International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

A Classification of Quran Verses Using Deep Learning

Abdelkareem M. Alashqar[1]

[1] *Faculty of Information Technology, Islamic University of Gaza, P.O. Box 108, Gaza, Palestine*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: Understanding the topics of Quran verses is considered as a main interest of Islamic Scholars, specialists of Quran studies and others. The traditional classification of Quran verses can be simplified and improved using the automated techniques such as Natural Language Processing (NLP) and Machine Learning (ML). While the majority of the current studies have used traditional ML approaches with small datasets, we used the Deep Learning (DL) algorithms with larger dataset for classifying Quran verses. This paper proposes a method for multi-label classification for accurately classifying Quran verses based on 12 predefined main topics using DL. We designed a structured method that consists of multiple steps for achieving the objective of this study. Firstly, a dataset of labeled Quran verses is collected, organized and converted to sequences of numbers to be understood by the DL models. The skip-gram algorithm of Word2Vec is used for considering the semantic of text to improve the models' performances. Then the embedding vectors are fed to two different DL models which are RNN and CNN to classify verses. The results of DL classifiers are evaluated based on accuracy, precision, recall, F1-score, and hamming loss where the cross-validation technique is used for more accurate results. The values of 90.38%, 96.98%, 92.49%, 93.81% and 0.0126 for accuracy, precision, recall, F1-score and hamming loss respectively were achieved as best results. The findings of this study help specialists of Quran studies to gain more insight for easily studying and teaching the topics discussed by Quran verses.

Keywords: Quran, Natural Language Processing, Multi-label classification, Deep learning, RNN, CNN, Word2Vec

1. INTRODUCTION

The Quran is the Word of Allah (God) revealed and sent down to Prophet Muhammad through the agency of the Archangel Gabriel. Quran was firstly transmitted orally between Muslims in the era of the Prophet and later was written in a book using Arabic language [1] by companions of the Prophet and affiliates. Quran is the main reference book for all Muslims within the globe. They used it for worship, reading, retrieving information and discussing topics. The Quran consists of 114 chapters called "Suras" and distributed into 30 fairly unified sized parts. While mostly each part includes more than one chapter, some chapters include more than one part such as "Al-Baqarah" and "Al-Imran" chapters. The Quran consists of 6,236 verses called (Ayat) with a total number of approximately 77,430 individual Arabic words. Some words are replicated so the total number of unique Arabic words is 14,870. The exact number of individual and unique words may vary depending on how the distinct words are defined and counted [2].

Each word in the Quran has diacritics which are cedillas written as signs above or below the word letter where the place of this diacritic not only affects the pronunciation of the word but also the meaning of it. So, diacritics are always considered as main part of the word as well as its original alphabet.

Quran corpus is of interest to many researchers. For instance, the research studies in [3] [4] provided significant results for using Arabic Quran corpus in comparing various algorithms of Part-of-Speech (POS) tagging.

Understanding the topics that included an embedded in the verses is considered as main interest for Islamic scholars and others who want to study and understand Quran. The Islamic scholars work on classifying Quran verses manually into specific topics where an individual verse can usually handle one or more topics at the same time. This means that a multi-label approach is used for classifying Quran verses. With the existence of Natural Language Processing (NLP) and Deep Learning (DL) methods it is more significant to apply the automatic classification of Quran verses into specific and predefined labels. There is no universal consensus about the topics of Quran verses because determining the number and the type of these topics are subject to the diligence of Islamic scholars and specialists of Quran studies. Additionally, the Quran verses can be classified into major topics where each of these major topics can be divided into sub-topics and each of these sub-topics can be additionally divided into more sub-topics.

While there are various traditional Machine Learning (ML) approaches used in the automatic classification of

E-mail address: aashgar@yahoo.com

Quran verses, the DL approaches can foster this classification and can produce more accurate results. Moreover, the DL approaches can be applied to process the sematic techniques such as word embeddings that have the capability of considering the relationships between the words of the natural language text.

The main objective of this paper is the use of the DL and word embedding techniques to automatically and accurately classifying Quran verses into predefined topics. This paper meaningfully contributes as follows:

- Prepare a dataset that contains considerable number of multi-labeled verses that reviewed and refined by an Islamic scholar. The number of verses reached 4,911 where each verse is classified into at least one of 12 main topics.
- Develop a skip-gram word embeddings technique to create data vectors of Quran verses for considering the semantic of text. The whole words of original Quran were used for training to create the word embedding.
- Build DL models for the automatic multi-label classification of Quran verses that take into account the word embeddings as well as the diacritized and undiacritized text of verses.

This contribution helps specialists of Quran studies to take more insight and knowledge about the discussed topics of Quran verses. Moreover, to facilitate information retrieval and to share knowledge between Muslims who are interested in studying Quran.

The remaining parts of this study are arranged as follows: the literature review of multi-label classification of Quran verses is provided in Section 2. Section 3 describes the scope and the adopted methodology of this study. The experimental results, evaluation and discussion are introduced in Section 4. Section 5 presents the threats to validity. Section 6 concludes the paper and provides future outlook.

2. RELATED WORK

There is a limited number of research works found in the literature for the automatic classification of Quran verses. A multi-label classification approach is always used in classifying the Quran verses that written originally in Arabic. While some researchers performed this classification on undiacritized text (text stripped out of diacritics) fewer researchers achieved it on diacritized text (text has diacritics). However, there is a considerable number of research studies for classifying Quran verses were achieved using English translated Quran.

As an early study in classifying Quran verses, the authors in [5] applied the approaches of lexical frequency of data and the hierarchical cluster analysis of Quran text in order to give understanding of thematic relationships between Quran chapters (Suras).

For the automatic classification of Quran verses the authors in [6] applied a traditional linear classification technique to categorize the verses of “Fatiha” and “Yaseen” chapters into general subjects. Later, the authors in [7] used four ML classifiers to classify sample of Quran into fourteen topics and they found that the Naïve Bayes (NB) achieved the best performance with comparison of the other experimented classifiers.

Recently the authors in [8] applied an ensemble multi-label ML approach using word embedding for classifying the verses of “Al-Baqarah” chapter into 393 topics. They manually labeled the verses based on “Mushaf Al-Tajweed” and built a Word2Vec algorithm of word embedding using classic Arabic corpus to help classifiers in understanding the semantic of input words. They also built a voting algorithm to evaluate the performance of three traditional ML classifiers were considerable performances results were reached.

