Classification of Legislations using Deep Learning

Sameerchand Pudaruth[1], Sunjiv Soyjaudah[2], and Rajendra Gunputh[3]
[1]ICT Department, University of Mauritius, Mauritius
[2]Soyjaudah Chambers, Mauritius
[3]Law Department, University of Mauritius, Mauritius

Abstract: *Laws are often developed in a piecemeal approach and many provisions of similar nature are often found in different legislations. Therefore, there is a need to classify legislations into various legal topics to help legal professionals in their daily activities. In this study, we have experimented with various deep learning architectures for the automatic classification of 490 legislations from the Republic of Mauritius into 30 categories. Our results demonstrate that a Deep Neural Network (DNN) with three hidden layers delivered the best performance compared with other architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). A mean classification accuracy of 60.9% was achieved using DNN, 56.5% for CNN and 33.7% for Long Short-Term Memory (LSTM). Comparisons were also made with traditional machine learning classifiers such as support vector machines and decision trees and it was found that the performance of DNN was superior, by at least 10%, in all runs. Both general pre-trained word embeddings such as Word2vec and domain-specific word embeddings such as Law2vec were used in combination with the above deep learning architectures but Word2vec had the best performance. To our knowledge, this is the first application of deep learning in the categorisation of legislations.*

Keywords: *Deep learning, neural networks, classification, legislations.*

Received October 17, 2019; accepted February 9, 2021
https://doi.org/10.34028/iajit/18/5/4

1. Introduction

Deep learning is a branch of artificial intelligence and a subset of machine learning which studies the application of deep neural networks in solving problems in the field of computer vision, computational linguistics and natural language processing. Deep learning is also known as deep structured learning or hierarchical learning. A deep neural network has at least two hidden layers. The role of each layer is to incrementally learn more complex representations from data. For example, in document classification, lower layers may identify characters and words while upper layers may identify more meaning items such as the main topic of the document [14]. In deep learning, the network learns from all layers at the same time in contrast to the idea of stacking several shallow models for each layer of learning. Deep learning is being used to achieve state-of-the-art performances on all types of problems and datasets, ranging from image recognition, speech recognition, document classification, drug discovery, recommendation systems and machine translation [3, 14]. With the continual increase in the size of datasets and a corresponding increase in the number of categories, the performance of traditional machine learning classifiers is degrading rapidly and have reached a plateau where further gain is very difficult, while it is commonly believed that the performance of a deep learning classifier gets better with an increase in the amount of available data [14].

Deep learning has one huge advantage compared with the traditional way of applying machine learning algorithms. Previously, in many situations, a disparate amount of time was being spent on feature engineering [3]. For example, in the computer vision field, previously in order to recognise an object from an image, various features such as colour, shape and texture information had to be extracted from each image before the training step. With deep learning, it is sufficient to feed the images directly to the classifier to build the training model. Similarly, in the field of natural language processing and machine translation, deep learning models are being preferred to brittle and reliable models based on hand-crafted linguistic rules. For example, machine translation models are being built to translate one language into another simply by providing the classifier with a huge parallel corpus. The focus in data science has moved from feature engineering and feature selection to data engineering [3]. With deep learning, more time is spent in preparing the data in the right format for input to the DNN. Be it audio data, videos, images or texts, pre-processing steps to remove noise, to resize or augment the data are still required and have become even more important. The availability of cheaper and faster Central Processing Units (CPUs) and GPUs, coupled with cloud-based solutions, have spearheaded the transition to deep learning architectures in all areas of research where machine learning was previously applicable. Moreover, deep learning is accelerating the

progress in the development of autonomous vehicles, smarter human-like robots, virtual worlds, image caption generation, abstractive summarisation and text generation.

Due to the wide interest in the use of deep learning, a number of deep learning frameworks has been developed in the last decade. The Deep Learning (DL) frameworks in order of popularity are: TensorFlow, Keras, PyTorch, MXNet, Theano, Caffe, Caffe2, CNTK, Chainer and Torch. TensorFlow is by far the most popular DL framework [23]. TensorFlow is implemented in C++ but it also has first-hand support for Python. In this work, we have chosen to use Keras, on a TensorFlow backend, to implement our deep learning architectures. Keras offers a simple and intuitive interface to programmers and other types of users to create deep learning architectures with a minimal amount of training and in very less time compared to most other frameworks [6]. Currently, a lot of documentation is available on the development and debugging of Keras models in Python.

Locating relevant legislations or similar legislations is often a difficult task for legal professionals or other users in the Republic of Mauritius. This is partly because the same issues have been discussed in several legislations which at first sight may not appear to be related. Furthermore, some laws in Mauritius, especially those preceding independence, were written in the French language while most of the post-independence laws are in English. This also makes the retrieval of relevant legislations even more difficult. Thus, in this study, we have investigated the performances of deep neural networks in the classification of legislations from the Republic of Mauritius. A deep neural network with 5 layers (1 input, 3 hidden, 1 output) provided the best classification accuracy. Adding more layers did not contribute to improving the accuracy, instead performance became worst in most cases, depending on the value of the hyperparameters. The processing time is also much longer when using more layers. CNNs, RNNs Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Bi-directional LSTM and GRU and pre-trained word embeddings did not improve the accuracy compared with the DNN but the CNN still did better than all the traditional machine learning classifiers on the legislation dataset. To our knowledge, this is the first work that studies the application of deep learning in the classification of legislations.

This paper proceeds in the following manner. In section 2, we provide an overview of the applications of deep learning in the law domain. Section 3 describes the dataset and the classification process. The description and implementation of the deep learning architectures are described in section 4. The experiments, results and evaluation of the results are presented in sections 5 and 6 concludes the paper with a brief note on limitations and future works.

2. Related Works

Although the application of deep learning in the field of natural language processing is very recent, a lot of work is being done in its sub-fields such as sentiment analysis, machine translation, document classification, document summarisation and question-answering systems [17, 20]. A lot of work has also been done on the retrieval of relevant documents from a legal knowledge base and the classification of court judgements into different areas of law [11, 27]. A number of works has also been done on the classification of provisions into various types [9, 10, 24, 28]. However, the classification of legislations has not received much attention from the Artificial Intelligence and Law community. In this section, we review works that have been done on the applications of machine learning and deep learning in document classification but with a particular focus on legal texts.

