IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 January 2024, accepted 4 April 2024, date of publication 11 April 2024, date of current version 26 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387860

RESEARCH ARTICLE

Data Augmentation With Semantic Enrichment for Deep Learning Invoice Text Classification

WEI WEN CHI[1], TIONG YEW TANG[1], (Member, IEEE), NARISHAH MOHAMED SALLEH[1], MUAADH MUKRED[1], HUSSAIN ALSALMAN[2], AND MUHAMMAD ZOHAIB[3]

[1]Department of Business Analytics, Sunway Business School, Sunway University, Bandar Sunway, Selangor 47500, Malaysia
[2]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[3]Software Engineering Department, Lappeenranta-Lahti University of Technology, 53851 Lappeenranta, Finland

Corresponding author: Tiong Yew Tang (tiongyewt@sunway.edu.my)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2024R244.

ABSTRACT Natural language processing (NLP) is a research field that provides huge potential to automate accounting tasks dealing with text data. This research studies the application of NLP in automatically categorizing invoices based on the invoice text description. The study employs semantic enrichment, data augmentation, and deep learning to address the NLP unique issues posed by the inherent short text and multi-class imbalance nature of invoice descriptions. Semantic enrichment was done using labels as an information source. Training data was artificially increased with either WordNet synonym replacement, Global Vectors for Word Representation (GloVe) word replacement, or the Bidirectional Encoder Representations from Transformers (BERT) word replacement method. Each training dataset was then supplied for training with one nondeep learning classifier and two deep learning classifiers respectively, namely Linear Support Vector Machine (LSVM), Bidirectional Long Short-Term Memory (Bi-LSTM), and BERT. Overall, the semantically enriched, WordNet augmented training set paired with the BERT classifier yielded the best results, successfully preserving semantics, reducing noise and overfitting while improving accuracy per class, achieving an increase of performance up to 20 percentage points (ppts) for macro F1 score and 6.7 ppts for accuracy.

INDEX TERMS Long short-term memory, data augmentation, deep learning, machine learning, global vectors for word representation, management accounting, natural language processing, semantics.

I. INTRODUCTION

The Industrial Revolution 4.0 (IR 4.0) brought about the growth of unstructured data in the past decade with a rate of up to 65% each year [2]. Unstructured data contains valuable information useful for elevating enterprises above their competition [2] and staying ahead in the areas of customer engagement, operational excellence, and product leadership [3]. Text data, a type of unstructured data, was introduced during the early days of digital computing [4]. With text data storage technology, Information Retrieval (IR) tasks became of great interest [5], and set the foundation for more automated and intelligent methods for textual-based processing like automated text classification tasks [6], [7]. Early on, researchers realized the need for machines to understand the natural language for more accurate indexing, instead of purely mathematical-based indexing [8], [9], fueling interest in natural language processing (NLP).

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

A. NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is the process of converting textual input into usable information by computers, making human-machine natural language interaction possible [1], [10], [11]. Technically, NLP converts semantic and syntactic relationships between textual data into a computationally represented knowledge model used for downstream tasks like statistics, clustering, and classification [12]. Today,

we observe many NLP applications in business, including the usage of sentiment analysis, spam detection, grammar checks, content summarization, and chatbots like Siri, Alexa, and more recently ChatGPT [13], [14].

B. NLP IN THE FINANCE AND ACCOUNTING DOMAIN

In finance and accounting, businesses have already used NLP with some success, particularly in the audit practice. Zhou reported on the successful use of NLP in Deloitte to automate the high volume checking of contracts to detect modification in ''control provisions'', speeding up the checking process from half a year to a month [15], [17]. Ernst & Young (EY) utilized NLP to assess available leases to find the impact in ''lease accounting rules'' owing to the Inland Revenue Service amendments (IRS) [15]. Zhang et al. proposed that NLP could extract risk results automatically, decreasing auditors' ''heavy reading'' to finish their audit assessment [16]. Li et al. noted that deep learning and NLP are increasing research momentum but still not yet to be mainstreamed in finance [18]. Current research opportunities are here.

C. RESEARCH MOTIVATION

The first reason for conducting this research is to apply NLP in automating business tasks and lowering operational costs. Volumes of unstructured data can be read by machines if NLP techniques and tools are configured correctly to read the text [19], potentially scaling up big data processing tasks and releasing human capacity for more intelligent tasks. The lack of automation in the accounting domain is the second motivation [20]. Accounting tasks have traditionally been performed manually, with a significant amount of repetitive work involved in reading and classifying unstructured text, such as invoice processing and invoice classification. These tasks are laborious and error-prone. Another key motivation is that prior research in the accounting domain has focused on sentiment analysis derived from texts in external financial reports and other publicly available financial sources [21], [22], [23], with little emphasis on NLP applications to accounting processes such as invoice classification or bank reconciliations [24]. As such, there is an incentive to investigate NLP applications in this area and address this research gap.

D. RESEARCH SCOPE

The business process in scope for this research is invoice classification based on the invoice text description. The nature of the invoice textual data introduces the expected problems to be faced when classifying invoices based on their respective text descriptions, namely ''Short Text Problems'' and ''Class Imbalance Problems''.

Invoice descriptions are expected to be brief, therefore also known as short text. This means that the text doesn't have as much context and meaning as a long text, is more ambiguous, and sparse, lacks grammatical structure, and has the synonym or homonym challenge [25], [26]. When using the term frequency-inverse document frequency (TF-IDF) method on short texts, each term tends to appear only once in each text, which makes it hard to assign weights [25], [27], [28]. In the past few years, deep learning models have become the most advanced way to sort text [29]. However, for the model to work well, deep learning requires a large training set volume [30]. Text augmentation may be needed if there is a need for more training data [31]. But for short text, the lack of data may make it hard to add contextual word embeddings such as Bidirectional Encoder Representations from Transformers (BERT) to improve the textual data. This is because there isn't enough contextual data in the original text to accurately add to the text while keeping the original semantic meaning. Most short-text vocabularies are informal or specific to a certain field, and terms are not standardized [32]. This could lead to a lot of abbreviations, misspelt words, typos, named entities, and reference numbers. If language models are used that can't figure out what abbreviations, named entities, and reference numbers are, this could cause problems with text classification. Instead of being useful semantic data, these data end up being noise.

Most investigations have used binary categorization. Multi-class classification of unbalanced data is a newer issue than binary classification imbalance problems due to its more complex nature compared to two-class datasets [33]. Invoice classification is multi-class. The majority classes will have far more instances than the minority classes. Most classifier learning methods assume a balanced distribution when modelling, so an imbalanced class distribution will make learning harder. Classification systems assume minority classes are rare, unknown, or unobserved, therefore they are misclassified more often than majority class cases [34] Due to their design, most machine learning algorithms optimize classification accuracy by sacrificing minority class accuracy [35]. In other cases, like invoice categorization, minority classes may have fewer incidences but high severity if ERP systems misclassify them. Improving accuracy in one class may hurt another [36]. Optimizing experiments is necessary to find the best configurations for classifier performance because descriptions, class labels, and learners are interdependent. Class overlaps with several groups, class label noise, and unclear class borders are issues which further complicate the classification problem.

This research aims to study the different NLP approaches to invoice text classification, in particular data augmentation, semantic enrichment, and learning models. Firstly, semantic enrichment is introduced to the short invoice text, using labels as the information source. This is meant to improve the semantic information of the short text. Next, the data is augmented using three different methods, namely the WordNet lexical synonym replacement, Global Vectors for Word Representation (GloVe) word embedding similar word replacement, or BERT contextual word embedding similar word replacement, resulting in different training sets to be used for classification performance comparison. This step is for increasing the training data for increased generalization

of the learning model and overcoming overfitting. Lastly, each training dataset is sent for modelling with one traditional classifier and two deep learning classifiers respectively, namely Linear Support Vector Machine (LSVM), Bidirectional Long Short-Term Memory (Bi-LSTM), and BERT. A point to remember is that the study employs the BERT model in two distinct areas of the methodology, which is in the augmentation process, where BERT is used to generate replacement words, and the text classification process, where BERT is used for label prediction.

The findings of this research can be broadly applied and used in a few key areas. First, a breakthrough approach in invoice text classification may be replicable in the accounting departments. This will help organizations reduce overhead costs for repetitive and mundane accounting tasks. The method could potentially be replicated for usage in other text classification use cases within the finance and accounting domain. The unique short text semantic enrichment technique shared in this study is a novel approach in the invoice description classification context, to the best of our knowledge. This can be added as a contribution to a growing number of studies on approaches to enrich short text description data semantically without altering the original meaning of the text. The class balancing technique in this study is not new, but the applicability of invoice text data will be useful for further research studies, particularly in the area of model generalization ability. Data augmentation for generating additional data sets may not be as simple as intended, as any augmentation technique not done well, will cause the synthetic data to lose its original meaning and increase noise within the model. In existing literature, we note that approaches to text classification tend to employ either semantic enrichment or data augmentation to improve classification tasks, but not together, to the best of our knowledge. The unique approach of combining semantic enrichment and text data augmentation to improve invoice state-of-the-art text classification is therefore a novel one and could be a noteworthy contribution to the research community.

The next section discusses the related research work for key components of this research. Section III discusses the methodology utilized for the experiment. Section IV shows the results of the experiment. Section V discusses the conclusion of the research. Finally, Section VI shows the research limitations and future works.

II. RELATED WORK

The discussion of the literature focuses on the related NLP and text classification methods employed in this study, namely short text semantic enrichment, text data augmentation, and text classification.

A. SHORT TEXT SEMANTIC ENRICHMENT

Semantic enrichment is the process of adding new information like named entities, topic tags, or emotion ratings to text data to enhance its context or meaning. It has been realized since the early 1960s that more semantic information would improve the accuracy and relevance of text being processed, even though the prevailing method for indexing at the time was purely mathematical [8], [37], [38], [39], [40], [41], [42]. Since 1965, a growing number of studies used dictionaries, pre-assigned term relationships, and knowledge bases as auxiliary data supplying semantic and syntactic information to the text [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. The relevance feedback method also gained popularity by using prior search result data for more accurate subsequent search results [55], [56], [57], [58], [59].

Short text issues are common in social media, and efforts have been made to incorporate more supplementary semantic information to improve text classification tasks. Utilizing supplementary sources such as well-known information sources like Wikipedia or relevant websites is a popular strategy [25], [26]. Phan et al. found that adding latent topics derived from large universal datasets to sparse datasets improved internet search and online contextual advertising matching accuracy [25]. Jin et al. highlighted that semantic enrichment using data outside of the existing datasets may introduce noise instead of improving semantics, resulting in a negative experimental outcome [26]. Mehanna and Mahmuddin proposed representing the tweet alongside supplementary prior conversation texts to improve sentiment detection [60]. For example, if the most recent tweet is too ambiguous, the model can infer the sentiment from previous conversations. Label embedding is another well-liked method of acquiring more information. Typically, this is achieved through an attention process. Dong et al. used ''self-interaction attention mechanisms'' and label embedding to improve the model's understanding of semantics [61].

Researchers agree that auxiliary sources should be used to improve semantic information. The same concept is used in this study, but the auxiliary source for semantic information is the class label itself. So far, there are no other methods that use the random augmented label data to add to the original text to improve semantic information and reduce data sparsity.

B. TEXT DATA AUGMENTATION

The data augmentation technique is a method that artificially increases the size and variety of training data by generating synthetic datasets based on real datasets to improve classification accuracy [62]. Through augmentation, the performance of machine learning models can be improved, introducing a healthy level of noise, reducing overfitting, and improving model generalization [63], [64]. The early motivation of augmentation was to overcome inherent classification problems by imbalanced dataset classes and was increasingly researched since the early 2000s primarily using over-sampling techniques [65], [66].

In recent times, technological advancement has brought deep learning architecture based on neural networks to the forefront of NLP tasks and requires large amounts of data to learn effectively [30]. It is not uncommon for deep

learning text classification research to utilize data sample sizes spanning from tens of thousands up to a few million samples [67], [68], [69], [70], However, due to the difficulties in obtaining large enough training datasets, researchers have studied the usage of data augmentation to overcome this problem [31], [64], [71]. Shorten et al. studied some notable emerging techniques, including 'rule-based augmentation', 'back-translation', and 'generative data augmentation' [64].

The rule-based text data augmentation, popularly represented by Easy Data Augmentation (EDA), was introduced by Wei and Zhou and is the inspiration for this study. This method is straightforward, quick, and simple to perform, focusing on swapping words, random word insertion or deletion, or synonym word replacements to generate new text data [63]. Duong and Nguyen-Thi proposed that EDA for Vietnamese could ''improve sentiment polarity'' because of its simplicity in comparison to other augmentation strategies [72]. Due to its simplicity, most studies use EDA as a baseline enhancement function to compare with a more complex approach [73], [74]. Ma introduced a variation of the EDA which has more complex augmentation Python functions like back translation, contextual word embedding augmentation, and word embedding augmentation in addition to the already available synonym augmentation [75].

Data augmentation was not always successful in improving NLP tasks. Zhou and Liu found that augmentation does not guarantee more accurate classification results for multi-class datasets and may even introduce negative performance [33]. There is also a possibility of introducing too much noise through text augmentation which creates additional challenges for the learning model [31]. Wei and Zou did not expect EDA to improve pre-trained models such as BERT [63]. This study challenges this assumption by supplementing invoice text training data with simple synonym replacements for sets of 100,000, 200,000, and 300,000, feeding them to the BERT text classifier for evaluation of performance.

