

ISO 9001:2015 Quality Management System Requirements and Audit Findings Classification Using Support Vector Machine and Long Short-Term Memory Neural Network: An Optimization Method

Ralph Sherwin A. Corpuz
Electronics Engineering Technology
Technological University of the Philippines
Ermita, Manila 1000 Philippines
ralphsherwin_corpuz@tup.edu.ph

Date received: August 22, 2020

Revision accepted: March 8, 2021

Abstract

Generating accurate and timely internal and external audit reports may seem difficult for some auditors due to limited time or expertise in matching the correct clauses of the standard with the textual statement of findings. To overcome this gap, this paper presents the design of text classification models using support vector machine (SVM) and long short-term memory (LSTM) neural network in order to automatically classify audit findings and standard requirements according to text patterns. Specifically, the study explored the optimization of datasets, holdout percentage and vocabulary of learned words called NumWords, then analyzed their capability to predict training accuracy and timeliness performance of the proposed text classification models. The study found that SVM (96.74%) and LSTM (97.54%) were at par with each other in terms of the best training accuracy, although SVM (67.96 ± 17.93 seconds [s]) was found to be significantly faster than LSTM (136.67 ± 96.42 s) in any dataset size. The study proposed optimization formulas that highlight dataset and holdout as predictors of accuracy, while dataset and NumWords as predictors of timeliness. In terms of actual implementation, both classification models were able to accurately classify 20 out of 20 sample audit findings at 1 and 3 s, respectively. Hence, the extent of choosing between the two algorithms depend on the datasets size, learned words, holdout percentage, and workstation speed. This paper is part of a series, which explores the use of Artificial Intelligence (AI) techniques in optimizing the performance of QMS in the context of a state university.

Keywords: ISO 9001:2015, long short-term memory, optimization, support vector machine, text classification

1. Introduction

A quality management system (QMS) is considered a fundamental element of an organization. It is comprised of activities used to define the objectives and processes of an organization in order to achieve desired results. In a QMS, organizations are required to consistently meet legal obligations, customer requirements, internal policies and relevant international standards; hence, they are bound to ensure customer satisfaction through the delivery of quality products and services (International Organization for Standardization [ISO], 2015a). The ISO 9001:2015 is the most popular standard published by the ISO with over one million certified companies and organizations in over 170 countries. The standard prescribes the adoption of a QMS, which is applicable to any type of business (ISO, 2020). Many organizations obtain ISO 9001:2015 certifications in order to improve their branding, synonymous to having a quality and customer-centric reputation (Poksinska *et al.*, 2002). In the Philippines, by virtue of Executive Order 605 series of 2007, numerous national and local government agencies, including state universities and colleges (SUC), have been pursuing certifications with the intent of improving their performances and essentially serve the general public better (Philippine Government Official Gazette, 2020). As of 2017, there are 361 certificates issued into various national and local government agencies in the country (Government Quality Management Committee, 2020).

The ISO 9001:2015 standard is composed of 10 clauses and 81 sub-clauses. These 91 total clauses and sub-clauses elaborate the requirements of the standard and serve as reference in planning, operating, evaluating and improving a QMS, except for clauses 1, 2 and 3, which are not auditable. Table 1 elaborates the clauses and sub-clauses of the ISO 9001:2015 standard.

Organizations seeking for ISO 9001 certifications are required to undergo series of audits to evaluate their level of conformance. An audit is defined as a systematic, independent and documented process of evaluating conformance with audit criteria, which are composed of, but not limited to, policies, procedures, work instructions or related documented information including compliance with legal and contractual obligations (ISO, 2015b). An audit can be categorized as internal and external.

Table 1. ISO 9001:2015 clauses and sub-clauses

Major Clause	4.0	5.0	6.0	7.0	8.0	9.0	10
Title	Context of the Organization	Leadership	Planning	Resources	Operation	Performance Evaluation	Improvement
Sub-Clause	4.1, 4.2, 4.3,	5.1, 5.1.1,	6.1, 6.1.1,	7.1, 7.1.1, 7.1.2,	8.1, 8.2, 8.2.1, 8.2.2,	9.1, 9.1.1,	10.1, 10.2,
	4.4, 4.4.1,	5.1.2, 5.2,	6.1.2, 6.2,	7.1.3, 7.1.4,	8.2.3, 8.2.3.1, 8.2.3.2,	9.1.2, 9.2,	10.2.1, 10.2.2,
	4.4.2	5.2.1, 5.2.2,	6.2.1,	7.1.5, 7.1.5.1,	8.2.4, 8.3, 8.3.1,	9.2.1, 9.2.2,	10.3
		5.3	6.2.2, 6.3	7.1.5.2, 7.1.6,	8.3.2, 8.3.3, 8.3.4,	9.3, 9.3.1,	
				7.2, 7.3, 7.4, 7.5,	8.3.5, 8.3.6, 8.4,	9.3.2, 9.3.3	
				7.5.1, 7.5.2,	8.4.1, 8.4.2, 8.4.3,		
				7.5.3, 7.5.3.1,	8.5, 8.5.1, 8.5.2,		
				7.5.3.2	8.5.3, 8.5.4, 8.5.5,		
					8.5.6, 8.6, 8.7, 8.7.1,		
					8.7.2		

An internal audit is conducted by the organization through its independent pool of internal auditors (IA), while external audits can be further classified as second-party or an audit conducted by or to parties that have interest in the organization such as external providers or suppliers; and the other one is called third-party audit conducted by independent certifying bodies (CB). Both first and third-party audits are required in QMS certification while second-party audit is optional (ISO, 2018).

To document the results of audits, auditors prepare and submit audit reports for review and decision of the top management. An audit report is one of the documented information required by the standard to be retained as an evidence of implementation of the audit program (ISO, 2015a, 2018). Table 2 shows a sample template used to document an audit report. Unfortunately, not all audit reports generated are accurately perfect.

Table 2. A sample flawed ISO 9001 audit report

Process	Findings	Clause	Title
Human Resource Management (HRM)	The standard requires that the organization shall provide competent persons to man the operation of the office as per Qualification Standard, however, during the time of audit, it was found that the clerk assigned has no eligibility records to work in the government sector. (Major Nonconformity; NC)	7.1.2/ 7.2	People/ Competence
Planning	The standard requires that quality objectives shall be communicated to relevant functions, however, during the time of audit, the process owner was not aware of the delegated quality objectives. (Minor Nonconformity; MiN)	6.2/ 7.3	Quality Objectives/ Awareness

As observed, there is an issue of ambiguity on the findings, which can be interpreted with multiple clauses of the standard. Inaccurately generated audit report, such as this, can lead to inaccurate analysis of root causes, proposal of

ineffective corrective actions, delay in submission of audit reports, or misunderstanding between auditors and auditees. Such issue is common among neophyte auditors who seem confused with the standard requirements and their equivalent clauses or whenever they have time constraints in generating audit reports. Of course, an accurate and concise audit report should have a clear statement of findings with only one applicable clause of the standard.

Fortunately, the emergence of artificial intelligence (AI), then later, machine learning (ML)- and deep learning (DL)-based text classification techniques has shown positive results in resolving similar issues. Text classification is a process of predicting natural language texts based on specific features (Sebastiani, 2002). In this method, texts are converted into their equivalent numerical vectors through preprocessing, feature engineering and modelling, using various ML or DL algorithms. The resulting model then automatically finds equivalent classes from large datasets; hence, text classification is considered an essential tool in reducing time and efforts of manual document classifications (Qing *et al.*, 2019).