The first attempt at the automatic classification of statutory documents was done by Curran and Thompson [8]. In those days, computing power was very limited and therefore, semi-automatic approaches were more common. A rigorous indexing mechanism was put in place for the manual classification of 149,655 statutory documents. The C4.5 algorithm was used for the automatic classification of the documents. The C4.5 algorithm is a type of decision tree which can be considered as a machine learning classifier. The researchers achieved a recall of 41.6% on 36 categories. However, there was a very huge variation between the recall for the different categories. About a decade later, Purpura and Hillard [25] classified 108,268 legislations from the United States into 20 law categories using Support Vector Machines (SVM). They achieved an accuracy of 82.2% on the first level of classification. These first level categories were further divided into varying numbers of sub-categories. The overall accuracy for the hierarchical classification was 71.0%. SVM was used at the second stage as well. In the context of the Eunomos project, Boella *et al*. [2] classified 223 legislative texts from the taxation domain into six sub-categories of tax laws. They achieved a weighted average recall of 76% using an SVM classifier from the Weka toolkit. At that time and up until recently, support vector machines were considered as state-of-the-art text classifiers.

In order to demonstrate the superiority of deep learning architectures in the field of text classification, Kowsari *et al*. [16] classified 46,985 abstracts from the Web of Science (WoS) into 134 categories in a hierarchical manner. In the first stage, there were 7 categories which were very distinct from each other while in the second stage, each first-level category was further divided into a varying number of areas. The

smallest number of sub-categories was 9 while the largest one was 53. Many experiments were carried out and comparisons were made with SVM-based approaches as well as with deep learning models from other researchers. Their best score of 76.6% was achieved by using a Recurrent Neural Network (RNN) at each stage. No information was provided on the average sizes of the abstracts. Based on testing on four different datasets, Lai *et al.* [18] had also come to the same conclusion that deep learning approaches deliver better performances than the traditional bag-of-words approaches. They achieved their best score by adding a recurrent layer to a Convolutional Neural Network (CNN). However, they did not provide results for a RNN alone.

Using a word-level and sentence-level attention mechanism inside a hierarchical deep learning network, Yang *et al*. [29] achieved state-of-the-art performances on six publicly available datasets. Their hierarchical attention network, as they dubbed it, is able to capture structural information about the documents and contextual information from words and sentences. A word embedding (dimension = 200) based on the Word2vec [1, 22] model was used to initialize the neural network in the input layer. A bi-directional GRU (a type of RNN) with a dimension of 50 in each direction was used. The gain in accuracy over previous approaches was significant for all the six datasets.

Deep learning has also been used for the extraction of entities from legal documents with a high rate of success. Chalkidis and Androutsopoulos [4] used a combination of LSTM, Bi-directional LSTM (Bi-LSTM), Conditional Random Field (CRF) and logistic regression layers to achieve much better performances than all their own previous approaches which were based on support vector machines and logistic regression. Luo *et al*. [21] have used deep learning for predicting the charge from Chinese criminal cases. Their architecture was based on a Bi-directional Recurrent Gated Unit (Bi-GRU) and a domain-specific word embeddings trained on 573,353 words with 100 dimensions. In most cases, the results were better than an SVM classifier trained on a bag-of-words model with tf-idf features.

John *et al*. [13] have used a deep learning algorithm for a question-answering task in the context of the annual Competition on Legal Information Extraction/Entailment (COLLIE) competition which is focused on information extraction and information retrieval. In particular, they have used a variant of the Long Short-Term Memory deep learning architecture, known as the Child Sum Tree- LSTM to predict whether the answer to a question is entailed in a piece of text. Their deep learning architecture achieved an accuracy of 70.1%, which was only 1.7% better than SVM. Collarana *et al*. [7] have used a two-step procedure to answer questions on the MaRisk regulatory document. This is a 62-page long document with about 24,000 words which describes financial management for financial institutions in Germany. In the first phase, the researchers have used StarSpace and fastText from Facebook research to create word embeddings, for the selection of the most relevant paragraphs from within the document. In the second step, they employed a deep learning architecture based on a Match-LSTM layer and an Answer-Pointer layer to select the most relevant span of text from within the paragraphs identified in the first stage. Although they achieved slightly better results in the top-3 and top-5 category with pre-trained word embeddings and LSTM, using a traditional information retrieval approach in the first phase lead to the best precision and recall for the top-1 category.

Very recently, Lippi *et al*. [19] have used deep learning architectures to identify unfair clauses from online terms of service. Their research is based on the understanding that online users rarely read the terms of service in the haste of consuming that service. Their dataset consists of 50 contract documents (terms of service) which were segmented into a total of 12,011 sentences of which only 1,032 were labelled as potentially unfair clauses. Surprisingly, support vector machines outperformed both the CNN and LSTM deep learning architectures by a significant margin. The authors also report that no improvement was observed when pre-trained word embeddings were used. However, it is not understood why results for a simpler deep learning network (based on multiple layers of dense connections) are not shown. Earlier, working on a similar problem, Goltz and Mayo [12] showed that the traditional bag-of-words model of representing documents performed better than their equivalent Word2vec representation.

In this section, we have given an overview of the different works that have been done on both legal and non-legal texts using both traditional machine learning classifiers and deep learning architectures. Deep learning techniques have yet to be applied to the classification of legislations. In the next section, we describe the dataset and the classification process.

3. Methodology

3.1. Dataset

Five hundred and six legislations from the Republic of Mauritius were manually classified into 35 categories by a legal professional who held a Legum Magister/Master of Laws (LLM) (Master of Laws) degree. In total, there are about 1000 legislations which are currently in force in Mauritius. Five of these categories had less than 5 samples and these were removed from the dataset. Our current and final dataset consists of 490 legislations classified into 30 categories. The list of categories and the number of documents in each category are described below in Table 1.

The category with the highest number of acts is Social security and welfare, with a total of sixty-eight acts which represents about 14% of the dataset. There are three categories with five acts, which is the least number of acts per category and each one of them represents 1% of the dataset. These categories are: Animal welfare, Emergency services and force majeure and Gambling. There is a total of 2,468,406 words in the dataset with a vocabulary of 85,041 words. The average number of words per act is 5,027 while the average number of words per sentence is 47. The total number of sentences is 53,054 and thus the average number of sentences per act is 108.