So far, no known studies have been done on performing EDA or its more complex variants on short text. This research focuses on three types of augment methods, namely synonym replacement, word embedding, and contextual word embedding technique. Generally, the common characteristic of all three methods is that each employs random word replacements. The difference lies in the basis on which the substitute word is derived. In the synonym replacement technique, a lexical database is used to source the synonym replacement. For the word embedding technique, a pre-trained word embedding model is used to find replacement words of similar meaning. Thirdly, for the contextual word embedding technique, a language model is used to generate replacement words based on the context of the sentence. In this study, the lexical database used is WordNet, the word embedding model method used is GloVe, and the language model used is BERT respectively for the augment techniques.

Miller et al. introduced the WordNet database reference designed as an online reference by computer systems mainly for quick indexing purposes, but with potential for other useful purposes, which in recent times has been proven useful for natural language tasks [52], [76], [77]. In the WordNet database, ''synsets'' are collections of words that are synonymous (or nearly synonymous). Each ''synset'' is represented by a unique concept and contains a list of words or phrases that can be used interchangeably within that concept. A taxonomy of concepts is formed by the hierarchical organization of ''synsets'' in WordNet. Understanding the relationships between various concepts, such as hypernyms (more general phrases) and hyponyms (more precise terms), is made possible by this hierarchy. The WordNet database contains more than 118,000 registered unique words with over 90,000 ''synsets'' [78].

The GLoVe method is an unsupervised learning technique that relies on word co-occurrence data from a sizable corpus of text. By factorizing a word co-occurrence data matrix, it creates word embeddings that capture both the semantic links and syntactic between words [79]. There are various pre-trained word vector sets for GloVe sourced from databases such as Wikipedia, Common Crawl, Twitter, and Gigaword. The vector set used in this study is based on 6 billion tokens trained from the English Gigaword Fifth Edition and 2014 Wikipedia dump [79]. The GloVe embedding substitution involves replacing a word in a sentence with another term that is closely associated with the original term. GloVe embeddings are based on co-occurrence statistics, which capture global word relationships, as opposed to BERT embeddings, which are learned using a neural network. Tan et al. improved sentiment analysis with GloVe-based minority oversampling [80].

BERT is a transformer-based neural-network model that has been pre-trained utilizing a masked language modelling objective on a huge amount of text data, in particular the 2,500 million words from English Wikipedia [81] and 800 million words from BooksCorpus. The BERT model learns through masking random tokens from input instances and trained to forecast the value of the masked tokens. Words or sentences that the BERT model has learned during pre-training are represented by BERT embeddings. A technique known as BERT embedding substitution entails replacing a word in a sentence with another term that has a similar meaning but differs slightly contextually. For NLP jobs, this method is used to increase the number of training data and enhance model performance. In the statement ''The dog chased the cat,'' for instance, the word ''dog'' might be changed to ''puppy,'' which has a similar meaning but a somewhat different context.

C. TEXT CLASSIFICATION

Before deep learning, traditional learning models used to be popular for text categorization. Support Vector Machine (SVM) was one the most effective traditional machine learning methods for textual analysis and classification with state-of-the-art results [5], [10], [82], [89], [98], [99] and

was in direct competition with neural networks in the 1990s. The foundation of SVMs is the idea of margin maximization, which entails locating the hyperplane that maximizes the distance between the two nearest data points from various classes. The theory behind this is that the classifier will be more noise-resistant and more general to new data if the margin is wide enough. SVMs are frequently employed for text categorization jobs like spam detection [92]. In text categorization, each document is often represented as a vector of word occurrences or word frequencies. These vectors may have tens of thousands, hundreds of thousands, or even millions of features, which may not be linearly separable. These high-dimensional feature spaces are easily handled by SVMs using kernels, which project data into a high-dimensional space where it may be linearly separated [93]. Although the usage of kernels or nonlinear SVMs are desirable, linear kernels or the linear SVM (LSVM) is also popular for text classification because they can be trained more quickly and easily, and the choice of kernel functions has little bearing on the classification performance [93], [94], [95]. SVM model performance is a good baseline for measuring the effectiveness of deep learning models like Bi-LSTM and BERT in recent works [96], [97]. As of this report, no research has used SVM for invoice text classification.

The early beginnings of deep learning was based on the basic artificial neural network ''Perceptron'' described in the late 1950s by Rosenblatt [83], and got significant interest in the mid-1980s [84], when Rumelhart et al. popularized back-propagation used for training multi-layer perceptron (MLP) with hidden layers where most architectures were feedforward networks and RNNs [85], [86], [87], [88]. In the late 1990s, deep learning architectures dramatically increased the performance of neural networks. The explosion of ''Big Data'', large unstructured and measured up to exabytes, generated through internet companies like Google and Facebook was a key driver of deep learning architecture adoption [30], [90]. Najafabadi et al. proposed that the ability of deep learning to learn deep patterns from large unstructured texts is beneficial to tackling the big data challenges, yet this remains an under-researched field [30]. In recent times, deep learning models like Bidirectional LSTM (Bi-LSTM) and BERT are replacing traditional NLP models with positive results [91].

LSTM is a sort of RNN architecture created to resolve the issue of vanishing gradients in conventional RNNs [100]. Memory cells in an LSTM network can retain data over time and selectively forget or remember it as required. Gate units that control the information flow into and out of the cell oversee these memory cells. The input gate, output gate, and forget gate are among the gate units. The input gate regulates the information that is input into the memory cell, while the forget gate regulates the data that should be erased from the cell. Based on the current input and the prior state of the memory cell, the output gate regulates the output from the memory cell [101]. LSTM was found to be superior to feedforward and RNNs [102]. The Bi-LSTM model was subsequently introduced by Graves & Schmidhuber, where the architecture analyses input sequences both forwards and backwards [103]. A Bi-LSTM has the benefit of being able to record both the past and the future context of a sequence. This can be especially helpful in jobs like voice recognition, where understanding the context both before and after a word might be crucial to correctly identifying it. Xu et al. found that Bi-LSTM outperforms CNN, RNN, LSTM, and NB. Bi-LSTM is frequently included for comparison with CNN, RNN, BERT, and SVM [96], [97], [74], [104]. The ''Keras'' library allows Bi-LSTM to be imported into Python. No study using Bi-LSTM for invoice text classification was found.

Devlin et al. created BERT, which can machine learn from unlabeled text bidirectionally and capture context [81]. This results in BERT learning models, containing valuable understanding of its underlying knowledge base. The BERT learning model's knowledge is represented in BERT embeddings which is useful for data augmentation. This knowledge is also able to be fine-tuned and leveraged for downstream tasks like text classification. BERT and its derivatives (RoBERTa, ALBERT, BART, etc.) are state-of-the-art models that often outperform deep learning and classical models due to their simple yet advanced technique [96], [105], [106]. The BERT model includes many parameters, operates on a large scale, and has high latency [107]. BERT is one of the largest natural language processing models in recent times. Due to these limitations, the model cannot operate without high computing requirements and incurs a long training period. In some real-time scenarios, the costs outweigh the benefits of improved accuracy. Gao et al. discovered that BERT performed worse than CNN and self-attention network models on papers over 400 words [70]. The Python ''transformers'' package loads BERT. No research has used BERT to classify invoice text.

III. METHODOLOGY

In this section, the popular Cross-Industry Standard Process for Data Mining (CRISP-DM) framework was utilized and will have the following sequence: the business understanding phase, the data understanding phase, the data preparation phase, the modelling phase, and the evaluation phase. Amani and Fadlalla referred to the use of this method in implementations including data mining [24]. According to Sharda et al., text mining approaches require more complex preprocessing steps than a data mining project [1].

A. BUSINESS UNDERSTANDING

The business process in the scope of this study is the process of filling up the electronic registration form and saving the invoice and associated Chart of Account (COA) data in the enterprise resource planning (ERP) system. The current process first starts at the point of the accounts staff receiving a supplier invoice from the supplier for some work done. To process the payment to the supplier, the staff must first register the invoice with the ERP system. The staff is expected to read the invoice and then summarize the business invoice details in the invoice transaction description within

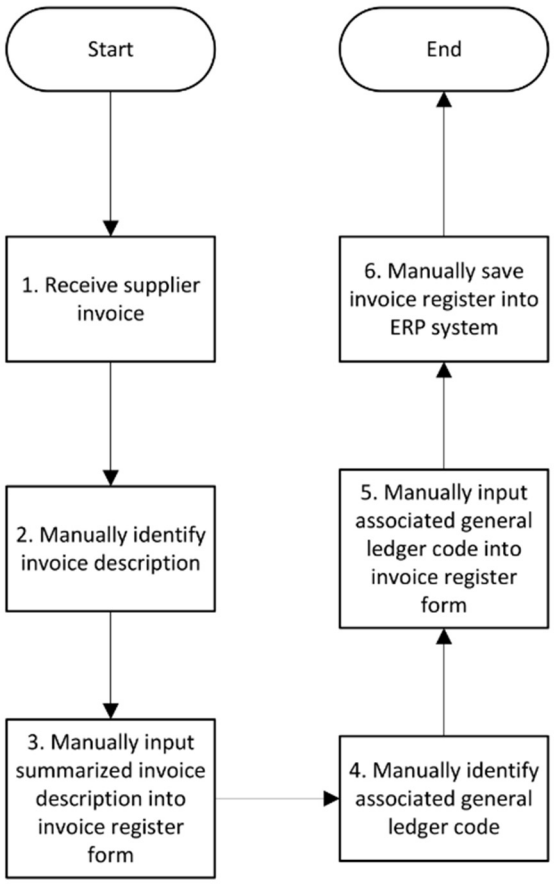

FIGURE 1. Current invoice classification manual process.

the electronic form. Then, the staff will identify the account code which is associated with the invoice description and input the code into the form. The system then auto-displays the description of the code. The staff finally saves the input data into the system. Fig. 1 displays the summary of the invoice classification process flow.

Potentially, steps 2-5 could be automated and yield a potential cost reduction of 90%. This is derived from a reduction of cycle time from 6 minutes per transaction to 0.6 minutes per transaction. Steps 2-3 and 5-6 can be automated using RPA technology. Step 4 is where the text classifier will automatically classify the account code on the invoice text description obtained, which is within the scope of this study. The next sections will be the research methodology for the text classifier scope.

B. DATA UNDERSTANDING

The data for this study originated from the invoice process, and the data was stored in the ERP system. The raw dataset was generated from this source from 2019 to 2021. In its raw format, the data has 13,933 rows and 2 columns. The total class is 77. The data definitions are in Table 1.

There were notable initial data observations. Firstly, the presence of uppercase (e.g., ''Claim 24.12.19-Lunch with Client''), special characters (e.g., ''12â€''), digits (e.g., ''18.11.20''), named entities, abbreviations (e.g., ''KUL''), and misspellings (''singking'' instead of ''sinking.'') were detected and will need to be removed as they do not contribute much to the context of the text description and are more likely to be noise and negatively impact the modelling process. Reference numbers (e.g., ''003-00000-HVLV'') and stop words (e.g., ''on'') were detected and should be removed to aid in model performance. There were class labels that were homogenous with slight differences, for example, ''Accommodation—Local'' and ''Accommodation—Travel.''. Some invoice descriptions appeared for multiple classes. These will be removed from the dataset as they will impact the modelling performance. Due to the multiple-class nature of the dataset, it is observed that the data is imbalanced. Fig. 2 displays the invoices class categories which were classified from the operation activities.

TABLE 1. Invoice data description.

Data	Data Type	Description
description	String	Invoice text description
label	String	General ledger code description

From the data preparation step up to the evaluation step, the scope is based on the framework displayed in Fig. 3. In short, the major steps taken in the data preparation phase include semantic enrichment, text cleaning, and text data augmentation. In each of these steps, datasets are produced for the modelling phase. In the modelling phase, text classification is performed before the evaluation phase. The operational steps were performed using Python language. The Python codes are published to GitHub for reference [108]. To make use of their available graphics processing unit (GPU), the Python codes were run on Google Colab and Kaggle.

