# Medical Speciality Classification System based on Binary Particle Swarms and Ensemble of One *vs.* Rest Support Vector Machines

Hossam Faris[a], Maria Habib[a,b], Mohammad Faris[b], Manal Alomari[b], Alaa Alomari[b]

[a]*King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*
*hossam.faris@ju.edu.jo,mar8160671@fgs.ju.edu.jo*
[b]*Altibbi (https://altibbi.com), Amman, Jordan*
*{maria.habib,mohammad.faris,manal,alaa.alomari}@altibbi.com*

**Abstract**

Nowadays, artificial intelligence plays an integral role in medical and healthcare informatics. Developing an automatic question classification and answering system is essential for coping with constant advancements in science and technology. However, efficient online medical services are required to promote offline medical services. This article proposes a system that automatically classifies medical questions of patients into medical specialties and supports the Arabic language in the MENA region. Text classification is not trivial, especially when dealing with a highly morphologically complex language, the dialectical form of which is the dominant form on the Internet. This work utilizes 15,000 medical questions asked by the clients of Altibbi telemedicine company. The questions are classified into 15 medical specialties. As the number of medical questions received daily by the company has increased, a need has arisen for an automatic classification system that can save the medical personnel much time and effort. Therefore, this article presents an efficient medical speciality classification system based on swarm intelligence (SI) and an ensemble of support vector machines (SVMs). Particle swarm optimization (PSO) is an SI-based and stochastic metaheuristic algorithm that is adopted to search for the optimal number of features and tune the hyperparameters of the SVM classifiers, which are deployed as one-versus-rest for multi-class classification. In addition, PSO is integrated with various binarization techniques to boost its performance. The experimental

results show that the proposed approach accomplished remarkable performance as it achieved an accuracy of 85% and a features reduction rate of 95.9%.

*Keywords:* Medical Text Classification, Swarm Intelligence, Support Vector Machines, Altibbi, Arabic Language Processing, One-Versus-Rest

## 1. Introduction

Artificial Intelligence (AI) is an integral part of modern healthcare informatics that saves efforts, time, and cost while enhancing the public's health services. Data mining is a fundamental aspect of AI, which concerned with extracting useful patterns of information from structured or unstructured data located in extensive data repositories and warehouses. However, the immense growth of web content makes it difficult to mine relevant information. Text mining and information retrieval are emerging technologies that reinforce the functionality of the online medical services. However, many information retrieval systems recapture a large number of documents that cannot realistically be comprehended and searched for in real time.

Question Answering (QA) systems are advanced techniques used by information retrieval systems. QA systems date back to the 1960s and provide accurate answers to intricate online domain-specific questions rather than merely extracting knowledge. Medical QA systems have attracted the attention of the medical community and have been included in various initiatives, such as the BioASQ challenge, to improve the performance of medical QA systems. A QA system consists of three modules: question analysis, information retrieval, and answer extraction. Question analysis involves question classification, keyword extraction, and expansion. Question classification is a fundamental preprocessing step that significantly influences the performance of QA systems. The objective of question classification is to assign a category to a question, which determines its answer type. Question classification plays a vital role in determining the most suitable answer extraction strategy. It also reduces the potential search space. A popular classification approach proposed by Li and Roth [1], categorizes questions into six types that are related to different answer types: abbreviation, entity, description, human, location, and numeric value.

The Development of a reliable QA and classification framework depends on the ability of the system to understand the questions. An understanding

of questions is demonstrated by the extraction of efficient features that are embedded in the textual questions. Textual data is processed by natural language processing techniques (NLPs). NLP methods employ machine learning and data mining tools to extract features hidden in raw data and transform them into useful knowledge. Various types of statistical or lexical features (e.g. syntactic, semantics, or word shape features) can be inferred from textual data. One of the most prominent advances in feature extraction is the word embedding. Word embedding is a process characterized by the numerical vector-like representation of words, in which similar words have similar vectors representations. Natural language processing in medical question classification is an active significant research area. This article sheds light on medical question classification in an Arabic context, which categorizes different types of questions into different medical classes of specialties.

The Arabic language is the fourth-most used language online and is the official spoken language of two billion people [2]. Generally, it can be used in two primary forms: Modern Standard Arabic or the dialectical Arabic, with dialectical Arabic being more commonly used on social networking and blogging sites. The Arabic language is a morphologically rich and sophisticated language that creates many challenges when using NLP techniques. The use of colloquial Arabic is especially challenging since people from different Arabic countries have different dialects, and these dialects can vary from city to city within a country. Moreover, dialectical Arabic involves many misspellings that differ morphologically and phonologically, which reduces the effectiveness of NLP processes. In addition, the Arabic language has a more abundant complex orthography and more morphosyntactic rules than other languages, and these are accompanied by a lack of Arabic lexical resources and tools [3]. Very few studies have been considering Arabic text classification in the medical natural language research. To the best of our knowledge, no previous works have studied the medical speciality classification of questions in Arabic. Therefore, research on Arabic medical QA analysis and classification requires more attention.

Several research studies have addressed biomedical question processing. Notably, question classification is performed using either rule-based techniques, machine learning methods, or a hybrid of both. Rule-based techniques largely depend on the extraction of rules via manual observations, which takes a long time, especially when large datasets are considered. On the other hand, machine learning methodologies use NLP techniques for text analysis and feature extraction. When this type is used, the reliability of

the final model is significantly influenced by the robustness of the feature extraction and analytical tools employed. These techniques are limited in their ability to understand the semantics of questions in the Arabic context.

The primary objective of this article is to promote the text classification process in the medical field to improve the efficiency and robustness of medical QA systems. Herein, Altibbi Company [1] is studied as a real-world case study. Altibbi provides telemedicine services such as answering customers' health and medical questions. A crucial step for answering questions is assigning the questions to the correct class of speciality and, in turn, to a relevant doctor. The most notable reason for automating the text classification process is the fact that Altibbi receives over 4,000 health questions per day, which makes the manual classification of questions cumbersome and a waste of time and resources. Furthermore, due to the nature of the speciality itself or ambiguity in the language in which the question is asked, classifying the type of question into one of a large number of specialties is not a trivial task.

Based on the above discussion, this article proposes an efficient speciality classification system to improve the classification of medical questions received from patients in the form of text messages. However, several challenges have emerged from the data collected from Altibbi, including issues with the processing of the Arabic language and the multi-class classification problem. The proposed approach addresses these problems through various intelligent components. In this approach, a binary version of PSO is used to search for the optimal values of different hyperparameters of the learning algorithms, while the one-versus-rest mechanism is utilized to handle multi-class classification process. The PSO algorithm is a metaheuristic optimization algorithm inspired by the social behavior of bird flocks. PSO is adopted in the present work to search for the optimal number of features and the minimum document frequency required to create the feature representation. Furthermore, PSO is used to tune the cost ($C$) parameter to train the SVM classifiers in the one-versus-rest ensemble. In addition, the PSO is integrated with different binarization mechanisms in an attempt to boost its search performance. This approach will be referred to as ($BPSO_{TF} - SVM_{OVR}$) throughout this work, and it will be compared with different well-known machine learning classification algorithms. The results show that ($BPSO_{TF} - SVM_{OVR}$) out-

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

performs other well-known techniques.