Text classification is widely used in various applications such as news categorization, sentiment analysis, predictive maintenance, information and labelling among others (Cai *et al.*, 2018). However, there seems a limited number of scientific papers relative to the use of AI-based text classification techniques specifically for ISO 9001 audit-related applications. One related study is that of Shirata and Sakagai (2008), wherein they used classification and regression tree (CART) to classify Japanese companies on their tendency to become bankrupt based on keywords found in financial audit reports. Another study is the design of the human factors/ergonomics maintenance audit classification framework used to detect aviation-related risks according to text patterns of safety audit findings (Hsiao *et al.*, 2013). While their proposed text classification models are found to be effective, the authors recommended to use more input data to establish deep learning and to conduct further testing to evaluate the robustness of the said models.

In a more closely related study, Tarnate *et al.* (2020) evaluated different types of recurrent neural networks (RNN) in classifying text-based ISO 9001 audit reports based on major clauses. They compared the effects of word encoding, word embedding and word encoding-embedding methods to optimize the learning performance of the classification model. The results found that bi-directional long short-term memory (Bi-LSTM) outperformed its sibling

LSTM using combined word encoding-embedding optimization technique. However, their study did not elaborate any causal relationship between word-encoding-embedding model and training classification accuracy. Meanwhile, a series of ISO 9001 related papers was published particularly on the use of text classification models for risk-based thinking (Corpuz, 2020), customer satisfaction analysis (Corpuz, 2021a, 2021b), and audit reports (Corpuz, 2019). Specifically, in one study, Corpuz (2019) used artificial neural network (ANN)-based scaled conjugate gradient (SCG) algorithm in classifying text based-ISO 9001:2015 audit reports. SCG has been found to be 95% accurate based on actual implementation. However, in this study, the performance of the SCG was not compared with other ML or DL algorithms; the factors that contributed to its classification accuracy was not determined. Moreover, in almost similar study, the author compared the performance of support vector machine (SVM) and long short-term memory (LSTM) neural network in classifying text-based feedback, suggestions, commendations, and complaints from customers. In here, SVM and LSTM algorithms were found to have almost similar training accuracy but SVM was faster than LSTM based on the data size used (Corpuz, 2021a).

Hence, this paper aimed to resolve the gaps of existing related studies. It specifically determined the better AI technique used to classify ISO 9001:2015 audit reports in timely and accurate fashion. It also investigated the predictors of training accuracy and timeliness performance using two of the most robust and popular AI algorithms – the SVM and LSTM neural network – through training, optimization and actual implementation methods.

2. Methodology

2.1 Datasets and Workflow

The study utilized a total of 2856 datasets, which were composed of internal and external audit reports coded through online spreadsheets, from the Technological University of the Philippines, Manila, Philippines, as of May 30, 2020, and the textual statements of ISO 9001:2015 standard requirements. Figure 1 shows the dataset used in this study. The “findings” are the textual descriptions of the nonconformity or the requirements of the standard, while the “classification” is the corresponding clause of each finding or requirement. Later in this section, the author further randomly grouped the dataset into 33.33% (1/3; 957) and 66.66% (1/2; 1904) of the total dataset to investigate

the effect of dataset size with training accuracy and timeliness performance of the proposed models. For each grouping, the dataset was restarted, and a holdout percentage (test data) was set to 10, 20, 30, 40 and 50% in order to ensure the validity of the training conducted.

Classification	Findings
Text	Text
10.3	Continual improvement The organization shall continually improve the suitability, adequacy and effectiveness of the quality management system in compliance to the Generally Accepted Accounting Principles (GAAP), the Government Accounting Standards Board (GASB), sub-systems should.
7.5.1	QMS requires that documented information shall be controlled to ensure it is available and suitable for use, where and when it is needed. The Act
7.5.3.1	The organization shall plan, implement and control the processes needed to meet the requirement and to implement the actions by establishing c
8.1	In conformity to the standard, the organization shall control the unique identification of the outputs when traceability is a requirement, and shall
8.5.2	According to the standards, the organization shall analyse and evaluate appropriate data and information arising from monitoring and measurem
9.1.3	According to the standard, the organization shall where applicable, take actions to acquire the necessary competence and evaluate the effectiven
7.2	The standard requires that the organization shall implement production and service provision under controlled conditions to include the validatio
8.5.1	The organization shall determine the processes needed for the QMS and their application throughout the organization and shall determine the inp
4.4	No Action plan to achieve quality objectives when they are not met
6.2	There was no documented information available during that time of visit for validation.
7.2	Information is cascaded or communicated verbally.
7.4	Upon validation from the QA office, the documented issues were submitted by the process office.
7.3	The Quality objectives shall be monitored. When asked about their monitoring system in terms of detailed office targets aside from their quality r
8.2.1	No QMS internal audit results for Q3 FY 2018
8.2.2	No Ss technical audit results for Q3 FY 2018
8.1	No Ss technical audit results for Q3 FY 2018

Figure 1. Sample dataset

Figure 2 illustrates the AI-based workflow conducted to realize the objectives of this study. To facilitate the modelling activities, the study utilized MATLAB computing software for training and testing, while the Statistical Package for the Social Sciences (SPSS) (version 27) was used for statistical analysis. The computer utilized in this study has specifications of 2.5 gigahertz (Intel Core i5) and 8 gigabytes, 1600 megahertz double data rate 3 synchronous dynamic random-access memory.

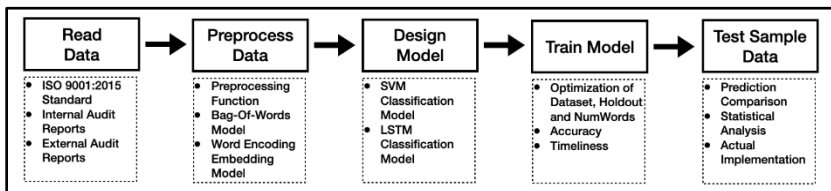


Figure 2. AI workflow

2.2 Data Processing

Considering the complexity of the text-based dataset used in this study, preprocessing procedures prior to modeling were conducted. Figure 3 shows the preprocessing function used for this purpose. Each text was converted into tokenized document or collection of words called tokens. To get rid of unwanted noise, the common stop-words were removed, then the texts were lemmatized into their root-word forms; punctuations and special characters were removed; and HTML or XML tags and those words with more than 15

and less than two characters were embedded to extract only the most helpful features.

```

1  function documents = preprocessText(textData)
2
3  % Tokenize the text.
4  documents = tokenizedDocument(textData);
5
6  % Remove a list of stop words then lemmatize the words. To improve
7  % lemmatization, first use addPartOfSpeechDetails.
8  documents = addPartOfSpeechDetails(documents);
9  documents = removeStopWords(documents);
10 documents = normalizeWords(documents, 'Style', 'lemma');
11
12 % Erase punctuation.
13 documents = erasePunctuation(documents);
14
15 % Remove words with 2 or fewer characters, and words with 15 or more
16 % characters.
17 documents = removeShortWords(documents, 2);
18 documents = removeLongWords(documents, 15);
19
20 % Convert to lowercase.
21 documents = lower(documents);
22
23 end