The following research works have used the English translated corpus of Quran for classifying the topics of verses. For instance, the authors in [9] adopted the approach of DL to classify the Quran verses into three main classes using the English translated Quran. They used 150 verses for training and 30 verses for evaluation. Also, the authors in [10] applied the multi-label classification approach to classify the verses of English translated Quran into 15 labels. They compared the classification results of NB, SVM and Artificial Neural Network (ANN) classifiers and found that the NB gives the best performance.

Additionally, the authors in [11] used the English translated Quran to achieve a multi-label classification of verses using 15 labels. They used the multinomial NB algorithm in classification and followed multiple steps of data preprocessing that include stemming, tokenization, case folding and Bag of Words (BOW) vectorization technique and reached a best hamming loss of 0.1247. While the authors in [12] used the tree augmented NB with mutual information techniques for multi-label classification of translated English Quran and reached a best hamming loss of 0.1121.

The authors in [13] proposed a Group-Based Feature Selection (GBFS) technique for improving the classification of Quran verses. They used this approach to label the verses of “Al-Baqarah” and “Al-Anaam” chapters into three different classes based on English translated Quran and Tafseer (commentary). They evaluated the verses classification using four ML classifiers where a considerable accuracy results were reached.

The authors in [14] applied the Back Propagation Neural Network (BPNN) for multi-label classification of Quran verses. They used the TF-IDF vectorization technique for feature extraction and evaluated the BPNN performance based on the Stochastic Gradient Descent (SGD) and "Adam" optimizers and reached a best hamming loss of 0.129.

The authors in [15] proposed an ANN with a Quasi-Newton updating procedure technique called ML4BFGS for multi-label classification for translated English Quran. They compared it with five other Newton training techniques where the ML4BFGS outperformed the compared techniques and performed 88%, 71%, 87% and 84% for precision, recall, F-score and hamming loss respectively.

From the previous stated related works, it is obvious that most studies of classifying Quran verses used English translated Quran. Although a very limited studies used the ANN approach in verses classification rather than the DL approach, this was performed only in English translated Quran corpus. For research studies that used the Arabic Quran and to the best of our knowledge the classification was performed on verses stripped out of diacritics.

Our study differs from previous studies in that it adopts the approach of DL in classifying Quran verses on Arabic Quran with diacritized text. Nonetheless, we will perform the classification on undiacritized Arabic text of Quran. Moreover, we used a considerable number of verses in our study that reaches 4,911 verses. The word embedding techniques also is adopted to consider the sematic of words and their relationships within the text context.

3. SCOPE AND METHODOLOGY

This section introduces the used dataset and the proposed research method that adopted in this paper.

A. The Dataset

The dataset is mainly created depending on the information provided from "Quran by Subject" web resource [16] and it is augmented by additional information collected from the Arabic Original Quran [17].

The Quran by Subject provided a text file that includes information for large number of Quran verses that categorized and titled by their related subtopics. More specifically there is a considerable number of verses that stored and written in the text file beneath their related subtopics. We processed the text file computationally using complex regular expression (RegEx) programming techniques to create an organized CSV (comma-separated values) file. This initial CSV file includes four columns which are Verse Text, Verse ID, Chapter Name and Subtopic. A new column is added to this initial file for representing the Chapter Id. To avoid any possible typos in the verse text the correct one is taken from the standard original Quran. Then each subtopic is replaced by its major topic based on the information provided by the main source of the text file and with the help of an Islamic Scholar who is specialist in Quran studies. More precisely a number of 167 subtopics are reduced to 12 major topics[1]. Examples of three major topics and their subtopics are provided in Table I. Each row in the CSV file represents information about one labeled verse where the number of individual rows in the CSV file reaches 6,246. A verse can appear more than one time when it handles more than one topic. So, a new processing is achieved on the CSV file so that a verse appears only in only one row in the file where its related topics is collected in a list[2]. Eventually each verse is classified into at least one of 12 major topics which are: "allah", "disbelief", "evils", "faith", "mohammad", "quran", "rulings", "stories", "struggle", "universe", "unseen" and "worship". This means that the verses are distributed over 12 major topics where the total number of individual classified verses became 4911. The major topics with Arabic and English are shown in Table II. It is important to note that the classified verses are belong to 112 from the 114 chapters of Quran where "Alfil" and "Almaoun" chapters have no classified verses in the dataset.

After analyzing the CSV file for some statistics, it is found that a verse has a minimum of one topic and a maximum of four topics. Figure 1 shows the distribution for the number of topics per verse. As shown in the Figure 1 there are 3,700 verses have one topic, 1,095 verses have two topics, 108 verses have three topics and 8 verses have four topics[3] . It is important to note that the dataset is unbalanced. Figure 2 depicts graphically the size of verses that belong to each major topic and Table III shows specifically the number of verses that discussed by each individual major topic.

B. Research Method

To achieve the main objective of this study the main steps that shown in Figure 3 will be performed.

1) Data Preprocessing

In this step the data preprocessing that stated previously in the "The Dataset" subsection was performed. An additional column on the CSV file was added that includes a verse text after removing diacritics in order to apply the classification process on diacritized and undiacritized verses text.

2) Vectorizes Verses Text

The machine learning models such as neural networks do not understand or accept input as text. Instead, the text must be converted to numbers. This type of conversion is called vectorization. So, the words of the dictionary that appeared in the verses in the topic-based Quran dataset must be converted to a sequence of integers where each integer represents a particular word in the dictionary. The neural network accepts verses as a fixed length vectors so the short

[1] The details of all subtopics are found in [18]: Section: 2.5.

[2] The processing steps of this section are provided as programming code in [18]: Sections 2.3 to 2.5.

[3] Programming source code for dataset statistics is provided in [18] Section 2.6.