C. DATA PREPARATION

Data preparation encompasses four key activities, namely text cleaning, train-test split, semantic enrichment, and text data augmentation. Refer to Fig. 4 for the data preparation process flow and sequence of activities. The original raw dataset is named 'D01' and is the basis for text cleaning. The dataset post text cleaning is named 'D02'. Both datasets 'D01' and 'D02' are divided into train and test datasets. 'D2_train' is then utilized for semantic enrichment where the 'D3_train' dataset is produced. The 'D3_train' dataset is then further used for text augmentation, specifically using WordNet (lexical synonym substitution), GloVe (word embedding substitution), and BERT (contextual word embedding substitution) methods. The output of the augmentation is 3 training datasets of 100,000, 200,000, and 300,000 instances for each augment technique respectively. At the end of the data preparation process, there will be 12 distinctive sets of training data and 3 unique sets of test data. All augmented training sets will be paired with the 'D2_test' test dataset, as the semantic and

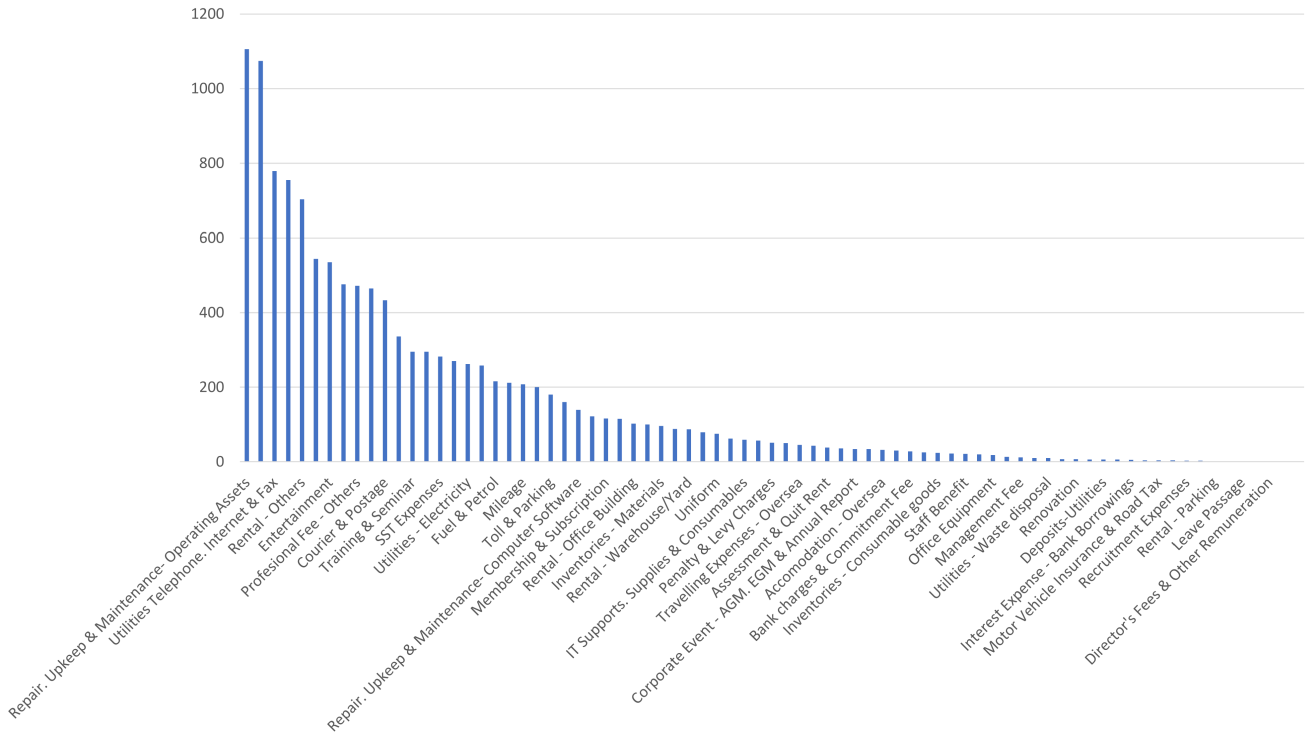

FIGURE 2. Invoice categories class imbalance.

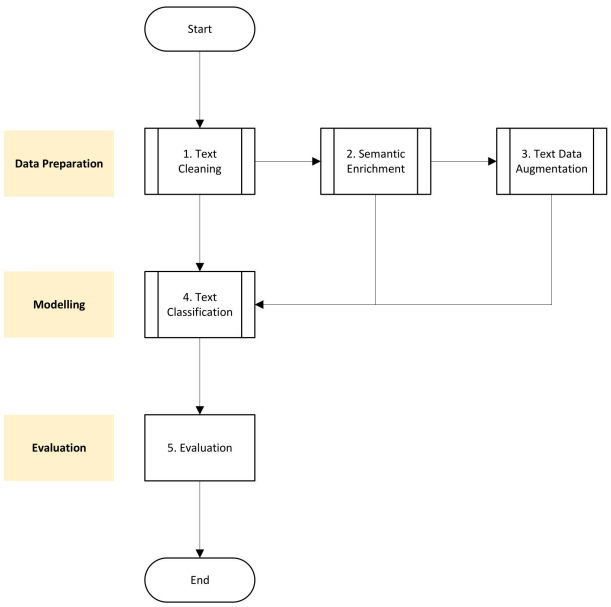

FIGURE 3. Overall methodology from data preparation to evaluation phase.

augmented sets were based on the 'D2_train' training dataset. The summary of train-test pairing sets is tabulated in Table 2.

In the text cleaning sub-process, the invoice descriptions are cleaned. This step aims to ensure only words with useful semantics are kept and to ensure the learning efficiency of

TABLE 2. Train-test pairing dataset description.

Set	Train Set Description	Test Set Description
D01	D1_train	D1_test
D02	D2_train	D2_test
D03	D3_train	D2_test
D04	D3_WNtrain100k	D2_test
D05	D3_GLtrain100k	D2_test
D06	D3_BTtrain100k	D2_test
D07	D3_WNtrain200k	D2_test
D08	D3_GLtrain200k	D2_test
D09	D3_BTtrain200k	D2_test
D10	D3_WNtrain300k	D2_test
D11	D3_GLtrain300k	D2_test
D12	D3_BTtrain300k	D2_test

the model. The process starts with the loading of the 'D01' dataset and ends with the output of the 'D02' dataset. This process sequence is visualized in Fig. 5. First, a text-cleaning function is used to clean the text. The function includes removal of digits, lowercasing, removing symbols, removing single characters, removing multiple spaces, removing stop words, and removing month names. Then, the cleaned text will be subjected to checking against the WordNet lexicon database to retain only English words. As short texts potentially contain words which may be acronyms, jargon or incomprehensible to the language model, a conservative approach is to only retain English words. Due to the cleaning steps, there will be entirely blank rows. Those rows will be

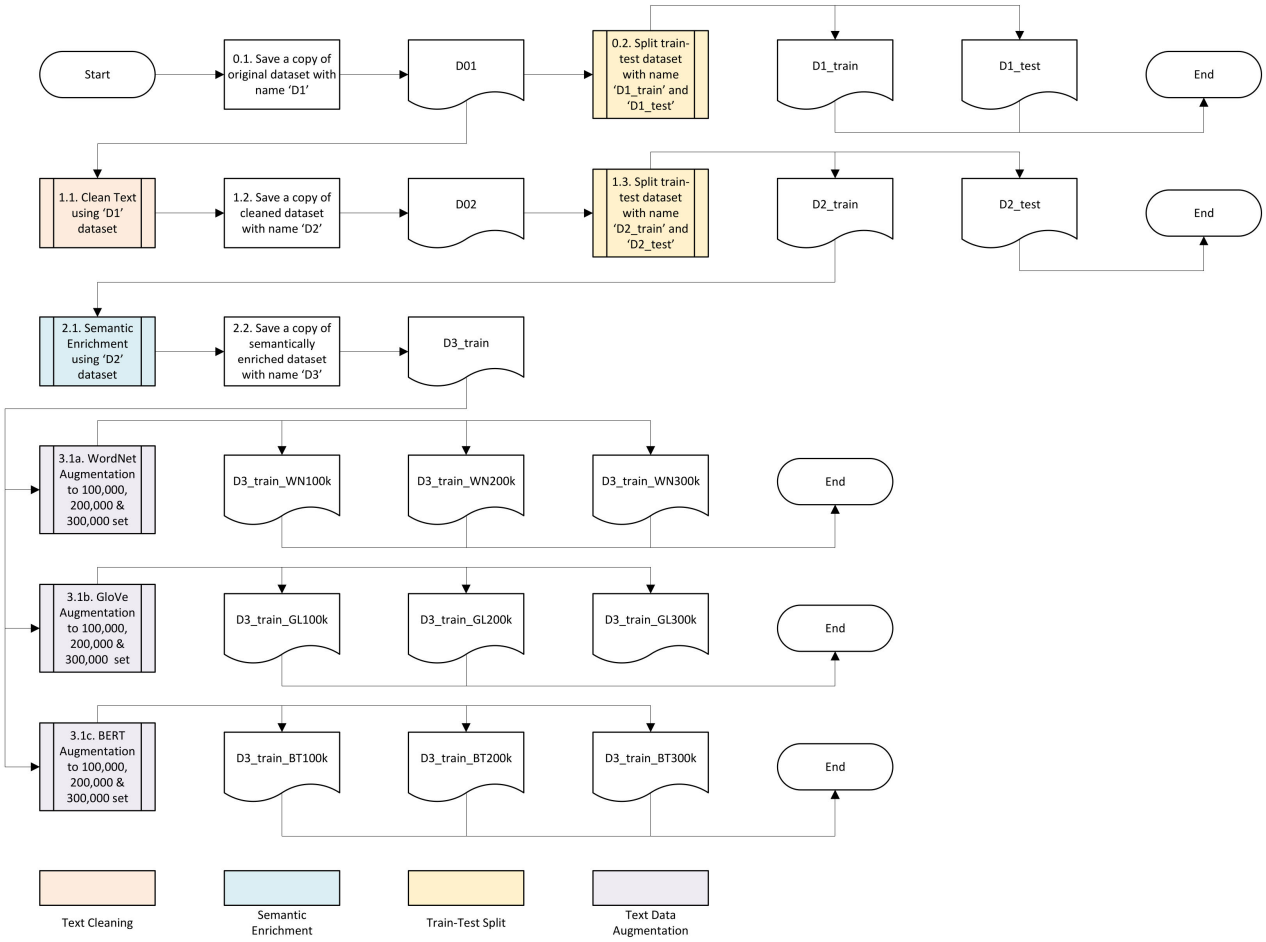

FIGURE 4. Data preparation process.

removed from the dataset. Next, instances tagged to more than one class were also removed from the dataset to reduce overlapping of classes. There is an exception to this, where within a multi-label scenario there are two labels for a single description, but one label was the minority with only one sample. In this case, the instances were not discarded. Instead, the minority sample will be reclassed to the majority label to resolve the issue. Lastly, instances where the class is tagged to only one instance were removed from the dataset to ensure at least one instance will be in the train and test set.

The datasets 'D01' and 'D02' are subjected to the train-test split procedure to prepare for the modelling phase. The general steps undertaken for the train test split are listed in Table 3. Before splitting the train and test set, any class labels with only 1 instance are removed from the dataset because it cannot split. This is to ensure both the training and testing datasets have representatives of all classes. The dataset is then split into the training and testing sets to the ratio of 85:15. This is to ensure there is sufficient training for effective learning and enough testing data for accurate evaluation. The outputs of this process are 'D1_train', 'D1_test', 'D2_train', and 'D2_test'.

TABLE 3. Train-test split steps.

Train-Test Split Steps	
01:	LOAD dataset 'df'
02:	REMOVE instances in 'df' with class WHERE the class is tagged to only one instance
03:	LIST values of 'description' column in 'df'
04:	LIST values of 'label' column in 'df'
05:	X_train LIST 85% of instances from 'x' with random selection
06:	X_test LIST 15% of instances from 'x' with random selection
07:	y_train LIST 85% of instances from 'y' with random selection
08:	y_test LIST 15% of instances from 'y' with random selection
09:	CREATE train set table with 'X_train' and 'y_train'
10:	CREATE test set table with 'X_test' and 'y_test'
11:	GENERATE 'train_set' in csv format
12:	GENERATE 'test_set' in csv format

The semantic enrichment process is where the cleaned invoice descriptions are enhanced with more semantic information from an auxiliary source, which is the class label in this research. The process starts with the loading of the 'D2_train' dataset and ends with the output of the 'D3_train' dataset. The process is visualized in Fig. 6. First, the label

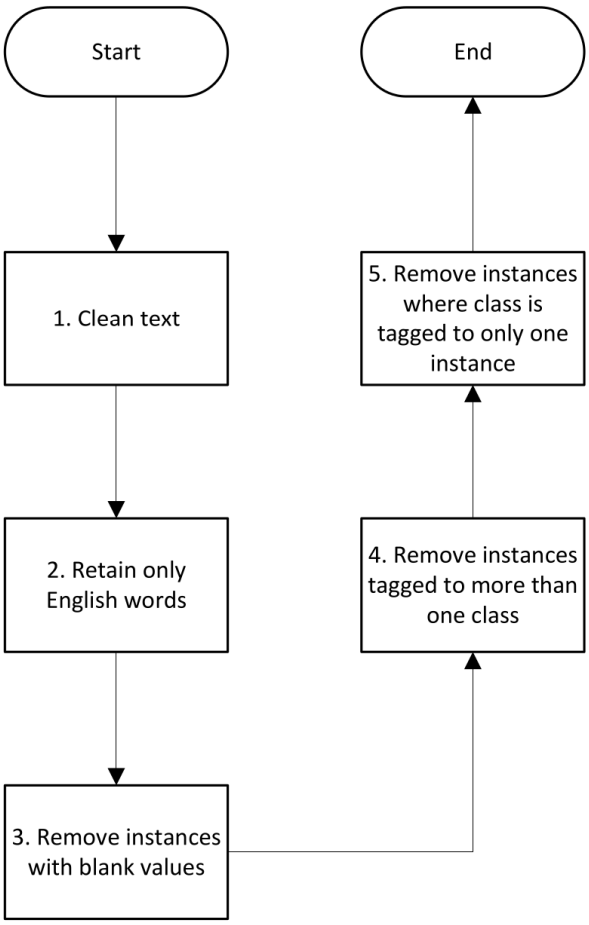

FIGURE 5. Text cleaning sub-process.

