IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 26, 2021, accepted June 1, 2021, date of publication June 8, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3087593

A Predictive Text System for Medical Recommendations in Telemedicine: A Deep Learning Approach in the Arabic Context

MARIA HABIB[1], MOHAMMAD FARIS[1], RANEEM QADDOURA[2], ALAA ALOMARI[1], AND HOSSAM FARIS[1,3,4]

[1]Altibbi.com, Amman 11831, Jordan
[2]Philadelphia University, Amman 1183, Jordan
[3]King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan
[4]School of Computing and Informatics, Al Hussein Technical University, Amman 11831, Jordan

Corresponding author: Hossam Faris (hossam.faris@ju.edu.jo)

ABSTRACT We are currently witnessing an immense proliferation of natural language processing (NLP) applications. Natural language generation (NLG) has emerged from NLP and is now commonly utilized in various applications, including chatting applications. The objective of this paper is to propose a deep learning-based language generation model that simplifies the process of writing medical recommendations for doctors in an Arabic context, to improve service satisfaction and patient-doctor interactions. The developed language generation model is a predictive text system intended for next word prediction in a telemedicine service. Altibbi—a digital platform for telemedicine and teleconsultations services in the Middle East and the North Africa (MENA) region—was utilized as a case study for the textual prediction process. The proposed model was trained using data obtained from Altibbi databases related to medical recommendations, particularly gynecology, dermatology, psychiatric diseases, urology, and internist diseases. Variants of deep learning models were implemented and optimized for next word prediction, based on the unidirectional and bidirectional long short-term memory (LSTM and BiLSTM), the one-dimensional convolutional neural network (CONV1D), and a combination of LSTM and CONV1D (LSTM-CONV1D). The algorithms were trained using two versions of the datasets (i.e., 3-gram and 4-gram representations) and evaluated in terms of their training accuracy and loss, validation accuracy and loss, and testing accuracy per their matching scores. The proposed models' performances were comparable. CONV1D produced the most promising matching score.

INDEX TERMS Altibbi, natural language processing, deep learning, telemedicine, Arabic, predictive text.

I. INTRODUCTION

NLG is a sub-field of NLP that combines computational linguistics and artificial intelligence to generate texts automatically. These texts are intended to exhibit the characteristics and intuition of natural texts—that is, they should be syntactically and semantically correct, as well as coherent. NLG techniques are broadly applied in various domains, such as in the media, education, and finance. One of the earliest applications was presented in 1966 in the form of "Eliza," a conversational bot and psychotherapist that chatted with users. However, early NLG approaches are rule-based and data-driven. Thus, they are not scalable and cannot handle large, complex datasets. Meanwhile, contemporary NLG techniques inspired by artificial intelligence and deep learning methods have demonstrated a remarkable propensity to handle a massive amount of data and extract informative features. NLG has numerous sub-applications, such as generating text in conversational chatbots [1], translating text from one language to another [2], generating stories [3], creating abstractive summaries of texts [4], providing automatic image captions [5], paraphrasing texts [6], and others [7].

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

Automatically predicting the next word is a specific text-generation subset of NLG in which the most likely next character, word, or phrase is determined as the user is typing. The automatic suggestions have several advantages, including reducing the keystroke rate, preventing misspellings, and saving time spent typing. However, it is challenging to consistently predict words that fit the context [8].

Text generation in the medical domain has various applications, including medical image captioning, conversational therapists and dialogue generation, medical report generation, and summarization. Next word prediction in medical reports is of particular interest in this paper, which examines Arabic text generation and prediction for medical recommendations while doctors are typing. Specifically, this paper investigates Altibbi,[1] a digital health platform that provides telemedicine services for the MENA region. Altibbi's primary service is to make telemedicine consultations accessible throughout the MENA region by connecting people with doctors from anywhere at any time. One of Altibbi's objectives is to help doctors produce medical recommendations by suggesting the possible next word as they type. These medical recommendations reveal the doctor's notes and the medications and treatments they prescribe for the patients. Typing such information can consume a significant amount of a doctor's time that could be spent consulting with patients. Therefore, this paper proposes a predictive text model for predicting and generating text for doctors' recommendations. This model is intended to improve service satisfaction, improve patient-doctor interactions, and save doctors' time. The proposed model requires a massive amount of data from Atibbi's databases, which contains more than two million documented consultations. Considering that the availability of such medical data in the Arabic context is scarce and rarely found.

Text generation in the Arabic language in the medical domain is crucial, yet it faces several serious challenges. Very few studies have been devoted to advancing text generation techniques in Arabic due to a lack of in-domain datasets and specialized processing tools. Moreover, Arabic is a highly complex and rich language that consists of 28 characters that differ morphologically and phonologically. For example, some characters have one or two dots below them, and some have one, two, or three above them, with each iteration having a different sound. The processing of Arabic scripts is also difficult because writing styles differ from one country to another and even from one city to another within the same country, depending on the dialect. Therefore, scripts often show different spellings and perhaps misspellings, which can influence the semantics. In this paper, a deep neural model has been developed based on Arabic recommendations collected from Altibbi. The proposed model undergoes several stages: data preprocessing, training data generation, model training and tuning, and, lastly, model evaluation. Different neural models have been constructed and trained based on convolutional and sequence-to-sequence models by utilizing convolutional neural networks (CNNs), LSTM, BiLSTM, and combinations of these. LSTM and BiLSTM are sequence-neural models for processing fixed and variable length of sequential data. They are popular methods for handling textual sequences since they can preserve relationships over long sequences. On the other hand, convolution networks extract features from texts, with each filter serving as a one-dimensional filter that generates a unique feature. The four deep learning models utilized in this work are LSTM, BiLSTM, CONV1D, and LSTM-CONV1D, which were investigated at three different embedding dimensions. These models were developed individually for the most common five medical specialties: gynecology, dermatology, psychiatry, urology, and internist diseases. Two versions of the training datasets were constructed per specialty based on a varying n-gram data model, including the 3-grams and 4-grams.

Evaluating NLG tasks is an open research direction, and there is no one standard way to measure the performance. Therefore, this paper compares the models by considering the training and validation of accuracy and loss, as well as testing accuracy, in terms of matching scores. Specifically, matching scores are used to evaluate the models by assessing what percentage of matching (1-gram overlapping) between the generated text (i.e., word) and the ground-truth text. The matching score metric only shows how much the models can find the exactly matched word with the ground-truth regardless of whether the generated word is correct and relevant to the context. Meanwhile, the aim is to generate five predictions that are all relevant to the respective context.

The rest of this paper is organized as follows. Section II provides an overview of recent related literature on text generation, particularly in the medical domain. Section III describes the methodology of the proposed approach, including a description of the dataset, the system architecture, and the evaluation criteria. Section IV presents the experimental details, and discusses the results. Finally, Section V offers conclusions regarding the methodology and results.

II. RELATED WORKS

This section describes related papers for NLG, including papers on general text generation and medical text generation.

A. TEXT GENERATION

Before the advent of NLG, text generation models were primarily based on recurrent sequence-to-sequence models that eventually evolved into convolution, reinforcement learning, and transformer-based models. This subsection discusses text generation approaches employed over various applications in recent studies.

Li *et al.* [9] developed a deep reinforcement approach for paraphrase generation. This approach consists of a neural generator and evaluator that is responsible for providing a reward. The proposed model was evaluated based on two datasets and relying on ROUGE, BLEU, and

[1] https://www.altibbi.com/

METEOR scores, and the model outperformed previously used approaches. The model also exhibited reasonable fluency and relevance of generated texts. Semeniuta *et al.* [10] created a variational autoencoder framework to generate text at the character level and over long sequences. Their model utilized a convolutional encoder and deconvolutional decoder with recurrent neural layers. However, the authors did not evaluate the model's performance on real downstream NLP tasks. Meanwhile, Marcheggiani and Perez-Beltrachini [11] proposed graph convolution encoders to generate text from structured figures. Their model was trained on WebNLG and SR11Deep and was evaluated based on BLEU, METEOR, and TER. The authors noticed that deep graph convolutions were more capable of generating texts than sequence models were. However, experiments with abstract figure types are more critical. In another study, Li *et al.* [12] proposed a neural text generation model for fitting various categories that was also suitable for supervised learning tasks. The proposed model integrates generative adversarial networks (GANs), the recurrent neural network (RNN), and reinforcement learning. It was evaluated based on its performance in a sentiment analysis task, on which it achieved an accuracy of almost 82%. Fan *et al.* [13] created a hierarchical neural model for story generation. The model fused a convolutional sequence-to-sequence model and a self-attention mechanism, and it was trained on a human-collected dataset of 303,358 stories. The model's evaluation was based on its perplexity and comparisons with human evaluations. The text generated by the model exhibited improved fluency and coherence. Further, Lee and Hsiang [14] designed a neural-based language model for textual patent claim generation. The designed model was built based on OpenAI's GPT-2 language model and was tested based on 90 samples that were coherent and free of obvious syntactical or semantics errors. However, the model was producing long, hard-to-read sentences.

Other authors [15] used BART and T5 models to evaluate two neural language models that generated text based on graphs. The models were trained on three benchmark datasets (i.e., LDC2017T10, WebNLG, and AGENDA), and the models were evaluated based on the BLEU scores. Even though the models obtained outstanding results on the used benchmarks, the authors explained that experimenting with richer graph structural bias has a significant impact. Liu *et al.* [16] discussed the potential benefits and limitations of procedural content generation in video games depending on deep learning methods. However, further investigations are required for events, goals, and character generations. Meanwhile, Yamshchikov and Tikhonov [17] proposed a variational recurrent autoencoder for music generation. Their approach showed diverse pleasing melodies, which were assessed by human evaluators. In [18], the authors constructed a deep learning approach for summarizing articles in the Arabic language. They utilized an encoder-decoder recurrent neural network with attention and coverage mechanisms trained on 300,000 articles collected from an Arabic website, "mawdoo3." Based on the ROUGE-1 scale of recall, precision, and f1-score, the model showed very favorable results (ROUGE-1(precision) = 62%). Moreover, Luu *et al.* [19] proposed a language model based on GPT-2 for citation generation trained on a large-scale dataset of scientific articles. The proposed model was assessed based on both human and automatic evaluations according to BLEU and ROUGE scores. Meanwhile, 71% of the generated text was correct. Although few research studies are devoted to text generation in the Arabic context, in one study [20], the authors developed a Transformer-based language model for Arabic language generation based on GPT-2. The authors created four versions (base, medium, large, and mega). The model was called ARAGPT2, and it was trained on a large corpus of Arabic articles and news stories. Further, the model was evaluated based on its perplexity, and it showed coherent grammatically correct results.

B. MEDICAL TEXT GENERATION

This subsection overviews recent research studies on text generation applications in the medical domain across different language contexts.

Jing *et al.* [21] constructed a deep learning approach to generate images for medical textual reports. Specifically, the proposed model used a hierarchical, attention-based LSTM to generate images to accompany written radiology and pathology reports. The model showed promising results according to BLEU, METEOR, ROUGE, and CIDER scores. Other authors in [22] developed a deep learning model for generating chest X-ray imaging reports. The model relied on a multi-attention and bidirectional LSTM to encode images and generate sentences. It showed excellent results according to various evaluation metrics. Other researchers [23] created a model for automatically completing texts entered into the Electronic Health Records (EHRs) database, focusing on records regarding colonoscopy, transesophageal echocardiogram, and anterior-cervical-decompression. The developed model was based on the Markov-chain statistical modeling and achieved outstanding results in terms of recall, precision, and time spent typing. Ginn and Unviersity [24] constructed "Smart Vet," an auto-complete system that helps veterinarians take medical notes. The proposed model was trained using two deep learning approaches: a sequence-to-sequence translation model and the OpenAI's GPT-2 model. The latter, yielded a better BLEU score of 1.19. Further, Van *et al.* [25] proposed "AutoMeTS," an auto-complete model designed to simplify medical texts. The proposed system was an ensemble of four language models (BERT, RoBERTa, XLNet, and GPT-2) and was trained using the English version of Wikipedia. The model achieved a predictive accuracy of 64.5%.