The rest of the paper is organized as follows. Section 2 describes related works on the topic of text classification and medical QA systems. Section 3 provides a background of the SVM algorithm and the PSO optimizer. Section 4 presents the problem description related to Altibbi's telemedicine service. Section 5 describes the dataset, the preprocessing stage, and feature extraction capabilities. Section 6 presents the proposed approach including particles' encoding, fitness evaluation, and binarization techniques, as well as the overall procedure of the designed approach. Section 7 explains the utilized evaluation measures. Section 8 presents the results, encompassing the experimental settings, the influence of fitness weighting parameters, the effect of variants of the transfer function, and a comparison between the proposed approach and popular machine learning classifiers. Finally, Section 9 provides general conclusions and offers recommendations for future works.

## 2. Related works

Previous studies on text classification have focused on implementing rule-based approaches to extract distinguishing patterns of features to develop classification rules. A prominent rule-based method involves targeting keywords of "wh questions" (e.g. why, where) and then finding words that are associated with these wh-words and determining whether they are nouns or verbs. However, rule-based methods are inefficient since they scale poorly with the data size and their computations are costly. On the other hand, machine learning techniques have been utilized widely in the era of text classification, particularly, in question classification. This section, in general, is devoted to describing related works in the area of text classification using machine learning algorithms that classify Arabic text where it is feasible. It also discusses previous research studies dedicated to medical or healthcare question classification and answering systems.

Machine learning methods are a remarkable type of AI that have been adopted for various applications including those in the industrial, financial, and medical sectors [4, 5, 6]. SVM algorithm is a popular machine learning algorithm for text classification. For instance, Zhang and Lee [7], developed an automatic question classification framework using a "TREC QA" data set by extracting the bag-of-words (BoW) and bag-of-ngrams features. In which, the SVMs algorithm showed efficient performance over naïve bayes (NB), decision tree (DT), and Sparse Network of Winnows algorithms. Meanwhile,

[8] performed an analytical study on the usage of statistical methods for fact-based question classification. Metzler and Croft found that the statistical methods performed better when they included semantics or syntactical features. Additionally, Li et al. [9] proposed a combination of rule-based and statistical methods for question classification that integrates linguistic features from WordNet. Further, [10] reported a state-of-the-art question classification method that considers headword features and WordNet hypernyms, along with maximum entropy (ME) classifier. Also, [11] proposed a novel question classification approach based on a semi-supervised and co-training framework. In this approach, the most frequent words were used as classification features and their semantic similarity were considered as feature weighting criteria. Their method led to a (2-4)% improvement in the results. In [12], the authors argued that analyzing the question structure could improve the performance of question classification systems. They justified this by using a new kernel function that relies on the syntactic dependency relationship and speech tagging. Cao et al. [13] developed an online clinical QA system (AskHERMES) that relies on linguistics features and machine learning algorithms. Moreover, Le-Hong et al. [14] created a constituency and dependency structure analysis of questions that increased the model's accuracy by over 8%.

Mohasseb et al. [15] proposed a grammar-based machine learning method for question classification. In their method, a hierarchical synthetic minority oversampling technique was integrated to resolve the class imbalance problem. In other work, Sarrouti and El Alaoui [16] implemented machine learning algorithms that classify biomedical questions into four types (yes-no, factoid, list, and summary) by relying on lexico-syntactical features. Their method achieved a maximum of 89% accuracy using SVMs. Similarly, Mohasseb et al. [17] developed a grammar-based approach for question categorization (GQCC). In this approach, questions are represented as grammatical patterns that are fed into machine learning methods. which performed well. There is little research on Arabic question classification and answering compared to those on English language. Such efforts include the work of Al-Bayan [18] for the holy book (Al-Quran) QA, and Arabic question classification using multinomial NB [19]. Remarkably, Hamza et al. [20] presented a new framework for Arabic question classification that reshapes the questions using a distributed representation of words with new words taxonomy, and their proposed approach achieved 90% accuracy using the SVM algorithm. López Seguí et al. [21] investigated the use of machine learning for primary

care teleconsultation using five algorithms and four text representation methods. The proposed strategy had very good performance results even when the amount of deployed data was relatively small. In addition, Wasim et al. [22] performed a multi-label question classification for answer type prediction in the biomedical field. Their approach depended on a label power set transformation method with logistic regression (LR) algorithm. Meanwhile, Sarrouti and El Alaoui [23] proposed a semantic biomedical QA system by adopting lexico-syntactic features and machine learning techniques, which achieved efficient performance and scaling abilities.

Our conducted literature review shows that there is a lack of research on the Arabic medical speciality classification of questions. Hence, this research work attempts to provide a medical speciality classification method by taking Altibbi telemedicine company as a case study.

## 3. Methods

### *3.1. Support vector machines*

The SVM algorithm is a popular, statistical, and supervised machine learning algorithm that is also known as the large margin classifier. It has been recognized widely among the research community for its ability to tackle diverse classification and regression problems in various fields, including image and text classification, image segmentation, and pattern recognition [24, 25].

Primarily, the SVM algorithm was developed by Boser et al. in 1992 [26], to integrate the support vectors and hyperplanes to formulate the decision boundaries for distinguishing classes. The purpose of decision boundaries is to maximize the marginal distance between a hyperplane and the respective data examples in case of a binary classification problem. They can also minimize the $\varepsilon$-deviations in cases of predictive learning (see Fig. 1). Originally, the SVM algorithm was proposed to address binary classification problems. However, later, it was adapted to address multi-class classification problems using different criteria, such as by utilizing each support vector machine to differentiate one class from all other classes by means of the (one-versus-rest) approach.

Generally, the objective of the SVM is to diminish the training error and promote the generalization ability to new unseen data based on the principle of structural risk minimization. It is intended to find a hypothesis ($I$) from

the hypothesis space ($H$) that ensures the minimal probability of error for a given training example in relation to a regularization function.