```

Figure 3. Preprocessing function

Figure 4 shows a sample screenshot of the tokenized documents as a result of preprocessing conducted to the dataset. In this particular example, out of 952 datasets, there were 857 tokens generated. Meanwhile, Table 3 details out the actual tokens generated in every dataset according to holdout percentage or the ratio of validation data as against the training data. As observed, the more holdout percentage is allocated, the lesser the tokens are generated.

```

857x1 tokenizedDocument:

  2 tokens: context organization
70 tokens: understand organization context organization shall determine ext
46 tokens: understand need expectation interested party due effect potentia
91 tokens: determine scope quality management system organization shall det
  4 tokens: quality management system process
87 tokens: organization shall establish implement maintain continually imp
  2 tokens: leadership commitment
107 tokens: general top management shall demonstrate leadership commitment
38 tokens: customer focus top management shall demonstrate leadership comm
19 tokens: communicate quality policy quality policy shall available mainta
64 tokens: organizational role responsibility authority top management sha
  4 tokens: action address risk opportunity
79 tokens: organization shall plan action address risk opportunity integrat
  4 tokens: quality objective planning achieve
31 tokens: planning change organization determine need change quality manag

```

Figure 4. Sample tokenized documents

Table 3. Tokenized document results

Dataset	Holdout (%)	Tokenized Documents
2856 (100%)	10	2571
	20	2285
	30	2000
	40	1714
	50	1428
1904 (66.67%)	10	1714
	20	1524
	30	1333
	40	1143
	50	952
952 (33.33%)	10	857
	20	762
	30	667
	40	572
	50	476

Figure 5 exhibits the word clouds as representations of the dataset before and after the preprocessing with a reduction rate of 75.68%. The bigger words mean that they appeared more frequently in the analysis.

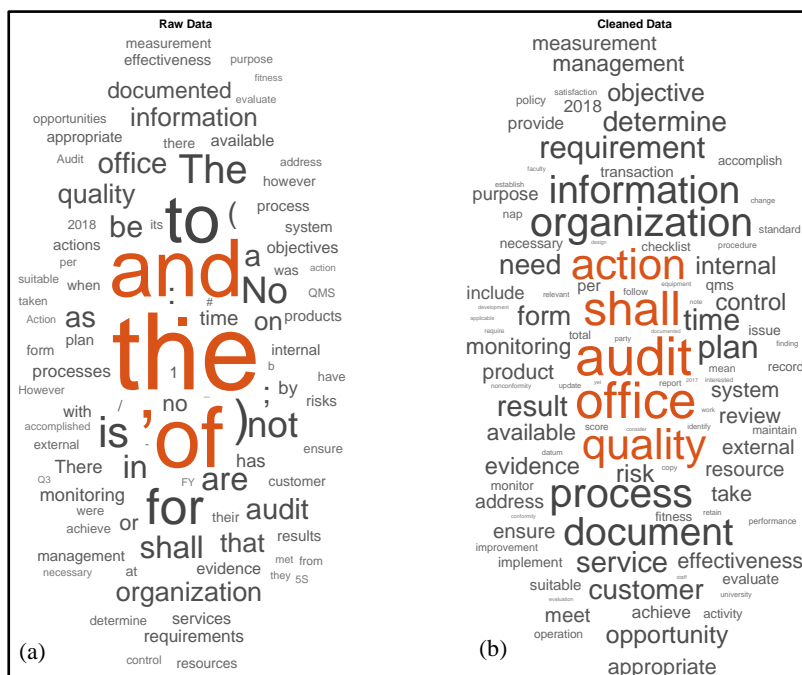


Figure 5. Word cloud before (a) and after (b) preprocessing

After the preprocessing, the tokenized documents were converted into numerical vectors. This was carried out by utilizing two types of text analysis models in consideration of the compatibility requirements of the MATLAB software. Initially, the author used bag-of-words (BOW) model for the SVM classification model. The BOW model records the number of times a word has appeared in a particular document; hence, it is called term-frequency counter. Figure 6 shows the BOW function wherein out of 857 tokens or initially scored words, only 745 *NumWords* or meaningful vocabularies were recorded. The 745 *NumWords* were the results of normalization where in the redundant tokens were removed such as organization, understand, shall, customer and determine among others. Also, listed in the figure are the top 20 most frequently-recorded words by the BOW model.

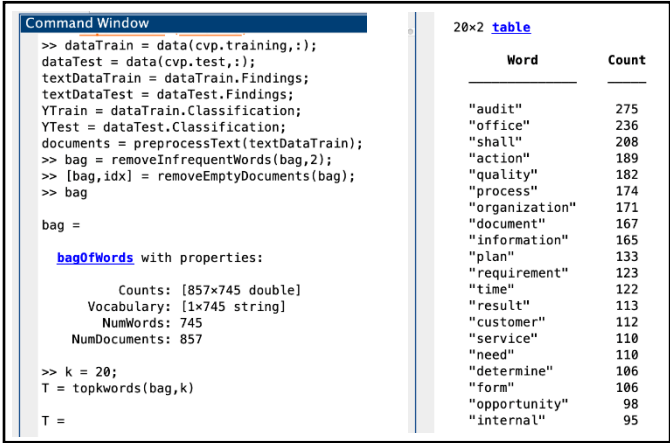


Figure 6. BOW model function and results

The second text analysis model employed was the word encoding (WE) model intended for the LSTM. The WE model maps out words of tokenized documents into sequences of numeric indices. Figure 7 shows a screenshot of the WE function and its results, which converted the learned words into numerical indices generated by the `word2ind` function. As shown, the *NumWord* “context” and “organization” have been assigned with “3” and “2” indices, respectively, while “quality” “management” “system” “processes” have been assigned with “18”, “19”, “20” and “69” indices, respectively. This is an important feature engineering technique for an LSTM to address the vanishing gradient effects of multidimensional data, such as texts, to the neural network.

```

Command Window
>> dataTrain = data(training(cvp),:);
dataValidation = data(test(cvp),:);
textDataTrain = dataTrain.Findings;
textDataValidation = dataValidation.Findings;
YTrain = dataTrain.Classification;
YValidation = dataValidation.Classification;
documentsTrain = preprocessText_LSTM(textDataTrain);
documentsValidation = preprocessText_LSTM_validation(textDataValidation);
enc = wordEncoding(documentsTrain);
enc =
wordEncoding with properties:
    NumWords: 2045
    Vocabulary: [1x2045 string]
>> %from 2 tokens = context organization
words = ["context" "organization"];
idx = word2ind(enc,words)
idx =
     3     2
>> % to 5 numeric indices
>>
%from 4 tokens = quality management system proces
words = ["quality" "management" "system" "process"];
idx = word2ind(enc,words)
idx =
    18    19    20    69
>> %to 126 numeric indices
    
```

Figure 7. WE model function and results

After the WE modelling, the target lengths of the tokenized documents were set to “40” since majority of these documents have less than 40 tokens. After which, those documents with lesser than 40 were left-padded and the ones that are greater than it were truncated. This is a standard procedure for the LSTM neural network to ensure the validity of the input data for the classification model. Figure 8 shows the histogram of the tokenized documents used as basis in setting the target.