Table 1. List of legal categories.

#	Legal categories	Sample legislations	Number of legislations
1	Agriculture, environment and natural reserves	The forest and reserves act 1983 The wildlife and national parks act 1993	12
2	Animal welfare	The control of stray dogs act 2000 The prevention of cruelty to animals act 1957	5
3	Arts and cultural heritage	The Aapravasi Ghat trust fund act 2001 The national heritage fund act 2003	18
4	Aviation	The civil aviation (fees and charges) act 1977 The civil aviation act 1974	7
5	Banking	The Bank of Mauritius act 2004 The foreign exchange dealers act 1995	11
6	Businesses and companies	The companies act 2001 The insolvency act 2009	28
7	Chemicals	The biological and toxic weapons convention act 2004 The dangerous chemicals control act 2004	11
8	Citizenship	The civil status act 1981 The Mauritius citizenship act 1968	11
9	Civil procedures	Code de procedure civile 1808 The courts (civil procedure) act 1856	12
10	Construction, buildings and development	The building act 1915 The town and country planning act 1995	15
11	Criminal procedure	The bail act 1999 The criminal procedure act 1853	17
12	Customs and excise	The customs act 1988 The foreign travel tax act 1978	9
13	Diplomatic immunities and privileges	The diplomatic relations act 1968 The official secrets act 1972	14
14	Education, training, research and standards	The education act 1957 The training and employment of disabled persons act 1996	38
15	Emergency services and force majeure	The fire services act 1954 The national disaster risk reduction and management act 2016	5
16	Finance	The financial services act 2007 The stock exchange act 1988	19
17	Gambling	The gambling regulatory authority act 2007 The horse racing board act 2003	5
18	Information and Communication Technologies	The computer misuse and cybercrime act 2003 The data protection act 2017	8
19	Judiciary	The court of Rodrigues jurisdiction act 1913 The judicial provisions act 2008	25
20	Labour issues	The employment rights act 2008 The recruitment of workers act 1993	12
21	Land laws	The cadastral survey act 2011 The survey of lands act 1972	7
22	Marine and ocean resources	The fisheries and marine resources act 2007 The territorial sea act 1970	10
23	Medical service, food, health and safety	The medical council act 1999 The occupational safety, health and welfare act 1988	22
24	Social security and welfare	The social integration and empowerment act 2016 The pensions act 1951	68
25	Social, cultural and religious activities	The islamic cultural centre trust fund act 1989 The roman catholic church act 1928	38
26	Sugar sector	The Mauritius sugar authority act 1984 The sugar industry efficiency act 2001	14
27	Tax laws	The income tax act 1995 The value added tax act 1998	15
28	Trade, industry and exports	The export processing zones development authority act 1990 The small scale industries act 1988	12
29	Transportation laws	The Mauritius land transport authority act 2009 The national transport corporation act 1979	9
30	Utilities	The central electricity board act 1963 The central water authority act 1971	13

3.2. Text Classification Process

Figure 1 shows all the different steps that must be followed to build and evaluate a machine learning model. Thus, the process starts with the conversion of legislations from the pdf format into text format. This is achieved using the PyPDF2 Python library. This library first split a pdf document into pages, convert each page to text and then concatenates all the content into a string variable. The text string is then segmented into sentences using the NLTK tokenizer.

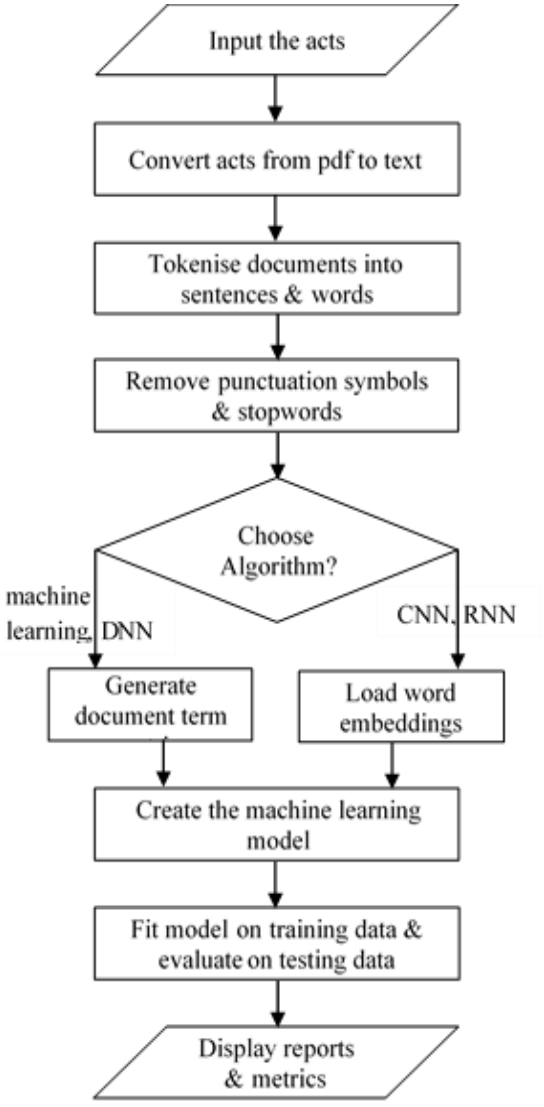


Figure 1. Process to build models for classification and evaluation.

The sentences are further tokenized into words. It is possible to tokenize the text string (representing the whole legislation) directly into words but the intermediary step is preferred in order to gather sentence statistics. Also, this provides the possibility to apply sentence-level cleaning operations such as removing very short sentences or very long ones, although this has not been used in our current text classification pipeline.

Once the document is available as a list of words, further operations are required in order to clean the text. Such operations include the removal of extra blank spaces, digits and punctuation symbols. All the stopwords from the Natural Language Toolkit (NLTK) list are also removed. Next, we extracted only the first *n* words from the cleaned text because deep learning algorithms are resource- intensive and they also need fixed-length inputs. Five different sizes of text are used: 500, 1000, 2000, 3000, and 4000 words.