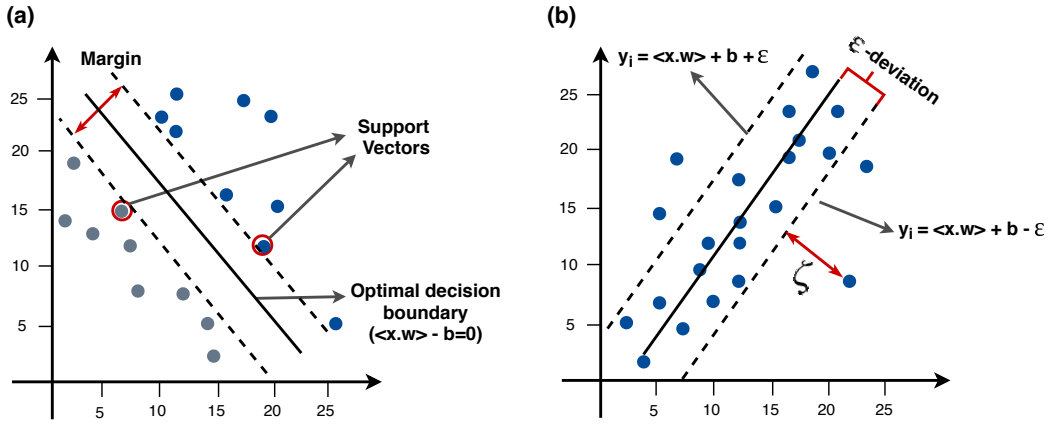



Figure 1: A description of the SVMs algorithm for classification and regression. The blue circles represent the positive class, and the grey circles represent the negative class. In (a), SVM utilizes the support vectors near the class boundary to maximize the margins, while in (b), it reduces the error deviations between the decision boundary and the actual target.

A linearly separable dataset can be differentiated by using a linear hyperplane in the form of ($f(x) = w \cdot x + b = 0$), where $w$ is a weight vector, and $b$ denotes a threshold between the hyperplane and the origin plane. Suppose a training dataset $X = (x^{(1)}, y^{(1)})...(x^{(m)}, y^{(m)})$ that has $m$ training examples and $n$ features. In this case, the objective of the SVM is to minimize the objective function characterized by Eq. 1, where $C$ is the regularization parameter, $A$ is the objective function presented by Eqs. (2-4), and $B$ is the regularization function (Eq. 5).

$$Min \quad C \cdot A + B \tag{1}$$

$$A = \sum_{i=1}^{m} \left[ y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] \tag{2}$$

In the above equation, $cost_1$ represents the cost regarding the positive class, and $cost_0$ corresponds to the negative class, as defined by Eq.3.

$$\begin{aligned} cost_1 &= -log(h_\theta(x^{(i)})) \\ cost_0 &= -log(1 - h_\theta(x^{(i)})) \end{aligned} \tag{3}$$

Where $h_\theta(x)$ is the hypothesis of the training data and given by Eq. 4.

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^T \geq 0 \\ 0 & otherwise \end{cases} \tag{4}$$

$$B = \frac{1}{2}\sum_{j=1}^{n} \theta_j^2 \tag{5}$$

A key aspect of the SVMs algorithm is the adoption of kernel functions to handle the non-linearly separable data, which transforms the data into a higher-dimensional space to grasp the non-linear relations within the data variables by linear-hyperplanes found in the higher spaces. The SVM utilizes various types of kernels, such as the linear kernel, the polynomial kernel, and the radial basis function (RBF). In other words, for example, the RBF kernel can be used to adapt non-linear variables and find non-linear decision boundaries. The RBF kernel is defined as $K(x^{(i)}, x^{(j)}) = exp(-\gamma||x^{(i)} - x^{(j)}||^2)$, where $\gamma$ is the gamma coefficient.

Even though the SVM algorithm is efficient, its performance is highly sensitive to the setting of the cost ($C$) regularization parameter and the ($\gamma$) coefficient in the case of non-linear SVMs. Hence, inappropriate values of $C$ and $\gamma$ can lead to poor generalization due to the overfitting or underfitting the data. To illustrate, setting a large value of the $C$ parameter results in a lower bias and higher variance, which is prone to overfitting. In contrast, a smaller value of $C$; increases bias and reduces variance, which causes the algorithm to underfit the data. Therefore, optimizing the values of $C$ and $\gamma$ to boost the performance of the SVM is an essential step to training the algorithm.

### *3.2. Particle Swarm Optimization*

The PSO algorithm is a kind of swarm intelligence optimizer, and a nature-inspired metaheuristic that was designed by Eberhart and Kennedy in 1995. The PSO algorithm is considered a global stochastic search algorithm that aims to find the reasonable solutions to an objective function. The search mechanism of the PSO algorithm is inspired by the search strategy of bird swarms while foraging. The PSO algorithm includes a collection of randomly generated solutions that are known by particles. Each particle is characterized by velocity and position; hence, the particles search for the optimal solutions by continuously updating their positions of flight, and their

velocities by following a leader particle. Primarily, the velocity and position components of particles are affected by the particle's (local) experience, as well as the particles' (global) experience. Subsequently, during the search process; each particle depends on its own experience, which is denoted by the "cognitive component" and is represented by ($pbest$). Moreover, the experiences of other particles are known as the "social component", and are represented by ($gbest$).

Exploring the search space corresponds to an iterative movement of particles, the velocities and positions of which are modeled mathematically by Eq. 6, and Eq. 7, respectively. Here, ($d \in D$) represents the $d-th$ dimension in the search space, ($w$) is the inertia weight coefficient that is responsible for balancing exploration and exploitation. ($r_1$) and ($r_2$) are two random numbers in the range of [0,1], ($c_1$) and ($c_2$) are the acceleration constants that control the randomness effect of the social and cognitive components. Furthermore, ($v_{id}(t)$) and ($x_{id}(t)$) are the current velocity and position of particle ($i$) at time ($t$) and the ($d-th$) dimension, respectively, while ($p_{id}$) presents the personal best position of particle (i), and ($g_d$) represents the global best particle among the population.

$$v_{id}(t+1) = w * v_{id}(t) + r_1 * c_1 * (p_{id}(t) - x_{id}(t)) + r_2 * c_2 * (g_d(t) - x_{id}(t)) \quad (6)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (7)$$

Over the years, the PSO algorithm has been used successfully in a wide-range of applications in various fields due to its simple implementation, the fact that there are few parameters to optimize, and its relatively fast convergence [28, 29, 30]. The typical classic PSO algorithm is illustrated in Algorithm 1.

Initially, the PSO algorithm was developed to cope with continuous optimization problems. However, later on, Kennedy and Eberhart [31] expanded the classic PSO algorithm so it could handle discrete (binary) problems. The binary PSO represents the positions of particles as vectors of binary bits 0,1, whereas, the velocities of particles act as the probabilities of a bit being a value of (1). In [31], Kennedy and Eberhart proposed the binary PSO by utilizing a type of transfer function known as the Sigmoid function. Transfer functions are mathematical models used to transform the continuous search space into a binary search space. Various transformation functions have been suggested. These include the V-shaped, and S-shaped transfer functions proposed by Mirjalili and Lewis [32].