TABLE I. Examples of three major topics and their related sub-topics

Major topic	Related sub-topics
Faith (الإيمان)	“وصف المؤمنون”، “الرؤي و تفسيرها”، “نصائح للمسلم ”، “فضل المؤمن علي الكافر”، “الحكمه”، “الإيمان”، “الأعراب”، “أولياء الله ”، “الدعوه إلي الله ”، “أوامر بأخلاق وأفعال حميده”، “السلم والسلام والاصلاح بين الناس”، “البر”، “الابتلاء و الاختبار”
Rulings (الأحكام)	“قصاص وحدود”، “طاعه الله و الرسول وأولي الامر”، “الحكم بما أنزل الله ”، “أحكام وأوامر للأمه والمسلمين”، “العدل والقسط والوفاء”، “الزكاه”، “الطلاق”، “الربا”، “أحكام وأصول العلاقه الزوجيه”، “الزنا”، “آداب الكلام مع رسول الله محمد عليه الصلاه والسلام”، “الأموال”، “الزواج والنكاح”، “الطعام”، “وحده المؤمنين والاعتصام بحبل الله ”، “القتل”، “الأمر بالمعروف والنهي عن المنكر والنصيحه”، “تغيير القبله”، “اليتامي”، “الورث واحكامه”، “حريه الاعتقاد”، “الارتداد”، “الخمر”، “الوصيه”، “الشورى”، “كفارات”، “الدَين”، “الميسر”، “تحريم ”، “لا إكراه في الدين”، “الزينه”، “حلف اليمين”
Quran “(القرآن)”	“قرآن”، “حروف أول السور”، “إعجاز قرآني”، “نسخ آيات”

TABLE II. The list of used major topics

Topic	Details	Topic	Details	Topic	Details
Rulings	أحكام الاسلام	Mohammad	عن النبي محمد	Disbelief	الكفار والأمم السابقة
Allah	الله في القرآن	Stories	القصص والتاريخ	Evils	مفاسد وموبقات
Faith	الإيمان والمؤمنون	Worship	العبادات	Universe	الكون ومخلوقات الله
Unseen	الغيب والبرزخ	Struggle	جهاد ومعارك	Quran	القرآن

sentences are padded to a length equals to the longest verse in the dataset.

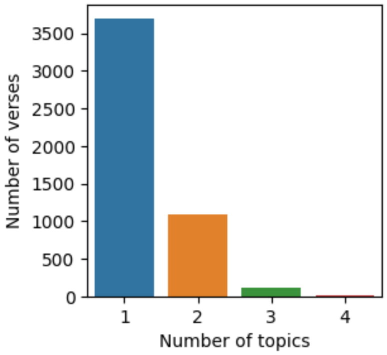


Figure 1. The Distribution of the number of topics per verse

3) Build Word Embeddings

Word embedding is a DL technique used to overcome the problem of considering the word independently when processing a sequence of words in the text. Based on word embeddings concepts the words that happen in similar contexts are likely have similar meanings. Word embeddings consider the semantic and the meaning of the word based on the technique of distributed representation. This technique tries to find the meaning of a word based on its relations with other words of its context [19]. Currently, word embedding is a foundational approach for various types of NLP techniques such as text classification. clustering, sentiment analysis and POS.

Word2Vec and GloVe are well known techniques used for static word embedding. The static embeddings are created based on large and finite number of words that form a dictionary where the keys of the dictionary are the

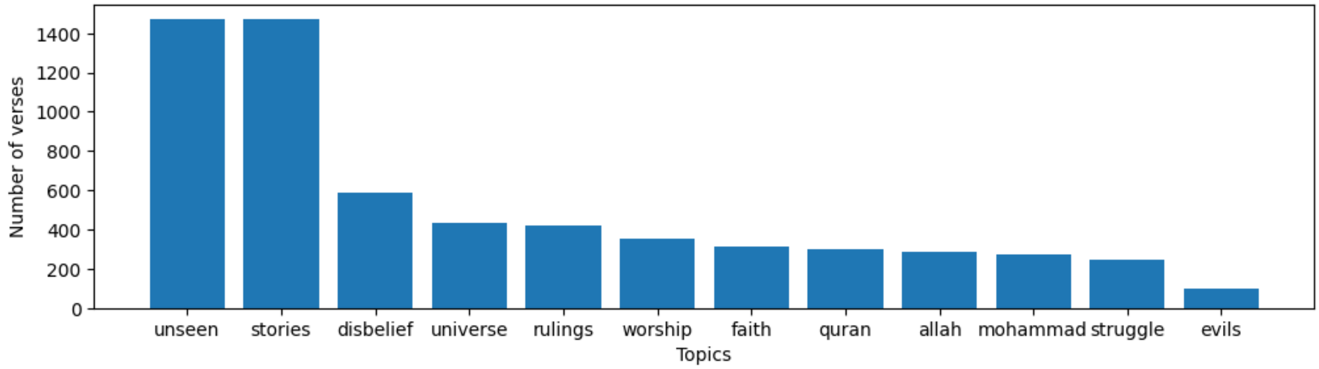


Figure 2. The number of verses for each individual topic

TABLE III. Details of the number of verses in each individual topic

Topic	Number of verses	Topic	Number of verses
Unseen	1474	Faith	310
Stories	1469	Quran	297
Disbelief	586	Allah	288
Universe	436	Mohammad	271
Rulings	421	Struggle	245
Worship	350	Evils	99

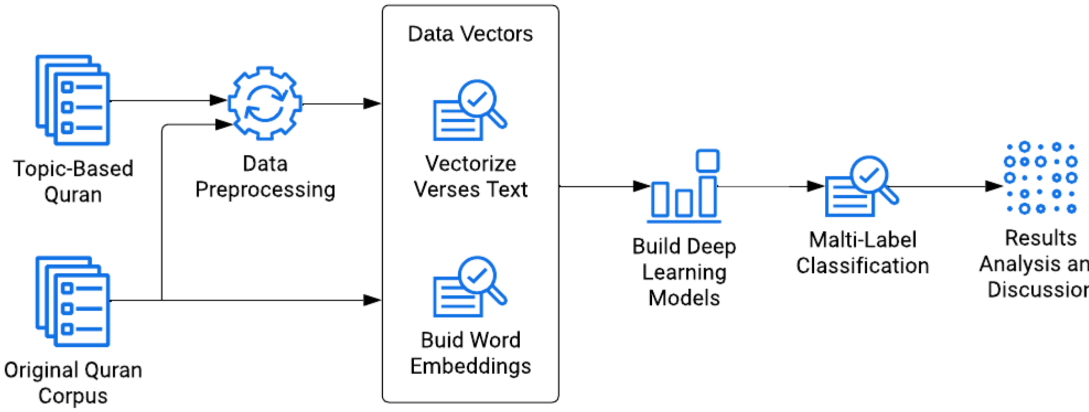


Figure 3. The main steps of the research method

words and the values are their corresponding vectors. The drawback of static embedding occurs when the needed word does not exist in the dictionary.

Word2Vec is a supervised model that considers the structure of the natural language to produce labeled training data. There are two algorithms for Word2Vec which are Continuous Bag of Words (CBOW) and skip-gram. CBOW predicts a word based on a given surrounding words where the order of surrounding words does not influence the word prediction. The skip-gram predicts the surrounding words based on the context word. While skip-gram gives better results in predicting infrequent words, the CBOW is faster.