The next step is to choose the machine learning classifier to build the model because the pre-processing steps will differ for traditional machine learning classifiers and deep learning models. A vanilla Deep Neural Network (DNN) is generally considered as a deep learning algorithm if it has more than one hidden layer. However, the inputs to this type of neural network are the same as for traditional machine learning classifiers, which is in the form of a document term matrix [14].

For building deep learning models such as CNNs and RNNs, the first step is usually the inclusion of a pre-trained word embeddings such as Word2vec and GloVe [17]. Deep learning classifiers deal with sequences of data and not single words, therefore, the tokenized and cleaned text has to be converted back to a sequence of words and ultimately to a sequence of numbers as neural networks can process only numbers [16]. A word index is kept in order to know which word maps to which number. This is easily achieved through the use of a Python dictionary where the key represents the word, and the value represents the corresponding number. Not all documents have the same length. However, deep learning models require that all inputs should be of the same length. Thus, all sequences which are less than the size of the longest sequence are padded with zeros.

Next, we need to build the classifier. For traditional machine learning classifiers, we simply have to call the relevant classifier and provide values for the different parameters. However, building deep learning models that perform well is slightly more challenging although the Keras library for Python hides a lot of the underlying complexities [6]. A neural network model has a minimum of three layers: an input layer, a hidden layer, and an output layer. However, it is more common for deep learning models to have several hidden layers. The dataset is usually divided into two parts: training and testing sets, where the training set is usually much larger than the testing set. A model is built on the training data and then evaluated on the testing data. The results can be interpreted by a confusion matrix, a classification report or individual performance metrics. In the next sections, we provide details on how the deep learning models were implemented and evaluated.

4. Implementation

In this section, three different deep learning architectures are described. Their hyperparameters and other customisations are also explained. We also

describe word embeddings as these are crucial for the proper functioning of deep neural networks.

4.1. Deep Neural Networks (DNNs)

Although the link is not often made, the simplest type of deep neural network is actually a multi-layer perceptron, a concept which has existed since many decades. In today's parlance, this is often called a vanilla deep neural network (vanilla DNN). A summary of the deep learning model is shown in Figure 2. Besides providing information on the names and types of each layer, it also indicates the output shape of each layer and the number of parameters at each layer. The parameters are the number of weights values that must be handled by the deep neural network at each layer. For the above scenario, the visible (input) layer has 4548 nodes (or neurons). This number is obtained by adding one to the size of the vocabulary which is 4547 in this case. The extra neuron is termed as the bias and is generally present in each layer except the last one. It is not always possible to exhaustively list all the factors which influence the outcome of a problem. The bias allows the neural network to model this unknown factor, and this is done simply by means of a constant value, which is updated on every iteration. To understand the purpose of a bias neuron, an analogy is often made to the c-intercept in the overwhelmingly well-known equation of a straight line: $y = mx + c$. If there is no c value, the line will always go through the origin which may not be the desired result. Also, experimentally, it has been found that the inclusion of a bias neuron generally improves the performance of a neural network.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	1164288
activation_1 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
activation_2 (Activation)	(None, 256)	0
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 256)	65792
activation_3 (Activation)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 30)	7710
activation_4 (Activation)	(None, 30)	0

Total params: 1,303,582
Trainable params: 1,303,582
Non-trainable params: 0

Figure 2. Summary of the deep neural network model.

Each neuron in this first layer is densely connected to the 256 neurons in the following (second) layer. Thus, there are 1,164,288 weight values (parameters) to handle between the visible and the first hidden layer. The bias neuron from one layer is not connected to the bias neuron in the following layer. Thus, in between the first set of hidden layers, we have 257 neurons from the first hidden layer which are connected to 256 neurons in the second hidden layer, which makes a total of 65,792 connections (weights or parameters). Similarly, we have the same number of connections between the second hidden layer and the third one. The third hidden layer is connected to the output layer in which there are only 30 neurons as we have 30 categories (outcomes or outputs). Thus, we have a total of 7,710 (256 x 30) connections between these last two layers.

Coming back to the Keras codes, a 'relu' activation function has been used in each layer, except the last one. ReLU stands for rectified linear unit and is currently the most used function in the world of deep learning. The 'relu' function is a simple algorithm which converts all negative values to zero but keeps all positive numbers to their actual values. However, in the last layer, a 'softmax' function has been used. This function converts the outputs from the last layer in the deep neural network (logits) into a series of probabilities which sums to 1. Thus, each category is assigned a probability and the one with the highest probability is generally considered as the correct prediction. When there are only two classes (binary problem), a 'sigmoid' activation function is used instead. This is a simpler function which outputs either a 0 or 1 depending on a threshold value, which is usually 0.5. For regression problems, a 'linear' activation function is used.

Once the deep neural network is defined, the next step is to compile it. The purpose of compilation is to transform the layers into a series of matrices which can efficiently be executed by either a CPU or a GPU. For multi-class problems, three parameters are often specified. The loss function for multi-class problems is "categorical_crossentropy". For binary problems (only two classes), we can either use "categorical_entropy" or "binary_crossentropy", while for regression problems, the 'mean_squared_error' loss function is used. The purpose of the loss function is to calculate the error between the actual values and the predicted values. The second parameter for the compilation process is about choosing an optimizer which primarily computes the weight gradients and decides on the direction for the next iteration. The 'adam' optimisation algorithm has been used but we have noticed that the 'adadelta' algorithm also provides similar performances on classification problems. Other popular options include the stochastic gradient descent (sgd) or the RMSprop (rmsprop) optimisation algorithms. The third parameter is the choice of metrics to estimate the performance of the model while the network is training. For classification problems, this is usually set to 'accuracy' while for regression problems, the 'mean_squared_error' is usually used. It is possible to include several metrics as this information is provided as a list of values. Furthermore, it is possible to create custom metrics for inclusion in this parameter.

The next step is to fit the network to the data. This is where the actual training happens using the backpropagation algorithm. The training data (x_train) and the targets (y_train) are supplied as separate variables. The network is trained for a specified number of iterations (epochs). A deep learning network with a large number of neurons converges very rapidly on small datasets and therefore a very large number of neurons is unnecessary as this may lead to overfitting. A good value for a specific dataset can only be found through some trial and error. Furthermore, not all the data is used in one go to compute the loss and accuracy in each epoch. It is more common to specify a 'batch_ size', which is the amount of training data (together with its corresponding target) which the network will use before updating the weights during an epoch. This is a convenient feature when dealing with huge data sizes. A value of 1 for 'verbose' allows the progress of the training to be monitored in real-time. It provides information about loss, accuracy and time taken for each batch and/or epoch. This feature can be turned off by setting it to 0.