**Algorithm 1** The classic PSO algorithm

```
Initialize a population of particles of size N
for each particle do
    Initialize the particle
end for
while (t< Maximum iterations) do
    for each particle do
        Evaluate the particle's fitness (J)
        if (J is better than the pBest) then
            set J as the new pBest
        end if
        Set the particle with the pBest value as gBest
    end for
    for each particle do
        Calculate particle's velocity as in equation 6
        Update particle's position as in equation 7
    end for
    t=t+1
end while
```

The Sigmoid function is defined as described in Eq. 8, where $T(v_{id}(t))$ is the probability of having a (1) bit value. Hence, the position of a particle in a binary PSO is formulated per Eq. 9 instead of Eq. 7, where $rand$ is a randomly generated number.

$$T(v_{id}(t)) = \frac{1}{1 + e^{-v_{id}(t)}} \tag{8}$$

$$x_{id}(t+1) = \begin{cases} 1 & \text{if } T(v_{id}(t)) \geq rand \\ 0 & \text{if } T(v_{id}(t)) < rand \end{cases} \tag{9}$$

## 4. Problem description

As primary care is at the core of Altibbi telemedicine service, it is Altibbi's doctors' responsibility to direct patients to the correct specialized doctor. This is because too many patients ask the question "Which doctor shall I visit?". Meanwhile, it must be ensured that patients are receiving a typical medical consultation, starting from the primary care stage until the process is completed when the patient is directed to a specialized doctor if needed.

This was easily detected by the general practitioner directly after the medical consultation with the patient. However, Altibbi provides asynchronous answers for medical questions that are answered by a specialized doctor within 24 hours. Altibbi receives thousands of questions every day that need to be answered asynchronously while avoiding spamming doctors' inboxes with questions that are not related to their speciality. Altibbi has carried out this routing process manually by having medical officers review questions and decide the corresponding speciality for each question, which was then routed to the correct doctor. However, this was a very time-consuming and cost-inefficient process. Furthermore, the manual process cannot be accomplished in real time and was not (100%) accurate. Many questions were routed incorrectly due to the high intersection among keywords and because some specialties overlap. Table 1 gives some examples on questions received by Altibbi Company in Arabic and their corresponding speciality type. In addition, the table exhibits two close questions in their type, which are related to the "Nutrition" and "Digestive System Diseases". This challenge raises the need to automate (speciality detection) by applying an intelligent computational detection module based on machine learning algorithms.

Table 1: Examples of different questions translated into English and their associated speciality.

| Question | Translated question | Speciality |
|---|---|---|
| متي يبدا الطفل بقول كلمات واضحة؟ | When does the child start by saying clear words? | Child health |
| ماهو العلاج الامثل للحساسية التلامسية الفقاعية التي ظهرت على يدي بعد استخدام كريمات الحساسية؟ | What is the best treatment for bullous contact allergies that appeared on my hands after using allergy creams? | Dermatology |
| الكربوهيدرات الموجودة في التمر معقدة ام بسيطة وكم حبة تمر استطيع ان اكل في النهار كرياضي؟ | Existing carbohydrates in dates are complex or simple, and how many units do I eat in the day as an athlete? | Nutrition |
| التقيء دائماعند اكل الفاكهه والخضار الغير مطبوخه، ما اسبابه وحلوله؟ | Always vomiting when eating uncooked fruits and vegetables, what are its causes and solutions? | Digestive System Diseases |
| ماهو علاج توسع الحدقه الدائم للعين اليسري مع ان الفحص عبر الرنين المغناطيسي وكل الاورده والشرايين سليمه؟ | What is the treatment for permanent pupil dilatation of the left eye, even though the examination through MRI and all the veins and arteries are intact? | Ophthalmology & Eye Diseases |

The artificial intelligent module is a profound automated phase, both in synchronous and asynchronous medical consultations. It provides a real-time and reliable step for asynchronous questions, thereby eventually minimize the doctor's efforts in addressing the correct speciality. Fig. 2 is a graphical illustration of the problem of medical speciality detection and question classification. An intelligent classification system involves processing, classifying, and answering questions by specialists.

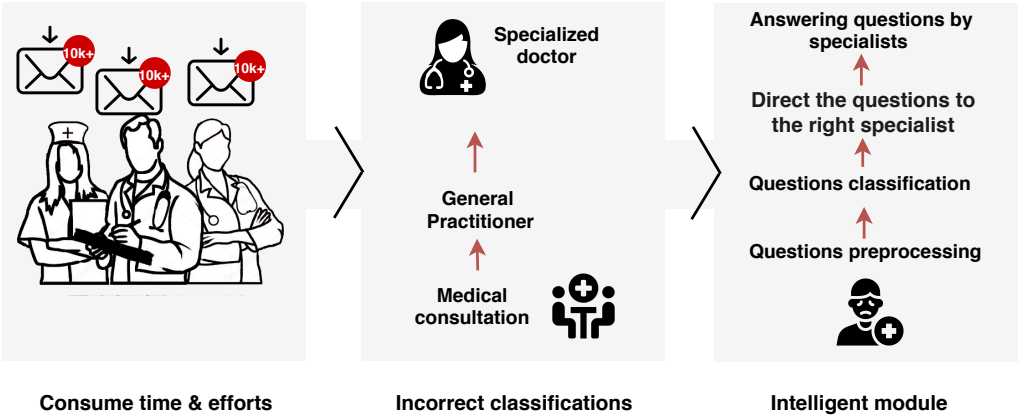



Figure 2: A description of the manual question classification and speciality detection. Also, the potential intelligent classification system.

## 5. Dataset collection and preparation

In this work, all data were obtained from Altibbi company. Altibbi is a digital health cloud-based platform that focuses on telemedicine services for primary care purposes, health management systems (HMS) and Arabic medical content, targeting the MENA region. The company has provided more than 1.2 million structured medical consultations, more than 3 million accredited and verified pieces of medical content, and about 1 million electronic medical records (EMR). Its telemedicine service is provided through multiple channels (video, live chat, GSM calls, and asynchronous responses).

For this work, Altibbi provided 15,000 questions, which were written in Arabic. Each question is classified by a team of experts from the company into one of 15 medical specialties. The questions were evenly distributed over the classes (i.e. 1000 questions per class).

### *5.1. Data preprocessing*

The data preprocessing stage includes several stages and prepares the data in a way that fits with the learning algorithms. Data preprocessing includes data cleaning and normalization, stemming, and tokenization. Cleaning primarily includes removing non-Arabic characters, numbers, symbols, diacritics, web addresses, and punctuation, as well as the Arabic stop words and negation words. The normalization or denoising of characters involves the alteration of various forms of Arabic characters into a common colloquial form. During tokenization, the questions are split into sets of words (tokens) based on the presence of a white space. The obtained tokens are then stemmed by removing the morphological affixes of terms. The IRSI stemmer is utilized from the Natural Language Toolkit (NLTK) library [33]. Consequently, the cleaned data is utilized for further analysis in the following experiments.