Moreover, Gopinath *et al.* [26] proposed a contextual auto-complete model intended to help medical doctors take notes. The authors applied a hierarchical inference language model to predict texts, which reduced the keystroke rate by 67%. Hoogi *et al.* [27] proposed an RNN model for generating

TABLE 1. Summary of related works.

Reference	Language	Objectives	Techniques applied	Performance evaluation
[15]	English	Graph-to-Text generation	BART & T5	BLEU (LDC2017T10) =49.7, BLEU (WebNLG)=59.7, BLEU (AGENDA)=25.66
[9]	English	Paraphrase generation	Deep reinforcement model	BLEU=45.74, METEOR= 20.18, ROUGE-1=42.15, and ROUGE-2= 24.73
[21]	English	The generation of medical imaging reports	Hierarchical LSTM	BLEU-1=0.517, BLEU-2=0.386, BLEU-3=0.306, BLEU-4=0.247, METEOR=0.217, ROUGE=0.447, and CIDER=0.327
[11]	English	Structured-graphs to text generation	Graph-convolution encoders	BLEU =0.535, METEOR=0.39, TER=0.44 (For the WebNLG dataset)
[12]	English	Category text generation	GANs, RNN, & reinforcement learning	Accuracy is almost 82%
[13]	English	Story generation	Hierarchical sequence to sequence and self-attention mechanism	perplexity= 36.56
[22]	English	Chest X-ray image report generation.	Multi-attention recurrent neural network.	BLEU-1=0.476, BLEU-2=0.430, BLEU-3=0.238, BLEU-4=0.169, ROUGE-L=0.347, CIDER=0.297, and MA=0.498
[14]	English	Patent claim generation	OpenAI's GPT-2	Qualitative assessment
[23]	English	EHRs auto-complete	MarKov models	Precision = 82%, recall = 93%, keystroke saving = 73.5%, reduction-in-typing time = 33.36%
[24]	English	The veterinarians medical notes completion.	OpenAI's GPT-2	BLEU = 1.19
[17]	English	Music & melody generation	Variational recurrent autoencoder	Human assessment
[18]	Arabic	Articles summarization	Encoder-decoder RNN with attention mechanism	ROUGE-1(precision)=61.36, ROUGE-1(recall)=32.59, ROUGE-1(f1-score)=42.47
[19]	English	Citation generation	Extended model of OpenAI's GPT-2	ROUGE-1=0.107, ROUGE-2=0.006, ROUGE-l= 0.084, BLEU=9.82
[25]	English	Text simplification	BERT, RoBERTa, XLNet, and GPT-2	Accuracy = 64.5%
[26]	English	Contextual auto-complete of clinical documentation	Named entity recognition (NER), term frequency-inverse document frequency (TF-IDF), and inference language model.	Mean reciprocal rank (MRR) = 0.28, mean average precision (MAP) = 0.27, based on the case of conditions.
[27]	English	Mammography report generation	LSTM-RNN	Accuracy of qualitative assessment is 75%.
[28]	English	Medical report generation	Encoder-decoder based on GPT.	BLEU-1=68.6, BLEU-2=60.8, BLEU-3=55.8, BLEU-4=52.3, ROUGE-L=64.1, and CIDER-D=324.5
[29]	English	Radiology report generation	Graph convolution based on attention	BLEU-1=0.441, BLEU-2=0.291, BLEU-3=0.203 , BLEU-4=0.147, ROUGE=0.367, CIDER=0.304, MIRQI-r=0.483, MIRQI-p=0.490, and MIRQI-F1=0.478
[30]	English & Chinese	Dialog generation of consultations related to COVID-19	Transformer, DialoGPT, and BERT-GPT	BLEU-2=0.046, BLEU-4=0.028, preplexity=10.8, NIST-4=0.36, METEOR=0.122, Entropy=8.5, Dist-1=0.079, Dist-2=0.395
[32]	Japanese	Text generation of chest radiography	CNN, LSTM and self-attention	ROUGE-1=0.8322, ROUGE-2=0.8304, ROUGE-3=0.8261, ROUGE-4=0.8214 (based on the original Japanese Social of Radiological Technology (JSRT) Dataset)
[33]	English	Automatic generation of retinal image captioning	CNN & Bidirectional-LSTM	BLEU-1=0.872, BLEU-2=0.659, BLEU-3=0.519 , BLEU-4=0.443
[34]	English	Generation of mental EHRs	Neural transformer	PPL=5.14, ROUGE-L=0.68, BLEU=37.28, and TER=0.49 (based on MIMIC-III dataset)
[35]	English	Text summarization of COVID-19 research articles	BERT & OpenAI's GPT-2	Visual inspection
[36]	English	Text summarization of biomedical text	BERT-embeddings & Hierarchical clustering	ROUGE-1=0.750, ROUGE-2=0.331

of mammography reports. The proposed ''LSTM-RNN'' yielded promising results when assessed based on the area under the curve (AUC). In [28], the authors proposed an auxiliary-signal encoder-decoder approach for generating radiology medical reports. The model was inspired by the generative pre-training (GPT) model and was trained on the CX-CHR and COVID-19 CT datasets. Evaluations based on the BLEU, AUC, and other scores revealed that the proposed model exhibited a highly favorable hit rate based on human evaluations. Zhang *et al.* [29] proposed a graph convolution-based on attention method for generating radiology reports. The proposed model was assessed based on BLEU, the Medical Image Report Quality Index (MIRQI) of recall, precision, and f1-score. The proposed model produced promising results when compared to models developed in other recent papers. Further, Yang *et al.* [30] adopted the transformer, DialoGPT, and BERT-GPT to generate dialogue to be used in consultations for COVID-19 diagnoses. The model was tested on English and Chinese versions of a medical dialog dataset and evaluated according to various metrics such as BLEU, METEOR, and perplexity. Han *et al.* [31] proposed a neural-symbolic approach for generating spinal medical reports. The constructed model achieved efficient capability in recognizing the spinal structures, even that the authors did not provide any evaluation metrics for assessing the generated texts. However, they reported the accuracy, specificity, and sensitivity of the radiological classification and semantic segmentation processes. In [32], the authors constructed a neural language model for generating text for chest radiography in the Japanese language. The designed encoder-decoder model utilized the CNN with LSTM and self-attention mechanism. The model provided favorable results based on BLEU scores. Moreover, Mishra and Banerjee [33] developed an automatic captioning model (a CNN accompanied by a self-trained bidirectional LSTM) for retinal images. The model was trained on the STARE database and evaluated using the BLEU score. The highest score of 87% was achieved for the BLEU-1. Meanwhile, Ive *et al.* [34] created a model for generating mental EHRs using the neural transformer model. The model was evaluated intrinsically based on perplexity, ROUGE-L, BLEU, and TER (the minimum number of edits), and extrinsically based on text classification. Overall, the model performed well.

Given that text summarization is a sub-field of text generation, Kieuvongngam *et al.* [35] developed an automatic text summarization model for medical research articles, using COVID-19 as a use case. The model is a neural language model based on BERT and OpenAI GPT-2 and was assessed based on its ROUGE score and a visual examination. Finally, Moradi *et al.* [36] proposed a deep learning-based summarizer of biomedical texts. The model utilized contextual BERT embeddings and a hierarchical clustering method for summarizing medical texts. Assessments based on ROUGE scores revealed promising results. However, evaluating such

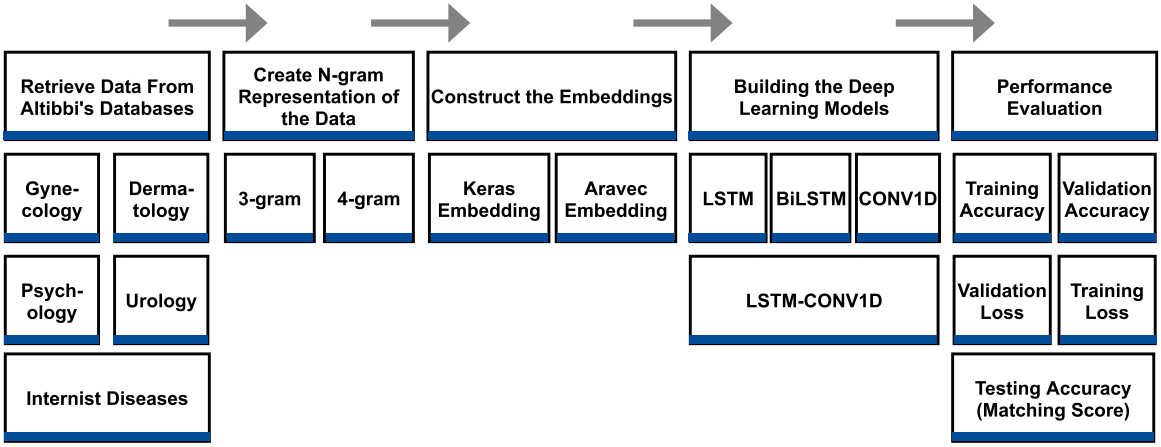


FIGURE 1. An abstract overview of the proposed methodology.

ر	ذ	د	خ	ح	ج	ث	ت	ب	ا
/r/	/ð/	/d/	/x/	/ħ/	/d͡ʒ/	/θ/	/t/	/b/	/a:/
r	ḏ	d	ḫ	ḥ	j	ṯ	t	b	ā
ف	غ	ع	ظ	ط	ض	ص	ش	س	ز
/f/	/ɣ/	/ʕ/	/ dˤ /	/tˤ/	/dˤ/	/sˤ/	/ʃ/	/s/	/Z/
f	ḡ	ʿ	ẓ	ṭ	ḍ	ṣ	š	s	z
		ي	و	ه	ن	م	ل	ك	ق
		/j/	/w/	/h/	/n/	/m/	/l/	/k/	/q/
		y	w	h	n	m	l	k	q

FIGURE 2. A presentation of the Arabic characters' scripts, transliteration, and IPA signs.

approaches lacks standard datasets and frameworks. Table 1 presents a summary of related papers.

III. METHODOLOGY

A. OVERVIEW

This section presents the proposed methodology dissected into three aspects: the data preparation, the system architecture, and the evaluation criteria. Figure 1 describes these aspects in details, where first it starts by collecting and retrieving the data, then the system architecture, which includes the preparation of the n-gram representation of data, constructing the embeddings (i.e., the Keras embeddings and Aravec embeddings), and then building variants of recurrent and convolutional deep learning models. Finally, the performance evaluation criteria are discussed.

B. DATASETS

The employed Arabic-language datasets were obtained from Altibbi. The Arabic language is a Semitic language and the mother tongue of more than 200 million speakers worldwide. It is so common because it is the language of Islam and the means of communication in daily Arabian life. Arabic has two standard forms: Classical Arabic and Modern Standard Arabic. However, among the Arabic countries, the most commonly spoken form of the language is Dialectical Arabic, which differs from one country to another. The Arabic language's 28 characters, along with their Arabic scripts, transliteration, and International Phonetic Alphabet (IPA) signs, are presented in Figure 2.