Global vectors for word representation (GloVe) is an unsupervised learning technique for producing vector representations for words. While Word2Vec is a predictive model the GloVe is a count-based model. In GloVe a large matrix is created where the rows represent the words and the columns represent their context which are the sequence of words [19].

We created our own embedding using "Gensim" which is an open-source library in Python used for extracting semantic meaning from text. We used the whole original Quran words as a corpus for training a Word2Vec model to create a binary file of words embedding. We used the following parameters in training the model which are: (sg=1) for skip-gram training, (window=5) for window size which is 5 words after and 5 words before the examined word, and (size=300) to produce 300 embeddings dimension[4]. The main reason for selecting the original Quran because it is the word of Allah that characterized by its inimitability and has miracle quality in syntax, form and content that cannot matched by human ability. However, we think that the existing algorithms of word embedding will stay have limited capability in considering the semantic and the root meaning of Quran verses. Additionally, to the best of our knowledge, the existing Arabic corpora used for word embedding are collected without considering the diacritics of the words since considering the diacritics of words is one of the main purposes of our study.

4) Build Deep Learning Models

Deep learning transformed machine learning in various fields such as NLP, speech recognition, computer vison and object detection. DL models differ from traditional models in that they can learn many features from data source rather than using limited features engineered by human. This is because DL models have a capability of recognizing the features using weights of large number of parameters. However, it is still hard to understand what features are learned by deep models and how they are learned [20].

A DL model is a deep artificial neural network of many layers. It essentially includes a visible input layer, many hidden interconnected layers in the middle and a visible output layer at the end. This type of deep models is called multi-layer perceptron (MLP) where a perceptron is a name given for a model that has only one single linear layer. The power of deep models in learning is the good features given by the hidden layers.

DL models applies a function using addition and multiplication to map inputs to their outputs. The learning of this function can be improved when it is combined with nonlinear activation function such as Rectified Linear Units (ReLUs). Additionally, the learning of the deep models can be optimized using techniques such as "Adam" optimizer where the rate of each individual parameter is updated for improvement. The flexibility of increasing the number of hidden layers and using regularizes such as normalization layers gives more improvement in the performance of DL models.

The deep neural networks apply the concept of backpropagation of the error signal by computing the gradients of the weights of a given layer regarding to the loss then pushing the updated weights in the reverse direction of the gradient for minimizing the loss and hence optimizing the network parameters.

There are various deep learning models found in the literature. These models can be separated into three groups. The non-sequential models that handle only a single input at a time for both training and prediction such as Conventional Neural Network (CNN), the sequential models that deal with sequences of inputs of arbitrary length such as Recurrent Neural Network (RNN) , and the attention-based models that handle the sequence at one time such as BERT [20].

To achieve the purpose of this study we used the CNN and RNN models which are described in the next subsections.

- Recurrent Neural Network (RNN)

An RNN is a type of deep learning algorithms that uses sequential data. It is usually used for NLP, language translation, and speech recognition. Whereas traditional deep neural networks process inputs and outputs independently through the processing steps, the RNN is characterized by its memory because its takes information from previous inputs to affect the current input and output.
Additionally, the RNN share parameters across each layer of the network and it has a capability to adjust the weights of the network nodes using the backpropagation technique in order to achieve reinforcement learning. There are various variants to RNN such as bidirectional RNN (BRNN), long short-term memory (LSTM) and gated recurrent units (GRUs). BRNN can use future data as well as previous inputs in order to improve accuracy. LSTM addresses the problem of long-term dependencies. It has cells of three types of hidden gates: input, output and forget gates to control the flow of information needed to predict the output. GRU also address the short-term memory problem such as the LSTM but it has two gates which are reset and update gates to control and retain the needed information [21] [22].

- Convolutional Neural Networks (CNN

Although the CNN is mostly used in image, speech, or audio signal inputs it also can be utilized in NLP such as text classification. CNN includes three types of layers which are convolutional, pooling and fully-connected (FC) layers. The convolutional is the first processed layer of the network that followed by more convolutional or pooling layers where the FC is the last layer of the network. The majority of computation is done in the convolutional layer that requires three components which are input data, a filter (kernel) and a feature map. The filter is called a feature detector which processes the input data to detect features and this process is called the convolution. The filter as a feature detector is usually a 3X3 matrix that represents part of data input where

[4] Details of programming source code for building word embedding from Quran are found in [18]: Section 2.3.

a dot product is computed between the filter and the input data to be stored in an output array. The process is repeated for other parts of the input data by shifting the filter by a stride until processing the entire input data. The fully built output array represents the feature map of convolutional feature. The initial convolutional layer can be followed by more convolutional layers. The pooling layer works such as the convolutional layer by sweeping the filter across the entire input but the filter does not hold weight. The pooling layer implements an aggregation function for producing the output array. There are two main types of pooling which are max pooling and average pooling. In max pooling the max value is selected from the filter that represents part of the data input to be stored in the output array. While in average pooling, the average is computed from the filter. In FC layer each node in the input layer is connected directly to an output node. While the convolutional layer and pooling layer partially connect input layer to output layer. Additionally, the convolutional and pooling layers use ReLu functions for activation while the FC layer uses softmax activation for classification [22] [23].

5) *Multi-Label Classification*

The topics handled and discussed by Quran verses are one of the main interests of Islamic scholars especially who are specialists in Quran studies. There are various and different verses classifications achieved by specialists in Quran studies. These respected achievements differ in the number of topics discussed by verses and the relationships between these topics such as a verse can handle a topic which is considered a subtopic of a more general one. Irrespective of the differences in classifying verses topics, and based on specialists of Quran studies, a verse in Quran often handles more than one topic at the same time. And hence the type of classification that will be used in the automatic classification of Quran verses using DL approach is a type of multi-label classification.

6) *Results Analysis and Discussion*

In this processing step the output results of DL classifiers will be evaluated and discussed. We used the metrics of precision, recall, f1-score, accuracy and hamming loss for evaluating the performance of the DL models when achieving multi-label classification on Quran verses.

Precision is the ratio of the number of samples classified correctly for a label (true positives – TP) to the total number of samples that both classified correctly (TP) and classified incorrectly (false positive – FP) for that label. The FP are classified as positive labels but actually belong to other labels. The precision is calculated by TP/(TP+FP) formula. The main purpose of the precision is to minimize the number of false positive [24].

Recall is the ratio of the number of samples classified correctly (TP) for a label to the total number of samples that classified correctly (TP) and classified incorrectly (false negatives FN) to other labels. The FN actually belong to the label but are classified incorrectly to other labels. Recall is calculated by using TP/(TP+FN) formula. The main purpose of recall is to minimize the number of false negatives.

