCode	City of the company
6	leadDescription	City of the company

- Customer Supplier; The customer/supplier element of the audit file contains an elaboration of the debtor and creditor data (Table A3).

Table A3. The customer/supplier element of the audit file.

Nr.	Field	Description
1	custSupID	Debtors or creditors number
2	type	Debtors or creditors type
3	taxRegistration	Debtors or creditors fiscal number
4	taxVerificationDate	VAT verification date
5	companyName	Debtors or creditors name
6	contact	Contact person
7	streetAddress	Shipping address
8	postalAddress	Invoice address
9	telephone	Debtors or creditors phone
10	fax	Debtors or creditors fax
11	email	Debtors or creditors e-mail
12	website	Debtors or creditors URL

- Transactions; The transaction data is recorded for each transaction with the following details in Table A4.

Table A4. The Transaction data.

Nr.	Field	Description
1	transactionID	Transaction number
2	description	Description of transaction
3	period	Fiscal period of transaction
4	transactionDate	Date of processing
5	sourceID	Person/application entering the transaction

- Journal; The transactions are split per journal and the journals contain the following details in Table A5.

Table A5. The Journal details.

Nr.	Field	Description
1	journalID	Journal number
2	description	Description of journal

- Transactionslines; The following data is recorded per transaction line (Table A6).

Table A6. The Transactionlines details.

Nr.	Field	Description
1	recordID	Unique row number
2	accountID	Ledger account code
3	custSupID	Debitors or creditors number
4	documentID	Document number
5	effectiveDate	Date of mutation
6	description	Description of transaction line
7	debitAmount	Debit amount in local currency
8	creditAmount	Credit amount in local currency
9	costDesc	Cost center
10	productDesc	Cost unit
11	projectDesc	Projectcode
12	vat	Vat code
13	currency	

- VAT; The VAT element is broken down as follows in Table A7.

Table A7. The VAT elements.

Nr.	Field	Description
1	vatCode	Code of VAT
2	vatPercentage	
3	vatAmount	

References

Abbas, Muhammad, Kamran Ali, Saleem Memon, Abdul Jamali, Saleemullah Memon, and Anees Ahmed. 2019. Multinomial Naive Bayes Classification Model for Sentiment Analysis. *International Journal of Computer Science and Network Security* 19: 40169. [CrossRef]

Abdurrahman, Ginanjar, and Mukti Sintawati. 2020. Implementation of xgboost for classification of parkinson's disease. *Journal of Physics: Conference Series* 1538: e012024. [CrossRef]

Akoglu, Haldun. 2018. User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine* 18: 91–3. [CrossRef] [PubMed]

Ali, Amal Al, Ahmed M. Khedr, Magdi El-Bannany, and Sakeena Kanakkayil. 2023. A Powerful Predicting Model for Financial Statement Fraud Based on Optimized XGBoost Ensemble Learning Technique. *Applied Sciences* 13: 2272. [CrossRef]

Arslan, Yusuf, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F. Bissyandé, Jacques Klein, and Anne Goujon. 2021. A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain. Paper presented at the Companion Proceedings of the Web Conference 2021, Madrid, Spain, May 25–28, pp. 260–68. [CrossRef]

Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz. 2020. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review* 54: 1937–67. [CrossRef]

Bergstra, James, Brent Komer, Chris Eliasmith, Daniel Yamins, and David D. Cox. 2015. Hyperopt: A python library for model selection and hyperparameter optimization. *Computational Science Discovery* 8: 014008. [CrossRef]

Bilal, Muhammad, and Abdulwahab Ali Almazroi. 2022. Effectiveness of Fine-tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews. *Electronic Commerce Research* 2022: 1–21. [CrossRef]

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence J. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16: 321–57. [CrossRef]

Chen, Hong, and Wei Wan. 2023. Analysis of E-Commerce Marketing Strategy Based on Xgboost Algorithm. *Advances in Multimedia* 2023: 1247890. [CrossRef]

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. Paper presented at the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17.

de Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv* arXiv:1912.09582. doi:10.48550/arXiv.1912.09582. [CrossRef]

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* arXiv:1810.04805v2.

González-Carvajal, Santiago, and Eduardo C. Garrido-Merchan. 2020. Comparing BERT against traditional machine learning text classification. *arXiv* arXiv:2005.13012.