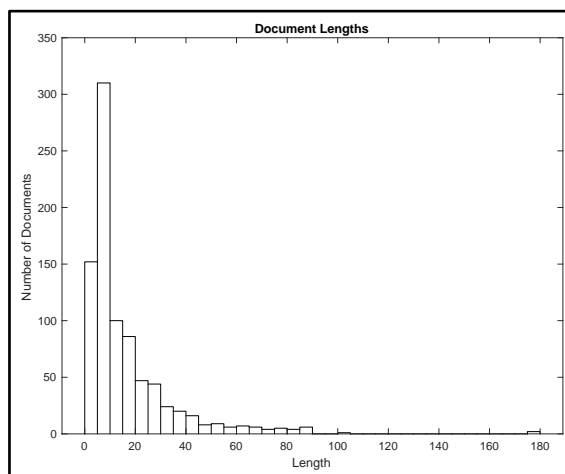


Figure 8. Tokenized documents histogram

Figure 9 exhibits the *doc2sequence* function used in padding and truncating the excess tokenized documents.

```
Command Window
>> documentLengths = doclength(documentsTrain);
figure
histogram(documentLengths)
title("Document Lengths")
xlabel("Length")
ylabel("Number of Documents")
sequenceLength = 40;
XTrain = doc2sequence(enc,documentsTrain,'Length',sequenceLength);
XTrain(1:5)
XValidation = doc2sequence(enc,documentsValidation,'Length',sequenceLength)
XValidation(1:5)

ans =

5x1 cell array

    {1x40 double}
    {1x40 double}
    {1x40 double}
    {1x40 double}
    {1x40 double}
```

Figure 9. *Doc2Sequence* function

2.3 SVM Classification Model

SVM, as introduced by Vapnik (1999), is a popular supervised machine learning technique owing to its strong theoretical foundation and powerful performance (Lessman and Vob, 2009; Coussement and Van den Poel, 2008). Illustrated in Figure 10, in its simplest form, is a SVM model used for binary classification.

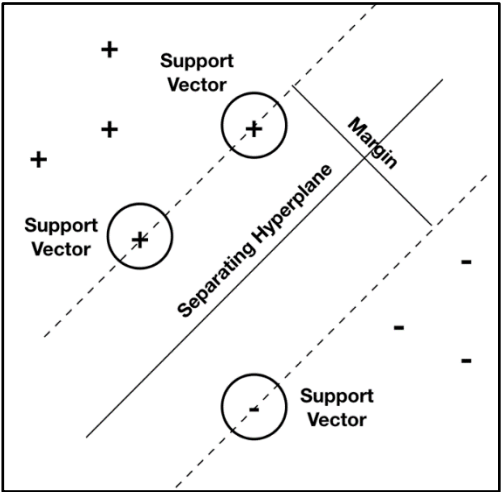


Figure 10. SVM model for binary classification

The SVM classifies data by finding the best hyperplane that separates all data points of one class from the other class. A hyperplane is considered “best” if it has the largest margin between two classes. Margin is the maximal width of the slab parallel to the hyperplane, which has no interior data points. Meanwhile, the data points that are closest to the separating hyperplane located in the boundary of the slab are called “support vectors”. The “+” in the figure indicates the data points for type 1 while “-” indicates the data points for type -1 (Hastie *et al.*, 2008; Christianini and Shawe-Taylor, 2000). Considering the 91 clauses and sub-clauses of the standard, the SVM classification model was designed using compact multiclass, error correcting output codes (ECOC) for SVM binary learners. Figure 11 shows the *CompactClassificationECOC* function used to create and train the SVM classification model. It utilized 4095 binary learners of “findings”, which served as “predictors” and 91 clauses or “classification” serving as “response”.

```

Command Window

>> YTrain(idx) = [];
XTrain = bag.Counts;
mdl = fitcecoc(XTrain,YTrain,'Learners','linear')

mdl =

classreg.learning.classif.CompactClassificationECOC
    ResponseName: 'Y'
    ClassNames: [1x91 categorical]
    ScoreTransform: 'none'
    BinaryLearners: {4095x1 cell}
    CodingMatrix: [91x4095 double]
    
```

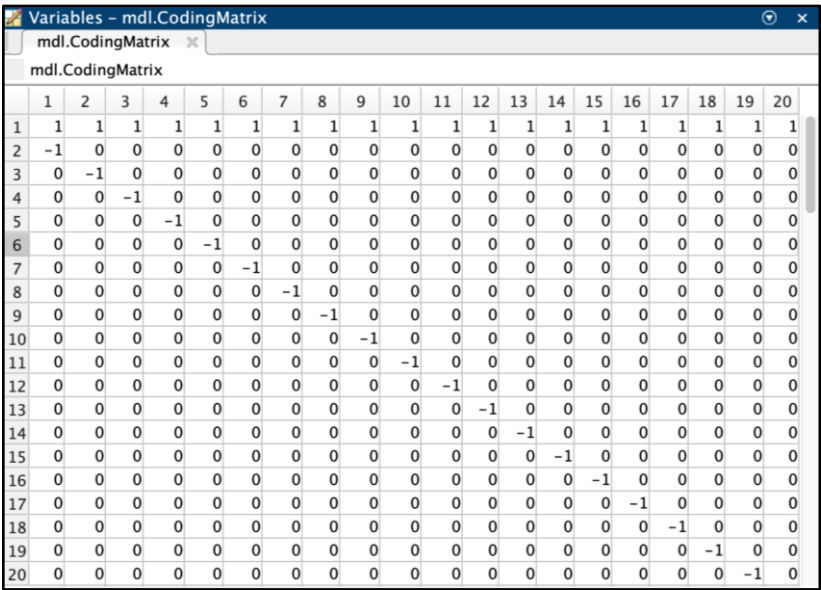
Figure 11. Compact multi class SVM function

The coding matrix of the SVM model is shown in Figure 12 with a sample screenshot of 20 x 20 elements. The matrix elaborates how the binary learners train and predict the classes. The columns and rows of the matrix correspond to the prediction results of the 4095 binary learners and the 91 classification classes, respectively.

The classification performance of the SVM model was determined by the following binary loss equation (Equation 1) (Fürnkranz, 2002; Escalera *et al.*, 2010).

$$\hat{p} = \underset{p}{\operatorname{argmin}} \frac{\sum_{b=1}^B |m_{pb}| l(m_{pb}, s_b)}{\sum_{b=1}^B |m_{pb}|} \quad (1)$$

where \hat{p} is the predicted class of observation used to minimize the aggregated losses of the binary learners B ; m_{pb} is element (p, b) of coding design matrix M where p is the corresponding class and b is the binary learner; s_b is the predicted score of the binary learner b ; and l is the binary loss function.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0

Figure 12. Sample 20 x 20 out of 4095 x 91 SVM coding matrix

2.4 LSTM Neural Network Classification Model

LSTM is the most popular type of recurrent neural network (RNN) that can learn long-term dependencies between time steps of sequence data. It is capable of addressing vanishing gradients, which is the main problem of artificial neural networks, through the use of memory cell to remember certain information over random time intervals, and gates to regulate the flow of information into and out of the cell (Hochreiter and Schmidhuber, 1997).