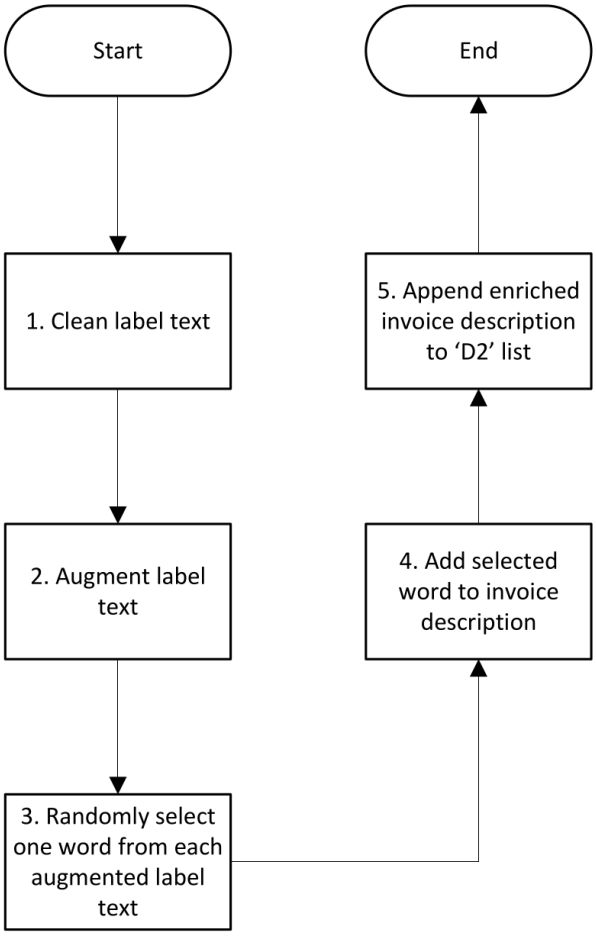

FIGURE 6. Semantic enrichment sub-process.

data is subjected to the same text-cleaning procedure in Fig. 5 to ensure standardization and consistency of label text. Subsequently, a new label column is generated to store the augmented text from the original labels. The method of augmentation is random swap, random deletion, or random insertion. The new label column with augmented texts is then converted to string format. The new label column is then tokenized. Only one word from the tokens will be selected randomly for all instances. Then, a new column is generated containing the original description text concatenated with the newly selected word for each row. This method creates semantically enriched synthetic instances using random auxiliary information from the class label. A new dataset is then generated where the semantically enhanced description text with associated labels is appended to the 'D2_train' set to generate a new 'D3_train' set with double the training data of the 'D2_train' train set.

The text data augmentation process is where the invoice text instances are artificially increased to improve model performance, particularly through resolving multi-class imbalance issues and increasing the learning effectiveness of deep learning models. The process starts with the loading of the 'D3_train' dataset and ends with the output of nine training datasets, with each augment approach producing three datasets containing approximately 100,000, 200,000, and 300,000 instances each. The purpose of having augmented datasets of different volumes of instances of 100,000 increments is to examine the impact of a larger training set on the classification model performance of the learning model. The reason for limiting the data augmentation to 300,000 instances is due to computing resource limitations where a significantly longer training time is needed to achieve the experiment objectives. The steps undertaken for data augmentation are visualized in Fig. 7.

Firstly, the 'D3_train' set instances are grouped by class. This is to determine how many synthetic instances are required to be generated to reach a near total of 300,000 instances combined across all classes while also achieving the same number of records for each class. After the required instances are determined per class, synthetic invoice texts are generated based on the 'D3_train' invoice texts in the same class group until the desired instances per class are achieved. This is done for the WordNet, GloVe, and BERT substitution augment techniques respectively. At this stage, the 'D3_WNtrain300k', 'D3_GLtrain300k', and 'D3_BTtrain300k' will have been produced. To produce

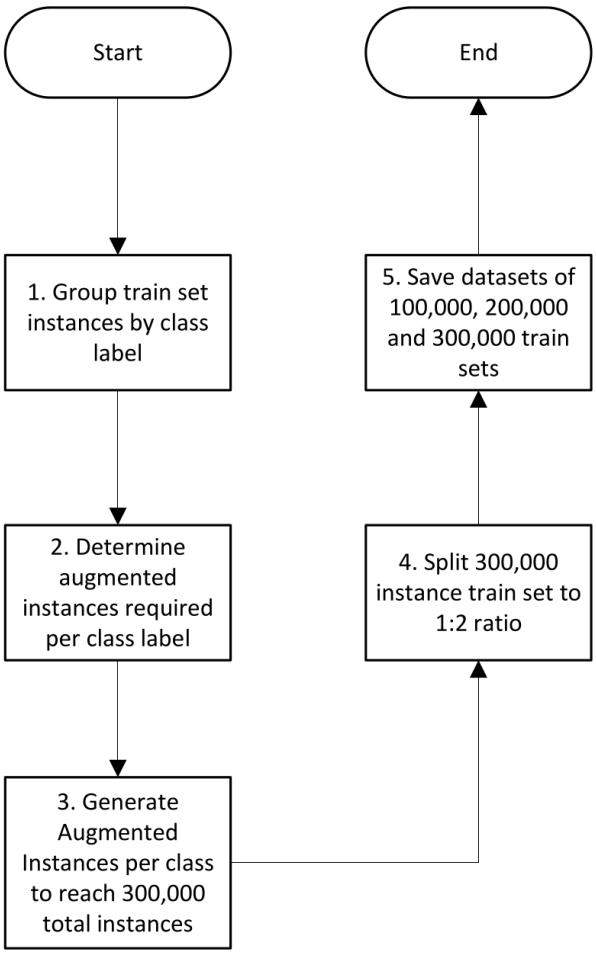

FIGURE 7. Text augmentation sub-process.

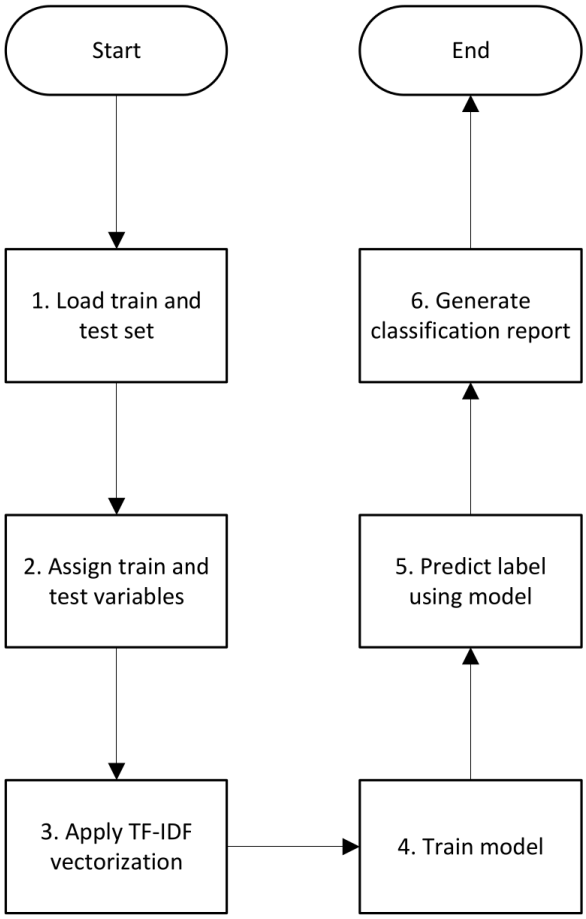

FIGURE 8. LSVM modelling sub-process.

the sets of 100,000 and 200,000 respectively, the set of 300,000 is split into a ratio of 1:2 for WordNet, GloVe and BERT training datasets. At this stage, 'D3_WNtrain100k', 'D3_GLtrain100k', 'D3_BTtrain100k', 'D3_WNtrain200k', 'D3_GLtrain200k', and 'D3_BTtrain200k' training sets will have already been produced.

D. MODELLING

The modelling phase covers the text classification activities for the three text classifiers in scope, namely LSVM, Bi-LSTM, and BERT. Here, each of the 12 train-test pairing datasets 'D01' to 'D12' listed in Table 2 is subjected to the three classifiers respectively. The output of the modelling is 12 output results for each classifier, with a total of 36 output results.

In the LSVM sub-process, the 12 train-test pairings are applied to the LSVM classifier to obtain 12 classification reports. This process sequence is visualized in Fig. 8. First, the training and testing set is loaded. The 'X_train', 'X_test', 'y_train', and 'y_test' variables are assigned accordingly. The 'X_train' and 'y_train' text are subjected to term frequency vectorization and subsequently applied with the TF-IDF transformer to apply larger weightage to more important words. The TF-IDF method is chosen as it factors in the importance of words rather than just word occurrences seen in the bag-of-words method. Next, the LSVM model is fitted to the vector text representation of the 'X_train' and 'y_train' variables for model training. After training, the 'X_test' is applied to the model for label prediction. Lastly, the predicted outputs are compared against the expected outputs 'y_test' and a classification report is generated for evaluation.

In the Bi-LSTM sub-process, the 12 train-test pairings are applied to the Bi-LSTM classifier to obtain 12 classification reports. This process sequence is visualized in Fig. 9. First, the training and testing set is loaded. Then, the label variables are codified in both train and test sets, as the model works with codes rather than label text descriptions. The 'X_train', 'X_test', 'y_train', and 'y_test' variables are assigned accordingly. Then, the train set is tokenized using the NLTK library, and a word index is created containing all the words in the train set descriptions. The NLTK library is used due to its popularity in working with NLP in Python. Next, the 'X_train' and 'X_test' word tokens are converted into integer sequences and padded to ensure the same vector sequence length for each instance. The padding step is needed as the model

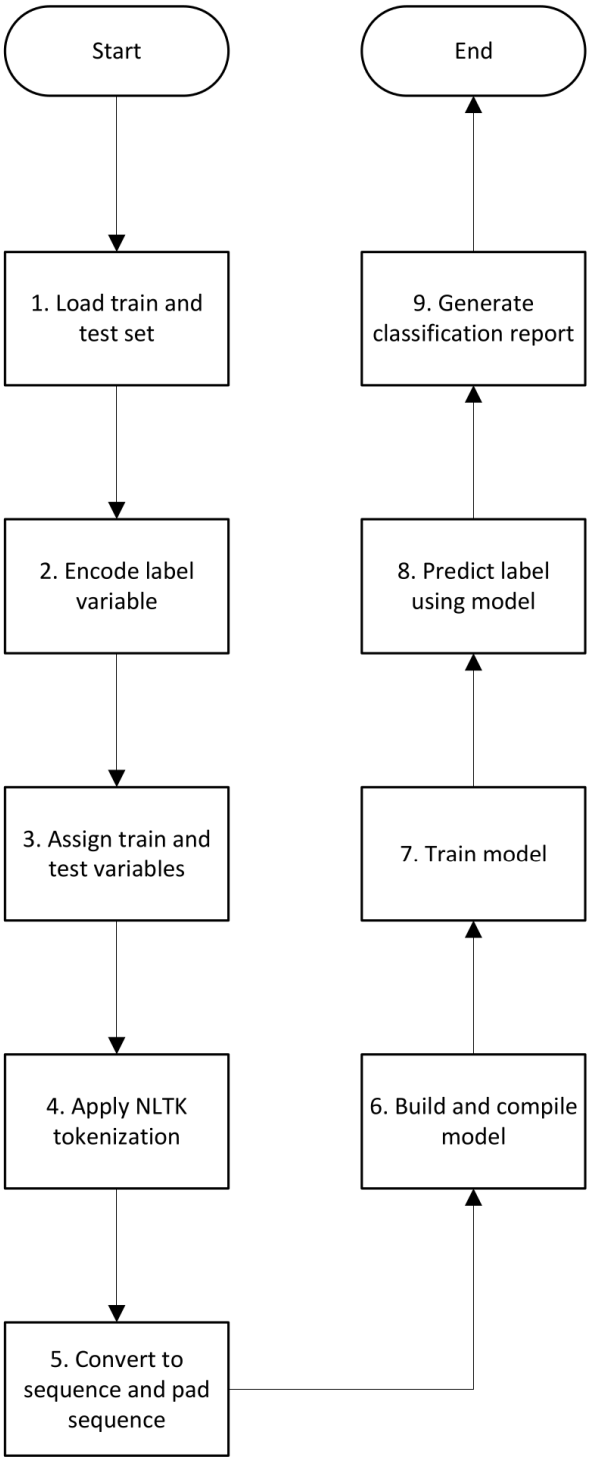

FIGURE 9. Bi-LSTM modelling sub-process.

requires input of the same sequence size for each instance. Subsequently, the Bi-LSTM model is built and compiled. The dropout rate is configured to a standard 0.5 across all the Bi-LSTM experiments in this study. This rate is selected to reduce overfitting, especially for the smaller train datasets. A learning rate of 0.001, although relatively small, is chosen to ensure stable learning and improve the chances of arriving at the optimal model, as opposed to a higher learning rate. A slow learning decay of 1e-6 is also chosen to ensure stable training. After compilation, the Bi-LSTM model is fitted to the vector text representation of the 'X_train' and 'y_train' variables for model training. A batch size of 128 was chosen. Larger batch size settings were attempted but not pursued due to larger GPU memory requirements for larger batch sizes. A train-validation split of 85-15 was set considering the large dataset volume used in this study where a 15% volume is sufficient for model performance reliability. Early stopping was set at 3 epochs past the epoch with the best validation loss. After training, the 'X_test' is applied to the model for label prediction. Lastly, the predicted outputs are compared against the expected outputs 'y_test' and a classification report is generated for evaluation.