F1-score is the combination of both precision and recall measures and used in the cases that not appropriate to consider one of the both metrics individually. F1-Score is calculated using 2TP/(2TP+FP+FN) formula which is the harmonic mean of precision and recall.

Accuracy is the ratio of the total number of samples classified correctly (TP and TN) to the total number of samples that classified correctly and incorrectly (TP, FP, TN and FN). The accuracy is calculated using (TP+TN)/(TP+FP+TN+FN) formula [24].

More detailed information for predictions are always provided by confusion matrices in order to be further easily analyzed and discussed [25]. For example, the confusion matrix in Figure 4 shows classifier results when experimenting three different labels which are L1, L2, and L3. The information provided by the main diagonal of the matrix TP1, TP2 and TP3 are the correct classifications of L1, L2, and L3 labels respectively. The remaining information of the matrix are the incorrect classifications where the rows represent the FNs and the columns represent the FPs. For instance, the F21 and F31 are the inputs of L1 that are classified incorrectly to L2 and L3 respectively. Whereas the F12 and F13 are the inputs of L2 and L3 respectively that are classified incorrectly to L1.

	L1	L2	L3
L1	TP1	F21	F31
L2	F12	TP2	F32
L3	F13	F23	TP3

Figure 4. A confusion matrix of classification results for three different labels

Hamming loss is the percentage of wrong labels divided by the total number of labels and computed as the hamming distance between the actual TP labels and the predicted TP labels and used in multi-label classification that castigates only the individual labels. The lower value of the hamming loss means a better multi-label classification results.

To avoid an inaccurate performance results due to the unbalanced dataset we used a 10-fold cross-validation (CV) approach. In this case the dataset is split randomly into 10 equal folds. We used 9 folds for the training process and remaining one for the testing process. This means that an 90% of dataset used for training an 10% is used for testing in each processing round of the 10 folds.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section provides a description of the used DL models and their neural network structures. It also provides the performance evaluation results of these models when conducting multi-label classification experiments on Quran verses in addition to the analysis and discussion of the experimental results.

We built the DL models using RNN and CNN algorithms for applying multi-label classification on Quran verses. Both models will be experimented using diacritized and undiacritized text where the skip-gram technique of word embedding is used.

We proposed the network structures that are shown in Table IV where each of both models includes the following main sequential layers:

1) An embedding layer which is the first layer that takes the verses as input to process their embeddings. An embedding size of 300 was chosen for both diacritized and undiacritized text of verses.

2) The model algorithm layer. The bidirectional LSTM (BiLSTM) for the RNN and one-dimensional convolution layer followed by pooling layer for the CNN were selected.

3) A dense layer of shape 24 was chosen where the number 24 is twice the number of classified topics (labels).

4) A dropout layer to randomly drop nodes from the output of the previous layer in order to reduce the overfitting problem of dataset training. A value of 0.2 was chosen which means that 20% of nodes will be dropped (removed).

5) A dense layer which is the final layer to produce and output of shape 12 that represents the classified labels of a verse.

More details about setting up these models are provided next.

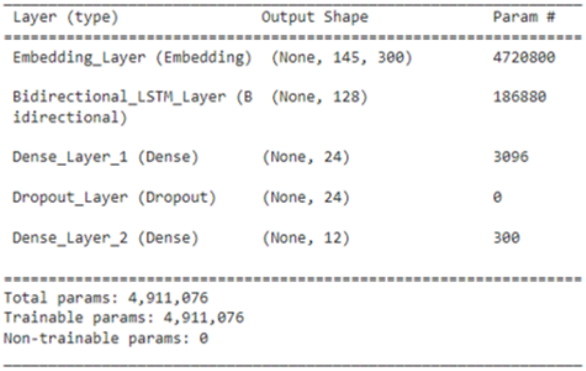
TABLE IV. The main layers of the used deep learning networks

Layer	RNN Network	CNN Network
1	Embedding layer	Embedding layer
2	BiLSTM Layer	1D Convolution Layer
3	Dense Layer	Max Pooling Layer
4	Dropout Layer	Dense Layer
5	Dense Layer	Dropout Layer
6	-	Dense Layer

As shown in Table IV each of both networks starts with an embedding layer to benefit from the sematic of dataset words and ends with a dense layer to produce the required multi-label classification of verses.

Because building an optimized DL networks with an appropriate configuration is subject to trial and error, we initially experimented our proposed DL networks using some variations of hyperparameters. We experimented both model algorithms in the the second layer using 32, 64 and 128 number of network nodes and found 64 gave better results than did 32 but 128 did not give significant increase on performance. However increasing the number of nodes will increase the complexity of DL model and hence will increase the learning processing time and this time will increase significantly when applying the cross-validation technique. It is noteworthy that the output nodes produced by BiLTSM are doubled because BiLTSM works bidirectional on text sequence by reserving the same number of nodes for processing the text from beginning to end and vice versa. So, the output of BiLSTM in our case becomes 128. The subsequent dense layer is used generally to improve model performance where a value of 24 is chosen for its number of nodes which is twice the the number of output nodes of the last layer. Where the output of the last layer is the classification of Quran verses. The dropout layer is the layer that precedes the last one and it is mainly used to reduce overfitting by randomly reducing the network nodes where a value of 20% is chosen for the percentage of reduction as used by many practitioners [19].

Figure 5 shows a summary for the RNN neural network and the characteristics of its layers when it is fed by the input of the vectorized Quran verses using diacritized text.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Embedding_Layer (Embedding)  (None, 145, 300)         4720800

 Bidirectional_LSTM_Layer (B  (None, 128)              186880
 idirectional)