An LSTM layer consists of output state and cell state, which learn the previous time steps. For each step, the LSTM layer increments or decrements information from cell state through the use of gates. Figure 13 shows a basic unit of an LSTM layer with four gates, namely (1) input gate (ig) that controls the level of updates; (2) forget gate (fg) that controls the level of reset also known as forget state; (3) cell candidate (cc) that adds information to cell state; and (4) output gate (og) that controls the level of cell state added to output state.

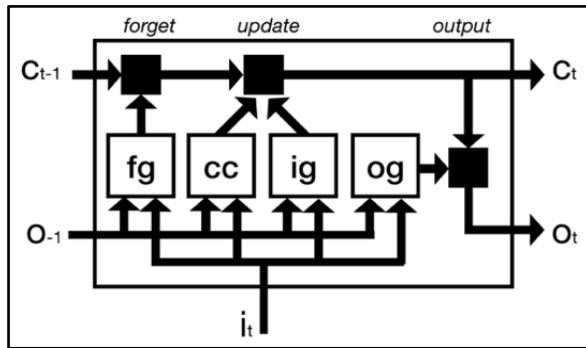


Figure 13. LSTM layer gates

The author designed the LSTM architecture shown in Figure 14, and then coded the equivalent function in Figure 15. Specifically, the author set the sequence input with one dimension followed by one word embedding layer, which mapped the tokenized words into a sequence of numerical indices, with 50 dimensions and 2033 unique words. The 2033 unique words were the results of the “word embedding” layer using the *doc2sequence* and *wrodEmbeddingLayer* functions of the MATLAB. From 745 *NumWords*, these remembered words were further mapped out into numeric vectors, which captured semantic details of the words, so that words with similar meanings have similar vectors. For example, the relationship “satisfaction is to customer as service is to organization” is described by the equation “customer – satisfaction + service = organization”.

The embeddings are the results of model relationships between words through vector arithmetic. Moreover, the author designed one LSTM layer with 80 hidden units; 116 fully connected layers to multiply the sequence input by weight matrix and add a bias vector into it; one *softmax* layer to compute the *softmax* function; and one classification output layer to determine the cross entropy (CE) loss of the network. To train the network, the author utilized adaptive moment estimation (Adam) solver (Kingma and Ba, 2014) with 16 *MiniBatchSize* and simulated with maximum 30 epochs; two gradient thresholds; and with an initial learning rate of 0.001.

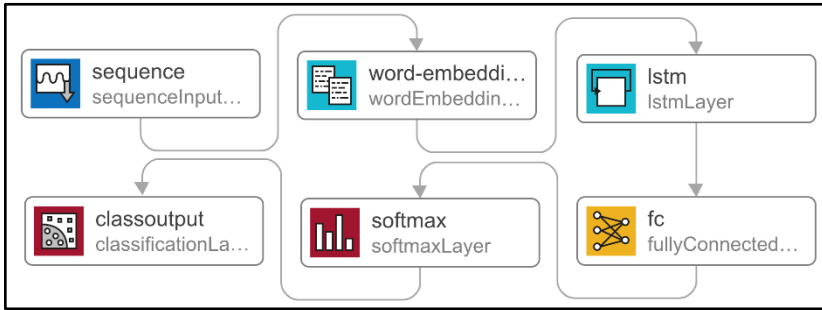


Figure 14. LSTM neural network architecture for text classification

```

Command Window
>> inputSize = 1;
embeddingDimension = 50;
numHiddenUnits = 80;
numWords = enc.NumWords;
numClasses = numel(categories(YTrain));
layers = [ ...
    sequenceInputLayer(inputSize)
    wordEmbeddingLayer(embeddingDimension,numWords)
    lstmLayer(numHiddenUnits,'OutputMode','last')
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer]
options = trainingOptions('adam', ...
    'MiniBatchSize',16, ...
    'GradientThreshold',2, ...
    'Shuffle','every-epoch', ...
    'ValidationData',{XValidation,YValidation}, ...
    'Plots','training-progress', ...
    'Verbose',false);

layers =

6x1 Layer array with layers:

1 '' Sequence Input      Sequence input with 1 dimensions
2 '' Word Embedding Layer Word embedding layer with 50 dimensions and 2033 unique words
3 '' LSTM                LSTM with 80 hidden units
4 '' Fully Connected     116 fully connected layer
5 '' Softmax              softmax
6 '' Classification Output crossentropyex
  
```

Figure 15. LSTM architecture function

Moreover, the *softmax* layer is based on the *softmax* function and computed after the fully connected layer as expressed in Equation 2 (Bishop, 2006).

$$S(c_p|x, \theta) = \frac{S(x, \theta|c_p)S(c_p)}{\sum_{j=1}^k S(x, \theta|c_j)S(c_j)} = \frac{\exp(a_p(x, \theta))}{\sum_{j=1}^k \exp(a_j(x, \theta))} \quad (2)$$

where $0 \leq S(c_p|x, \theta) \leq 1$; $\sum_{j=1}^k S(c_j|x, \theta) = 1$; $a_p = \ln(S(x, \theta/c_p)S(c_p)$; $S(x, \theta/c_p)$ is the conditional probability of the sample given class p and $S(c_p)$ is the class prior probability.

Meanwhile, the classification output layer was used to compute the CE loss for multi-classification and infer the number of classes from the output size of the previous layer. The CE loss was computed using Equation 3 (Bishop, 2006).

$$l = - \sum_{i=1}^S \sum_{j=1}^C x_{ij} \ln o_{ij} \quad (3)$$

where S is the number of samples, C is the number of classes, x_{ij} is the indicator that the i^{th} sample belongs to the j^{th} class, and o_{ij} is the output for sample i for class j based on the softmax function wherein the probability of the network is associated with the i^{th} input and class j .

2.5 Training, Optimization, and Actual Implementation

Consistent with the preprocessing method conducted in the preceding section, for each dataset, the holdout percentage or the validation threshold was increased from 10, 20, 30, 40 up to 50%. The more the holdout percentage was, the more data were divided to validate the training samples. Each combination of dataset and holdout percentage would generate a number of vocabularies called *NumWords*. Altogether, the dataset, holdout percentage and the *NumWords* were optimized and recorded in which the resulting data were used in the statistical analysis. For both SVM and LSTM models, the training classification accuracy was determined by the MATLAB using Equation 4.

$$a = \frac{(tp + tn)}{(tp + fp + fn + tn)} \quad (4)$$

where a is the training classification accuracy; tp is the number of true positive; tn is the number of true negative; fp is the number of false positive; and fn is the number of false negative.

After the training, the performance of SVM and LSTM models was compared in terms of accuracy and timeliness. This was done by determining their mean differences using independent samples T-test equation (Equation 5) (Villanueva and Corpuz, 2020).

$$t = \frac{\bar{s}1 - \bar{s}2}{\sqrt{\left(\frac{d_1^2}{o1} + \frac{d_2^2}{o2}\right)}} \quad (5)$$

where $\bar{s}1$ is the SVM dataset average; $\bar{s}2$ is the LSTM dataset average; $o1$ is

the number of observations of SVM; $o2$ is the number of observations of LSTM; $d1$ is the SVM standard deviation; and $d2$ is LSTM standard deviation. The resulting t value is compared with critical t-value distribution table wherein the degrees of freedom df was computed using Equation 6.

$$f = \frac{\left(\frac{d_1^2}{o1} + \frac{d_2^2}{o2} \right)^2}{\frac{1}{o1-1} \left(\frac{d_1^2}{o1} \right)^2} \quad (6)$$

Consequently, the author determined the predictors of accuracy and timeliness performance based on the effect of dataset, holdout, and *NumWords*. This approach was done to determine the best set of parameters that could help in optimizing the performance of the classification models. The prediction analysis was done using the following multiple regression equation (Equation 7) (Neter *et al.*, 1996; Corpuz, 2016).

$$r_i = \beta_0 + \beta_1 A_{i1} + \beta_2 A_{i2} + \dots + \beta_p A_{ip} + \epsilon_i, \quad i = 1, \dots, n, \quad (7)$$

where r_i is the i^{th} response; β_c is the c^{th} coefficient; β_k is the constant term in the model; A_{ij} is the i^{th} observation on the j^{th} predictor variable $j=1, \dots, p$; ϵ_i is the i^{th} random error.