After training is completed and a model is generated, it must be evaluated. An evaluation must be performed on an unseen dataset (a segment of the original dataset that was not used for training). If the accuracy of the model is above a certain required value, the model can be saved (to disk) and loaded later to make predictions. This sample network is not the only one that can be used for text classification. In fact, any number of hidden layers (minimum is one) can be used between the visible and the output layers. However, through experiments on our dataset, adding a third hidden layer only minimally improved the accuracy at the cost of more training time while a fourth hidden layer actually created a decline in the accuracy.

4.2. Convolutional Neural Networks (CNNs)

Kim [15] was one of the first researchers who employed the Word2vec model on top of a CNN network for text classification. He achieved state-of-the-art performances on several of the datasets. However, it is important to point out that the highest number of categories in the dataset was six and that most of these datasets dealt with very short texts, such as single sentences. Undavia *et al*. [26] achieved their best results on 15 legal categories using Word2vec on CNN.

The network starts by instantiating an object of type Sequential to create a sequential model. The first layer is an Embedding layer which uses pre-trained word embeddings based on Word2vec. The vocab_size parameter is the size of the vocabulary although it is possible to constraint the vocabulary to a smaller size. The embedding_dims parameters is the dimension of the word_embeddings which is 300 for Word2vec. The weights parameter is a list of values which encodes starting values for each word from the vocabulary based on the word embeddings. Input_length is the size of the longest sequence of text which is present in the dataset. When using pre-trained word embeddings, the trainable parameter must be set to false, otherwise, the network will override these weight values on the next iteration. There are three 1-dimensional convolution layers between the input and the output layers. The first parameter in the Conv1D layer is the number of filters (i.e., the number of convolutions) and the second one is the kernel size (i.e., the number of words in the sliding window). In order to reduce overfitting, a dropout mechanism is included after every layer, except the last one.

4.3. Recurrent Neural Networks (RNNs)

Unlike the previous deep neural networks that we have considered so far, a recurrent neural network has memory. This means that it maintains information about content it has already processed and not just on the current segment. There are two main types of RNN: LSTM and GRU. These two types of networks have been designed to solve the vanishing gradient problem, whereby a network may become untrainable as more and more layers are added to it. Although such types of networks are more complex, they do not perform so well on text classification problems, but their representational power has become more apparent in more complex domains such as machine translation and question answering systems.

Undavia *et al*. [26] obtained their worst results with LSTM but results were better with GRU. LSTM and GRU were found to work well on sentiment analysis problems where the texts are relatively short and where only a few categories are used. The other downside of recurrent networks is that they take much more time to train than CNNs or vanilla DNNs. We also have bidirectional LSTM (Bi-LSTM) and bidirectional GRU (Bi-GRU). They are similar to the regular LSTM and GRU but they process the input sequence in both directions (from start to end and then in the reverse way). The representations are then merged into a single one. Bi-GRU and Bi-LSTM usually deliver slightly better performances than their unidirectional counterparts.

5. Experiments, Results and Evaluation

A large number of experiments were conducted using the three different types of neural networks (DNN, CNN and RNN) in combination with different types of pre-trained word embeddings (Word2vec, GloVe and Law2Vec). Table 2 shows the classification accuracy for five different segment sizes of legislations. Eighty percent of the legislation dataset was used for training and the remaining was used for testing. Eight different classifiers were used for comparison purposes. Because of the random initialization step in neural networks, the results are not the same on each run. Thus, all

experiments with deep neural networks (1, 2, 3) were repeated five times and a mean value was calculated. This is also the case for decision trees and random forests which use random numbers in some of their steps. The traditional machine learning classifiers were used with their default parameters.

Table 2. Text classification results with deep neural networks.

#	Classifier	490 Legislations: No. of words				
		500	1000	2000	3000	4000
1	DNN	**59.9**	**60.9**	**60.5**	**60.5**	**59.9**
2	EM + CNN	**55.1**	**55.1**	**55.1**	**55.4**	**56.5**
3	EM + LSTM	33.7	26.5	25.5	24.5	22.5
4	SVM	52.0	45.9	44.9	41.8	40.8
5	KNN	31.6	33.7	33.7	33.7	34.7
6	Naïve Bayes	45.9	46.9	51.0	46.9	48.9
7	Decision Trees	39.5	38.8	36.7	34.3	33.3
8	Random Forests	34.3	36.1	37.1	31.0	39.8

*EM: Word2vec word embeddings.

The vanilla DNN, in which the inputs are in the form a document term matrix delivers the best performance in all the five scenarios, with a mean accuracy of about 60% in all the runs. However, increasing the size of the input had no impact on the accuracy, which means that the first 500 words of legislations have enough predictive power to classify them. The second-best classifier is the CNN which uses word embeddings. The results shown were obtained with the Word2vec model. The classification accuracies obtained with GloVe were slightly less. Moreover, using a domain-specific word embeddings such as the Law2Vec model did not perform better than the general Word2vec model [5]. There is a slight improvement in accuracy when the document size becomes larger with CNN but this is not substantial. The same is true for K-Nearest Neighbour (KNN).

For SVM, there is a significant drop in accuracy from 52% to 40.8% when moving from a size of 500 words to 4000 words. The same holds for Decision Trees although the drop is only about half in magnitude compared with SVM. There is no clear trend for Naïve Bayes and Random Forests. The worst results were obtained with LSTM. LSTM uses word embeddings as input and there is a drop in the accuracy with an increase in document size. Furthermore, training using the LSTM network is extremely slow compared with CNN, DNN and other traditional classifiers. It took 27 minutes to train on a data size of 500 words per document using a laptop with an Intel Core-i3 processor, 16GB RAM and an SSD of 100 GB. It took 10 hours to train 4000 words on a set of 392 documents. Experiments were also performed using GRU, Bidirectional-LSTM and Bidirectional-GRU but similar results to LSTM were obtained. GRU is about twice as fast as LSTM but LSTM is faster than Bi-GRU.