### *5.2. Features extraction*

The main objective of natural language processing techniques is to find useful information related to the raw data by adopting intelligent machine learning methods. However, a key aspect of training machine learning algorithms is preparing the statistical and numerical features that reveal the hidden patterns within textual data. Several approaches have emerged in the literature for reshaping textual data into numerical vector-like representations (e.g., the bag-of-words method, term frequency, TF-IDF) [34].

TF-IDF is a textual vectorization technique that is used in various fields such as document and text classifications [35, 36]. It is an evolved version of term frequency (TF) vectorization strategy that assigns rare words in a document a high weight value. The advantage of TF-IDF over the TF approach is its ability to mitigate the influence of frequent words that are not especially important or informative and can mislead the learning process. The TF method represents the ratio of the occurrences ($f_k$) of each word ($k$) over the number of unique keywords in the document, as in Eq. 10.

$$TF = \frac{n_k}{n} \tag{10}$$

Differently, the IDF presents a measure to quantify the frequency rate of a word across all documents. Hence, higher IDF values indicate more frequently used words. Eq. 11 provides the formula of IDF, where ($N$) is the

number of documents, and ($df_k$) is the number of documents that include the keyword ($k$).

$$IDF = log_2(\frac{N}{df_k}) \tag{11}$$

However, keywords with a high IDF value have a low weighting score in TF-IDF. TF-IDF is given based on the cross-product of $TF$ and $IDF$. Eq. 12 shows the formula to calculate the weight of the term ($t_k$) based on TF-IDF [34].

$$Weight\ (t_k) = \frac{f_k}{n}\ X\ log_2(\frac{N}{df_k}) \tag{12}$$

### *5.3. Exploratory data analysis*

In machine learning, the preprocessing step of exploring the data is known as exploratory data analysis (EDA). The objective of EDA is to explore data visually and capture the characteristics that aid its analysis. It is very challenging to interpret high-dimensional datasets to find relationships among features. In such cases, natural language textual data must be dealt with, whereby the words are the features. As such, dimensionality reduction techniques are concerned with diminishing the feature space by preserving principal features to visualize and examine the data. In the literature, different mechanisms such as the t-Distributed Stochastic Neighbor Embedding (tSNE) have been introduced to reduce the number of features [37]. The tSNE is a non-linear feature mapping system that depends on the conditional probability of similarities between points in the higher space and the conditional probability of similarities for points in the lower spaces. Its objective is to reduce the difference in the lower and higher conditional probabilities using Kullback-Leibler divergence [37]. The tSNE is primarily used to visualize vast datasets.

Fig. 3 and 4 show a graphical projection of the features of the medical questions obtained before and after preprocessing and elimination of stopwords, respectively. Fig. 3 presents the capability of the tSNE algorithm to group similar points together. Some classes are clearly observable, while others overlap. The ("Nutrition") and ("Tumors") classes are apparent, while ("Psychiatric Diseases"), ("Gynecology & Women Diseases"), and ("Sexual Health") are highly interfering. Fig. 4 describes the projection of 15,000 samples after preprocessing and the removal of the stopwords. It is evident here that more classes has formed a clearer shape. For example, the ("Dental

Medicine"), ("Eye Diseases"), and ("Child Health") became more pronounced, while ("Gynecology & Women Diseases"), and ("Sexual Health") became more compact. However, ("Dermatology") and ("Ear, Nose, & Throat") still experienced extensive overlapping. It is clear from the second figure (Fig. 4) that the removal of stopwords or the meaningless words results in more clear compact clusters of classes, whereas, the existence of redundant words makes the classes less predictable.

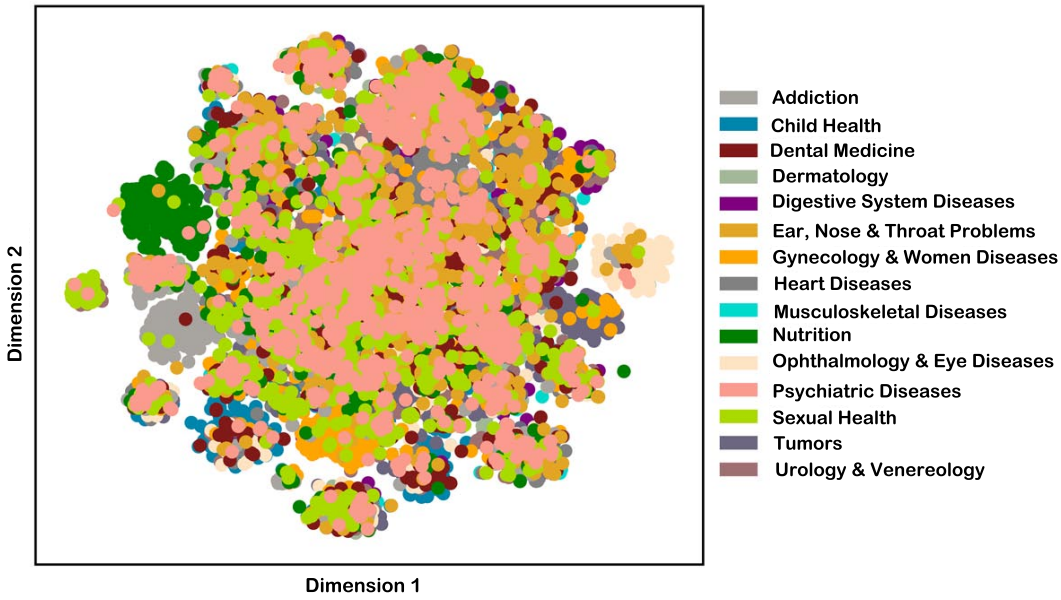



Figure 3: The tSNE projection of 15,000 questions before the preprocessing and the removal of stopwords. In addition, it incorporates the total number of features (words) that are produced from the TF-IDF vectorizer.

On the other hand, after carrying out the preprocessing steps, the question length distributions for the 15 classes were depicted as boxplots (Fig. 5). The lengths of questions are represented by the number of tokens in the respective tokenized questions. It can be seen that most of the classes have very similar distributions with their length medians ranging from 50 to 75 tokens.