After the series of modelling steps, both SVM and LSTM models were deployed for actual implementation using 20 sample audit findings, which were independent from the training and validation datasets. The main intent of this testing was to establish the effectiveness of the design process conducted for both classification models.

3. Results and Discussion

In this study, AI-based text classification models using SVM and LSTM neural network were designed to generate accurate and timely ISO 9001:2015 audit reports. To optimize the performance of the said models, the author investigated the predictors of accuracy and timeliness performance then compared the effects of these parameters with the classification models.

The SVM and LSTM classification models were trained by increasing the number of datasets and holdout percentage. The effect of this approach was observed in the resulting *NumWords* or vocabulary of words learned by the models as well as on the training accuracy and timeliness performance results. Summarized in Table 4 is the training performance of both SVM and LSTM models. As observed, SVM generally had better classification accuracy than LSTM in smaller datasets, while the latter fared better with the larger ones. Likewise, SVM performed faster in larger datasets than LSTM.

Table 4. Training performance comparison

Dataset	Holdout (%)	SVM			LSTM		
		<i>NumWords</i>	Accuracy (%)	Timeliness (s)	<i>NumWords</i>	Accuracy (%)	Timeliness (s)
2856 (100%)	10	1832	94.74	79.08	2132	97.54	306
	20	1620	93.70	82.28	2130	95.80	281
	30	1476	90.42	80.66	2115	94.39	260
	40	1281	88.88	90.18	2094	94.83	247
	50	1071	85.50	79.48	2033	88.10	211
1904 (66.67%)	10	1153	88.42	75.35	2131	92.11	137
	20	1061	86.84	75.40	2119	92.37	108
	30	977	80.39	77.19	2054	84.24	93
	40	935	78.58	66.98	1967	79.24	80
	50	762	73.00	80.87	1881	76.47	83
952 (33.33%)	10	745	48.42	61.91	2045	49.47	55
	20	689	52.11	54.50	1923	48.95	60
	30	645	48.77	50.54	1814	51.93	48
	40	558	50.79	31.99	1599	48.95	41
	50	479	50.00	32.99	1524	46.22	40

Notably, the best accuracy performance was captured at 2856 dataset and 10 holdout percentage for both models. As such, SVM generated 1832 *NumWords*, 94.74% accuracy and 79.08 s elapsed time, while LSTM generated 2132 *NumWords*, 97.54% accuracy and 306 s elapsed time. So far, the best classification performance was obtained by LSTM as shown in Figure 16, which exhibits the training progress data on accuracy and loss but with slower training time of 5 min and 6 seconds (s).

To support the above findings, Table 5 highlights the results of the independent samples T-test used to inferentially compare the two models. While both SVM and LSTM performed generally more accurate in smaller and larger datasets, respectively, the study found that there was no statistically significant difference in terms of training classification accuracy between the

two models with $t(28) = -.281$; $p = .781$. This means that both SVM and LSTM were at par with each other in terms of classification accuracy criteria.

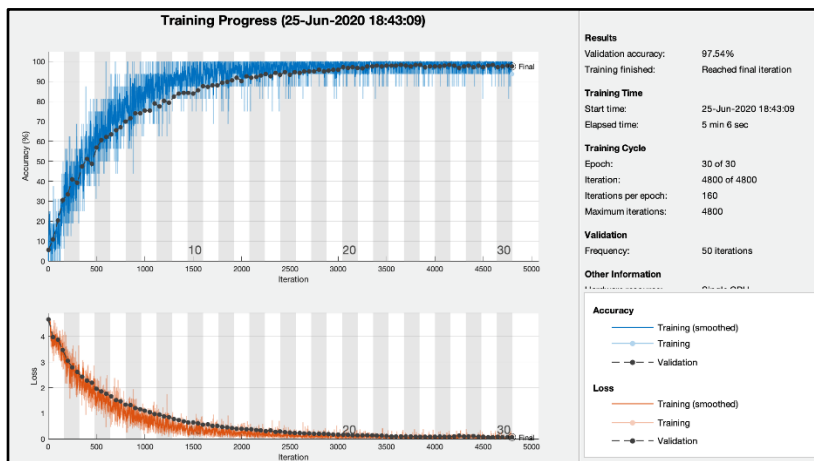


Figure 16. LSTM best training progress

Meanwhile, in terms of timeliness, it was found out that there was a statistically significant difference between the two models. The results revealed that SVM performed faster (67.96 ± 17.93 s) than LSTM (136.67 ± 96.42 s) where $t(14.971) = -2.713$ and $p = .000$.

Table 5. Independent samples T-test results

	Levene's Test of Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Diff	Std. Error Diff.	95% Confidence Interval of the Difference	
								Lower	Upper
Accuracy Equal Variances Assumed	.352	.558	-.281	28	.781	-2.003	7.137	-16.623	12.616
Accuracy Equal Variances Not Assumed			-.281	27.667	.781	-2.003	7.137	-16.623	12.616
Timeliness Equal Variances Assumed	34.701	.000	-2.713	28	.011	-6870667	25.323	-120.578	-16.836
Timeliness Equal Variances Not Assumed			-2.713	14.971	.016	-6870667	25.323	-122.690	-14.724

The author explored the causal relationship between dataset, holdout and *NumWords* as independent variables with training accuracy and timeliness as dependent variables. As such, it was revealed that the dataset and holdout percentage were the predictors of accuracy while dataset and *NumWords* were the predictors of timeliness. Specifically, Table 6 shows the model summary used to establish how the regression model can fit the required data. The values of $R = .939$ and $.746$ represent the multiple correlation coefficient for which both signify good prediction levels for training accuracy and timeliness, respectively. The R^2 value of $.882$ and $.562$, also known as the coefficient of determination, indicates that the independent variables can explain 88.20 and 56.20% of the variability of the dependent variables, respectively.

Table 6. Model summary

Dependent Variable	R	R ²	Adj. R ²	Std. Error Estimate	R ² Change	F Change	df 1	df 2	Sig. F Change
Accuracy	.925 ^a	.855	.850	7.457	.855	164.911	1	28	.000
	.939 ^b	.882	.873	6.857	.027	6.113	1	27	.020
Timeliness	.673 ^a	.453	.433	57.657	.453	23.157	1	28	.000
	.749 ^c	.562	.529	52.542	.109	6.717	1	27	.015

a. Predictors: (Constant), Dataset

b. Predictors: (Constant), Dataset, Holdout

c. Predictors: (Constant), Dataset, *NumWords*

As listed in Table 7, the F-ratios of $F(2, 27) = 100.568$, $p = 0.000$ for accuracy and $F(2, 27) = 17.301$, $p = 0.000$ for timeliness both denote that the identified independent variables statistically significantly predicted the dependent variables of the model. This further signifies that the regression model is a good fit of the data.