Figures 3 and 4 show how the training loss, validation loss, training accuracy and validation accuracy vary with the number of epochs. From Figure 3, we can see that the training loss decreases very rapidly to a value close to zero in less than 50 iterations (epochs), while the validation loss reaches its minimum value only after about 20 steps. From Figure 4, we can see that the validation accuracy is at its highest value only after 24 epochs. The training accuracy starts from a value close to zero to reach a value of close to 100% in less than 50 epochs. Although, the neural network is able to understand the training data fully, we can see that the validation accuracy does not follow the same trend but rather stagnates as a value close to 60%. This is a case of overfitting and this shows that training a neural network beyond a certain number of steps does not necessarily improve the accuracy.

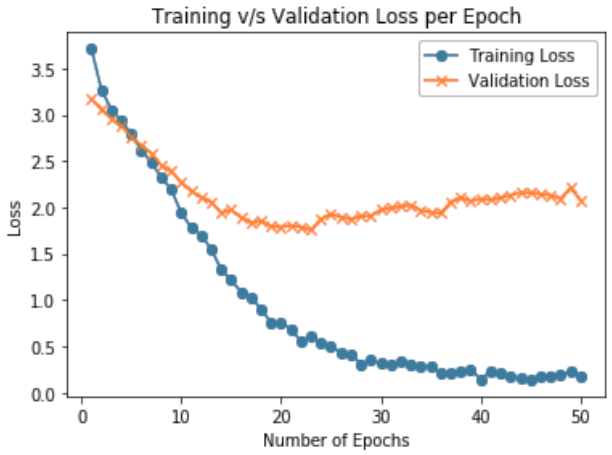


Figure 3. Training and validation loss for DNN on 500 words.

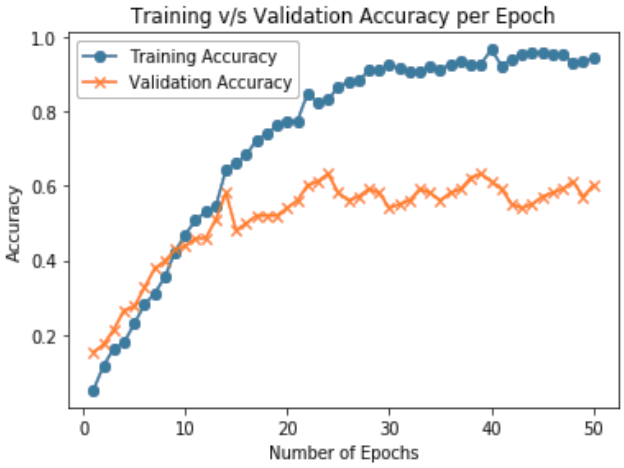


Figure 4. Training and validation accuracy for DNN on 500 words.

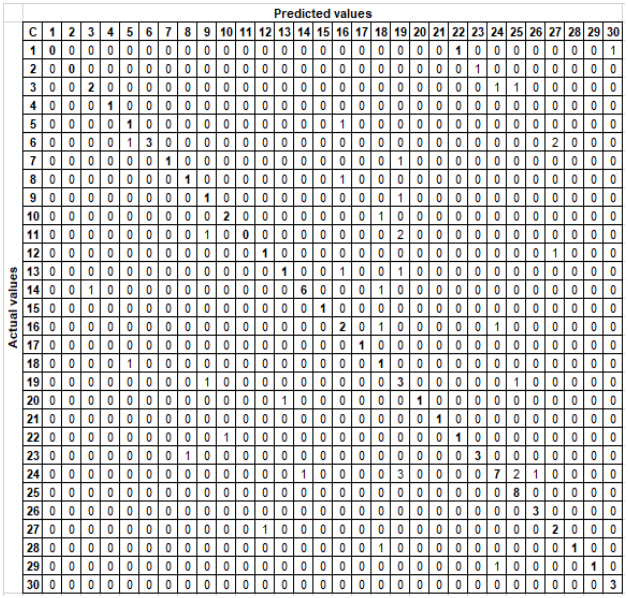
Table 3. Classification report for DNN on 500 words.

#	Category of Legislations	No. of docs for training	No. of docs for testing	Precision	Recall	F1-Score
1	Agriculture, environment and natural reserves	10	2	0.0	0.0	0.0
2	Animal welfare	4	1	0.0	0.0	0.0
3	Arts and cultural heritage	14	4	0.7	0.5	0.6
4	Aviation	6	1	1.0	1.0	1.0
5	Banking	9	2	0.3	0.5	0.4
6	Businesses and companies	22	6	1.0	0.5	0.7
7	Chemicals	9	2	1.0	0.5	0.7
8	Citizenship	9	2	0.5	0.5	0.5
9	Civil procedures	10	2	0.3	0.5	0.4
10	Construction, buildings and development	12	3	0.7	0.7	0.7
11	Criminal procedure	14	3	0.0	0.0	0.0
12	Customs and excise	7	2	0.5	0.5	0.5
13	Diplomatic immunities and privileges	11	3	0.5	0.3	0.4
14	Education, training, research and standards	30	8	0.9	0.8	0.8
15	Emergency services and force majeure	4	1	1.0	1.0	1.0
16	Finance	15	4	0.4	0.5	0.4
17	Gambling	4	1	1.0	1.0	1.0
18	Information & Comm. Technologies	6	2	0.2	0.5	0.3
19	Judiciary	20	5	0.3	0.6	0.4
20	Labour issues	10	2	1.0	0.5	0.7
21	Land laws	6	1	1.0	1.0	1.0
22	Marine and ocean resources	8	2	0.5	0.5	0.5
23	Medical services, food, health and safety	18	4	0.8	0.8	0.8
24	Social security and welfare	54	14	0.7	0.5	0.6
25	Social, cultural and religious activities	30	8	0.7	1.0	0.8
26	Sugar sector	11	3	0.8	1.0	0.9
27	Tax	12	3	0.4	0.7	0.5
28	Trade, industry and exports	10	2	1.0	0.5	0.7
29	Transportation laws	7	2	1.0	0.5	0.7
30	Utilities	10	3	0.8	1.0	0.9
Total/Average		**392**	**98**	**0.63**	**0.59**	**0.58**