In the BERT sub-process, the 12 train-test pairings are applied to the BERT classifier to obtain 12 classification reports. This process sequence is visualized in Fig. 10. First, the training, validation and testing set is loaded. The train-validation split is 85-15. Then, the label variables are codified in both train and test sets, as the model works with codes rather than label text descriptions. The 'X_train', 'X_val', 'X_test', 'y_train', 'y_val', and 'y_test' variables are assigned accordingly. Then, the train set is tokenized using the BERT tokenizer, the ''BERT-Base-Uncased'' pre-trained model variant. The uncased variant is chosen because the dataset does not seem to have reliable and proper casing information. The larger BERT pre-trained variants were not chosen due to limited computational resource availability for this study. Next, the input features, attention masks and labels are initialized and set from the encoded dataset. The input, attention masks and labels are then converted to tensor structures for the train, validation and test sets respectively. This step is necessary as the BERT model works with tensor structures. Then, the datasets are split into batches for model training. Due to limited GPU memory, the batch of 16 is used. Subsequently, the BERT model is trained. The BERT architecture for classification is used along with the ''BERT-Base-Uncased'' pre-trained model for the modelling process. The Adam optimizer, a well-known optimizer, is used with a common setup of a learning rate of 1e-6 and epsilon of 1e-8. This setup is to ensure stable and moderate training for the model. Early stopping was set at 3 epochs past the epoch with the best validation loss. After training, the test tensor dataset is applied to the model for label prediction. Lastly, the predicted outputs are compared against the expected outputs and a classification report is generated for evaluation.

E. EVALUATION

The evaluation step is the last step for the CRISP-DM framework, where there are 36 classification report results are examined using the accuracy and macro F1-score with 80% as the benchmark for good performance. The standard accuracy metric is the primary metric used for evaluation. This metric gives a general indicator of how well the model performs. The

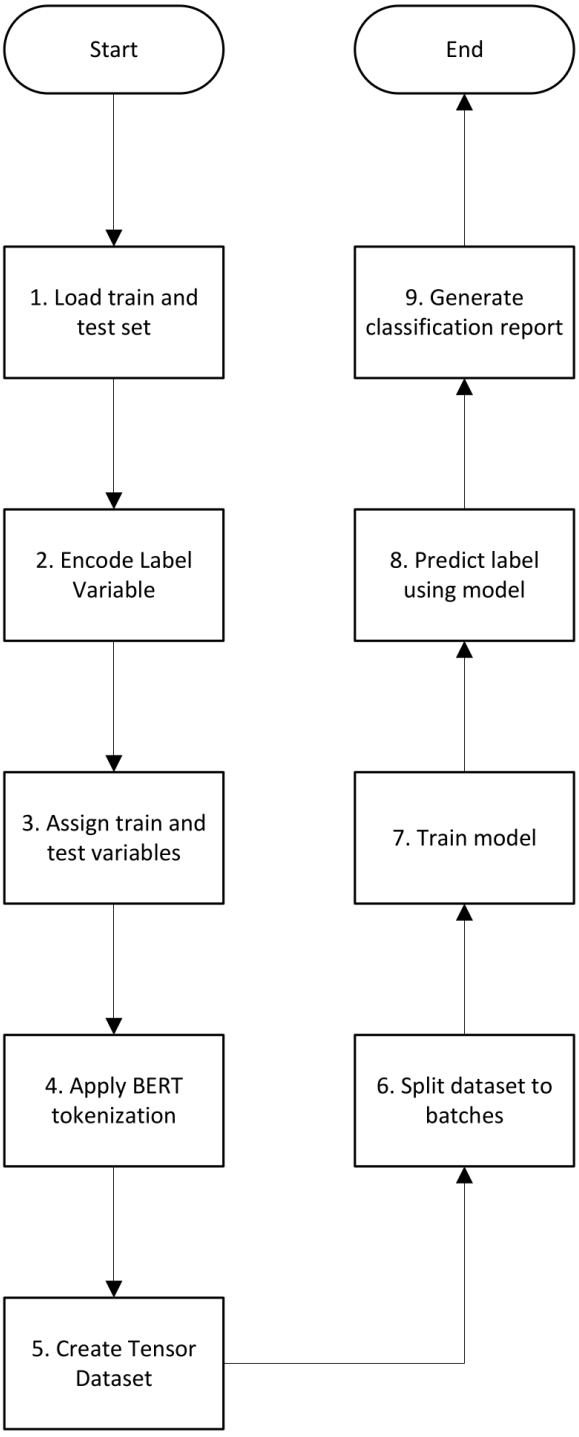

FIGURE 10. BERT modelling sub-process.

measurement is derived by calculating the percentage of the total instances correctly classified, as in (1). TP and TN refer to true positives and true negatives respectively, while FP and FN refer to false positives and false negatives respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

The accuracy metric is used due to its ease of interpretation, extensive usage across text classification studies and reliability when the classes are balanced. However, some of the datasets are imbalanced, particularly the pre-augmented datasets, which may provide biased accuracy results. This is caused by high prediction accuracy for the majority classes and low prediction accuracy for the minority classes. As such, the macro F1-score is utilized as the next classification evaluation metric. Each class, regardless of size or prevalence in the dataset, is considered equally in the macro F1-score computation. A high F1-score will show that the model is not biased towards any class. The macro F1 score is measured by calculating the F1 score for each class independently, and the unweighted average of all classes is then computed, as in (2). N refers to the total class count and F1_i is the F1 score for the class_i.

$$\text{MacroF1Score} = \frac{\sum_{i=1}^{N} F1_i}{N} \tag{2}$$

The F1-score is known as the harmonic mean of the precision and recall evaluation scores, as in (3). The macro F1-score accounts for both precision and recall for each class separately.

$$\text{F1}_i = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Recall counts how many of the true positive instances were accurately predicted, as in (4). Precision counts how many of the predicted positive examples were correctly identified, as in (5).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

IV. RESULTS AND DISCUSSION

In this section, the research results are presented and discussed from the data preparation and modelling perspective.

A. DATA PREPARATION OUTPUTS

The text cleaning, semantic enrichment, train-test split, and text data augmentation were performed successfully. Table 4 describes the volume of instances for the train and test dataset outputs for the 12 train-test paired datasets. It is observed that the text cleaning reduced the train instance count to 10,013 as shown in the 'D02' dataset. This was due to blank rows appearing post-text cleaning which had to be removed. The semantic enrichment doubled the volume of the 'D2_train' dataset to 20,026 instances in the 'D3_train' set. From 'D04' to 'D12', the train sets are augmented to instances approximately at 100,000, 200,000, and 300,000 respectively for each augment method. Due to the focus on having the same number of training instances for every class, the augmented training instances for each dataset do not total up to exactly 100,000, 200,000, and 300,000 figures. However, they are slightly higher or lower than the said amounts. The train

TABLE 4. Dataset volumes post data preparation.

Set	Train Set Volume	Test Set Volume
D01	11,837	2,090
D02	10,013	1,768
D03	20,026	1,768
D04	99,993	1,768
D05	99,993	1,768
D06	99,993	1,768
D07	200,018	1,768
D08	200,018	1,768
D09	200,018	1,768
D10	300,012	1,768
D11	300,012	1,768
D12	300,012	1,768

TABLE 5. Text classification performance.

Set	Accuracy			Macro F1-Score		
	LSVM (%)	Bi-LSTM (%)	BERT (%)	LSVM (%)	Bi-LSTM (%)	BERT (%)
D01	77.9	78.3	81.1	67.0	61.0	61.0
D02	84.4	83.2	84.1	75.0	68.0	63.0
D03	84.6	85.7	86.4	76.0	78.0	73.0
D04	81.4	83.8	86.7	72.0	78.0	75.0
D05	79.6	81.3	84.2	67.0	71.0	77.0
D06	78.5	80.2	82.8	67.0	68.0	75.0
D07	82.9	84.7	86.9	74.0	76.0	78.0
D08	80.6	83.8	86.1	69.0	74.0	78.0
D09	79.3	83.1	85.0	69.0	71.0	76.0
D10	82.7	85.0	87.7	74.0	75.0	81.0
D11	80.4	81.8	86.6	68.0	70.0	79.0
D12	80.0	78.5	86.6	70.0	66.0	79.0

datasets post-augmentation are class-balanced. See Fig. 11 for a visualization of the balanced classes post-augmentation.

B. MODELLING OUTPUTS

The LSVM, Bi-LSTM, and BERT text classification were successfully performed. The results from the modelling stage are summarized in Table 5 and visualized in Fig. 12.

The baseline results for pre-cleaning were around a 77%-81% accuracy score range with the BERT classifier as the top performer. The Macro F1 score was a minimum of 10 ppts and a maximum of 20 ppts behind the accuracy scores at the 61%-67% score range, which indicates the models were heavily biased toward the majority classes. This supports the phenomenon where the minority classes are misclassified more than the majority classes, and that the model sacrifices the minority class accuracy for overall accuracy [34], [35]. LSVM obtained the highest Macro F1-score at 67%, 6 ppts higher than Bi-LSTM and BERT classifiers, supporting the finding that SVM works well with small and sparse datasets.

Post cleaning, the accuracy of all three classifiers improved to the range of 83%-84% accuracy, with LSVM achieving the best performance at 84.4%. This confirms that the raw text has a lot of noise, containing many specific industry terms and named entities [32]. After retaining only English words, the performance improved by up to around 7 ppts from the baseline accuracy scores. The Macro F1-scores improved up to a maximum of 8 ppts with measures up to 75% achieved by LSVM while deep learning classifiers lag at below the 70% mark, indicating that deep learning classifiers still do not have enough information to detect good patterns for better classification performance.

The accuracy results for post-semantic enrichment improved to up to 2 ppts to 84%-86% range. However, a noteworthy observation is that the macro F1-scores improved up to around 10 ppts average for the Bi-LSTM and BERT deep learning classifiers, achieving 78% and 73% respectively, while LSVM registered only a 1 ppt increase to 76%. This indicates that before the semantic enrichment, some of the minority classes had higher errors, but post-semantic enrichment, these classes performed better in accuracy. This supports the concept that getting auxiliary semantic information from labels is beneficial for reducing the data sparsity issue and improving the text classification performance. The hypothesis is supported [61]. However, this only applies to the deep learning classifiers where a larger training dataset is beneficial but adversely impacts the LSVM performance, which does not perform well with larger datasets.

Post text augmentation, the overall top classifier is BERT, where its accuracy and macro F1-score performance were superior for all the datasets post augmentation except the 'D04' dataset Macro F1-score where Bi-LSTM performed better. The BERT classifier has an accuracy rate of more than 80% across all post-augment datasets. This supports the findings of recent studies where the BERT classifier outperforms other deep learning and traditional classifiers [96], [105], [106]. The datasets driving the high classification accuracy with the BERT classifier were from 'D04', 'D07', and 'D10', logging 86.7%, 86.9%, and 87.7% respectively. LSVM underperformed post-text augmentation due to the inherent limitation of the model dealing with larger datasets. Bi-LSTM, although performing better than LSVM, did not perform better than BERT due to the lack of an underlying knowledge base.