The datasets were obtained from doctors' recommendations. However, several cleaning steps were required. These steps included removing single characters, English

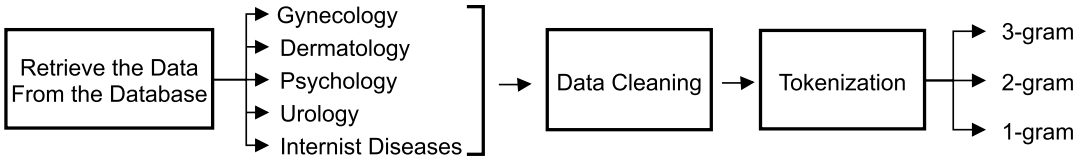


FIGURE 3. Phases of data preparation.

a) He continues the treatment for a week and follows up with the attending physician, and it is necessary to stop smoking if he smokes.

b) يستمر على العلاج لمده اسبوع ويتابع مع الطبيب المعالج ويلزم إيقاف التدخين لو مدخن

يستمر على العلاج لمده	يستمر على العلاج	المعالج ويلزم إيقاف
على العلاج لمده اسبوع	على العلاج لمده	الطبيب المعالج ويلزم
العلاج لمده اسبوع ويتابع	العلاج لمده اسبوع	ويلزم إيقاف التدخين
لمده اسبوع ويتابع مع	لمده اسبوع ويتابع	إيقاف التدخين لو
مع الطبيب المعالج ويلزم	اسبوع ويتابع مع	التدخين لو مدخن
ويلزم إيقاف التدخين لو	ويتابع مع الطبيب	
إيقاف التدخين لو مدخن	مع الطبيب المعالج	

Trigram (3-grams)
Quadgram (4-grams)

FIGURE 4. Description of n-gram dataset: a) an example of a medical recommendation translated into English and b) the recommendation in Arabic. The green tokens represent 3-gram training examples, while blue tokens represent the 4-gram training examples.

characters, recommendations written in English; eliminating punctuation marks, symbols, and Latin symbols; removing blanks and duplicate records, removing regular expressions and emojis. Furthermore, most of the numbers needed to be translated into textual representations. The recommendation information was stored based on the specialty type in the database. As such, five datasets were created, one for each of the most active specialties: gynecology, dermatology, psychology, urology, and internist diseases. Preparing the data for deep learning models requires transforming text sequences into tokens. In the present study, tokens were created using Keras [37] tokenizer to split the sequences based on the space delimiter, where each word is a token. Subsequently, all unique tokens in the corpus were indexed to adapt to the required shape of the learning algorithms. Figure 3 depicts the stages of data preparation. As can be seen, two versions of the datasets were created based on an n-gram formulation.

Language models, such as statistical or neural language models, provide probabilities for words or text sequences. The n-gram model is a sparse representation of text used to build language models. Accordingly, a sequence of n words represents an n-gram model. For example, if the sequence contains two words, then the corresponding n-gram is called a bigram ''2-gram'' model. If the sequence contains three words, then it is a trigram ''3-gram'' language model. In this regard, for each specialty-based dataset, two new datasets were created (i.e., a trigram and a quadgram (4-gram)). The trigram sequence considers two words from the history to predict the third word, whereas the quadgram considers the previous three words.

Figure 4 shows two examples of the created datasets, including the 3-gram and the 4-gram.

C. SYSTEM ARCHITECTURE

Several variants of deep learning models are implemented to process two different versions of the n-gram datasets of the five specialties. As shown in Figure 5, 3-gram and 4-gram datasets were constructed from a corpus of data related to psychiatric diseases, gynecology, dermatology, urology, and internist diseases. The eight versions of datasets were fed into LSTM, BiLSTM, CONV1D, and a combination of LSTM and CONV1D. They were then evaluated quantitatively and qualitatively. This section introduces the networks used and the details of their implementation.

1) EMBEDDINGS

Embeddings are dense, low-dimensional representations of words, documents, or pieces of text that are presented as real-valued vectors. Embedding models are classified into frequency-based and predictive-based models.

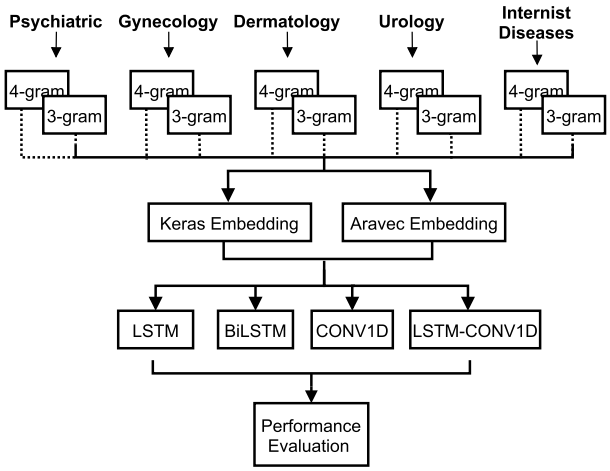


FIGURE 5. An illustration of the methodology.

The frequency-based models are better suited to capture the syntactical and statistical features hidden in text (such a model is the term frequency-inverse document frequency). Whereas, the predictive models are evolved from neural network models such as the Word2Vec, where they can represent the semantics hidden in texts. Aravec [38] is a pre-trained embedding model for the Arabic language. It is developed on data collected from Twitter and Wikipedia with a total number of vocabularies is 3 billion. The developed models were based on the Word2Vec model and based on two types of word embeddings models: the skip-gram (SG) and continuous bag-of-words (CBOW).

2) LSTM AND BILSTM

LSTM [39] is a type of recurrent neural network (RNN) capable of learning the long-term dependencies of data. It solves the problem of vanishing gradients, which is typically experienced by RNNs, by replacing each hidden unit with a memory cell and neural gates to maintain and control its different states. An LSTM network is a chain of connected units as shown in Figure 6-(a). In the figure, x_t is the input at time t, h_t is the hidden state at time t, and y_t is the output at time t. Each LSTM unit consists of three gates as shown in Figure 6-(b). The purple arrow represents the cell state (i.e., the long-term memory). The cell is responsible for storing (remembering) the information provided during the previous interval, and the three gates are responsible for regulating the flow of information by adding or removing (forgetting) information from the cell state. The three gates are the forget, input, and output gates, which are described below the figure.

- *Forget Gate:* A sigmoid function picks the values of the previous hidden state ($h_{(t-1)}$) and the input (x_t) to generate an output value between 0 and 1. When the output of the sigmoid function is 0, the previous cell state (C_{t-1}) is forgotten during the multiplication operation; when it is 1, the previous cell state is remembered.
- *Input Gate:* A sigmoid function takes the values of the previous hidden state ($h_{(t-1)}$) and the input (x_t) to decide which values should be used to update the cell state. The output of the sigmoid function is a value between 0 and 1. Values close to 0 indicate that the information is not important, whereas values close to 1 mean the data is important and should be retained. Then, a tanh function (which is the input modulator) holds the values of the previous hidden state and the input to generate an output vector value between -1 and 1. The data is either written on the cell state or forgotten based on this value. The output of multiplication operation the outputs of the sigmoid and tanh functions were multiplied and added to the output of the forget gate to update the cell state.
- *Output Gate:* It determines the next hidden state. The previous hidden state ($h_{(t-1)}$) and the input (x_t) are fed into a sigmoid function, whereas the modified cell state (C_t) is fed into a tanh function. The multiplication of the outputs of the sigmoid and tanh functions forms the new hidden state ($h_{(t+1)}$), which is the output of the LSTM unit.

On the other hand, the BiLSTM includes two distinct LSTM networks. The first one moves from the left to the right, forms a forward layer, and considers the historical data in the context of a left-to-right language. The other one flows from the right to the left, forms a backward layer, and considers future data. This allows the network to preserve information from previous and subsequent states, thus improving the understanding of the context.

LSTM and BiLSTM are suitable for processing sequences of data, as they recognize the dependency of a sequence based on the order in which the information is presented. Hence, they apply to the textual auto-completion (prediction) of medical recommendations.

3) CNN NETWORK

The convolutional neural network (CNN) was proposed by [40] to predict and classify different kinds of high-dimensional data presented mainly in a grid format (e.g., images, texts, audio, and video files). CNN networks have an input layer and an output layer, as well as a large number of hidden layers. However, they are different from traditional neural networks, as they incorporate a convolution operation in one or more of their hidden layers (called convolution layers). The convolution operation is a mathematical operation used to identify and extract features, though it is different conceptually from the convolution operation used in other fields (i.e., engineering, and pure mathematics). Generally, a CNN consists of three layers: the convolution, activation, and pooling layers (the pooling operation is not mandatory). The convolution operation convolves two functions of real input values, performing a weighted averaging process at each time step t as shown in Equation 1. The asterisk in the equation refers to the convolution operation, the x is the input data, the k is the kernel (filter), and the

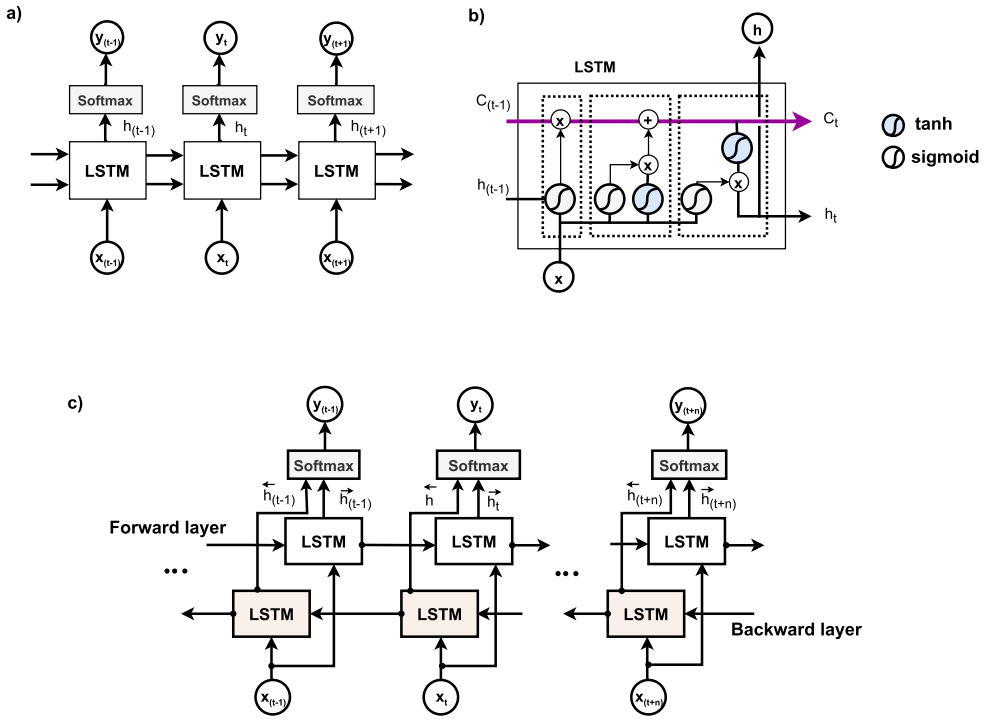


FIGURE 6. LSTM and BiLSTM networks; (a) LSTM network, (b) LSTM cell, and (c) BiLSTM.

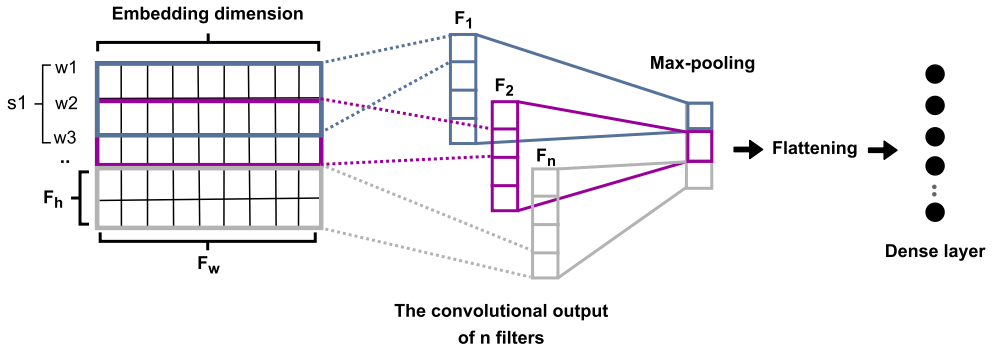


FIGURE 7. The convolution operation for an NLP task. F_h is the height of the filter, F_w is the width of the filter, and n is the number of the filters.

output f is the "feature map."

$$f(t) = (x * k)(t) = \int x(a)k(t-a)da \tag{1}$$

In practice, the real data is multidimensional (tensors), not continuous. Therefore, a discrete convolution operation (as defined in Equation 2) is usually used. This operation considers the input x as 2-dimensional data, as well as the kernel k. The kernel convolves over the input by performing different operations, including the strides and padding.

$$f(t) = (x * k)(i,j) = \sum_n \sum_m x(n,m)k(i-n)(j-m) \tag{2}$$

Convolution is employed to improve computational efficiency and memory utilization, especially when extremely large multidimensional data are involved. This is achieved by using filters that can transform sparse input data into more compact feature sets. Also, the filters reduce the number of weight parameters needed and handle variable lengths of the input data [41]. The activation part of the convolution maintains the positive generated features, which are used to proceed further in the learning process, while negative values are discarded. Meanwhile, at the pooling layer, a downsampling technique is performed to shrink the feature maps by summarizing the statistical properties of the nearby points. Figure 7 describes the convolution and pooling operations of the CNN network.