Table 7. ANOVA

Dependent Variable		Sum of Sq	df	Mean Square	F	Sig.
Accuracy	Regression	9169.788	1	9169.788		
	Residual	1556.927	28	55.605	164.911	.000 ^a
	Total	10726.716	29	-		
	Regression	9457.204	2	4728.602		
	Residual	1269.512	27	47.019	100.568	.000 ^b
	Total	10726.716	29	-		
Timeliness	Regression	76982.553	1	76982.553		
	Residual	93080.531	28	3324.305	23.157	.000 ^a
	Total	170063.084	29	-		
	Regression	95525.223	2	47762.612		
	Residual	74537.860	27	2760.661	17.301	.000 ^c
	Total	170063.084	29	-		

a. Predictors: (Constant), Dataset

b. Predictors: (Constant), Dataset, Holdout

c. Predictors: (Constant), Dataset, *NumWords*

Meanwhile, the following model coefficients in Table 8 indicate how much the dependent variable varies with a specific independent variable while the other independent variables are set constant. The unstandardized coefficient values of 0.022 and 0.050 denoted that for every one element of dataset was added, accuracy and timeliness were increased by approximately 0.022% and 0.050 s, respectively. On the other hand, the unstandardized coefficient value of -.219 means that for every percentage increase of holdout percentage, the accuracy was decreased by approximately .219%. Lastly, the unstandardized coefficient value of 0.049 denoted that for every *NumWord* or word remembered by the classification models, the timeliness was likewise increased by approximately 0.049 s.

Table 8. Coefficients

Dependent Variable		Unstandardized Coefficients		Std. Coefficient	t	Sig.
		B	Std. Error	Beta		
Accuracy	(Constant)	32.229	3.601	-	8.950	0.000
	Dataset	0.022	0.002	0.925	12.842	0.000
	(Constant)	38.795	4.245	-	9.140	0.000
	Dataset	0.022	0.002	0.925	13.965	0.000
	Holdout	-0.219	0.089	-0.164	-2.472	0.020
Timeliness	(Constant)	-21.726	27.843	-	-0.780	0.442
	Dataset	0.065	0.014	0.673	4.812	0.000
	(Constant)	-66.286	30.650	-	-2.163	0.040
	Dataset	0.050	0.014	0.521	3.718	0.001
	<i>NumWords</i>	0.049	0.019	0.363	2.592	0.015

Hence, as the result of multiple regression analysis, the proposed equations to predict the training accuracy and timeliness, based on the optimization of predictors, are Equations 8 and 9, respectively.

$$a = 38.795 + (.022d) - (.219 h) \tag{8}$$

$$t = - 66.286 + (.050d) + (.049 n) \tag{9}$$

where *a* is the predicted accuracy; *t* is the predicted timeliness; *d* is the dataset value; *h* is the holdout percentage; and *n* is the *NumWord* value.

Figures 17 and 18 illustrate the normal probability plots and equivalent histogram for both accuracy and timeliness, respectively.

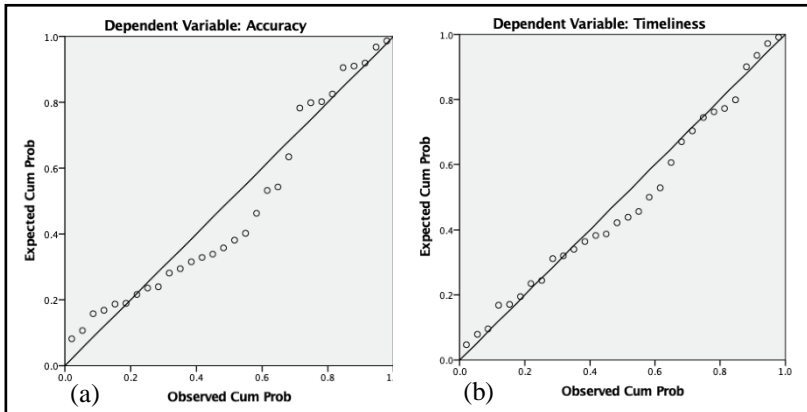


Figure 17. Normal P-P plot of regression standardized residual: accuracy (a) and timeliness (b)

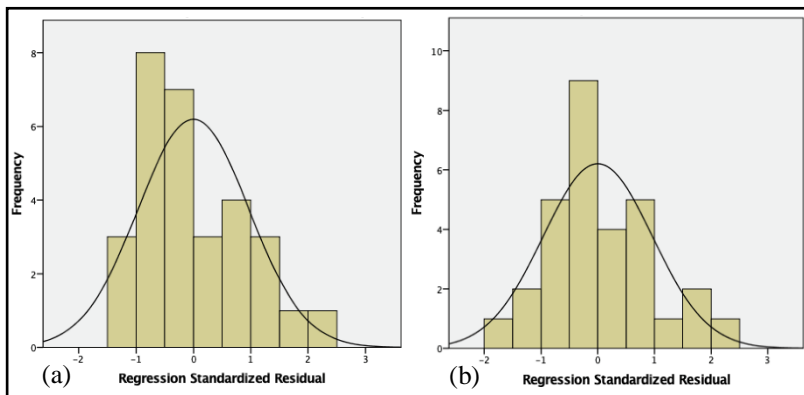


Figure 18. Multiple regression histogram: accuracy (a) and timeliness (b)

2.6 Actual Implementation Results

In consideration of the preceding results of training and statistical analyses, the best classification models were deployed, which utilized 2856 dataset, 10% holdout with 1832 (SVM) and 2132 (LSTM) *NumWords* for actual implementation. Figures 19 and 20 show the actual prediction made by SVM and LSTM classification models, respectively. Interestingly, both models predicted 20 out of the 20 samples accurately with a classification time of 1 and 3 s for SVM and LSTM, respectively. Hence, both classification models validated the effectiveness of the proposed design process.

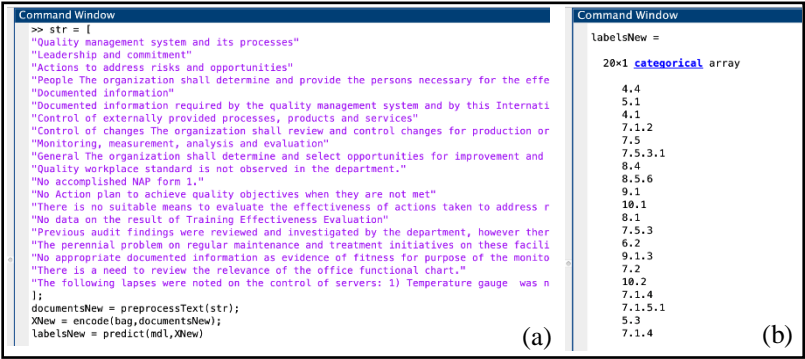


Figure 19. SVM classification model actual implementation results:
SVM input (a) and SVM output (b) data