A common trait among these 3 datasets is that they were augmented using the WordNet lexical substitution method. This supports the relevance of the WordNet database as an auxiliary source for short text augmentation despite its simpler approach compared to other augmentation strategies [72]. The finding also supports the notion that EDA can improve pre-trained models such as BERT, exceeding Wei and Zhou's expectations [63]. The highest macro F1-scores with BERT classifier were caused by datasets 'D10', 'D11', and 'D12', all that were augmented to 300,000 instances using the WordNet, GloVe, and BERT augmenters, with scores of 81%, 79%, and 79% respectively The results support the findings of Chawla et al. and Estabrooks et al. where oversampling augmentation addresses the class imbalance issues inherent in the dataset like bias towards the majority class [65], [66]. The Bi-LSTM model showed the most

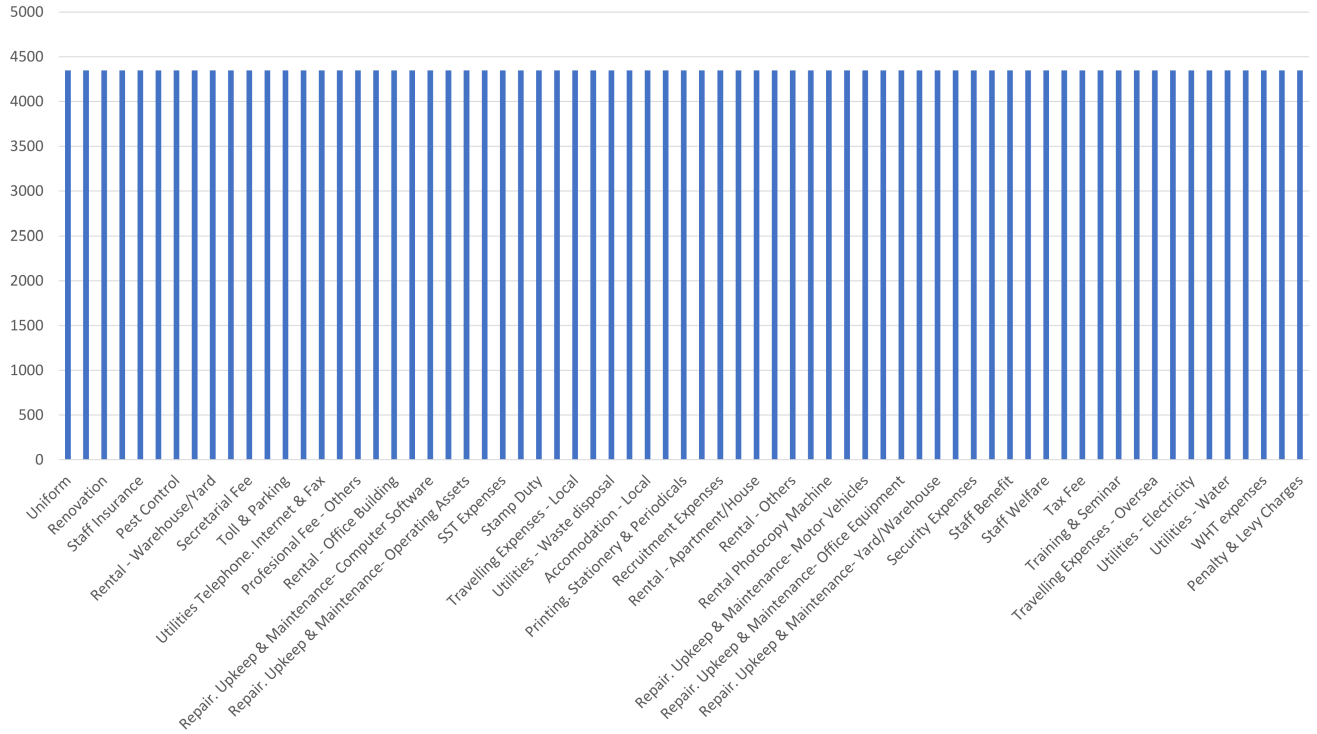

FIGURE 11. Balanced classes after augmentation.

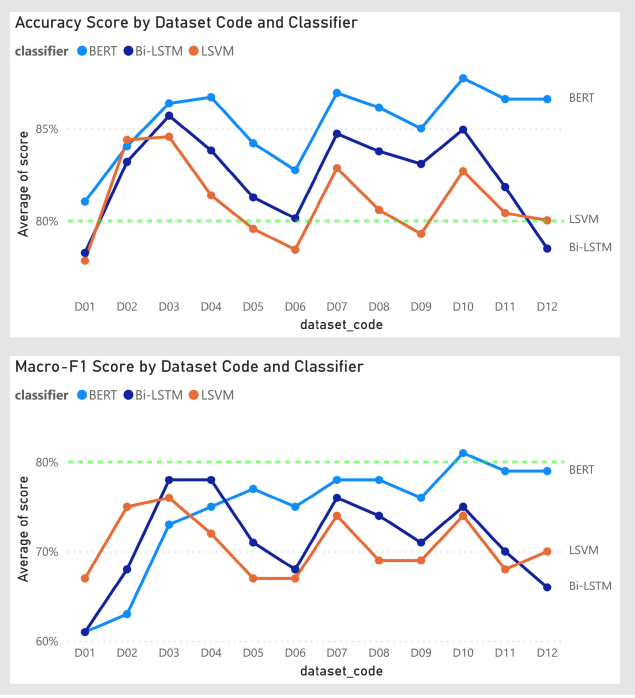

FIGURE 12. Accuracy and macro F1-score by dataset and classifier.

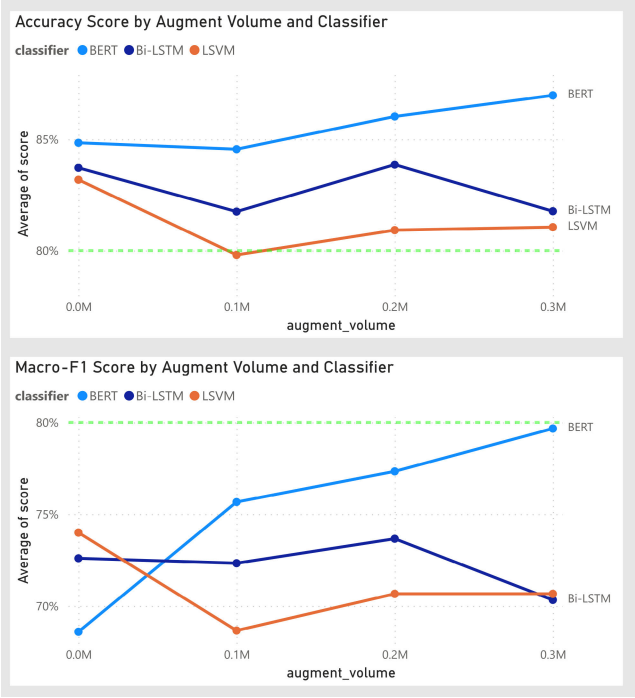

FIGURE 13. Accuracy and macro F1-scores by the classifier for various augmentation volume levels.

gain in accuracy scores, increasing by 6.7 ppts, whereas the BERT classifier showed the largest improvement in Macro F1 scores, increasing by up to 20 ppts. GloVe and BERT augmented datasets underperformed against WordNet due to the augmented words generated resulted in more noise rather than useful words for the classification models.

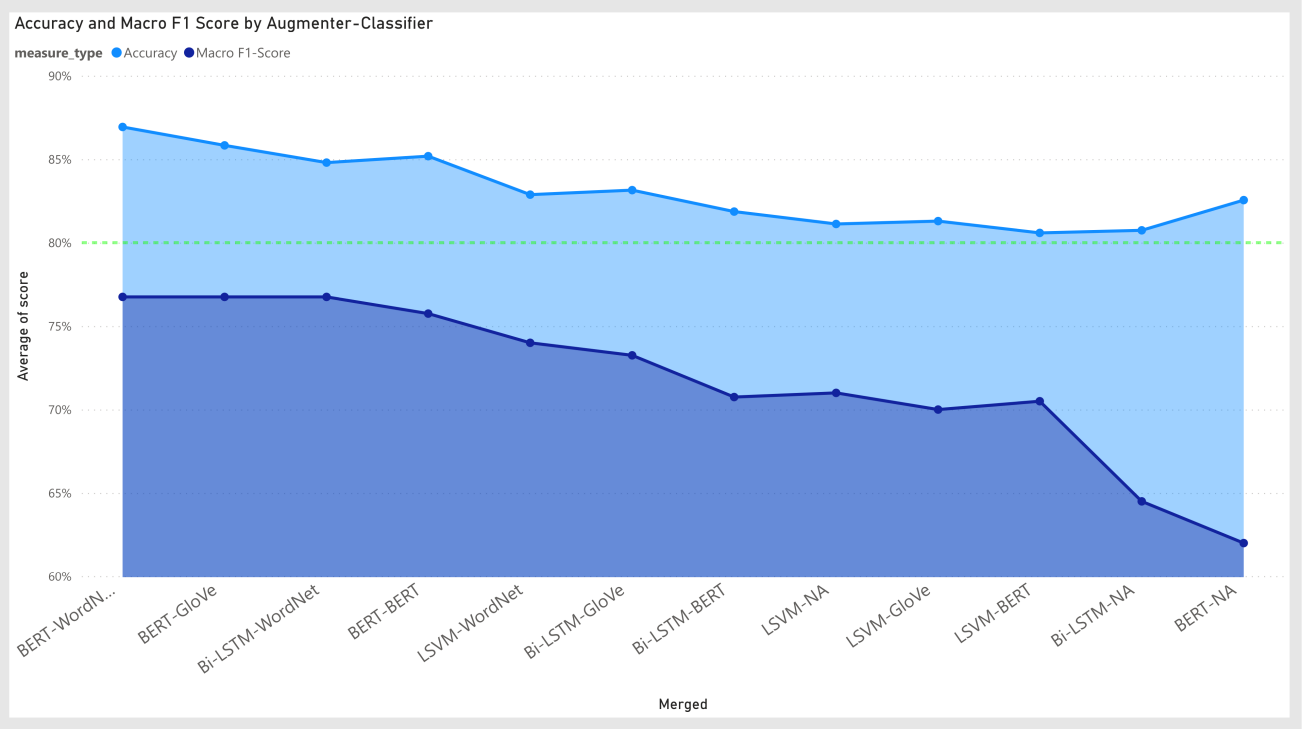

FIGURE 14. The macro F1-scores and accuracy for several classifier-augmenter combinations in average scores.

Fig. 13 demonstrates that the BERT classifier frequently attained accuracy ratings of more than 80% at various augmentation volumes and showed better performance with increasing augmented volume. At 300,000 augmented instances, the BERT classifier's macro F1 scores were trending upward and just shy of 80%. With the 100,000-instance dataset, BERT classifier Macro-F1 results significantly outperformed the non-augmented results. This result is consistent with Najafabadi et al.'s study that deep learning models learn more effectively with larger datasets [30]. However, this applies to only the BERT classifier. With the growth of synthetic data, Bi-LSTM and LSVM trended inconsistently, eventually trending down or flat at the 300,000-instance dataset. The performance of the Bi-LSTM and LSVM supports the study finding of Zhou and Liu and Sahin where augmentation may not guarantee better classification performance due to too much noise generated in augmented datasets [31], [33].

The visualization chart to compare the average accuracy and Macro-F1 scores of the classifier-augmenter combination in decreasing order is shown in Fig. 14. The BERT-WordNet classifier-augmenter is the best performer. In terms of performance, the WordNet augmenter and BERT classifier were usually found in the top five classifier-augmenter list, demonstrating the importance of these two methods for the success of the classification assignment.

V. CONCLUSION

The results of the experiment revealed a few key findings. The phenomenon of minority classes being misclassified more than majority classes was discovered during pre-cleaning, and the model sacrifices minority class accuracy for overall accuracy, as supported by prior studies [34], [35]. The classification results improved to more than 80% after cleaning. This confirms that the raw text carries a lot of noise and a lot of industry terms [32], which caused the model to perform poorly before cleaning. After semantic enrichment, macro F1 scores improved by up to 10 ppts on average. This supports the hypothesis that obtaining auxiliary semantic information from labels [61] is beneficial for reducing data sparsity and improving text classification performance in general. The findings also support BERT as the best performer, which is consistent with previous research [96], [105], [106].

BERT classifier results also demonstrated a gradual improvement in scores from 100,000 to 300,000 training datasets for both accuracy and macro-F1 scores, demonstrating the importance of large datasets for deep learning [73]. An intriguing discovery was that WordNet synonym augmentation was the best augmenter for short text, outperforming GloVe and BERT augmentations, which were rumored to perform better in NLP tasks. The unexpected BERT-WordNet combination also contradicted Wei and Zou's prediction that EDA would not improve the performance of pre-trained models like BERT [63]. The objectives of the study were met. To begin, the findings indicate that semantic enrichment improves invoice classification performance. Second, the results indicate that expanding it to a larger training set improves invoice classification performance. Third, the findings indicate that WordNet synonym augmentation is the best data augmentation approach for reducing class imbalance

in short text scenarios. Fourth, BERT is the most effective classifier in terms of invoice classification performance.

One notable finding was that short text classification performance is affected by what was done during the preprocessing stage. If the text was properly preprocessed and the semantic information was preserved, the classifier can perform well, as in the case of invoice text preprocessing. The original invoice text was noisy, with many non-English-named entities that pre-trained models like BERT struggled to handle. However, keeping only the English words, which Bert excels at, significantly improved BERT's performance.

Another discovery was that the method for dealing with short text differs from the method for dealing with long text. Using a contextual word embedding augmentation when there isn't much context in the original text, according to the findings, may backfire, with the augmentation overcompensating and providing augmented texts that are semantically different from the original text. In this case, a more conservative approach to augmentation, such as simple synonym substitution, may be sufficient and necessary to preserve the semantic information within the text, resulting in state-of-the-art performance. This research aims to produce a novel approach to invoice text classification with practical application for business organizations, and the objective has been met. The implications of success in invoice classification may be helpful to replicate in actual business practice, particularly within the accounting departments. This model can also be integrated with other solutions such as RPA and OCR where OCR reads the invoice text from the source document, and the RPA updates the ERP system while the model predicts the account category from the invoice text. The value of this study, if it is easily applied to invoice classification, will help organizations reduce overhead costs for repetitive and mundane accounting tasks. The usage of this method could potentially be extended for usage in other text classification use cases within the finance and accounting domain.

VI. LIMITATION AND FUTURE WORK

In this study, only one traditional learner was used to compare against deep learning models. It is recommended that more traditional learners be compared against the deep learning models for more concrete findings on the question of the effectiveness of traditional versus deep learning models. Furthermore, due to the huge 2,346 total combination evaluations of 69 multi-class conversions into binary class for the Area Under Curve (AUC) evaluation, we will explore this complex AUC evaluation in future work. In terms of the language supported, the novel approach only covers the scope of the English language given the retention of English words during the data cleaning stage, English-based augmentation, and English-based pre-training language model for the BERT classification model. As such, it is recommended for future works to cover a wider linguistic scope using the same process with a different language variation.