This convolution, activation, and pooling process is repeated many times throughout all convolutional layers to

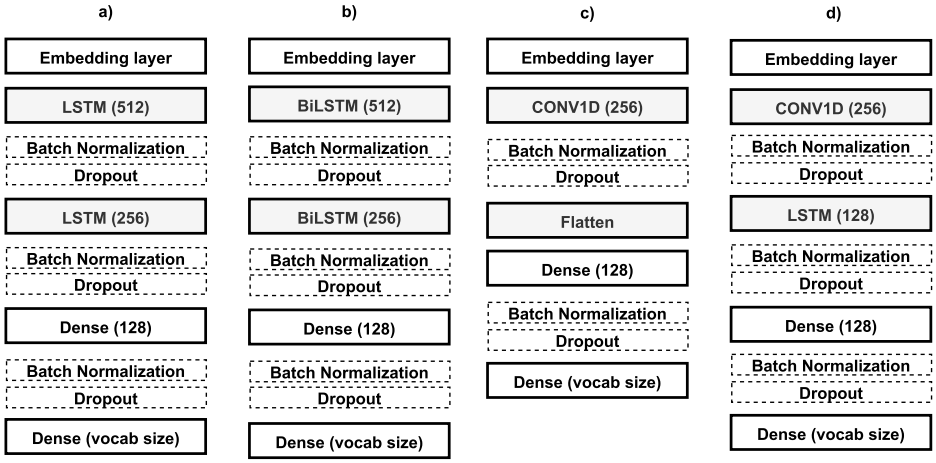


FIGURE 8. Description of the implemented structures of a) LSTM, b) BiLSTM, c) CONV1D, and d) LSTM-CONV1D.

produce a highly robust representation of features. The generated feature maps are flattened to vectors and then entered into a fully connected neural network layer for classification and prediction. The combination of convolution and sequence models can also be considered for processing textual data sequences. The consolidated features generated from a convolutional model are considered spatial features that serve as the input for the following sequence model. The sequence model is designed to handle the produced features from the convolution as sequences and subsequently extract new chronological features of the text.

4) MODELS TUNING AND SELECTION

The variants of deep learning models (i.e., LSTM, BiLSTM, CONV1D, and LSTM-CONV1D) used in this work were initially implemented to process sequences of data like textual data. Hence, the implemented methodology used altered versions of LSTM, BiLSTM, CONV1D, and LSTM-CONV1D for next word prediction. At first, the four models were tested at roughly 10 epochs to determine which model is most suitable for next word prediction. For this purpose, all models were configured at a learning rate of 0.001. The batch size was 128, the activation function was the rectified linear unit (ReLU), the optimizer was Adam (derived from the adaptive moment estimation method), and the filter size was either 3 or 2 depending on which n-gram model was utilized. Moreover, the models were tested at three different dimensions of embeddings (100, 200, and 300), at different number of epochs, and at two embedding types (i.e., Keras embedding, and Aravec embedding). Besides, evaluated based on the training and validation of accuracy and loss. The Keras embedding is offered by the Keras library, in which the Keras embedding layer is initialized by random weights and then tuned during the training process.

The best-obtained model of each LSTM, BiLSTM, CONV1D, and LSTM-CONV1D was further tuned and optimized. However, optimizing the models included adding regularization and dropout parameters, batch normalization to ensure that the output of each layer is correctly normalized and scaled. This provided more stable networks and sped up the learning process. Furthermore, the models were implemented at increased number of epochs (30 epochs) to expand the capacity of the algorithms to improve the convergence over a large number of epochs. These enhanced models were evaluated based on the training and validation accuracy and loss, as well as the accuracy of their matching scores at testing.

The optimized models are shown in Figure 8. The structures of LSTM, BiLSTM, CONV1D, and LSTM-CONV1D are presented in sub-figures (a) and (b). The structure of LSTM comprises two stacked LSTM layers. First, it takes the embedding layer as input with three main parameters (i.e., the vocab size; the embedding dimension; the weights, and the input length, which is the maximum sequence length). The two stacked layers of LSTM consist first of 512 units and then 256 units, which then followed by a dense layer of 128 neurons. The LSTM layers and the dense layers are separated by two layers of batch normalization and dropout with a percentage of 40%. The number of neurons in the final dense layer represents the number of classes. As the problem of interest is the next word prediction, the number of classes in this case is the number of unique vocabularies in the dataset. Meanwhile, its form of activation is Softmax activation. The softmax layer is responsible for computing the probabilities of the output classes according to Equation 3, where z represents the weighted sum of the input at layer l, j is the number of neurons at the current layer, and k is the number of neurons in the previous layer.

$$a_j^l = \frac{e^{z_j^l}}{\sum_k e^{z_k^l}} \tag{3}$$

The structure of the BiLSTM network is similar (as shown in Figure 8-(b)), it consists of an embedding layer, two stacked BiLSTM layers (of 512 and 256 units), a dense layer of 128 neurons, and a fully connected layer of the vocabulary size. It also contains the separating layers of batch normalization and dropout. The structure implemented for the CONV1D network is described in Figure 8-(c). This network includes an embedding layer, a convolution layer of one dimension (capable of processing textual data with 256 filters), and a flattening layer in which the output is shaped into an acceptable format so that it can enter the next dense layer of 128 neurons. A batch normalization layer, a dropout layer, and the last fully connected dense layer are also part of this network. The structure of the LSTM-CONV1D is illustrated in Figure 8-(c), where the output of a convolution layer of 256 filters is fed into an LSTM network with 128 units before entering a dense layer with a size of 128. As with the other networks, the LSTM-CONV1D network adopts batch normalization and dropout layers, as well as an embedding layer and the final fully connected layer.

The models described above were trained on two versions of datasets (i.e., the 4-gram, and 3-gram) for each of the five specialties. They were then evaluated quantitatively based on the matching scores (discussed in subsequent sections).

D. EVALUATION CRITERIA

The models were evaluated based on the testing accuracy, formulated in terms of the matching score. The matching score compares the actual labels y of the testing part of the dataset with the highest five likelihood predictions of the developed learning model. To illustrate, for each class, the learning model produces a probability that any given word will be the next word. Hence, each testing sample in the dataset has five possible predictions as in Equation 4, which are the highest probabilities. Consequently, any match between the five predictions and the truth label is substituted by a value (1). This is described by Equations 5, where the x variable presents the truth value, and $\hat{y}$ is the predicted value. All predictions that match the truth labels across the testing dataset are summed and averaged to form a matching score (as in Equation 6 denoted by $Accuracy_{ms}$). In which, m is the number of samples in the testing set. However, matching is done syntactically. Therefore, any prediction that is slightly different from the truth label in terms of syntax due to normalization is considered a non-match.

$$\hat{Y} = \{\hat{y_1},\ \hat{y_2},\ \hat{y_3},\ \hat{y_4},\ \hat{y_5}\} \tag{4}$$

$$f(x) = \begin{cases} 1 & if\ \ \hat{y} = x\ \wedge\ \exists(\hat{y} \in \hat{Y}) \\ 0 & otherwise \end{cases} \tag{5}$$

$$Accuracy_{ms} = \frac{1}{m}\sum_{i}^{m} f(x_i) \tag{6}$$

For instance, considering a dataset of three sentences ($m = 3$), which are the ''The best treatment is'', ''Do a sonar on'', and ''Wash with warm water''. And the proposed model predicted the potential next word for the first sentence as {''The peel'', ''The laser'', ''With'', ''If'', and ''Sessions''}. For the second sentence: {''The belly'', ''The testicles'', ''The testis'', ''The prostate'', and ''The pelvis''}. And for the third sentence, they are {''Calming down'', ''And take a lot of'', ''Or'', ''On'', and ''Use''}. If the truth value of the first sentence is ''The laser'', and for the second is ''The chest'', and for the third is ''Calming down''. Thus, as the first and third sentences match, then the $Accuracy_{ms} = (1 + 0 + 1)/3 = 67\%$.

TABLE 2. The parameters settings for implementing deep learning models.

Name of parameter	Value
Learning rate	0.001
Optimizer	Adam
Loss function	Sparse categorical crossentropy
Batch size	128
Filter size	Height is 3 for 4-gram & 2 for 3-gram, Width is the embedding dimension
Stride size	1
Activation	Relu
Regularizer	L1-L2 (l1=1e-5, l2=1e-4)
Dropout rate	0.4

IV. EXPERIMENTS AND RESULTS

This section presents the details of the experiments set up, and discusses the results in regard to three different analyses: the first subsection studies the influence of the embedding dimension, the second interprets the effect of the number of epochs, and the third investigates the embedding type.

A. EXPERIMENTAL DETAILS

Regarding the hardware settings of the experiments, the development platform was Google Colaboratory (Colab). Regarding the Colab, the processor was Intel(R) Xeon(R) CPU @ 2.00GHz, and the memory was 27 GB. Regarding the cloud server, the Python version 3.7.3 was used on Ubuntu-1804-bionic-64 cloud server, the memory was 64 GB, and the processor was an Intel(R) Core(TM) i7-7700 with a speed of 3.6 GHz. Meanwhile, the used GPU was GeForce GTX 1080 (8 GB). Furthermore, the utilized deep learning framework was Keras [37], which was built on top of Tensorflow [42]. Table 2 describes the parameters according to the settings of the deep learning models. In the table, $l1$ and $l2$ are the penalties of the regularizers. Besides, the utilized embedding models are the Keras embedding, and the Aravec-Twitter-CBOW at dimension 300.

B. EFFECT OF EMBEDDING DIMENSION

This subsection presents a discussion of the performance of LSTM, BiLSTM, CONV1D, and LSTM-CONV1D based on their training accuracy (T. Accuracy), validation accuracy (V. Accuracy), training loss (T. Loss), and validation

TABLE 3. A comparison of training and validation accuracy and loss based on the 3-gram datasets of gynecology, dermatology, psychology, urology, and internist diseases. (E.D.) is the embedding dimension.