## 6. Proposed classification system ($BPSO_{TF} - SVM_{OVR}$)

Various algorithms have been developed to deal with binary classification problems, such as logistic regression and SVM. However, binary classification

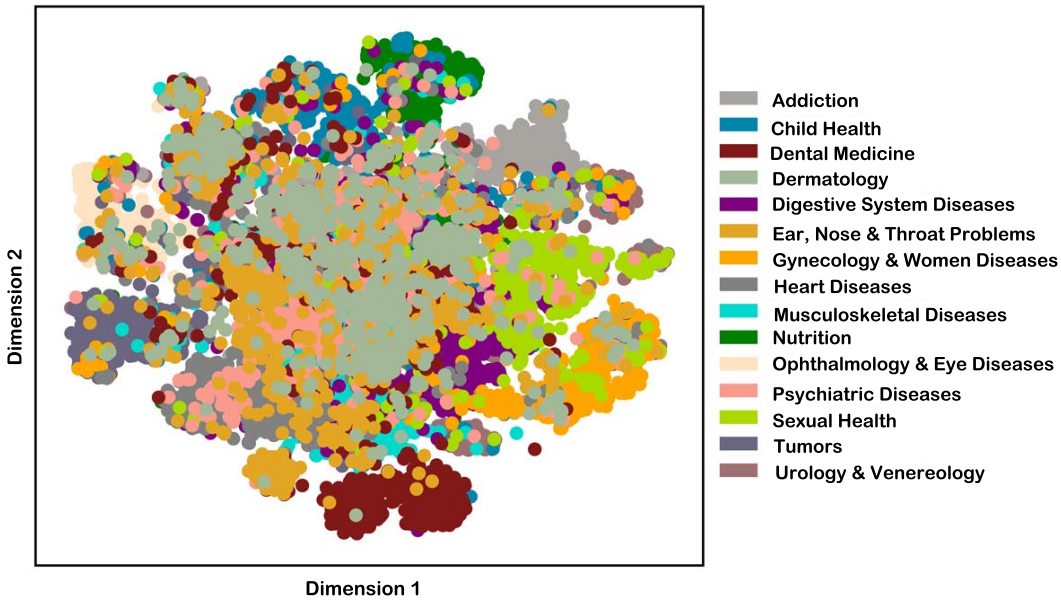



Figure 4: The tSNE projection of 15,000 questions after questions' preprocessing and by considering the total number of features that are produced from the TF-IDF vectorizer.

algorithms can be adapted to handle multi-class classification problems by dividing the multi-class classification problem into several binary classification problems. Essentially, in the literature, there are two basic approaches for transforming the multi-class classification into multiple binary classifications, which are the one-versus-one (OVO), and the one-versus-rest (OVR) methods. The former is concerned with training a classifier for each pair of classes, while the predicted class is the one with the most votes. However, a downside of the OVO strategy is that it requires the training of $(N_c * (N_c - 1)/2)$ classifiers, for which $N_c$ is the number of classes.

Meanwhile, the latter requires training one classifier for each class. Thus, each classifier learns a class against all other classes. As such, each classifier predicts a probability of a class, and, finally, the predicted class is that with the highest probability score. Markedly, OVR has some noteworthy advantages over the OVO approach. One such advantage is that it learns smaller number of classifiers ($N_c$), which makes its computations easier [38].

### *6.1. Particle encoding*

Each particle in the proposed methodology is represented by a binary vector that consists of three parts. The first part of the vector represents the

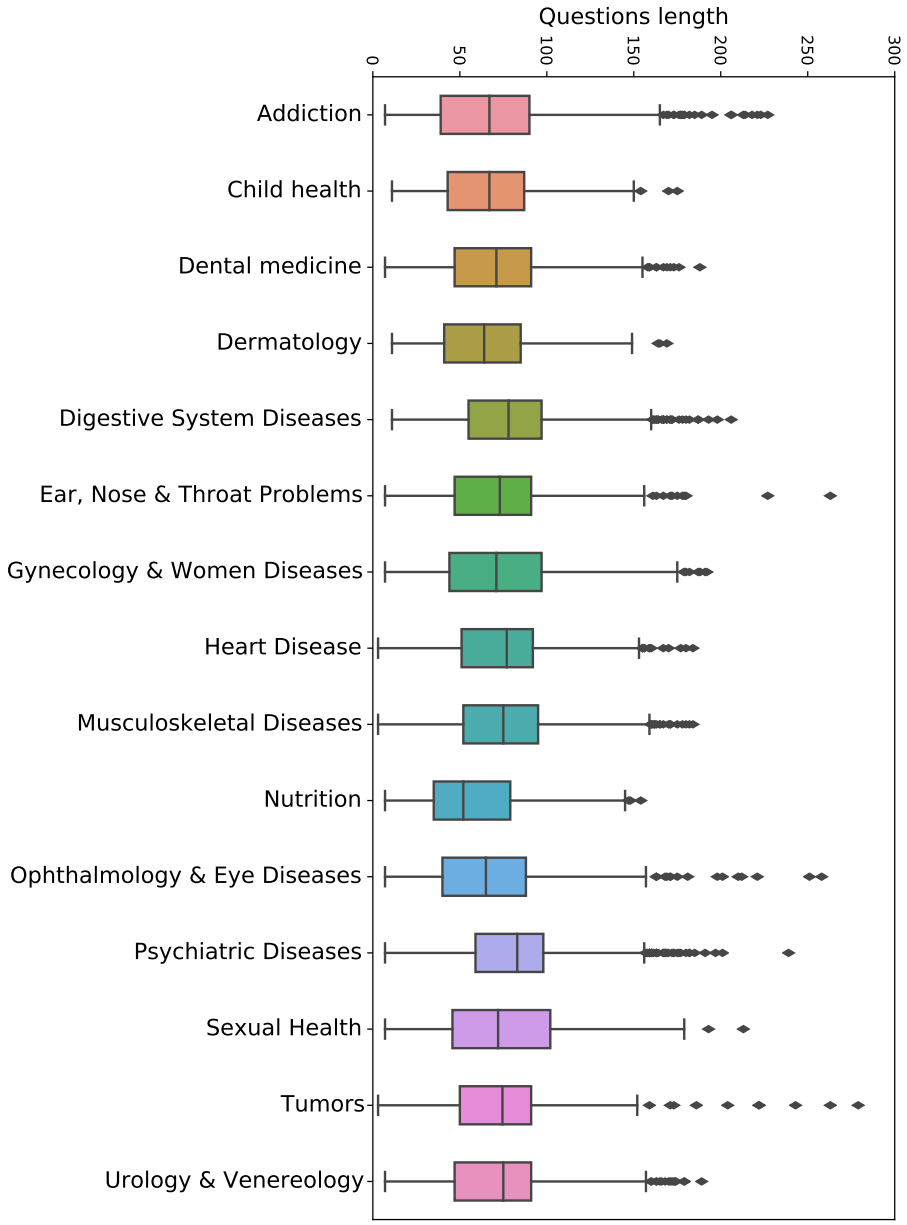



Figure 5: Questions length distribution in terms of the number of words for the 15 medical classes.