 Dense_Layer_1 (Dense)       (None, 24)                3096

 Dropout_Layer (Dropout)     (None, 24)                0

 Dense_Layer_2 (Dense)       (None, 12)                300

=================================================================
Total params: 4,911,076
Trainable params: 4,911,076
Non-trainable params: 0
_________________________________________________________________
```

Figure 5. The layers specification of RNN network

As shown in Figure 5 each verse (sample) in the input is represented as a fixed length vector constructed of a sequence of 145 integer values. As shown in Figure 5, the first layer of the network is the embedding layer that takes this input and creates an embedding vector of size 300 for each verse (sample). This means that the output of the embedding layer are samples of two-dimensional vectors of shape (145,300). It is important to note that, as shown in the summary, the number of the processed parameters (features) in this layer reached 4,720,800. The next layers are: an bidirectional LSTM layer for producing output vectors of shape 128, a dense layer followed by a dropout layer for producing an output vectors of shape 24. The final layer is a dense layer that produces output vectors each of which includes 12 individual values that represent the probabilities of the classified labels (topics) of a verse (sample).

Additionally, a value of 60 is set to the epoch hyperparameter which represents the number of processing times performed in the entire dataset. And a value of 64 is set for the batch size hyperparameter that represents the number of samples that fed to the neural network each time. Figure 6 and Figure 7 show the values of accuracy and loss respectively that produced through the executing sequence of epochs when conducting the RNN model on the diacritized text of Quran verses. As shown in Figure 6 the accuracy reached its highest values for training and validation samples of the dataset after 17 epochs approximately. To minimize the risk of producing less accurate results that may affect the classification performance results especially for the loss function algorithm a larger epoch value is selected.

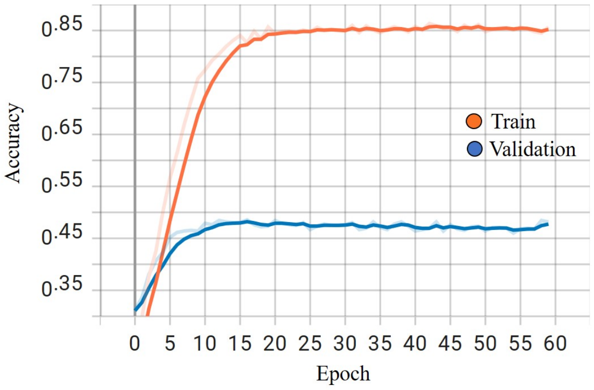


Figure 6. Accuracy per epoch

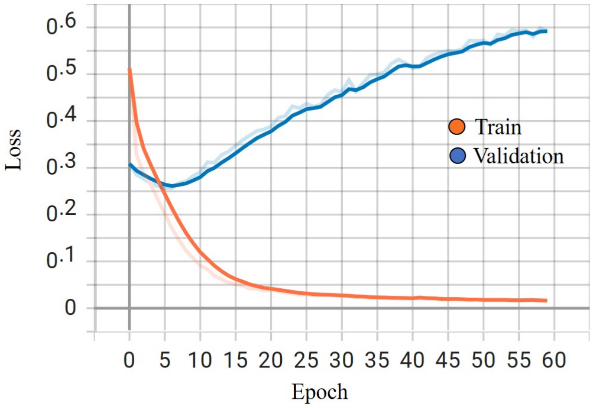


Figure 7. Loss per epoch

The neural networks of the two models described previously were applied on the Quran dataset for the purpose of multi-label classification of verses into 12 topics where the approach of 10-fold cross-validation is used. It is important to note that the DL classifier produces a probability value for each label in every round of the 10-folds so the probability value is rounded to 0 or 1 where the value of 1 means that the topic (label) is classified to the verse (sample).

When building the words sequences on the 4,911 verses of the Quan dataset with diacritized text a vocabulary of 15,736 words was created where the length of each verse vector is 145 which represents the longest verse. While the created vocabulary of undiacritized text of Quran verses includes 13,345 words each of which is a vector of shape 129 of integer values.

The results for accuracy, precision, recall, F1-score and hamming loss are provided in Table V As shown from the results we reached considerable performances using both of used DL models. It is clear that the DL classifiers produced better results when using diacritized text of Quran verses. The RNN model achieves the best accuracy and recall with values of 90.38% and 92.49% respectively while the CNN model produced the best precision and F1-Measure with values of 96.98% and 93.81%. Additionally, the CNN produced the lowest hamming loss with 0.0126 that is the best ratio for the multi-label classification results.

The details of evaluation results for each label (topic) using diacritized and undiacritized text are provided in Table VI and Table VII when experimenting Quran verses using RNN and CNN models respectively. By comparing the performance results between the classified labels, it is obvious that the RNN achieved the best performance for the “stories” label using both diacritized and undiacritized text. Also, the CNN achieved the best recall and F1-score values for the “stories” label using both types of text while it performed the best precision for “moahmmad” label using diacritized text and best precision for “struggle” label using undiacritized text. The best performance results of precision, recall and F1-score are highlighted in bold text of numerical values in TableVI and Table VII.

By analyzing these results, it clear that the RNN and CNN classifiers mostly performed better with the “stories” label which represents the topic that has the largest number of labeled verses. Nonetheless, the two DL classifiers did not perform the lowest performance for the topic that has the lowest number of labeled verses. For instance, both DL classifiers performed the lowest precision for “faith” and “disbelief” respectively using diacritized text although these labels did not have the lowest number of labeled verses. The lowest performance results of precision, recall and F1-score are highlighted in bold an italic text of numerical values in Table VI and Table VII.

For visualizing the classification results, Figure 8 shows the confusion matrices for 12 classified labels when con-