Hajek, Petr, Mohammad Zoynul Abedin, and Uthayasankar Sivarajah. 2022. Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework. *Information Systems Frontiers*. [CrossRef] [PubMed]

Lei, Shimin, Ke Xu, Yizhe Huang, and Xinye Sha. 2020. An Xgboost based system for financial fraud detection. *E3S Web of Conferences* 214: 2042. [CrossRef]

Lorena, Ana Carolina, André. C. P. L. F. de Carvalho, and João M. P. Gama. 2008. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review* 87: 19–37. [CrossRef]

Martínez-Castaño, Rodrigo, Amal Htait, Leif Azzopardi, and Yashar Moshfeghi. 2021. BERT-Based Transformers for Early Detection of Mental Health Illnesses. Paper presented at the 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, pp. 189–200. [CrossRef]

McHugh, Mary L. 2013. The Chi-square test of independence. *Biochemia Medica* 2013: 143–49. [CrossRef]

Miranda, Fábio M., Niklas Köhnecke, and Bernhard Y. Renard. 2023. Hiclass: A python library for local hierarchical classification compatible with scikit-learn. *Journal of Machine Learning Research* 24: 1–17.

Ng, Andrew Y., and Michael. I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. *Neural Information Processing Systems* 14: 841–48. Available online: https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf (accessed on 3 May 2023).

Nurhayati, Hidayati Nasrah, and Iskandar Muda. 2022. The General Ledger and Reporting Systems Cycle: Traditional vs. Digital Accounting Information Systems Era in Pharmacy Issuers and Implementation of Internal Control Procedures That Enable Cost Savings in Dealing with Threats in the Cycle. *Journal of Pharmaceutical Negative Results* 17: 3558–65. Available online: https://www.pnrjournal.com/index.php/home/article/view/5155 (accessed on 17 June 2023).

Paleczek, Anna, Dominik Grochala, and Arthur Rydosz. 2021. Artificial breath classification using XGBoost algorithm for diabetes detection. *Sensors* 21: 4187. [CrossRef]

Porter, Martin F. 2001. Snowball: A Language for Stemming Algorithms. Available online: http://snowball.tartarus.org/texts/introduction.html (accessed on 4 June 2023).

Potdar, Kedar, Taher S. Pardawala, and Chinmay D. Pai. 2017. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications* 4: 7–9. [CrossRef]

Putatunda, Sayan, and Kiran Rama. 2018. A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost. Paper presented at the 2018 International Conference on Signal Processing and Machine Learning, Shanghai, China, November 28–30, vol. 1, pp. 332–40. [CrossRef]

Rahman, Saifur, Muhammad Irfan, Muhammad Raza, Khawaja Moyeezullah Ghori, Shumayla Yaqoob, and Muhammad Awais. 2020. Performance Analysis of Boosting Classifiers in Recognizing Activities of Daily Living. *International Journal of Environmental Research and Public Health* 17: 1082. [CrossRef] [PubMed]

Rizinski, Maryan, Hristijan Peshov, Kostadin Mishev, Lubomir T. Chitkushev, Irena Vodenska, and Dimitar Trajanov. 2022. Ethically Responsible Machine Learning in Fintech. *IEEE Access* 10: 97531–54. [CrossRef]

Sajjad, Ahmed, Knut Hinkelmann, and Flavio Corradini. 2022. Development of Fake News Model using Machine Learning through Natural Language Processing. *arXiv* arXiv:2201.07489. [CrossRef]

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* arXiv:1910.01108. [CrossRef]

Silla, Carlos N., and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22: 31–72. [CrossRef]

Stein, Roger Alan, Patricia A. Jacques, and João Francisco Valiati. 2019. An Analysis of Hierarchical Text Classification Using Word Embeddings. *Information Sciences* 471: 216–32. [CrossRef]

Tang, Jiali, and Khondkar E. Karim. 2017. Big data in business analytics: Implications for the audit profession. *CPA Journal* 87: 34–39. Available online: https://www.cpajournal.com/2017/06/26/big-data-business-analytics-implications-audit-profession (accessed on 10 January 2023).

van Buuren, Joost, and Wiebren Wijma. 2022. Over kwaliteitsborging van datagedreven controlemethodologie. *Maandblad voor Accountancy en Bedrijfseconomie* 96: 15–25. [CrossRef]

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv* arXiv:1910.03771.

Zibran, Minhaz Fahim. 2007. Chi-Squared Test of Independence. pp. 1–7. Available online: http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/topic-fahim-CHI-Square.pdf (accessed on 17 June 2023).

Zimek, Arthur, Fabian Buchwald, Eibe Frank, and Stefan Kramer. 2008. A Study of Hierarchical and Flat Classification of Proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 7: 563–71. [CrossRef] [PubMed]