Table 3 shows the detail classification results for the legislation dataset for a DNN and document size of 500 words. The number of training documents, number of testing documents, precision, recall and the f1-score are shown for each category of legislation. We have used a stratified sampling strategy to make sure that at least one legislation from each category is present in the testing set. The mean precision for the overall testing set is 63%. However, a perfect precision score of 100% was obtained for nine categories. There were eight categories for which the precision was less than 50%. Similarity, the mean recall is 59% and a perfect score of 100% was obtained for seven categories. There were only four categories for which the recall was less than 50%. It is interesting to note that out of the five categories which has the least number of training samples in the dataset, four of them have achieved a perfect score for both recall and precision. If is often believed that with deep learning approaches, the more data that you have the better is the performance and if you have only a small amount of data, the results are not usually so good. However, experiments on our legislation dataset do not necessarily support this belief. Firstly, we saw that increasing the document size did not increase the accuracy with DNN. Moreover, there was only a minimal improvement in the performances of the CNN model. Secondly, the Pearson product moment correlation coefficient between the number of training documents per category and the f1-score was less than 10%, showing that there is no linear relationship between these two variables. In many earlier studies on deep learning, researchers have intentionally removed categories with less than a certain number of instances as it was believed that this would negatively impact the overall performance of the models.

Figure 5 shows the corresponding confusion matrix for the classification report shown in Table 3. The confusion matrix allows a much deeper incursion into the results and allows us to understand the misclassifications in greater depth. For example, the two testing documents for the Agriculture, environment and natural reserves categories were not correctly classified. One of them was classified into the Marine and ocean resources category and the other one into the Utilities category. Both misclassifications are quite understandable as there is definitely some natural overlap between the 1st and the 22nd category. Utilities also is a fairly broad category which englobes energy, electricity, water, wastewater and post office services. In the Banking category, there were two documents which were in the testing set. One of them has been correctly classified while the one has been misclassified into the Finance category. We also notice a value of 1 in rows 6 and 18 in column 5 (Banking). This means that one legislation from the Businesses and companies category and one from the Information and Communication Technologies (ICT) category have been misclassified into the Banking category. These

misclassifications can provide helpful insights into the suitability of categories that have been used in this study. They may also help us to identify legislations that have been wrongly classified by the human expert.



Actual values \ Predicted values	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	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	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	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	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	0	0	0	0	0	0	0	0	6	0	0	0	1	0	0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	1	0	0	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	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3	0	0	0	0	0	7	2	1	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Figure 5. Confusion matrix for DNN on 500 words.

6. Conclusions

Deep learning is a machine learning approach in which successive layers of meaningful representations are learnt from raw or processed data using neural networks. The major benefit of deep learning is the possibility to do away with feature engineering, which is one of the main bottlenecks in machine learning pipelines. In this work, we have used deep learning for the classification of 490 legislations from the Republic of Mauritius. Various deep learning architectures were considered but we achieved our best results with what is commonly known as a vanilla deep neural network, as this is the simplest type of neural network that is possible to develop. Thus, we obtained an accuracy of 60.9% for the legislation dataset with 30 categories with a deep neural network consisting of four dense layers. The convolutional neural network also delivered very good performances but in general it was 5% less than the DNN but still better than most traditional machine learning classifiers. The long short-term memory recurrent neural networks and its variants were not appropriate for this dataset and had the worst performances. They were also extremely slow. The general pre-trained Word2vec embeddings did slightly better than Law2Vec, which is a domain-specific word embeddings but Law2Vec is trained on at least ten times less data than Word2vec and GloVe models.

This work will be very beneficial to barristers, notaries, attorneys, judges, magistrates, court officers and other legal professionals who have to deal with all types of legislations on a daily basis. Legal research is a major component of their work and this classification based on deep learning will help them locate related legislations in less time. It is possible to further improve this work by using a larger dataset and updated word embeddings trained on larger corpora. Further improvements may be possible through the creation of hybrid deep learning architectures based on the combination of the more basic types.

References

[1] Bilgin M. and Koktas H., "Sentiment Analysis with Term Weighting and Word Vectors," *The International Arab Journal of Information and Technology*, vol. 16, no. 5, pp. 953-959, 2019.

[2] Boella G., Caro L., and Humphreys L., "Using Classification to Support Legal Knowledge Engineers in the Eunomos Legal Document Management System," *in Proceedings of 5th International Workshop on Juris-informatics*, Kagawa, 2011.

[3] Burns G., Li X., and Peng N., "Building Deep Learning Models for Evidence Classification from the Open Access Biomedical Literature," *Database*, vol. 1019, 2019.

[4] Chalkidis I. and Androutsopoulos I., "A Deep Learning Approach to Contract Element Extraction," *in Proceedings of 30th International Conference on Legal Knowledge and Information Systems*, Luxembourg, pp. 155-164, 2017.

[5] Chalkidis I. and Kampas D., "Deep Learning in Law: Early Adaptation and Legal Word Embeddings Trained on Large Corpora," *Artificial Intelligence and Law*, vol. 27, pp. 171-198, 2019.

[6] Chollet F., *Deep Learning with Python*, Manning Publications Co, 2018.

[7] Collarana D., Heuss T., Lehmann J., Lytra I., Maheshwari, G., Nedelchev R., and Trivedi P., "A Question Answering System on Regulatory Documents," *in Proceedings of 31st Annual Conference on Legal Knowledge and Information Systems*, Groningen, pp. 41-50, 2018.

[8] Curran T. and Thompson P., "Automatic Categorization of Statute Documents," *in Proceedings of the 8th ASIS SIGICR Classification Research Workshop*, Washington, pp. 19-30, 1997.

[9] De Maat E. and Winkels R., "Automatic Classification of Sentences in Dutch Laws," *in Proceedings of 21st Annual Conference on Legal Knowledge and Information Systems*, Florence, pp. 207-216, 2008.

[10] Francesconi E. and Passerini A., "Automatic Classification of Provisions in Legislative Texts," *Artificial Intelligence and Law*, vol. 15, no. 1, pp. 1-17, 2007.