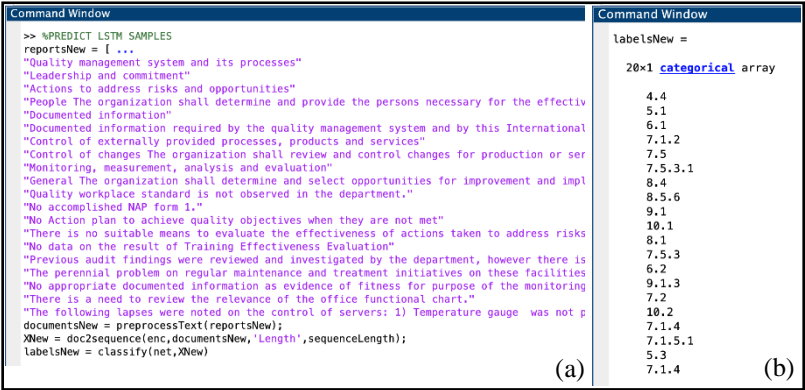


Figure 20. LSTM classification model actual implementation results:
LSTM input (a) and LSTM output (b) data

4. Conclusion and Recommendation

The study found that while LSTM generally performed better (97.54% [best rating]) than SVM (94.74% [best rating]) in terms of training accuracy in larger datasets, their difference, however, is not statistically significant. On the other hand, SVM performed significantly faster with an average of 67.96 ± 17.93 s in all training instances than LSTM, which had an average of 136.67 ± 96.42 s. The training performances of these algorithms can be improved by optimizing their datasets, holdout percentages and *NumWords*.

Moreover, it was found out that the predictors of accuracy were dataset and holdout, while the predictors of timeliness were dataset and *NumWords*.

Specifically, dataset was found to have positive effect on both accuracy and timeliness performances, while holdout percentage had negative effect on accuracy and *NumWords* had negative effect on the timeliness only. This means that the more datasets are used, the better will be the accuracy but with slower training time; the more holdout percentage will result in lesser accuracy; and the more *NumWords* will result in slower training time.

Using the above-mentioned optimization techniques, both SVM and LSTM classification models were proven to be effective in predicting 20/20 of the correct clauses of the standard based on the actual implementation results. Hence, the extent of choosing between SVM and LSTM algorithms for ISO 9001 standard clauses and audit findings classifications would depend on the number of available datasets, number of learned words, optimization of holdout percentage, familiarization to various text analysis models to generate more *NumWords* and speed capability of the workstation.

Future research should add more audit findings, preferably from other similar organizations, to improve the learnability of the classification models and validate the proposed prediction equation. Likewise, it is recommended to identify the maximum number of dataset, holdout and *NumWords* that could be optimized to avoid overfitting.

5. Acknowledgement

The author would like to thank the support of the top management of the Technological University of the Philippines in ensuring the needed resources of the QMS of the university; to the internal auditors and the process owners for their continued support during audits; and to the quality assurance staff for their relentless efforts and contributions.

The author dedicates this paper to his wife, Maria Teresa Carmela B. Garcia-Corpuz, and first-born daughter, Athera Ampere G. Corpuz, for their unconditional love and inspiration.

6. References

Bishop, C.M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Cai, J., Li, J., Li, W., & Wang, J. (2018). Deep learning model used in text classification paper. Proceedings of the 15th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 123-126.

Christianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge, UK: Cambridge University Press.

Corpuz, R.S.A. (2016). Live chat technology optimization framework for online businesses. *Prism*, 21(2), 15-33.

Corpuz, R.S.A. (2019). Implementation of artificial neural network using scaled conjugate gradient in ISO 9001:2015 audit findings classification. *International Journal of Recent Technology and Engineering*, 8(2), 420-425.

Corpuz, R.S.A. (2020). ISO 9001:2015 risk-based thinking: A framework using fuzzy-support vector machine. *Makara Journal of Technology*, 24(3), 149-159.

Corpuz, R.S.A. (2021a). Categorizing natural language-based customer satisfaction: An implementation method using support vector machine and long short-term memory neural network. *International Journal of Integrated Engineering* (article in press).

Corpuz, R.S.A. (2021b). An application method of long short-term memory neural network in classifying English and Tagalog-based customer complaints, feedbacks, and commendations. *International Journal on Information Technologies and Security*, 13(1), 89-100.

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34(1), 313-327.

Escalera, S., Pujol, O., & Radeva, P. (2010). On the decoding process in ternary error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7), 120-134.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning Research*, 2, 721-747.

Government Quality Management Committee. (2020). Agencies with ISO 9001 certification. Retrieved from <https://www.gqmc.gov.ph/index.php/reports-references/agencies-with-iso-9001-certification>.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). The elements of statistical learning (2nd ed.). New York: Springer.

Hsiao, Y-L., Drury, C., Wu, C., & Paquet, V. (2013). Predictive models of safety based on audit findings: Part 1: Model development and reliability. *Applied Ergonomics*, 44 261-273. <https://doi.org/10.1016/j.apergo.2012.07.010>

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

International Organization for Standardization (ISO). (2015a). Quality management systems - Fundamentals and vocabulary (ISO Standard No. 9000:2015). Retrieved from <https://www.iso.org/standard/45481.html>

International Organization for Standardization (ISO). (2015b). Quality management systems – Requirements (ISO Standard No. 9001:2015). Retrieved from <https://www.iso.org/standard/62085.html>

International Organization for Standardization (ISO). (2018). Guidelines for auditing management systems (ISO Standard No. 19011:2018). Retrieved from <https://www.iso.org/standard/70017.html>

International Organization for Standardization (ISO). (2020). ISO 9000 family quality management. Retrieved from <https://www.iso.org/iso-9001-quality-management.html>

Kingma, D. & Ba, J. (2015). Adam: A method for stochastic optimization. Retrieved from <https://arxiv.org/abs/1412.6980>

Lessman, S., & Vob, S. (2009). A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research*, 199(2), 520-530. <https://doi.org/10.1016/j.ejor.2008.12.017>

Neter, J., Kutner, M.H., Nachtsheim, C.J., & Wasserman, J. (1996). *Applied linear statistical models*. Irwin: The McGraw-Hill Companies, Inc.

Philippine Government Official Gazette. (2020). Executive order no. 605, s. 2007. Retrieved from <https://www.officialgazette.gov.ph/2007/02/23/executive-order-no-605-s-2007>

Poksinska, B., Dahlgaard J., & Antoni M. (2002). The state of ISO 9000 certification: A study of Swedish organisations. *The TQM Magazine*, 14(5), 297-306.

Qing, L., Linhong, W., & Xuehai, D. (2019). A novel neural network-based method for medical text classification. *Future Internet*, 11(255), 1-13. <https://doi.org/10.3390/fi11120255>

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34 (1), 1-47.

Shirata, C., & Sakagami, M. (2008). An analysis of the “going concern assumption”: Text mining from Japanese financial reports. *Journal of Emerging Technologies in Accounting*, 5, 1-16.

Tarnate, K., Madhavi, D., & De Goma, J. (2020). Overcoming the vanishing gradient problem of recurrent neural networks in the ISO 9001 quality management audit reports classification. *International Journal of Scientific and Technology Research*, 9 (3), 6683-6686.

Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed.). Berlin, Germany: Springer.

Villanueva, A.B., & Corpuz, R.S.A. (2020). Design and development of fire evacuation system using fuzzy logic control. *International Journal of Scientific and Technology Research*, 9(4), 2096-2103.