399 maximum number of features to be considered by the TF-IDF vectorizer,
400 where its length is (19) elements. This number is selected to represent the

total number of possible features in the dataset (102,994). The second part of the vector consists of two elements that represent the minimum document frequency parameter. This parameter determines the threshold according to which all terms have document frequency lower than it will be removed from the process when building the vocabulary in TF-IDF. Whereas, the third part presents the cost ($C$) parameter of SVM algorithms included in the OVR mechanism, where ($C$) has (8) elements in length. Hence, a vector of dimension (29) is utilized to encode the number of features, the document frequency parameter of TF-IDF, and the $C$ hyperparameter of SVMs. Fig. 6 represents the adopted structure of PSO particles.

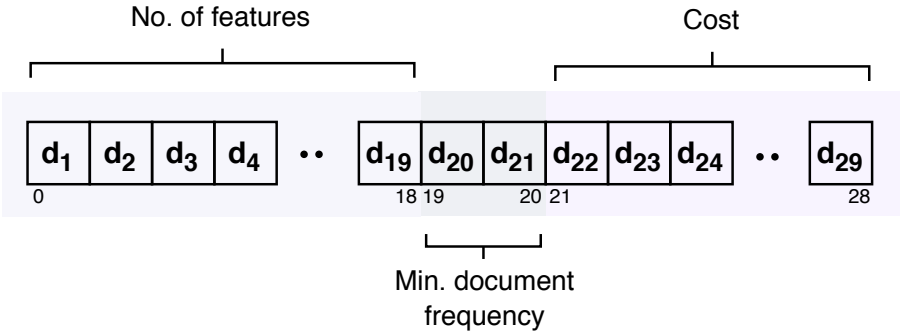



Figure 6: An illustration of PSO particle dissected into three parts.

### *6.2. Fitness evaluation*

Assessing the goodness of the generated particles requires decoding the vector's elements into three parameters; the maximum number of features ($f_s$), the minimum document frequency ($f_d$), and the cost parameter. Therefore, ($f_s$) and ($f_d$) were adopted to build the TF-IDF vectorizer, while the $C$ parameter is used to create the SVM classifiers. The classification model of the SVMs were trained based on the preprocessed training dataset and validated using a separate transformed validation set. It is worth to note that the OVR method generated 15 different SVMs classifier for each class label.

The evaluation of the trained model depends on a weighted-sum fitness approach that relies on the classification performance given by (1-accuracy) and the selected features rate. The fitness is formulated by Eq. 13, where $\alpha$ and $\beta$ are two weighting parameters, in which, ($\beta = 1 - \alpha$). ($f_s$) is the

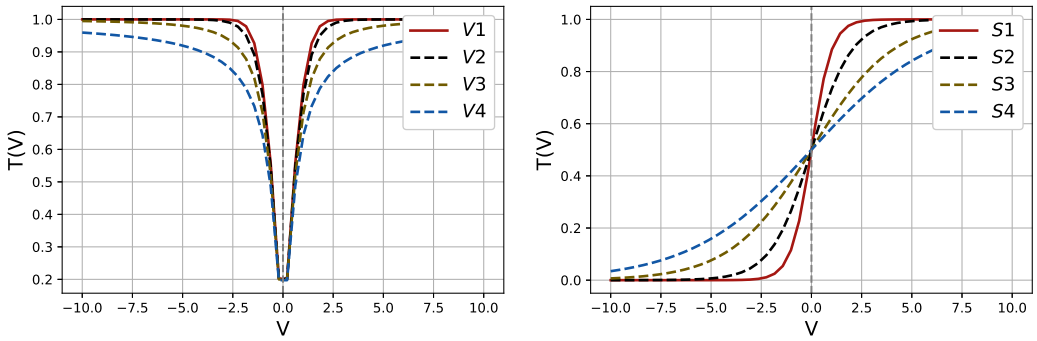
number of selected features, and $(F)$ represents the total number of features (unique words) [39, 40].

$$Fitness = \alpha(1 - Accuracy) + \beta\frac{f_s}{F} \tag{13}$$

### *6.3. Binarization mechanism*

The objective of the binarization mechanism is to convert the real search space into a binary search space so that it can be adapted to binary optimization problems. It was first introduced by Kennedy and Eberhart, who used it to transform the particles of PSO from real-valued vectors into binary vectors.

Two variants of Sigmoid transfer functions are used extensively in the literature. These are the S-shaped and V-shaped functions [32]. The transfer function assigns a probability for each element in the vector, which indicates its likelihood of being (1). If the probability is greater than a randomly generated number, then the corresponding element of the particle is assigned a value of (1); otherwise, it is assigned a value of (0). Eq. 8 shows the S2 transfer function (Sigmoid), while Eq. 9 presents the transformation process based on the probability produced by Eq. 8. Fig. 7 depicts the behavior of the V-shaped and S-shaped transfer functions.




(a) (b)

Figure 7: A description of two families of transfer functions; (a) is the V-shaped function, and (b) is the S-shaped function

### 6.4. Procedure

The procedure of ($BPSO_{TF} - SVM_{OVR}$) starts by creating a random population of particles that has random positions and velocities. Each particle is evaluated using an objective (fitness) function, as illustrated by Eq. 13. Hence, depending on the computed fitness values, each particle updates its (*bpest*) of the obtained solutions. Meanwhile, the global best (*gbest*) solution found among all particles is stored in external memory. Afterward, the particles adjust their positions and velocities using Eqs. 9 and 6, respectively. Before updating the position vectors, the transfer function transforms the velocity values into potential probabilities to encode the binary position vector. The processes of generating solutions (particles), evaluating them, and adjusting their positions and velocities are repeated until the maximum number of iterations is reached.

The fitness evaluation phase involves five critical operations, which are decoding the particles, dividing the data into training and validation sets, creating the TF-IDF vectorizer, building the SVM classifiers using OVR mechanism, and evaluating the fitness of the corresponding particle.

Decoding the particles from binary to real values results in three optimization parameters: the maximum number of features ($max\_features$), minimum document frequency ($min\_df$), and the cost parameter of SVMs. The first two parameters ($max\_features$,$min\_df$) are utilized to construct the TF-IDF vectorizer, while the cost parameter is considered when building a linear SVM models. The TF-IDF vectorizer tokenizes the set of questions and calculates the occurrences of words. Therefore, to build a TF-IDF features matrix of shape (samples#, features#), a TF-IDF transformer is applied to the data.

Regarding the classification model, the OVR approach is employed for training 15 SVM classifiers that handle the multi-class classification problem. A subset of the training data is left for validation, which accounts for (25% of the entire training dataset). The training subset is adopted for fitting the SVM classifiers in the OVR, while the validation set is used to evaluate the trained models and to compute the particles' fitness. Further, the applied fitness is a weighted sum of the error and features rate, as described by Eq. 13. Upon evaluating the particles, the fitness is returned to the PSO cycle to adjust the particles and guide the search toward the optimal solutions. Fig. 8 illustrates the methodology steps of the proposed ($BPSO_{TF} - SVM_{OVR}$).