Specialty	Model	T. Accuracy	V. Accuracy	T. Loss	V. Loss	E.D.
Gynecology	LSTM	0.324	0.245	3.595	**7.408**	100
	LSTM	0.340	0.250	3.446	7.602	200
	LSTM	**0.344**	**0.251**	**3.414**	7.611	300
	BiLSTM	0.327	0.248	3.578	**7.382**	100
	BiLSTM	0.338	**0.253**	3.431	7.731	200
	BiLSTM	**0.348**	0.252	**3.369**	7.687	300
	CONV1D	**0.365**	**0.255**	**3.209**	7.977	100
	CONV1D	0.361	0.254	3.252	**7.649**	200
	CONV1D	**0.365**	0.253	3.217	7.661	300
	LSTM-CONV1D	0.334	0.248	3.547	**7.512**	100
	LSTM-CONV1D	0.343	0.248	3.485	7.627	200
	LSTM-CONV1D	**0.352**	**0.250**	**3.383**	7.544	300
Dermatology	LSTM	0.359	0.246	3.425	**9.056**	100
	LSTM	0.387	0.246	3.193	9.611	200
	LSTM	**0.401**	**0.253**	**3.034**	9.602	300
	BiLSTM	0.373	0.249	3.289	**9.392**	100
	BiLSTM	0.397	0.257	3.093	9.720	200
	BiLSTM	**0.408**	**0.258**	**3.017**	9.639	300
	CONV1D	0.504	0.268	2.319	10.192	100
	CONV1D	0.533	0.267	2.129	**10.188**	200
	CONV1D	**0.537**	**0.272**	**2.085**	10.440	300
	LSTM-CONV1D	0.390	0.253	3.263	**9.038**	100
	LSTM-CONV1D	0.412	0.255	3.090	9.062	200
	LSTM-CONV1D	**0.414**	**0.258**	**3.066**	9.206	300
Psychology	LSTM	0.3641	0.229	3.406	**11.011**	100
	LSTM	0.3987	0.238	3.158	11.443	200
	LSTM	**0.4264**	**0.243**	**2.936**	11.398	300
	BiLSTM	0.4063	0.237	3.062	**11.018**	100
	BiLSTM	0.4368	**0.245**	2.850	11.709	200
	BiLSTM	**0.4439**	**0.245**	**2.772**	11.969	300
	CONV1D	0.556	0.259	2.074	**11.988**	100
	CONV1D	0.597	0.258	1.846	12.265	200
	CONV1D	**0.624**	**0.263**	**1.665**	12.198	300
	LSTM-CONV1D	0.367	0.231	3.583	**10.292**	100
	LSTM-CONV1D	0.428	**0.248**	3.025	11.103	200
	LSTM-CONV1D	**0.433**	0.247	**3.021**	10.932	300
Urology	LSTM	0.416	0.311	3.117	**8.750**	100
	LSTM	0.444	0.315	2.874	8.918	200
	LSTM	**0.482**	**0.323**	**2.569**	9.012	300
	BiLSTM	0.452	0.322	2.768	**8.627**	100
	BiLSTM	0.483	**0.330**	2.548	8.883	200
	BiLSTM	**0.497**	0.330	**2.453**	9.298	300
	CONV1D	0.595	0.340	1.833	**9.379**	100
	CONV1D	0.617	0.342	1.683	9.552	200
	CONV1D	**0.634**	**0.343**	**1.566**	9.718	300
	LSTM-CONV1D	0.457	0.320	2.894	**8.486**	100
	LSTM-CONV1D	0.479	0.326	2.726	8.739	200
	LSTM-CONV1D	**0.493**	**0.330**	**2.602**	8.624	300
Internist diseases	LSTM	0.640	0.592	1.890	**4.408**	100
	LSTM	0.651	**0.594**	1.789	4.578	200
	LSTM	**0.656**	0.593	**1.753**	4.586	300
	BiLSTM	0.647	0.591	1.801	**4.554**	100
	BiLSTM	0.655	0.592	1.716	4.673	200
	BiLSTM	**0.667**	**0.596**	**1.631**	4.717	300
	CONV1D	0.728	0.603	1.188	**5.341**	100
	CONV1D	0.741	0.604	1.106	5.517	200
	CONV1D	**0.743**	**0.604**	**1.095**	5.487	300
	LSTM-CONV1D	0.663	0.597	1.753	4.682	100
	LSTM-CONV1D	0.669	0.599	1.691	4.705	200
	LSTM-CONV1D	**0.676**	**0.600**	**1.627**	**4.665**	300

loss (V. Loss) for gynecology, dermatology, psychology, urology, and internist diseases at 4-grams and 3-grams. A sensitivity analysis was also conducted based on the embedding dimension for the four models over the five specialties.

Table 3 presents data regarding the performance of LSTM, BiLSTM, CONV1D, and LSTM-CONV1D based on the 3-grams formulations of the datasets. Regarding gynecology, the training accuracy was showing an increasing relationship with the increasing number of the embedding dimension. This was found even for the CONV1D at the same dimensions of 100 and 300 with a value of (36.5%). Meanwhile, the LSTM attained the lowest training accuracy of 34.4%. In terms of validation accuracy, both LSTM and LSTM-CONV1D were increasing with the embedding dimension, presenting close maximum accuracy values of 25.1% and 25%, respectively. BiLSTM and CONV1D did not perform at the same level. BiLSTM achieved the best at dimension 200 (52.3%), while CONV1D performed the best at dimension 100 (25%). Similar behavior was noticed for the training loss, as the best score obtained by CONV1D (loss = 3.209).

Regarding validation loss, the best scores of LSTM, BiLSTM, and LSTM-CONV1D were achieved at dimension 100, while CONV1D produced its best score at dimension 200. All models obtained relatively close loss values, with the best value of 7.382 achieved by BiLSTM. Meanwhile, CONV1D outperformed the other models in terms of training accuracy, training loss, and validation accuracy (36.5%, 3.209, and 25.5%, respectively) at embedding dimension 100.

For the dermatology specialty, training accuracy increased as the embedding dimension increased. The maximum accuracy of 53.7% was achieved by CONV1D at dimension 300. Regarding validation accuracy, each model attained its best performance at dimension 300. CONV1D performed the best, as it yielded a validation accuracy of 27.2%. LSTM, with an accuracy of 25.3%, was the worst model in this regard. Each model achieved its lowest training loss score at dimension 300. CONV1D obtained the best training loss of 2.085. For validation loss, LSTM-CONV1D achieved the lowest minimal loss of 9.038 at dimension 100, while LSTM-CONV1D produced the highest validation loss of 10.188 at dimension 200. Among all models, CONV1D performed the best in terms of training accuracy, training loss, and validation accuracy.

For the psychiatric dataset, the models performed similarly as for the dermatology dataset. Specifically, the models' training accuracy increased alongside the embedding dimension. LSTM achieved a lower training accuracy (42.64%) than LSTM-CONV1D (43.3%) and BiLSTM (44.39%), while CONV1D had the highest accuracy of all (62.4%). Considering validation accuracy, LSTM, BiLSTM, and LSTM-CONV1D achieved accuracies around 24%. CONV1D outperformed the other models, achieving a validation accuracy of 26.3% at dimension 300. In terms of training loss, all models presented their best scores at embedding dimension 300. At this dimension, LSTM-CONV1D attained a score of 3.021, LSTM attained a score of 2.936, BiLSTM attained a score of 2.772, and CONV1D presented the best score of 1.665. All four models obtained their best validation loss performance at embedding dimension 100. In this case, CONV1D performed the worst, with a score of 11.988. LSTM and BiLSTM did slightly better of (11.011, and 11.018, respectively), whereas, LSTM-CONV1D obtained the best score of 10.292.

Regarding the urology dataset, the training accuracy and loss increased when the embedding size was maximized. CONV1D achieved the best accuracy and loss of 63.4%, and 1.566, respectively, when the embedding dimension was 300. Considering validation accuracy, CONV1D, LSTM-CONV1D, and LSTM achieved their best performances (34.3%, 33%, and 32.3%, respectively) at dimension 300. Meanwhile, the BiLSTM model achieved its best validation accuracy of 33% at dimension 200. LSTM-CONV1D achieved its best validation loss score of 8.486 at dimension 100, and the LSTM and BiLSTM models yielded similar scores. CONV1D performed the worst, producing a validation loss score of 9.379. For the internist diseases dataset, training accuracy and loss increased as the embedding dimension number increased, and all models presented their best performance at embedding dimension 300. LSTM, BiLSTM, and LSTM-CONV1D achieved similar training accuracies of 65.6%, 66.7%, and 67.6%, respectively. CONV1D performed the best, yielding an accuracy of 74.3%. CONV1D also produced the best training loss score of 1.095. BiLSTM, LSTM-CONV1D, and CONV1D obtained their highest validation accuracy scores (59.6%, 60%, and 60.4%, respectively) at embedding dimension 300. Meanwhile, LSTM performed the best at dimension 200, presenting an accuracy of 59.4%. Regarding validation loss, LSTM, BiLSTM, and CONV1D produced their best results at dimension 100, with LSTM providing the best score of 4.408. LSTM-CONV1D achieved its best loss (4.665) at dimension 300.

Table 4 shows the performance of the four deep learning models based on the 4-gram representation and over the five specialties. Regarding the gynecology dataset and the LSTM model, the best training accuracy and loss results were achieved at embedding dimension 300. However, this model presented its best validation accuracy at embedding dimension 200 and its highest validation loss at embedding dimension 100. The model clearly exhibited overfitting, yet its training accuracy increased when the dimension number increased. Similar outcomes were found for BiLSTM, with the best training results observed at dimension 300 and the best validation results found at dimension 200. Nevertheless, the training accuracy and loss performances increased as the value of embedding increased. The CONV1D model presented its best training accuracy, training loss, and validation accuracy results at dimension 200, while it achieved its best validation loss at dimension 300. No clear relationship was detected between this model's performance and the embedding dimension, as it peaked at dimension 200 before dropping at dimension 300. Meanwhile, the LSTM-CONV1D model produced its best training accuracy and loss scores at embedding dimension 200 and its best validation performance at dimension 300. The maximum validation accuracy score obtained by LSTM, BiLSTM, and CONV1D was 25.3%, and the minimum validation loss (7.163) was achieved by CONV1D.

Regarding dermatology, increasing the embedding dimension improved training accuracy, training loss, and validation accuracy for all four tested models. Differently, validation loss is the best at embedding dimension 100. The best results for LSTM, BiLSTM, and LSTM-CONV1D in terms of training accuracy, training loss, and validation accuracy were observed at dimension 300. Meanwhile, CONV1D's performance fluctuated across the dimensions. Among all models, the best training and validation accuracies (66.3% and 27%, respectively) were obtained by CONV1D. This model also produced the best training loss score of 1.493. The best validation loss score (9.95) was achieved by LSTM-CONV1D. The same behavior was exhibited for the psychiatric diseases dataset, as scores for all four metrics increased when

TABLE 4. A comparison of training and validation accuracy and loss based on the 4-gram datasets of gynecology, dermatology, psychology, urology, and internist diseases.

Specialty	Model	T. Accuracy	V. Accuracy	T. Loss	V. Loss	E.D.
Gynecology	LSTM	0.348	0.249	3.394	**7.955**	100
	LSTM	0.366	**0.253**	3.242	8.247	200
	LSTM	**0.379**	0.250	**3.170**	8.458	300
	BiLSTM	0.366	**0.253**	3.267	**8.237**	100
	BiLSTM	0.378	**0.253**	3.152	8.446	200
	BiLSTM	**0.389**	0.252	**3.071**	8.631	300
	CONV1D	0.391	0.252	3.152	8.145	100
	CONV1D	**0.411**	**0.253**	**2.971**	8.311	200
	CONV1D	0.357	**0.253**	3.510	**7.163**	300
	LSTM-CONV1D	0.375	0.246	3.331	8.204	100
	LSTM-CONV1D	**0.393**	0.247	**3.163**	8.311	200
	LSTM-CONV1D	0.354	**0.248**	3.554	**7.472**	300
Dermatology	LSTM	0.417	0.249	2.993	**10.163**	100
	LSTM	0.457	0.253	2.694	10.661	200
	LSTM	**0.472**	**0.255**	**2.608**	10.910	300
	BiLSTM	0.420	0.251	2.947	**10.252**	100
	BiLSTM	0.460	0.257	2.685	10.751	200
	BiLSTM	**0.475**	**0.258**	**2.596**	10.716	300
	CONV1D	0.606	0.264	1.828	**11.930**	100
	CONV1D	0.657	**0.270**	1.526	12.010	200
	CONV1D	**0.663**	0.266	**1.493**	12.319	300
	LSTM-CONV1D	0.424	0.249	3.139	**9.950**	100
	LSTM-CONV1D	0.465	0.249	2.825	10.470	200
	LSTM-CONV1D	**0.494**	**0.257**	**2.597**	10.614	300
Psychology	LSTM	0.426	0.234	2.970	**12.079**	100
	LSTM	0.468	0.234	2.691	12.441	200
	LSTM	**0.489**	**0.240**	**2.566**	12.817	300
	BiLSTM	0.449	0.239	2.761	**12.071**	100
	BiLSTM	0.494	0.245	2.478	13.095	200
	BiLSTM	**0.521**	**0.249**	**2.297**	13.299	300
	CONV1D	0.639	0.253	1.724	**14.138**	100
	CONV1D	0.702	0.255	1.383	14.626	200
	CONV1D	**0.727**	**0.260**	**1.233**	14.974	300
	LSTM-CONV1D	0.435	0.234	3.119	**11.732**	100
	LSTM-CONV1D	0.472	0.235	2.796	12.677	200
	LSTM-CONV1D	**0.491**	**0.240**	**2.709**	12.300	300
Urology	LSTM	0.465	0.316	2.751	**9.619**	100
	LSTM	0.513	**0.329**	2.418	10.131	200
	LSTM	**0.540**	0.328	**2.242**	10.142	300
	BiLSTM	0.496	0.329	2.525	**9.421**	100
	BiLSTM	0.529	0.334	2.280	10.121	200
	BiLSTM	**0.552**	**0.337**	**2.130**	10.191	300
	CONV1D	0.691	0.341	1.415	**11.336**	100
	CONV1D	0.760	**0.350**	1.036	11.656	200
	CONV1D	**0.768**	0.347	**1.000**	12.036	300
	LSTM-CONV1D	0.487	0.318	2.814	**9.590**	100
	LSTM-CONV1D	0.550	0.331	2.341	10.163	200
	LSTM-CONV1D	**0.586**	**0.340**	**2.086**	10.351	300
Internist diseases	LSTM	0.670	0.613	1.742	**4.474**	100
	LSTM	0.689	0.615	1.591	4.695	200
	LSTM	**0.698**	**0.618**	**1.536**	4.774	300
	BiLSTM	0.681	0.618	1.626	4.556	100
	BiLSTM	0.676	0.619	1.690	**4.401**	200
	BiLSTM	**0.703**	**0.621**	**1.465**	4.836	300
	CONV1D	0.742	0.625	1.231	4.903	100
	CONV1D	0.736	**0.627**	1.301	**4.864**	200
	CONV1D	**0.773**	**0.627**	**1.032**	5.228	300
	LSTM-CONV1D	0.690	0.614	1.694	**4.755**	100
	LSTM-CONV1D	0.713	0.616	1.506	5.144	200
	LSTM-CONV1D	**0.715**	**0.618**	**1.499**	4.810	300

the value of the embedding dimension increased. CONV1D performed the best in terms of training and validation accuracy (72.7% and 26%, respectively), as well as training loss (1.233). LSTM-CONV1D achieved the minimum validation loss score (11.732).