REFERENCES

[1] R. Sharda, D. Delen, E. Turban, J. E. Aronson, T.-P. Liang, and D. King, *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*, 4th ed. Harlow, U.K.: Pearson, 2018.

[2] B. Marr. (2019). *What is Unstructured Data and Why is it so Important to Businesses? An Easy Explanation for Anyone*. Forbes, Jersey City, NJ, USA. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2019/10/16/what-is-unstructured-data-and-why-is-it-so-important-to-businesses-an-easy-explanation-for-anyone/?sh=242a95b615f6

[3] M. Treacy and F. Wiersema. (Jan. 1993). *Customer Intimacy and Other Value Disciplines*. Harvard Bus., Brighton, MA, USA. [Online]. Available: https://hbr.org/1993/01/customer-intimacy-and-other-value-disciplines

[4] G. Salton, "The past thirty years in information retrieval," *J. Amer. Soc. Inf. Sci.*, vol. 38, no. 5, pp. 375–380, Sep. 1987, doi: 10.1002/(SICI)1097-4571(198709)38:5<375::AID-ASI5>3.0.CO;2-3.

[5] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002, doi: 10.1145/505282.505283.

[6] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM J. Res. Develop.*, vol. 1, no. 4, pp. 309–317, Oct. 1957, doi: 10.1147/rd.14.0309.

[7] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, Apr. 1958, doi: 10.1147/rd.22.0159.

[8] P. B. Baxendale, "Machine-made index for technical literature—An experiment," *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 354–361, Oct. 1958, doi: 10.1147/rd.24.0354.

[9] M. E. Maron and J. L. Kuhns, "On relevance, probabilistic indexing and information retrieval," *J. ACM*, vol. 7, no. 3, pp. 216–244, Jul. 1960, doi: 10.1145/321033.321035.

[10] L. Guo, F. Shi, and J. Tu, "Textual analysis and machine leaning: Crack unstructured data in finance and accounting," *J. Finance Data Sci.*, vol. 2, no. 3, pp. 153–170, Sep. 2016, doi: 10.1016/j.jfds.2017.02.001.

[11] J. Weizenbaum, "ELIZA—A computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.

[12] J. Barnett, K. Knight, I. Mani, and E. Rich, "Knowledge and natural language processing," *Commun. ACM*, vol. 33, no. 8, pp. 50–71, Aug. 1990, doi: 10.1145/79173.79177.

[13] B. Marr. (Sep. 11, 2020). *4 Simple Ways Businesses Can Use Natural Language Processing*. Forbes, Jersey City, NJ, USA. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2020/09/11/4-simple-ways-businesses-can-use-natural-language-processing/?sh=2576804e3a5f

[14] J. Leggatt. (Mar. 22, 2023). *What Is ChatGPT? A Review Of The AI in Its Own Words*. Forbes, Jersey City, NJ, USA. [Online]. Available: https://www.forbes.com/advisor/business/software/what-is-chatgpt/

[15] A. Zhou. (Nov. 4, 2017). *EY, Deloitte and PWC Embrace Artificial Intelligence for Tax and Accounting*. Forbes, Jersey City, NJ, USA. [Online]. Available: https://www.forbes.com/sites/adelynzhou/2017/11/14/ey-deloitte-and-pwc-embrace-artificial-intelligence-for-tax-and-accounting/?sh=34921e7e3498

[16] Y. Zhang, F. Xiong, Y. Xie, X. Fan, and H. Gu, "The impact of artificial intelligence and blockchain on the accounting profession," *IEEE Access*, vol. 8, pp. 110461–110477, 2020, doi: 10.1109/ACCESS.2020.3000505.

[17] Deloitte. (2019). *Making Deals Successful: The Impact of Analytics in M&A and Value Creation*. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/finance/Summary%20Making%20Deals%20Successful.pdf

[18] L. Li, Y. Feng, Y. Lv, X. Cong, X. Fu, and J. Qi, "Automatically detecting peer-to-peer lending intermediary risk—Top management team profile textual features perspective," *IEEE Access*, vol. 7, pp. 72551–72560, 2019, doi: 10.1109/ACCESS.2019.2919727.

[19] D. Baviskar, S. Ahirrao, V. Potdar, and K. Kotecha, "Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions," *IEEE Access*, vol. 9, pp. 72894–72936, 2021, doi: 10.1109/ACCESS.2021.3072900.

[20] T. Korhonen, E. Selos, T. Laine, and P. Suomala, "Exploring the programmability of management accounting work for increasing automation: An interventionist case study," *Accounting, Auditing Accountability J.*, vol. 34, no. 2, pp. 253–280, Nov. 2020, doi: 10.1108/aaaj-12-2016-2809.

[21] B. Back, J. Toivonen, H. Vanharanta, and A. Visa, "Comparing numerical data and text information from annual reports using self-organizing maps," *Int. J. Accounting Inf. Syst.*, vol. 2, no. 4, pp. 249–269, Dec. 2001, doi: 10.1016/s1467-0895(01)00018-5.

[22] S. R. Das and M. Y. Chen, "Yahoo! For amazon: Sentiment extraction from small talk on the web," *Manage. Sci.*, vol. 53, no. 9, pp. 1375–1388, Sep. 2007, doi: 10.1287/mnsc.1070.0704.

[23] A. Kloptchenko, T. Eklund, J. Karlsson, B. Back, H. Vanharanta, and A. Visa, "Combining data and text mining techniques for analysing financial reports," *Intell. Syst. Accounting, Finance Manage.*, vol. 12, no. 1, pp. 29–41, Apr. 2004, doi: 10.1002/isaf.239.

[24] F. A. Amani and A. M. Fadlalla, "Data mining applications in accounting: A review of the literature and organizing framework," *Int. J. Accounting Inf. Syst.*, vol. 24, pp. 32–58, Feb. 2017, doi: 10.1016/j.accinf.2016.12.004.

[25] X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, and Q.-T. Ha, "A hidden topic-based framework toward building applications with short web documents," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 961–976, Jul. 2011, doi: 10.1109/TKDE.2010.27.

[26] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2011, doi: 10.1145/2063576.2063689.

[27] S. S. Samant, N. L. B. Murthy, and A. Malapati, "Improving term weighting schemes for short text classification in vector space model," *IEEE Access*, vol. 7, pp. 166578–166592, 2019, doi: 10.1109/ACCESS.2019.2953918.

[28] I. Alsmadi and G. K. Hoon, "Term weighting scheme for short-text classification: Twitter corpuses," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3819–3831, Jan. 2018, doi: 10.1007/s00521-017-3298-8.

[29] J. Wang, Y. Li, J. Shan, J. Bao, C. Zong, and L. Zhao, "Large-scale text classification using scope-based convolutional neural network: A deep learning approach," *IEEE Access*, vol. 7, pp. 171548–171558, 2019, doi: 10.1109/ACCESS.2019.2955924.

[30] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, Feb. 2015, doi: 10.1186/s40537-014-0007-7.

[31] G. G. Sahin, "To augment or not to augment? A comparative study on text augmentation techniques for low-resource NLP," *Comput. Linguistics*, vol. 48, no. 1, pp. 5–42, Apr. 2022, doi: 10.1162/coli_a_00425.

[32] S. García-Méndez, M. Fernández-Gavilanes, J. Juncal-Martínez, F. J. González-Castaño, and Ó. B. Seara, "Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus," *IEEE Access*, vol. 8, pp. 61642–61655, 2020, doi: 10.1109/ACCESS.2020.2983584.

[33] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006, doi: 10.1109/TKDE.2006.17.

[34] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Nov. 2011, doi: 10.1142/s0218001409007326.

[35] J. Ge, H. Chen, D. Zhang, X. Hou, and L. Yuan, "Active learning for imbalanced ordinal regression," *IEEE Access*, vol. 8, pp. 180608–180617, 2020, doi: 10.1109/ACCESS.2020.3027764.

[36] L. B. Letaifa and M. I. Torres, "Perceptual borderline for balancing multi-class spontaneous emotional data," *IEEE Access*, vol. 9, pp. 55939–55954, 2021, doi: 10.1109/ACCESS.2021.3071485.

[37] H. P. Edmundson and R. E. Wyllys, "Automatic abstracting and indexing—Survey and recommendations," *Commun. ACM*, vol. 4, no. 5, pp. 226–234, May 1961, doi: 10.1145/366532.366545.

[38] M. E. Maron, "Automatic indexing: An experimental inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, Jul. 1961, doi: 10.1145/321075.321084.

[39] F. B. Baker, "Information retrieval based upon latent class analysis," *J. ACM*, vol. 9, no. 4, pp. 512–521, Oct. 1962, doi: 10.1145/321138.321148.

[40] H. Borko and M. Bernick, "Automatic document classification," *J. ACM*, vol. 10, no. 2, pp. 151–162, Apr. 1963, doi: 10.1145/321160.321165.

[41] H. E. Stiles, "The association factor in information retrieval," *J. ACM*, vol. 8, no. 2, pp. 271–279, Apr. 1961, doi: 10.1145/321062.321074.

[42] G. Salton, "Associative document retrieval techniques using bibliographic information," *J. ACM*, vol. 10, no. 4, pp. 440–457, Oct. 1963, doi: 10.1145/321186.321188.

[43] G. Salton and M. E. Lesk, "The SMART automatic document retrieval systems—An illustration," *Commun. ACM*, vol. 8, no. 6, pp. 391–398, Jun. 1965, doi: 10.1145/364955.364990.

[44] P. A. W. Lewis, P. B. Baxendale, and J. L. Bennett, "Statistical discrimination of the synonymy/antonymy relationship between words," *J. ACM*, vol. 14, no. 1, pp. 20–44, Jan. 1967, doi: 10.1145/321371.321374.

[45] G. Salton and M. E. Lesk, "Computer evaluation of indexing and text processing," *J. ACM*, vol. 15, no. 1, pp. 8–36, Jan. 1968, doi: 10.1145/321439.321441.

[46] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969, doi: 10.1145/321510.321519.

[47] F. J. Damerau, "Automatic parsing for content analysis," *Commun. ACM*, vol. 13, no. 6, pp. 356–360, Jun. 1970, doi: 10.1145/362384.362495.

[48] R. Grishman, L. Hirschman, and N. T. Nhan, "Discovery procedures for sublanguage selectional patterns: Initial experiments," *Comput. Linguistics*, vol. 12, no. 3, pp. 205–215, Jul. 1986. [Online]. Available: https://aclanthology.org/J86-3002

[49] K. Dahlgren, J. Mcdowell, and E. P. Stabler, "Knowledge representation for commonsense reasoning with text," *Comput. Linguistics*, vol. 15, no. 3, pp. 149–170, Sep. 1989. [Online]. Available: https://aclanthology.org/J89-3002

[50] L. F. Rau, P. S. Jacobs, and U. Zernik, "Information extraction and text summarization using linguistic knowledge acquisition," *Inf. Process. Manage.*, vol. 25, no. 4, pp. 419–428, Jan. 1989, doi: 10.1016/0306-4573(89)90069-1.

[51] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd, "Cyc: Toward programs with common sense," *Commun. ACM*, vol. 33, no. 8, pp. 30–49, Aug. 1990, doi: 10.1145/79173.79176.

[52] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An on-line lexical database," *Int. J. Lexicography*, vol. 3, no. 4, pp. 235–244, 1990, doi: 10.1093/ijl/3.4.235.

[53] S. W. McRoy, "Using multiple knowledge sources for word sense discrimination," *Comput. Linguistics*, vol. 18, no. 1, pp. 1–30, Mar. 1992. [Online]. Available: https://aclanthology.org/J92-1001

[54] I. Dagan and A. Itai, "Word sense disambiguation using a second language monolingual corpus," *Comput. Linguistics*, vol. 20, no. 4, pp. 563–596, Dec. 1994. [Online]. Available: https://aclanthology.org/J94-4003

[55] G. Salton, "Recent studies in automatic text analysis and document retrieval," *J. ACM*, vol. 20, no. 2, pp. 258–278, Apr. 1973, doi: 10.1145/321752.321757.

[56] K. S. Jones, "Experiments in relevance weighting of search terms," *Inf. Process. Manag.*, vol. 15, no. 3, pp. 133–144, 1979, doi: 10.1016/0306-4573(79)90060-8.

[57] R. Attar and A. S. Fraenkel, "Local feedback in full-text retrieval systems," *J. ACM*, vol. 24, no. 3, pp. 397–417, 1977, doi: 10.1145/322017.322021.

[58] C. Vernimb, "Automatic query adjustment in document retrieval," *Inf. Process. Manag.*, vol. 13, no. 6, pp. 339–353, 1977, doi: 10.1016/0306-4573(77)90054-1.

[59] J. Xu and W. B. Croft, "Improving the effectiveness of information retrieval with local context analysis," *ACM Trans. Inf. Syst.*, vol. 18, no. 1, pp. 79–112, Jan. 2000, doi: 10.1145/333135.333138.