TABLE V. Performance experimental results of multi-label classification of Quran verses into 12 topics

	RNN		CNN	
Metric	Diacritized	Undiacritized	Diacritized	Undiacritized
Accuracy	**0.9038**	0.8959	0.8974	0.8894
Precision	0.9445	0.9423	**0.9698**	0.9675
Recall	**0.9249**	0.9156	0.9092	0.9030
F1-Score	0.9343	0.9286	**0.9381**	0.9337
Hamming Loss	0.0138	0.0149	**0.0126**	0.0135

TABLE VI. Detailed multi-label classification performance results of Quran verses using RNN model

	Diacritized			Undiacritized			
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	of verses
allah	0.8909	0.8507	***0.8703***	0.9091	***0.8333***	***0.8696***	288
disbelief	0.8865	0.9198	0.9028	***0.9049***	0.9096	0.9072	586
evils	0.9231	***0.8485***	0.8842	0.9438	0.8485	0.8936	99
faith	***0.8786***	0.8871	0.8828	0.9073	0.8839	0.8954	310
mohammad	0.9684	0.9041	0.9351	0.9275	0.8967	0.9118	271
quran	0.9454	0.9327	0.9390	0.9067	0.9158	0.9112	297
rulings	0.9200	0.9287	0.9243	0.9550	0.9074	0.9306	421
stories	**0.9805**	**0.9564**	**0.9683**	**0.9675**	**0.9510**	**0.9591**	1469
struggle	0.9414	0.918	0.9298	0.9444	0.9020	0.9228	245
universe	0.9525	0.8739	0.9115	0.9319	0.8784	0.9044	436
unseen	0.9556	0.9498	0.9527	0.9553	0.9430	0.9491	1474
worship	0.9538	0.8857	0.9185	0.9404	0.8571	0.8969	350

TABLE VII. Detailed multi-label classification performance results of Quran verses using CNN model

	Diacritized			Undiacritized			
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	of verses
allah	0.9717	0.8333	0.8972	***0.9469***	***0.8056***	***0.8705***	288
disbelief	***0.9597***	0.8942	0.9258	0.9548	0.9010	0.9271	586
evils	0.9872	***0.7778***	***0.8701***	0.9765	0.8384	0.9022	99
faith	0.9746	0.8677	0.9181	0.9576	0.8742	0.9140	310
mohammad	**0.9878**	0.8967	0.9400	0.9839	0.9004	0.9403	271
quran	0.9855	0.9158	0.9494	0.9851	0.8923	0.9364	297
rulings	0.9612	0.8836	0.9208	0.9736	0.8765	0.9225	421
stories	0.9764	**0.9578**	**0.9670**	0.9810	**0.9476**	**0.9640**	1469
struggle	0.9782	0.9143	0.9451	**0.9866**	0.9020	0.9424	245
universe	0.9598	0.8761	0.9161	0.9515	0.8555	0.9010	436
unseen	0.9614	0.9294	0.9452	0.9566	0.9281	0.9421	1474
worship	0.9739	0.8543	0.9102	0.9735	0.8400	0.9018	350

ducting experiments using the RNN algorithm with the diacritized text of Quran verses. As shown in Figure 8, for instance 1,405 verses of "stories" label are classified correctly, while 64 verses of the "stories" label are classified to other labels and 28 verses from other labels are classified as "stories" label.

Comparing our results of this study with other related studies is not provided due to the diversity of used datasets and predefined classified topics. Although we reached significant performance results in multi-label classification of Quran verses using a dataset that is larger than those used of other studies.

Although we developed our embedding using original Quran corpus which is the word of Allah, there still a challenge in creating such embedding using larger corpora because most of the available public corpora were created for other domains and mostly written as undiacritized text.

Although the selection of DL network structure, number of its layers and the characteristics of these layers are subject to the concept of trial and error, our approach

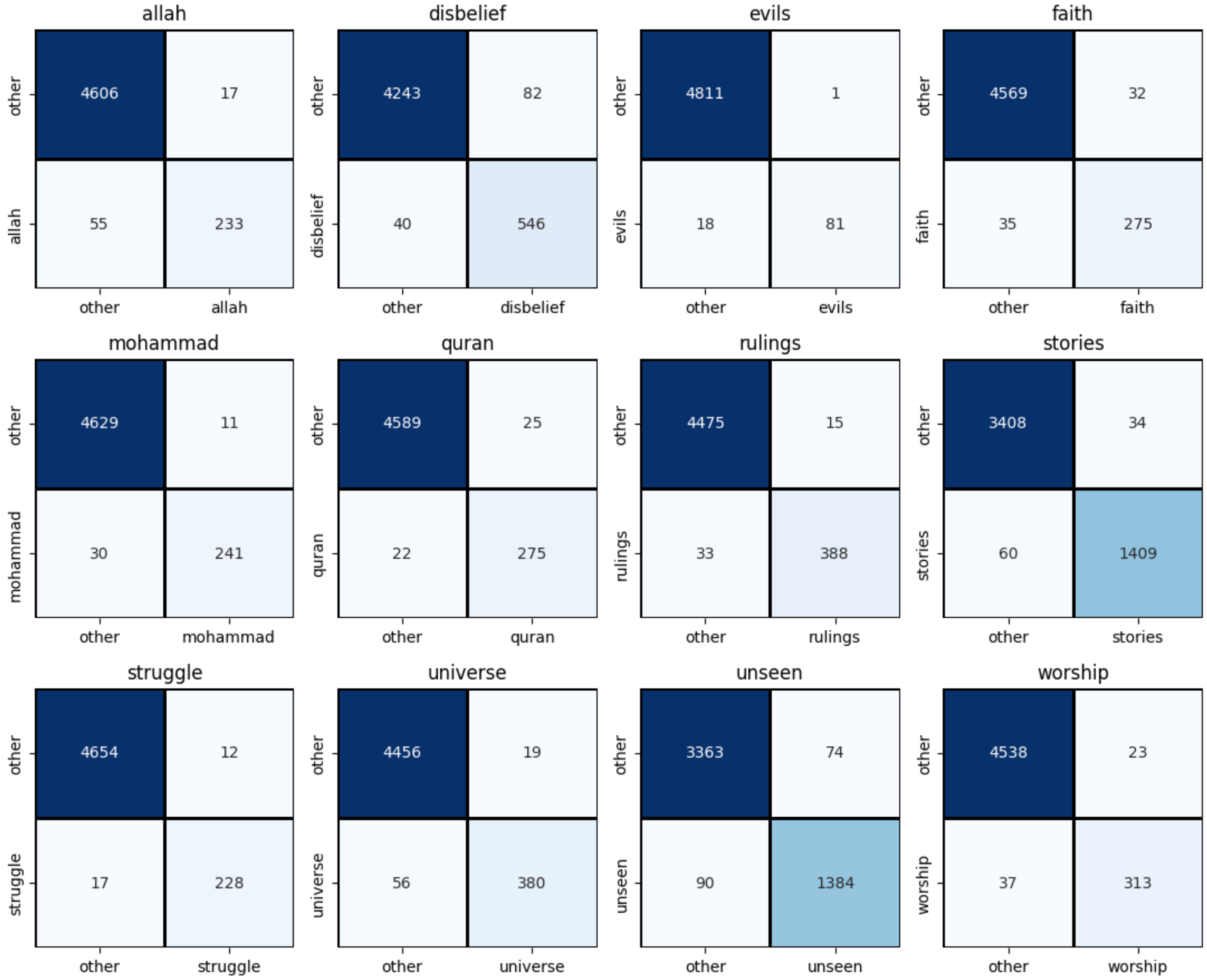


Figure 8. Confusion matrices for 12 classified topics of Quran verses

of building an appropriate model takes into account the tradeoff between simplicity and complexity to minimize underfitting and overfitting respectively.

The DL algorithms always needs more computational times for dataset training than do the traditional methods. Nonetheless we tried to reach suitable configurations through considerable number of training iterations.

5. THREATS TO VALIDITY

In this section we provide the threats to validity focusing on the potential limitations of this study.

1) This paper used the DL approach for multi-label classification of Quran verses into 12 predefined topics and to mitigate the biased results due to using one specific DL learning algorithms, we used two DL algorithm which are RNN and CNN and built an appropriate configured number of neural network layers in our experiments.
2) The DL classifiers are always affected by dataset size and to mitigate the bad effect on the performance results we used larger dataset that includes 4,911 labeled verses that distributed among almost all chapters of Quran except “Alfil” and “Almaoun” chapters.
3) To mitigate the negative effect of classification performance results due to unbalanced dataset we adopted the 10-fold cross-validation in our experiments using well preprocessed data and new versions of DL algorithms. While the traditional folding is used, some experiments can be repeated in future studies using the stratified type of folding.

6. CONCLUSION AND FUTURE WORK

In this paper we applied the deep learning (DL) approach for multi-label classification of Quran verses. We created a dataset of 4,911 labeled verses. A method of multiple steps is designed for achieving the purpose of this paper. It includes data preprocessing, creating words sequences and word embeddings using skip-gram Word2Vec, building DL neural networks using RNN and CNN, producing experimental results for evaluating the DL models and finally analyzing and discussing the performance results. Our experiments were done using diacritized and undiacritized text of Quran verses. We also applied the 10-fold cross validation to produce better performance results for the unbalanced dataset. The values of 90.38%, 96.98%, 92.49%, 93.81%, and 0.0126 for accuracy, precision, recall, F1-score and hamming loss were achieved respectively.