For the urology dataset, the training accuracy, training loss, and validation accuracy performances of BiLSTM and LSTM-CONV1D gradually improved. While LSTM-CONV1D produces the best scores for training accuracy, training loss, and validation accuracy, BiLSTM produced the best validation loss score (9.421). LSTM and CONV1D did not exhibit a uniform trend regarding the relationship between the evaluation metrics with the value of the embedding dimension. CONV1D performed the best in terms of training accuracy, training loss, and validation accuracy (76.8%, 35%, and 1.000, respectively). The models behaved similarly for the internist diseases dataset as they did for the dermatology and psychiatric diseases datasets. Specifically, their performances tended to improve as the value of the embedding dimension increased. CONV1D achieved the maximum

TABLE 5. A comparison of the best models based on the training and validation accuracy and loss for five specialties based on Keras embeddings.

Specialty	Model	T. Accuracy	V. Accuracy	T. Loss	V. Loss	E.D.	N-gram	Epochs
Gynecology	LSTM	0.339	0.251	4.222	6.624	200	4-grams	30
	BiLSTM	0.366	0.253	4.055	6.931			
	CONV1D	**0.455**	**0.263**	**3.172**	6.634			
	LSTM-CONV1D	0.324	0.249	4.368	**6.230**			
Dermatology	LSTM	0.440	0.267	3.391	7.851	300	4-grams	30
	BiLSTM	0.490	0.269	3.222	8.576			
	CONV1D	**0.622**	**0.279**	**2.131**	7.864			
	LSTM-CONV1D	0.411	0.264	3.567	**7.510**			
Psychology	LSTM	0.494	0.260	2.965	9.378	300	4-grams	30
	BiLSTM	0.557	0.262	2.762	10.339			
	CONV1D	**0.707**	**0.273**	**1.688**	8.910			
	LSTM-CONV1D	0.463	0.256	3.129	**8.768**			
Urology	LSTM	**0.743**	**0.363**	**1.435**	7.492	300	4-grams	30
	BiLSTM	0.634	0.356	2.378	8.266			
	CONV1D	0.560	0.354	7.633	**2.582**			
	LSTM-CONV1D	0.517	0.349	7.180	7.227			
Internist diseases	LSTM	0.702	0.631	1.949	3.790	300	4-grams	30
	BiLSTM	0.738	0.629	1.834	4.192			
	CONV1D	**0.780**	**0.632**	**1.308**	3.787			
	LSTM-CONV1D	0.683	0.627	2.044	**3.467**			

training accuracy of 77.3%, the maximum validation accuracy of 62.7%, and the best training loss score of 1.032. Meanwhile, BiLSTM outperformed the other models in terms of validation loss, achieving a minimum score of 4.401.

In conclusion, comparing the performances of the models, both at 3-grams and 4-grams, revealed that all models performed better in the 4-gram representation of the datasets than the 3-gram representation. Therefore, the best of the 4-grams models for each of the five specialties were used for further optimization to maximize their performance. This optimization process is discussed in the following subsection.

C. EFFECT OF NUMBER OF EPOCHS

This subsection discusses the performance of the models after optimization. The best model (between LSTM, BiLSTM, CONV1D, and LSTM-CONV1D) for each specialty was optimized based on its structure and then implemented at an increased number of epochs (30). Table 5 represents the models' performances in terms of the training accuracy and loss, and validation accuracy and loss (the 4-gram model was considered in all cases). Regarding the gynecology dataset, CONV1D achieved superior results in terms of training accuracy, training loss, and validation accuracy (45.5%, 3.172, and 26.3%, respectively), at embedding dimension 200. Meanwhile, LSTM-CONV1D achieved the lowest validation loss score of 6.230. For the dermatology dataset, CONV1D again performed the best in terms of training accuracy, training loss, and validation accuracy (62.2%, 2.131, and 27.9%, respectively) at dimension 300. LSTM-CONV1D performed the best in terms of validation loss, with a score of 7.510. CONV1D also produced the best training accuracy, training loss, and validation accuracy scores, both for the psychology dataset (70.7%, 1.688, and 27.9%, respectively) and the internist diseases dataset (78%, 1.308, and 63.2%, respectively). LSTM-CONV1D once again achieved the best validation loss scores for these datasets (8.768 for psychology and 3.467 for internist diseases). Finally, for the urology dataset, LSTM performed the best in terms of training accuracy, training loss, and validation accuracy at dimension 300 (74.3%, 1.435, and 36.3%, respectively). CONV1D achieved the best validation loss score (2.582).

Figure 9 shows Bar chart visualization of the training and validation accuracies for gynecology, dermatology, psychology, urology, and internist diseases.

Figure 10 presents the convergence curves based on the training and validation accuracies of the best models for each of the five specialties. The convergence curves illustrate the models' performances in terms of accuracy over 30 epochs. The figure clearly shows that, for all datasets, the models' training accuracies increased smoothly over the epochs. CONV1D produced the best results for the gynecology, dermatology, psychology, and internist diseases datasets, while LSTM was the best performer for the urology dataset. Also, the maximum accuracy was achieved for the internist diseases dataset (approximately 80%), while the gynecology dataset exhibited the worst performance (approximately half of the internist dataset). Moreover, inspecting the convergence of the validation accuracy shows that for the gynecology, dermatology, psychology, and urology datasets, a maximum convergence of nearly 30% on average was achieved. In contrast, the best convergence of validation accuracy of the internist diseases dataset was roughly 60%. Table 6 displays examples of the predictions generated by the best models (i.e., LSTM, BiLSTM, and CONV1D) for the gynecology, dermatology, psychiatric diseases, urology, and internist diseases datasets. The table presents the testing samples, their English

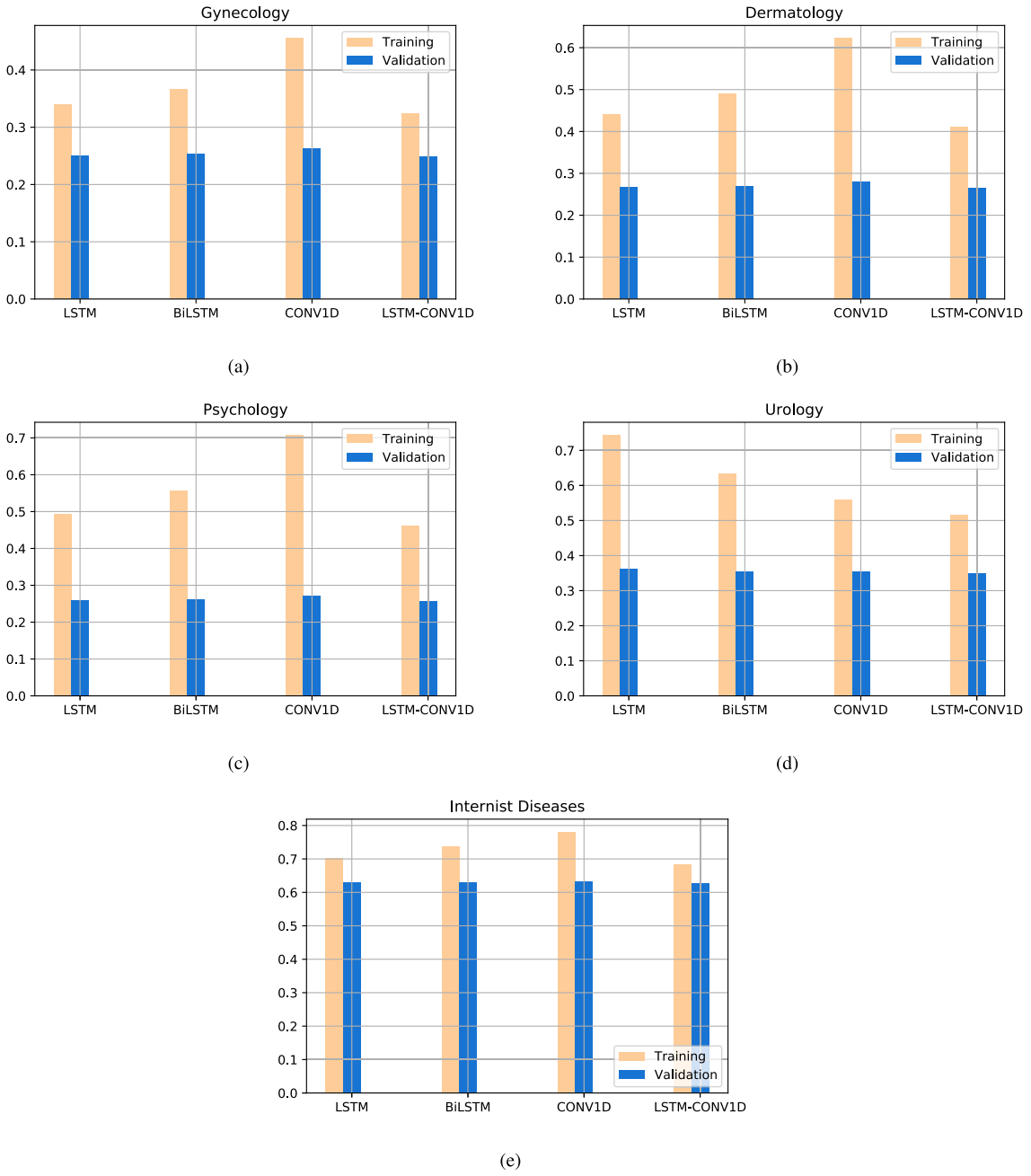


FIGURE 9. Bar chart depiction of the validation and training accuracy for five medical specialties; a) gynecology, b) dermatology, c) psychology, d) urology, and e) internist diseases.

translations, and their predictions (both in Arabic and English). Quantitatively, the best-performing model for the internist diseases dataset was the CONV1D model. However, the BiLSTM model generated more relevant predictions than the CONV1D model.