[60] Y. S. Mehanna and M. B. Mahmuddin, "A semantic conceptualization using tagged bag-of-concepts for sentiment analysis," *IEEE Access*, vol. 9, pp. 118736–118756, 2021, doi: 10.1109/ACCESS.2021.3107237.

[61] Y. Dong, P. Liu, Z. Zhu, Q. Wang, and Q. Zhang, "A fusion model-based label embedding and self-interaction attention for text classification," *IEEE Access*, vol. 8, pp. 30548–30559, 2020, doi: 10.1109/ACCESS.2019.2954985.

[62] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets," *Soft Comput.*, vol. 13, no. 3, pp. 213–225, May 2008, doi: 10.1007/s00500-008-0319-7.

[63] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," presented at the 9th Int. Joint Conf. Natural Lang. Process., Nov. 2019, doi: 10.18653/v1/D19-1670.

[64] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *J. Big Data*, vol. 8, no. 1, Jul. 2021, doi: 10.1186/s40537-021-00492-0.

[65] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[66] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, Jan. 2004, doi: 10.1111/j.0824-7935.2004.t01-1-00228.x.

[67] L. Cai, Y. Song, T. Liu, and K. Zhang, "A hybrid BERT model that incorporates label semantics via adjustive attention for multi-label text classification," *IEEE Access*, vol. 8, pp. 152183–152192, 2020, doi: 10.1109/ACCESS.2020.3017382.

[68] H. Huan, J. Yan, Y. Xie, Y. Chen, P. Li, and R. Zhu, "Feature-enhanced nonequilibrium bidirectional long short-term memory model for Chinese text classification," *IEEE Access*, vol. 8, pp. 199629–199637, 2020, doi: 10.1109/ACCESS.2020.3035669.

[69] J.-S. Lee and J. Hsiang, "Patent classification by fine-tuning BERT language model," *World Pat. Inf.*, vol. 61, Jun. 2020, Art. no. 101965, doi: 10.1016/j.wpi.2020.101965.

[70] S. Gao, M. Alawad, M. T. Young, J. Gounley, N. Schaefferkoetter, H. J. Yoon, X.-C. Wu, E. B. Durbin, J. Doherty, A. Stroup, L. Coyle, and G. Tourassi, "Limitations of transformers on clinical text classification," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 9, pp. 3596–3607, Sep. 2021, doi: 10.1109/JBHI.2021.3062322.

[71] M. Bayer, M.-A. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and C. Reuter, "Data augmentation in natural language processing: A novel text generation approach for long and short text classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 1, pp. 135–150, Apr. 2022, doi: 10.1007/s13042-022-01553-3.

[72] H.-T. Duong and T.-A. Nguyen-Thi, "A review: Preprocessing techniques and data augmentation for sentiment analysis," *Comput. Social Netw.*, vol. 8, no. 1, Jan. 2021, doi: 10.1186/s40649-020-00080-x.

[73] J. Luo, M. Bouazizi, and T. Ohtsuki, "Data augmentation for sentiment analysis using sentence compression-based SeqGAN with data screening," *IEEE Access*, vol. 9, pp. 99922–99931, 2021, doi: 10.1109/ACCESS.2021.3094023.

[74] S. Lee, L. Liu, and W. Choi, "Iterative translation-based data augmentation method for text classification tasks," *IEEE Access*, vol. 9, pp. 160437–160445, 2021, doi: 10.1109/ACCESS.2021.3131446.

[75] E. Ma. (Apr. 12, 2019). *Data Augmentation in NLP*. Towards Data Sci. [Online]. Available: https://towardsdatascience.com/data-augmentation-in-nlp-2801a34dfc28

[76] C. Li, S. Feng, Q. Zeng, W. Ni, H. Zhao, and H. Duan, "Mining dynamics of research topics based on the combined LDA and WordNet," *IEEE Access*, vol. 7, pp. 6386–6399, 2019, doi: 10.1109/ACCESS.2018.2887314.

[77] F. Li, L. Liao, L. Zhang, X. Zhu, B. Zhang, and Z. Wang, "An efficient approach for measuring semantic similarity combining WordNet and Wikipedia," *IEEE Access*, vol. 8, pp. 184318–184338, 2020, doi: 10.1109/ACCESS.2020.3025611.

[78] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: 10.1145/219717.219748.

[79] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," presented at the EMNLP, Oct. 2014, doi: 10.3115/v1/D14-1162.

[80] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, "Sentiment analysis with ensemble hybrid deep learning model," *IEEE Access*, vol. 10, pp. 103694–103704, 2022, doi: 10.1109/ACCESS.2022.3210182.

[81] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[82] B. Ramesh and J. G. R. Sathiaseelan, "An advanced multi class instance selection based support vector machine for text classification," *Proc. Comput. Sci.*, vol. 57, pp. 1124–1130, Jan. 2015, doi: 10.1016/j.procs.2015.07.400.

[83] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[84] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996, doi: 10.1017/CBO9780511812651.

[85] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[86] D. C. Plaut and G. E. Hinton, "Learning sets of filters using back-propagation," *Comput. Speech Lang.*, vol. 2, no. 1, pp. 35–61, Mar. 1987, doi: 10.1016/0885-2308(87)90026-x.

[87] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," *Phys. Rev. Lett.*, vol. 59, no. 19, pp. 2229–2232, Nov. 1987, doi: 10.1103/physrevlett.59.2229.

[88] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Netw.*, vol. 1, no. 4, pp. 339–356, Jan. 1988, doi: 10.1016/0893-6080(88)90007-x.

[89] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer-Verlag, 1995.

[90] *Frontiers in Massive Data Analysis*, Nat. Res. Council, Nat. Academies Press, Washington, DC, USA, 2013, doi: 10.17226/18374.

[91] V. Balakrishnan, Z. Shi, C. L. Law, R. Lim, L. L. Teh, and Y. Fan, "A deep learning approach in predicting products' sentiment ratings: A comparative analysis," *J. Supercomput.*, vol. 78, no. 5, pp. 7206–7226, Nov. 2021, doi: 10.1007/s11227-021-04169-6.

[92] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999, doi: 10.1109/72.788645.

[93] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?" *Mach. Learn.*, vol. 46, no. 1, pp. 423–444, Jan. 2002, doi: 10.1023/a:1012491419635.

[94] A. Sun, E.-P. Lim, and Y. Liu, "On strategies for imbalanced text classification using SVM: A comparative study," *Decis. Support Syst.*, vol. 48, no. 1, pp. 191–201, Dec. 2009, doi: 10.1016/j.dss.2009.07.011.

[95] C. Wan, Y. Wang, Y. Liu, J. Ji, and G. Feng, "Composite feature extraction and selection for text classification," *IEEE Access*, vol. 7, pp. 35208–35219, 2019, doi: 10.1109/ACCESS.2019.2904602.

[96] F.-Z. El-Alami, S. O. El Alaoui, and N. En Nahnahi, "Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8422–8428, Nov. 2022, doi: 10.1016/j.jksuci.2021.02.005.

[97] R. Xiang, E. Chersoni, Q. Lu, C. Huang, W. Li, and Y. Long, "Lexical data augmentation for sentiment analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 11, pp. 1432–1447, Jun. 2021, doi: 10.1002/asi.24493.

[98] T. Zhang and F. J. Oles, "Text categorization based on regularized linear classification methods," *Inf. Retr.*, vol. 4, no. 1, pp. 5–31, Apr. 2001, doi: 10.1023/a:1011441423217.

[99] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004. [Online]. Available: https://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf

[100] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[101] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[102] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015, doi: 10.1109/TASLP.2015.2400218.

[103] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005, doi: 10.1016/j.neunet.2005.06.042.

[104] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019, doi: 10.1109/ACCESS.2019.2909919.

[105] J. J. Bird, A. Ekárt, and D. R. Faria, "Chatbot interaction with artificial intelligence: Human data augmentation with T5 and language transformer ensemble for text classification," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 18, pp. 3129–3144, Aug. 2021, doi: 10.1007/s12652-021-03439-8.

[106] L. Khan, A. Amjad, N. Ashraf, and H. T. Chang, "Multi-class sentiment analysis of Urdu text using multilingual BERT," *Sci. Rep.*, vol. 12, no. 1, p. 5436, Mar. 2022, doi: 10.1038/s41598-022-09381-9.

[107] X. Gong, W. Ying, S. Zhong, and S. Gong, "Text sentiment analysis based on transformer and augmentation," *Frontiers Psychol.*, vol. 13, May 2022, doi: 10.3389/fpsyg.2022.906061.
[108] D. Chi. (2023). *Semantic Enrichment, Data Augmentation and Deep Learning for Boosting Invoice Text Classification Performance: A Novel Natural Language Processing Strategy*. [Online]. Available: https://github.com/cwwdaniel/invoice-text-classification

WEI WEN CHI was born in Selangor, Malaysia, in 1990. He received the B.Com. degree in accounting from University Tunku Abdul Rahman (UTAR), Malaysia, in 2011, and the master's degree in business analytics from Sunway University, Malaysia, in 2023.

He was the Head of Operational Excellence at Zurich Insurance, Malaysia, specializing in digitalization, simplification, and automation of insurance processes and project delivery. He is currently a Senior Consultant with Deloitte Southeast Asia. His research interests include deep learning, natural language processing, machine learning, and their practical applications to business processes.

TIONG YEW TANG (Member, IEEE) received the bachelor's degree (Hons.) in information system engineering from University Tunku Abdul Rahman (UTAR), in 2006, and the Ph.D. degree in information technology from Monash University, in 2016. He is an inventor and author of the Patent Cooperation Treaty (PCT) patent. He is actively engaging with research projects on artificial intelligence in the business research domain. His research interests include bibliometric analysis, artificial intelligence, deep learning, robotic process automation, business intelligence, big data analytics, and computational intelligence. He is a member of the research committee at Sunway University, Malaysia. He was a recipient of the Book Price Award, in 2006, and the Dean's List Award, in 2004 and 2005, for his academic achievements in UTAR. He was also a recipient of the PVC's Award for Excellence in Research, Teaching, and Administration: Round 2, Monash University, in 2011.

NARISHAH MOHAMED SALLEH received the B.Sc. degree in electrical and electronics engineering from the University of Missouri, Columbia, MO, USA, in 1996, the master's degree in manufacturing system engineering, in 2013, and the Ph.D. degree in software engineering from National University Malaysia (UKM), in 2020, with a focus on requirement engineering for software development. She has extensive experience and knowledge in data engineering, data science, and system development from manufacturing, supply chain management and logistics, accounting and finance, maintenance and inventory management, and material science domains which led to the development and deployments of many ERP, the IoT, big data, RPA, and Industrial 4.0 based on projects to improve organization efficiency and enhance the automation process.

MUAADH MUKRED received the bachelor's degree in computer science from Al-Mustansiriyah University, in 2002, the M.Sc. degree in computer science from the University of Technology Malaysia, in 2011, and the Ph.D. degree from UKM. He is currently a Lecturer with the Department of Business Analytics, Sunway Business School, Sunway University. He is also an Associate Fellow with the Cyber Security Center, Faculty of Information Science and Technology, University Kebangsaan Malaysia (UKM), Malaysia, where he has been a Postdoctoral Researcher. He has worked for more than four years as a Software Developer with the Al-Noor Foundation, Putrajaya. Previously, he was a Lecturer with the Computer Science Department, Sana'a Community College, for more than eight years. He has joined the industry for five years, working in computer science, data analysis, and programming. His permanent position at Sana'a Community College. Then, he moved to Malaysia after getting a scholarship. In 2011, he joined SCC, where he started a new position as the Head of the Higher Professional Education Division and served as a Lecturer. In 2013, he got another scholarship. He was involved in some research grant projects, some in artificial intelligence and machine learning, and others in analyzing and architecting big data. He has recently been involved as an academic partner with some research projects funded by some universities in Malaysia and Middle East. He has supervised some Ph.D. and master's students and some other undergraduate final-year projects. He has published several scientific/research papers in well-known international journals and conferences. His research interests include big data analytics, artificial intelligence, natural language processing, technology adoption, and database architecture. He received the Outstanding Researcher Award as one of the best students in his batch. He has also been awarded an outstanding publication. He has also been invited as an external examiner for many Ph.D. and master's students and served as a reviewer for many high-impact journals.

HUSSAIN ALSALMAN received the B.Sc. and M.Sc. degrees in computer science from King Saud University (KSU), Riyadh, Saudi Arabia, and the Ph.D. degree in artificial intelligence, U.K. From 2009 to 2014, he chaired the Department of Computer Science, College of Computer and Information Sciences, KSU. He worked for several years as a Consultant for several companies in the private sector and institutes in the government sector in Saudi Arabia. He is currently a Staff Member with the Department of Computer Science, KSU. His main research interests include medical image processing, machine learning algorithms, neural networks, computational methods for health care monitoring, and ensemble and deep learning models for medical analysis and diagnosis. He was a member of the review board of the *Journal of King Saud University-Computer and Information Sciences*, from 2004 to 2014.

MUHAMMAD ZOHAIB is currently a Teacher with the Software Engineering Department, Lappeenranta-Lahti University of Technology, Lappeenranta, Finland. He taught several courses, such as software quality in software development and software development skills. His main research interests include software engineering, machine learning algorithms, computational methods for health care monitoring, ensemble, and deep learning models for medical image analysis.