The findings of this paper can help specialists of Quran studies and others to gain more insights and knowledge for easily studying and teaching the topics discussed by Quran verses.

An extension to this study can be performed using transfer learning such as pretrained DL models. Larger dataset can be also utilized in addition of using more algorithms of DL models.

References

[1] S. H. Nasr, C. K. Dagli, M. M. Dakake, J. E. Lumbard, and M. Rustom, *The Study Quran. A new translation and commentary, 19*, (2015).

[2] “ChatGpt,” https://chat.openai.com/chat, (Accessed: February 2, 2023).

[3] A. M. Alashqar, “A comparative study on arabic pos tagging using quran corpus,” *In 2012 8th International Conference on Informatics and Systems (INFOS)*, pp. NLP 28–33, (2012).

[4] K. Alrajhi and M. A. ELAffendi, “Automatic arabic part-of-speech tagging: Deep learning neural lstm versus word2vec,” *International Journal of Computing and Digital Systems*, vol. 8(03), pp. 307–315, (2019).

[5] N. Thabet, “Understanding the thematic structure of the qur’an: an exploratory multivariate approach,” *In Proceedings of the ACL Student Research Workshop*, pp. 7–12, (2005, June).

[6] M. N. Al-Kabi, G. Kanaan, K. Al-Shalabi, R.and Nahar, and B. Bani-Ismail, “Statistical classifier of the holy quran verses (fatiha and yaseen chapters),” *Journal of Applied Sciences*, vol. 5(3), pp. 580–583, (2005).

[7] M. N. Al-Kabi, B. M. A. Ata, H. A. Wahsheh, and I. M. Alsmadi, “A topical classification of quranic arabic text,” *In Proceedings of the 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, pp. 252–257, (2013, December).

[8] E. H. Mohamed and W. H. El-Behaidy, “An ensemble multi-label themes-based classification for holy qur’an verses using word2vec embedding,” *Arabian Journal for Science and Engineering*, vol. 46, pp. 3519–3529, (2021).

[9] S. K. Hamed and M. J. Ab Aziz, “Classification of holy quran translation using neural network technique,” *Journal of Engineering and Applied Sciences*, vol. 13(12), pp. 4468–4475, (2018).

[10] F. S. Nurfikri, “A comparison of neural network and svm on the multi-label classification of quran verses topic in english translation,” *In Journal of Physics: Conference Series*, vol. 1192(1), p. 012030, (2019).

[11] R. A. Pane, M. S. Mubarok, and N. S. Huda, “A multi-lable classification on topics of quranic verses in english translation using multinomial naive bayes,” *In 2018 6th International Conference on Information and Communication Technology (ICoICT)*, pp. 481–484, (2018, May).

[12] M. S. Mubarok and Huda, “A multi-label classification on topics of quranic verses in english translation using tree augmented naïve bayes,” *In 2018 6th International Conference on Information and Communication Technology (ICoICT)*, pp. 103–106, (2018, May).

[13] A. O. Adeleke, N. A. Samsudin, A. Mustapha, and N. M. Nawi, “A group-based feature selection approach to improve classification of holy quran verses,” *In Recent Advances on Soft Computing and Data Mining: Proceedings of the Third International Conference on Soft Computing and Data Mining (SCDM 2018), Johor, Malaysia, February 06-07, 2018*, pp. 282–297, (2018).

[14] N. S. Huda and M. S. Mubarok, “A multi-label classification on topics of quranic verses (english translation) using backpropagation neural network with stochastic gradient descent and adam optimizer,” *In 2019 7th International conference on information and communication technology (ICoICT)*, pp. 1–5, (2019, July).

[15] M. Borhani, “Multi-label log-loss function using l-bfgs for document categorization,” *Engineering Applications of Artificial Intelligence, 91, 103623*, (2020).

[16] “Quran By Subject,” https://quranbysubject.com/categories.php, (Accessed: February 5, 2023).

[17] “Tanzil - Quran Navigator,” https://tanzil.net, (Accessed: February 9, 2023).

[18] “A Classification of Quran Verses Using Deep Learning: Complete Code,” https://colab.research.google.com/drive/1j3dhIsU7p7jU-WfxY2W73WSqYl4PzhrV, (Accessed: April 2, 2023).

[19] A. Kapoor, A. Gulli, S. Pal, and F. Chollet, *Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models*. Packt Publishing Ltd., 2022.

[20] T. Ganegedara, *Natural Language Processing with TensorFlow*. Packt Publishing Ltd., 2022.

[21] “IBM: What is recurrent neural networks?” https://www.ibm.com/topics/recurrent-neural-networks, (Accessed: March 15, 2023).

[22] M. Ekman, *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers Using TensorFlow*. Addison-Wesley Professional, (2021).

[23] B. W. are Convolutional Neural Networks?, “https://www.ibm.com/topics/convolutional-neural-networks,” (Accessed: March 15, 2023).

[24] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*.

[25] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17(1), pp. 168–192, (2021).

Abdelkareem M. Alashqar is an assistant professor at the department of software development, the Islamic University of Gaza, Palestine. He received his PhD in Information Systems from Mansoura University, Egypt. His research and teaching interests include software engineering, software quality evaluation, natural language processing and machine/deep learning.