Furthermore, comparing the performance of the best-obtained models in terms of their matching scores is presented in Table 7. The table shows the testing accuracies of the best models, in terms of the matching scores, for each of the five specialties. It can be seen that the best matching score for the gynecology dataset was obtained by CONV1D (44.9%). This score means that the model can predict the exact words as provided in the ground-truth nearly half the time. For the dermatology dataset, the LSTM, BiLSTM, CONV1D, and LSTM-CONV1D models performed similarly, with the CONV1D achieving the best of a matching score of 26.8%. Similar findings were revealed for the psychology dataset, as the best matching score of 26.8% was achieved by CONV1D. For the urology dataset, the four models presented very similar results, but the CONV1D

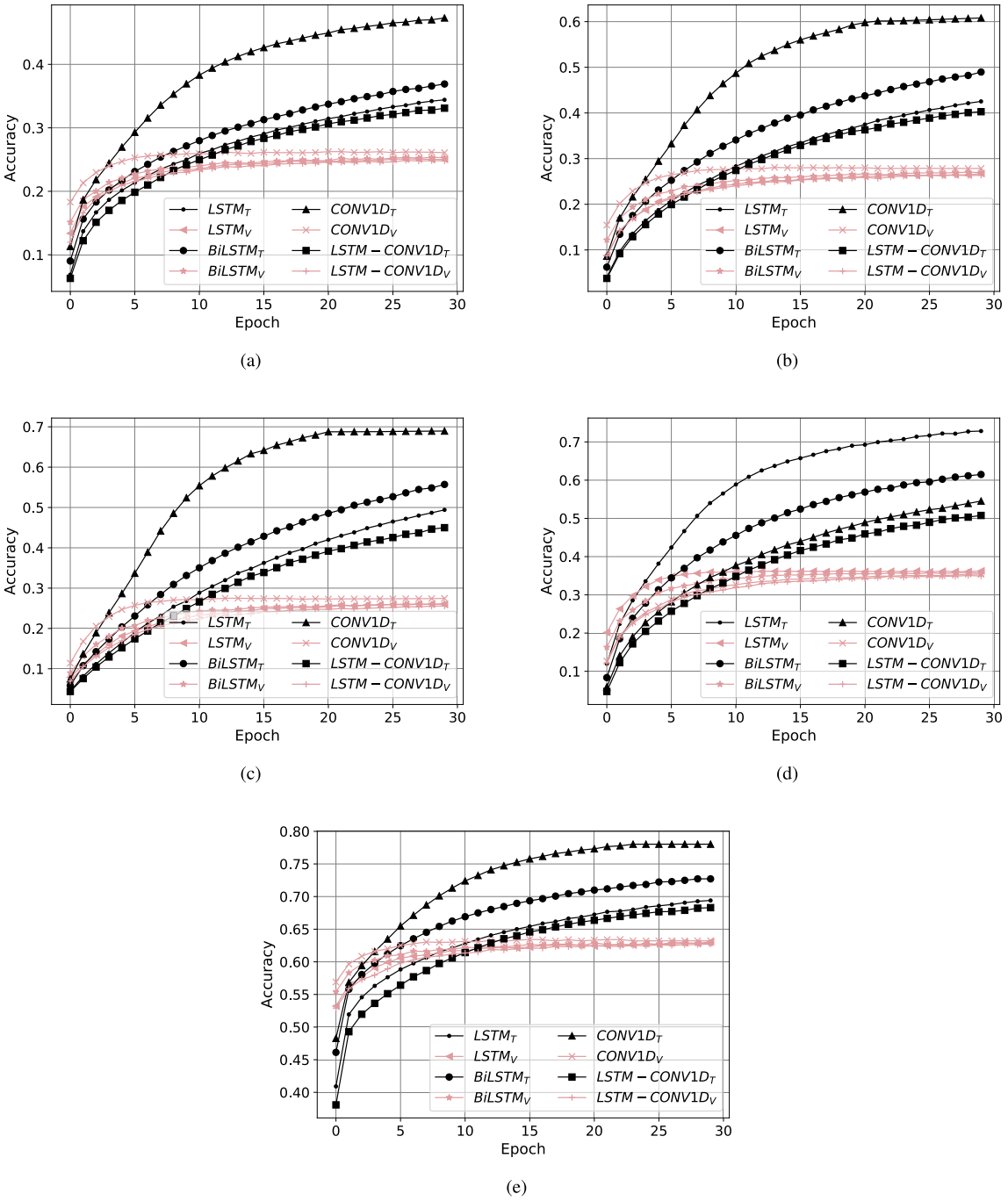


FIGURE 10. **Comparison of the validation and training accuracy for five medical specialties; a) gynecology, b) dermatology, c) psychology, d) urology, and e) internist diseases.**

once again provided the best result, with a value of 39%. Regarding the internist diseases dataset, LSTM, CONV1D, and LSTM-CONV1D achieved matching scores of 29.7%, 29.5%, and 29.7%, respectively. The BiLSTM model outperformed others, attaining a matching score of 30.3%.

D. EFFECT OF EMBEDDING TYPE

This subsection presents the effect of the embedding type on the performance results. The Aravec embedding was implemented to study its influence on the models' performance over the Keras embedding. Table 8 shows the results of the developed deep learning models based on the Aravec embedding and regarding the training accuracy and loss, and validation accuracy and loss, when the embedding dimension is 300, the n-gram representation is 4, and the epochs are 30.

For gynecology, the best training and validation accuracies were obtained by the BiLSTM model (19.5%, 21.1%, respectively). Whereas, the best training and validation

TABLE 6. Examples of predictions of the best-obtained models for the five specialties.

Specialty	Model	Testing samples	Translation	Predictions				
				1	2	3	4	5
Gynecology	CONV1D	هي أعراض التهاب	Are symptoms of inflammation	الحوض The pelvis	المهبل The vagina	بعنق In a neck	في In	الرحم The womb
		تحليل الحمل بعد	Pregnancy analysis after	اسبوع Week	موعد A time of	يومين Two days	ايام Days	يوم A day
		يمكن تناول حبوب	You can take pills	منع Suppressing	دوفاستون Duphaston	دافلون Daflon	كليمين Climen	بسكوبان Buscopan
Dermatology	CONV1D	الغسل بالماء الدافي	Wash with warm water	يهدئ Calming down	والاكثار And take a lot of	او Or	على On	استخدام Use
		تجنب التعرض لاشعه	Avoid exposure to radiation	الشمس Of the sun	شمس Of sun	شمسي Sunny	عند At	الاحتكاك The friction
		العلاج الأفضل هو	The best treatment is	التقشير The peel	الليزر The laser	مع With	اذا If	جلسات Sessions
Psychiatric	CONV1D	تعانين من اضطراب	You suffer from a disorder	القلق The Anxiety	قلق Anxiety	نفسي Psychological	سلوكي Behavioural	الهلع Panic
		نصحت المريضة بالمعالجة	The patient advised for treatment	السلوكية The behavioral	وعمل And do	النفسي The psychological	النفسي The psychological	النفسية The psychological
		انصحك بالتفكير بالامور	I advise you to think about things	الايجابيه The positive	التي Which	الممتعة Interesting	وليست And not	الإحباط The frustration
Urology	LSTM	يجب عمل تحليل	You should do a lab test	بول Urine	البول The urine	للبول For urine	السائل The liquid	هرمون Hormone
		الي وجود التهاب	There is inflammation	في In	مع With	البول The urine	او Or	بعد After
		وعمل سونار على	Do a Sonar on	البطن The belly	الخصيتين The testicles	الخصيه The testis	البروستاتا The prostate	الحوض The pelvis
Internist	BiLSTM	لمعرفه سبب تكرار	To find out the cause of recurrence	الاعراض The symptoms	الالم The pain	الصداع The headache	الضغط The pressure	فقر Anemia
		عمل تحليل الغدة	Do gland analysis	الدرقية The thyroid	والمتابعة And the follow	لاستشاري For a consultant	وصورة And image	ورسم And do image
		انصح بمراجعة طبيب	I recommend seeing a doctor	الباطنيه The internist	باطنية Internist	باطني Internist	الباطنة The internist	باطنيه Internist

TABLE 7. A comparison of the models' accuracies based on the matching score of the best-obtained models for five specialties based on Keras embedding.

	Matching score				
Model	Gynecology	Dermatology	Psychology	Urology	Internist diseases
LSTM	0.429	0.260	0.255	0.389	0.297
BiLSTM	0.442	0.260	0.259	0.388	**0.303**
CONV1D	**0.449**	**0.268**	**0.268**	**0.390**	0.295
LSTM-CONV1D	0.432	0.259	0.256	0.389	0.297

TABLE 8. A comparison of the best models based on the training and validation accuracy and loss for five specialties based on Aravec embeddings.

Speciality	Model	T. Accuracy	V. Accuracy	T. Loss	V. Loss
Gynecology	LSTM	0.180	0.207	5.884	6.353
	BiLSTM	**0.195**	**0.211**	5.884	6.550
	CONV1D	0.183	0.203	**5.526**	**6.310**
	LSTM-CONV1D	0.142	0.174	6.174	6.474
Dermatology	LSTM	0.212	0.225	5.566	6.903
	BiLSTM	0.237	**0.233**	5.552	7.241
	CONV1D	**0.238**	0.228	**4.977**	6.906
	LSTM-CONV1D	0.146	0.189	6.069	**6.890**
Psychology	LSTM	0.231	0.227	5.315	7.781
	BiLSTM	0.275	**0.235**	5.261	8.338
	CONV1D	**0.287**	0.226	**4.573**	7.905
	LSTM-CONV1D	0.155	0.181	5.948	**7.777**
Urology	LSTM	0.310	0.311	4.689	6.622
	BiLSTM	**0.364**	**0.319**	4.678	7.245
	CONV1D	0.345	0.308	**4.086**	6.577
	LSTM-CONV1D	0.216	0.260	5.381	**6.519**
Internist diseases	LSTM	0.562	0.584	3.283	3.705
	BiLSTM	**0.578**	**0.593**	3.357	3.919
	CONV1D	0.575	0.588	**2.867**	**3.541**
	LSTM-CONV1D	0.515	0.561	3.597	3.726

loss values were attained by the CONV1D model by having 5.526 and 6.310, respectively. Regarding dermatology, the best training accuracy and loss gained by the CONV1D model by having 23.8% and 4.977, respectively. Also, the best validation accuracy obtained by the BiLSTM model was 23.3%, while the best validation loss was 6.890. For psychology, the CONV1D model obtained the best training accuracy of 28.7% and loss of 4.573. Also, the BiLSTM model obtained the highest validation accuracy of 23.5%, while the LSTM-CONV1D gained the lowest validation loss of 7.777. Besides, for the urology and internist diseases, the BiLSTM achieved the best training and validation accuracies by having 36.4%, 31.9%, and 57.8%, 59.3%, respectively. Furthermore, the CONV1D model performed the best in terms of the training loss for urology and internist diseases. The LSTM-CONV1D achieved the best in terms of the validation loss for the urology (6.519), while for the internist diseases, the CONV1D did the best of 3.541.

To sum up, regarding the validation accuracy, when the embedding is Aravec, the BiLSTM performed the best. Whereas, at the Keras embedding, the CONV1D achieved the best over the five specialties. Generally, the Keras embedding presented in Table 5 performed better than the Aravec embedding in terms of training accuracy and loss and validation accuracy and loss.

Table 9 shows the accuracy based on the matching score when the used embedding is Aravec. It can be seen from the table that the BiLSTM model performed the best over the five specialties. The gynecology had (40.6%), the dermatology attained 26.5%, the psychology achieved 26%, the urology obtained 38%, and the internist diseases achieved 29%.

TABLE 9. A comparison of the models' accuracies based on the matching score of the best-obtained models for five specialties based on Aravec embeddings.

	Matching score				
Model	**Gynecology**	**Dermatology**	**Psychology**	**Urology**	**Internist diseases**
LSTM	0.397	0.262	0.257	0.371	0.278
BiLSTM	**0.406**	**0.265**	**0.260**	**0.380**	**0.290**
CONV1D	0.393	0.253	0.253	0.367	0.271
LSTM-CONV1D	0.357	0.240	0.227	0.321	0.252

This is slightly was less performing than the matching score presented in Table 7, which was depending on the Keras embedding.

V. CONCLUSION

Recently, NLG has been adopted for various applications, but it is insufficiently studied in the Arabic context. An NLG-based model is proposed in this paper and applied as a one-time, one-word, predictive text model for Arabic medical recommendations. The objectives are to save doctors' time, improve service satisfaction, and improve patient-doctor interactions. Variants of deep learning models were utilized to predict the next word of text-based medical recommendations across various specialties. Hence, 3-gram and 4-gram representations of different datasets were incorporated. The implemented deep learning models in this study were the LSTM, BiLSTM, CONV1D, and LSTM-CONV1D models, where they were trained for the most common consultation types in Altibbi (i.e., gynecology, dermatology, psychiatric, urology, and internist diseases). A sensitivity analysis was conducted based on the embedding dimension, epochs, and embedding type to boost the models' performances. The best-obtained models were developed in this analysis and compared based on their training accuracy, training loss, validation accuracy, validation loss, and testing accuracy (measured as a matching score). All deep learning models achieved encouraging results, as they produced relevant suggestions for the next word approximately half the time.