[11] Giri R., Porwal Y., Shukla V., Chadha P., and Kaushal R., "Approaches for Information

Retrieval in Legal Documents," *in Proceedings of 10th International Conference on Contemporary Computing*, Noida, pp. 1-6, 2017.

[12] Goltz N. and Mayo M., "Enhancing Regulatory Compliance by Using Artificial Text Mining to Identify Penalty Clauses in Legislation," *in Proceedings of the International Workshop on Mining and Reasoning with Legal Text*, London, pp. 1-9, 2017.

[13] John A., Luigi D., Guido B., and Cesare B., "An Approach to Information Retrieval and Question Answering in the Legal Domain," *in Proceedings of 10th International Workshop on Juris-informatics*, Kanagawa, pp. 1-14, 2016.

[14] Kastrati Z., Imran A., and Yayilgan S., "the Impact of Deep Learning on Document Classification Using Semantically Rich Representations," *Information Processing and Management*, vol. 56, no. 5, pp. 1618-1632, 2019.

[15] Kim Y., "Convolutional Neural Networks for Sentence Classification," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, pp. 1746-1751, 2014.

[16] Kowsari K., Brown D., Heidarysafa M., Meimandi K., Gerber M., and Barnes L., "HDLTex: Hierarchical Deep Learning for Text Classification," *in Proceedings of the 16th IEEE International Conference on Machine Learning and Applications*, Cancun, pp. 364-371 2017.

[17] Kowsari K., Meimandi K., Heidarysafa M., Mendu S., Barnes L., and Brown D., "Text Classification Algorithms: A Survey," *Information*, vol. 10, no. 4, 2019.

[18] Lai S., Xu L., Liu K., and Zhao J., "Recurrent Convolutional Neural Networks for Text Classification," *in Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Austin, pp. 2267-2273, 2015.

[19] Lippi M., Palka P., Contissa G., Lagioia F., Micklitz H., Sartor G., and Torroni P., "CLAUDETTE: An Automated Detector of Potentially Unfair Clauses in Online Terms of Service," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 117-139, 2019.

[20] Liu L., Liu K., Cong Z., Zhao J., Ji Y., and He J., "Long Length Document Classification by Local Convolutional Feature Aggregation," *Algorithms*, vol. 11, no. 8, pp. 108, 2018.

[21] Luo B., Feng Y., Xu J., Zhang X., and Zhao D., "Learning to Predict Charges for Criminal Cases with Legal Basis," *in Proceedings of Conference on Empirical Methods in Natural Language Processing*, Copenhagen, pp. 2727-2736, 2017.

[22] Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., "Distributed Representations of Words and Phrases and their Compositionality," *in Proceedings of the 27th Conference on Advances in Neural Information Processing Systems*, Lake Tahoe, pp. 3111-3119, 2013.

[23] Nguyen G., Dlugolinsky S., Bobak M., Tran V., Garcia A., Heredia I., Malik P., and Hluchy L., "Machine Learning and Deep Learning Frameworks and Libraries for Large-Scale Data Mining: A Survey," *Artificial Intelligence Review*, vol. 52, pp. 77-124, 2019.

[24] O'neill J., Buitelaar P., Robin C., and Brien L., "Classifying Sentential Modality in Legal Language: A Use Case in Financial Regulations, Acts and Directives," *in Proceedings of the 16th International Conference on Artificial Intelligence and Law*, London, pp. 159-168, 2017.

[25] Purpura S. and Hillard D., "Automated Classification of Congressional Legislation," *in Proceedings of the 7th Annual International Conference on Digital Government Research*, San Diego, pp. 219-225, 2006.

[26] Undavia S., Meyers A., and Ortega J., "A Comparative Study of Classifying Legal Documents with Neural Network," *in Proceedings of the Federated Conference on Computer Science and Information Systems*, Poznan, pp. 515-522, 2018.

[27] Van-Noortwijk K., Visse J., and De Mulder R., "Ranking and Classifying Legal Documents using Conceptual Information," *Journal of Information, Law and Technology*, vol. 2006, pp. 1-15, 2006.

[28] Waltl B., Bonczek G., Scepankova E., and Matthes F., "Semantic Types of Legal Norms in German Laws: Classification and Analysis Using Local Linear Explanations," *Artificial Intelligence and Law*, vol. 27, no. 1, pp. 43-71, 2019.

[29] Yang Z., Yang D., Dyer C., He X., Smola A., and Hovy E., "Hierarchical Attention Networks for Document Classification," *in Proceedings of the Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, pp. 1480-1489, 2016.

Sameerchand Pudaruth is a Senior Lecturer and Head of the ICT Department at the University of Mauritius. He has a PhD in Artificial Intelligence (AI) from the University of Mauritius. He is a Senior member of IEEE, founding member of the IEEE Mauritius Subsection and the current Vice-Chair of the IEEE Mauritius Section. He is also a member of the Association for Computing Machinery (ACM). His research interests are Artificial Intelligence, Machine Learning, Data Science, Machine Translation, Computer Vision, Robotics, Mobile Applications, Web Technologies, Multimedia, Blockchain and Information Technology Law. He has written more than 70+ papers for national and international journals and conferences. He has been in the organising committee of many successful international conferences.

Sunjiv Soyjaudah obtained his LLB (Hons) degree from the University of London. He was called to the Bar of England and Wales at the Honourable Society of the Middle Temple and subsequently joined the Mauritian Bar. He is presently the Head of Soyjaudah Chambers. He has been the Dean of the Faculty of Engineering at the University of Mauritius. He successfully supervised over 20 PhD students and 2 post-doctoral fellows. He has published over 250 research papers in international journals and conferences. He was successively appointed Executive Director of the Tertiary Education Commission and the Information and Communications Technologies Authority.

Rajendra Gunputh is the Dean of the Faculty of Law and Management at the University of Mauritius with a Personal Chair in International Comparative Law with Specialisation in Employment Law and Trade Union Law. He holds a double PhD (Law) from the l'Université Paris V-René Descartes (France) in Public Law and l'Université de la Réunion in Private Law both with the highest distinctions (First Class). He is currently the Dean of the Faculty of Law and Management at the University of Mauritius. He has authored more than 100 articles which have been published worldwide and several treaties on Mauritian Law and various books and manuals on Law (Labour Law, Media Law, International Humanitarian Law).