This work can be extended further to not just predict the next consequent word, but also to predict a phrase or a longer sequence of words. Furthermore, enhancing the quality of the developed models can be achieved by training the models on more large-scale datasets. Besides, implementing state-of-the-art models such as attention and transformers models plays a significant role in promoting the performance of such predictive text models.

REFERENCES

[1] J. Gao, B. Peng, C. Li, J. Li, S. Shayandeh, L. Liden, and H.-Y. Shum, "Robust conversational AI with grounded text generation," 2020, *arXiv:2009.03457*. [Online]. Available: http://arxiv.org/abs/2009.03457

[2] F. A. Khan and A. Abubakar, "Machine translation in natural language processing by implementing artificial neural network modelling techniques: An analysis," *Int. J. Perceptive Cognit. Comput.*, vol. 6, no. 1, pp. 9–18, 2020.

[3] P. Jain, P. Agrawal, A. Mishra, M. Sukhwani, A. Laha, and K. Sankaranarayanan, "Story generation from sequence of independent short descriptions," 2017, *arXiv:1707.05501*. [Online]. Available: http://arxiv.org/abs/1707.05501

[4] H. Zhang, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," 2019, *arXiv:1902.09243*. [Online]. Available: http://arxiv.org/abs/1902.09243

[5] K. Loganathan, R. S. Kumar, V. Nagaraj, and T. J. John, "CNN & LSTM using Python for automatic image captioning," *Mater. Today, Proc.*, Dec. 2020, doi: 10.1016/j.matpr.2020.10.624.

[6] M. H. Shakeel, A. Karim, and I. Khan, "A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts," *Inf. Process. Manage.*, vol. 57, no. 3, May 2020, Art. no. 102204.

[7] W. Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H. Ji, and M. Jiang, "A survey of knowledge-enhanced text generation," 2020, *arXiv:2010.04389*. [Online]. Available: http://arxiv.org/abs/2010.04389

[8] R. Sharma, N. Goel, N. Aggarwal, P. Kaur, and C. Prakash, "Next word prediction in hindi using deep learning techniques," in *Proc. Int. Conf. Data Sci. Eng. (ICDSE)*, Sep. 2019, pp. 55–60.

[9] Z. Li, X. Jiang, L. Shang, and H. Li, "Paraphrase generation with deep reinforcement learning," 2017, *arXiv:1711.00279*. [Online]. Available: http://arxiv.org/abs/1711.00279

[10] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," 2017, *arXiv:1702.02390*. [Online]. Available: http://arxiv.org/abs/1702.02390

[11] D. Marcheggiani and L. Perez-Beltrachini, "Deep graph convolutional encoders for structured data to text generation," 2018, *arXiv:1810.09995*. [Online]. Available: http://arxiv.org/abs/1810.09995

[12] Y. Li, Q. Pan, S. Wang, T. Yang, and E. Cambria, "A generative model for category text generation," *Inf. Sci.*, vol. 450, pp. 301–315, Jun. 2018.

[13] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," 2018, *arXiv:1805.04833*. [Online]. Available: http://arxiv.org/abs/1805.04833

[14] J.-S. Lee and J. Hsiang, "Patent claim generation by fine-tuning OpenAI GPT-2," 2019, *arXiv:1907.02052*. [Online]. Available: http://arxiv.org/abs/1907.02052

[15] L. F. R. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, "Investigating pretrained language models for graph-to-text generation," 2020, *arXiv:2007.08426*. [Online]. Available: http://arxiv.org/abs/2007.08426

[16] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," *Neural Comput. Appl.*, pp. 1–19, Oct. 2020.

[17] I. P. Yamshchikov and A. Tikhonov, "Music generation with variational recurrent autoencoder supported by history," *Social Netw. Appl. Sci.*, vol. 2, no. 12, pp. 1–7, Dec. 2020.

[18] M. Al-Maleh and S. Desouki, "Arabic text summarization using deep learning approach," *J. Big Data*, vol. 7, no. 1, pp. 1–17, Dec. 2020.

[19] K. Luu, R. Koncel-Kedziorski, K. Lo, I. Cachola, and N. A. Smith, "Citation text generation," 2020, *arXiv:2002.00317*. [Online]. Available: http://arxiv.org/abs/2002.00317

[20] W. Antoun, F. Baly, and H. Hajj, "AraGPT2: Pre-trained transformer for arabic language generation," 2020, *arXiv:2012.15520*. [Online]. Available: http://arxiv.org/abs/2012.15520

[21] B. Jing, P. Xie, and E. Xing, "On the automatic generation of medical imaging reports," 2017, *arXiv:1711.08195*. [Online]. Available: http://arxiv.org/abs/1711.08195

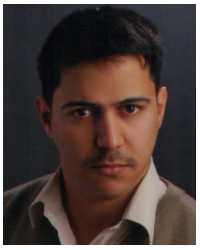
[22] X. Huang, F. Yan, W. Xu, and M. Li, "Multi-attention and incorporating background information model for chest X-ray image report generation," *IEEE Access*, vol. 7, pp. 154808–154817, 2019.

[23] A. Yazdani, R. Safdari, A. Golkar, and S. R. N. Kalhori, "Words prediction based on N-gram model for free-text entry in electronic health records," *Health Inf. Sci. Syst.*, vol. 7, no. 1, p. 6, Dec. 2019.
[24] S. Ginn, "Smart vet: Autocompleting sentences in veterinary medical records," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2019.
[25] H. Van, D. Kauchak, and G. Leroy, "AutoMeTS: The autocomplete for medical text simplification," 2020, *arXiv:2010.10573*. [Online]. Available: http://arxiv.org/abs/2010.10573
[26] D. Gopinath, M. Agrawal, L. Murray, S. Horng, D. Karger, and D. Sontag, "Fast, structured clinical documentation via contextual autocomplete," in *Proc. Mach. Learn. Healthcare Conf.*, 2020, pp. 842–870.
[27] A. Hoogi, A. Mishra, F. Gimenez, J. Dong, and D. Rubin, "Natural language generation model for mammography reports simulation," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2711–2717, Sep. 2020.
[28] M. Li, F. Wang, X. Chang, and X. Liang, "Auxiliary signal-guided knowledge encoder-decoder for medical report generation," 2020, *arXiv:2006.03744*. [Online]. Available: http://arxiv.org/abs/2006.03744
[29] Y. Zhang, X. Wang, Z. Xu, Q. Yu, A. Yuille, and D. Xu, "When radiology report generation meets knowledge graph," 2020, *arXiv:2002.08277*. [Online]. Available: http://arxiv.org/abs/2002.08277
[30] W. Yang, G. Zeng, B. Tan, Z. Ju, S. Chakravorty, X. He, S. Chen, X. Yang, Q. Wu, Z. Yu, E. Xing, and P. Xie, "On the generation of medical dialogues for COVID-19," 2020, *arXiv:2005.05442*. [Online]. Available: http://arxiv.org/abs/2005.05442
[31] Z. Han, B. Wei, X. Xi, B. Chen, Y. Yin, and S. Li, "Unifying neural learning and symbolic reasoning for spinal medical report generation," *Med. Image Anal.*, vol. 67, Jan. 2021, Art. no. 101872.
[32] K. Sakka, K. Nakayama, N. Kimura, T. Inoue, Y. Iwasawa, R. Yamaguchi, Y. Kawazoe, K. Ohe, and Y. Matsuo, "Character-level japanese text generation with attention mechanism for chest radiography diagnosis," 2020, *arXiv:2004.13846*. [Online]. Available: http://arxiv.org/abs/2004.13846
[33] S. Mishra and M. Banerjee, "Automatic caption generation of retinal diseases with self-trained RNN merge model," in *Proc. Adv. Comput. Syst. Secur.* Singapore: Springer, 2020, pp. 1–10.
[34] J. Ive, N. Viani, J. Kam, L. Yin, S. Verma, S. Puntis, R. N. Cardinal, A. Roberts, R. Stewart, and S. Velupillai, "Generation and evaluation of artificial mental health records for natural language processing," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–9, Dec. 2020.
[35] V. Kieuvongngam, B. Tan, and Y. Niu, "Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2," 2020, *arXiv:2006.01997*. [Online]. Available: http://arxiv.org/abs/2006.01997
[36] M. Moradi, G. Dorffner, and M. Samwald, "Deep contextualized embeddings for quantifying the informative content in biomedical text summarization," *Comput. Methods Programs Biomed.*, vol. 184, Feb. 2020, Art. no. 105117.
[37] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[38] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of arabic word embedding models for use in arabic NLP," *Procedia Comput. Sci.*, vol. 117, pp. 256–265, Jan. 2017.
[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[40] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[42] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," San Francisco Bay Area, Silicon Valley, CA, USA, Tech. Rep., 2015. [Online]. Available: https://www.tensorflow.org

MARIA HABIB received the bachelor's degree in computer engineering from the Faculty of Engineering and Technology, The University of Jordan, and the master's degree in web intelligence from the Department of Information Technology, King Abdullah II School of Information Technology. She was a Research Assistant with The University of Jordan and a Former Graduate Research Trainee in bioinformatics and big data analysis with the Bioinformatics Laboratory, Department of Parasitology, McGill University, supervised by Jianguo (Jeff) Xia. She is currently a Data Science Engineer and a Researcher with Altibbi, Amman, Jordan. She is also a member of the (Evo-ML.com) Research Group.

MOHAMMAD FARIS received the degree (Hons.) in computer information systems from Al Albayt University, Jordan. He is currently a Data Scientist with Altibbi Telemedicine Company. His main technical skills include Python, TensorFlow, Flask, and PHP.

RANEEM QADDOURA received the Ph.D. degree in computer science in the fields of machine learning and data mining. She has 14 years of experience in total in both academic and industrial experience. She is currently an Assistant Professor with Philadelphia University. She is also an Active Research Member of the Evolutionary and Machine learning Group, which focuses on evolutionary algorithms, machine learning, and their applications for solving important problems in different areas. Her current research interests include evolutionary computation, and data clustering and classification.



ALAA ALOMARI received the B.Sc. degree in computer science from Yarmouk University, Jordan, and the M.Sc. degree in computer science from the Jordan University of Science and Technology, Jordan. He is currently the Chief Information Officer (CIO) and the Product Director of Altibbi (Telemedicine platform for MENA region). He is leading the development, planning, and administration of an innovative, robust, and secure information technology environment throughout the platform and systems of Altibbi. He is also a Technical Director of Unifonic Cloud Communication. He spent 15 years in information technology roles in managing and administering web-based projects and servers in different areas, such as cloud communication, NGO, online games, media, and online recruitment. His primary responsibilities as the CIO are to setup a technical strategic plan that covers and governs policies, resource allocation, information technology protocols, and security compliances. He received technical and management certificates, such as ZCE, CMDBA, and AWS Solution Architect.

HOSSAM FARIS received the B.A. degree in computer science from Yarmouk University, Jordan, in 2004, the M.Sc. degree in computer science from Al-Balqa' Applied University, Jordan, in 2008, and the Ph.D. degree in e-business from the University of Salento, Italy, in 2011. In 2016, he worked as a Postdoctoral Researcher with the Information and Communication Technologies Research Center (CITIC), GeNeura Team, University of Granada, Spain. He co-founded the Evolutionary and Machine Learning (Evo-ML.com) Research Group. He is currently a Professor with the School of Computing and Informatics, Al Hussein Technical University, and the Department of Information Technology, King Abdullah II School for Information Technology, The University of Jordan, Jordan. His research interests include applied computational intelligence, evolutionary computation, knowledge systems, data mining, semantic web, and ontologies. He was awarded a full-time Competition-Based Scholarship from the Italian Ministry of Education and Research to pursue the Ph.D. degree.