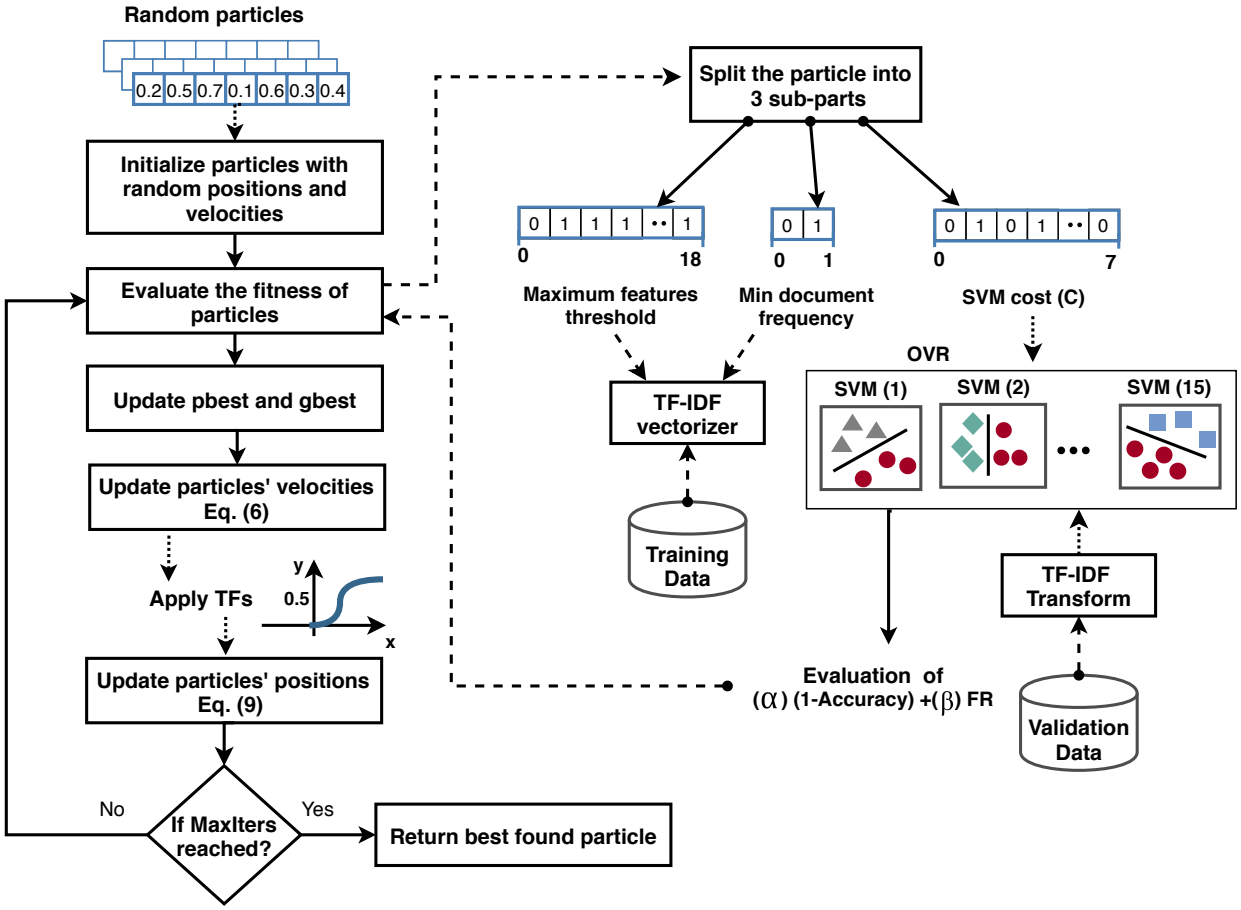



Figure 8: A description of the methodology in which the MaxIters is the maximum number of iterations, and FR is the features rate.

## 7. Evaluation measures

A set of evaluation measures were utilized to quantify the performance of the multi-class classification problem, including the accuracy, macro-recall, macro-precision, macro-f1, and the features reduction rate (FRR). The macro measure converts binary evaluation metrics into multi-class evaluation measures that averages the computation of the binary metric across all the classes. The accuracy is the fraction of correct predictions over the total number of samples ($n_{\text{samples}}$), which is defined in Eq. 14, where $y$ is the actual value of

sample $(i)$, and $\hat{y}$ is the predicted value.

$$Accuracy(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \tag{14}$$

The macro-recall ($Recall_m$) calculates the average recall of each class, for which, the recall expresses how much the model can identify the positive examples. The macro-recall is given by Eq. 15. In which, $(L)$ is the set of all classes, $(y_l)$ is the proportion of predicted data with label $l$, and $\hat{y}_l$ represents the data samples that have true labels.

$$Recall_m = \frac{1}{|L|} \sum_{l \in L} R(y_l, \hat{y}_l), \quad R(y_l, \hat{y}_l) = \frac{|y_l \cap \hat{y}_l|}{|\hat{y}_l|} \tag{15}$$

Similarly, the macro-precision ($Precision_m$) finds the mean precision across all classes. In this case, the precision describes the ratio of correctly identified positive examples over the actual number of positive examples (Eq. 16).

$$Precision_m = \frac{1}{|L|} \sum_{l \in L} P(y_l, \hat{y}_l), \quad P(y_l, \hat{y}_l) = \frac{|y_l \cap \hat{y}_l|}{|y_l|} \tag{16}$$

The macro f1-score ($F1 - score_m$) calculates the score for each class and then returns their unweighted average. The F1-score is the harmonic mean of precision and recall, and it indicates the balance level between them. The mathematical formula of $F1 - score_m$ is given by Eqs. (17-18).

$$F1 - score_m = \frac{1}{|L|} \sum_{l \in L} F_\beta(y_l, \hat{y}_l) \tag{17}$$

$$F_\beta(y_l, \hat{y}_l) = \left(1 + \beta^2\right) \frac{P(y_l, \hat{y}_l) R(y_l, \hat{y}_l)}{\beta^2 P(y_l, \hat{y}_l) + R(y_l, \hat{y}_l)} \tag{18}$$

The FRR indicates the capability of the classifier in removing the irrelevant features and promoting its classification performance. The FRR is given by Eq. 19, where $F$ is the total number of unique features, and $f_s$ is the number of selected features.

$$FRR = \frac{F - f_s}{F} \tag{19}$$

## 8. Experiments and results

### *8.1. Experimental setup*

All experiments are implemented using Python version (3.7) on a Windows server (2012) platform with a random-access memory (RAM) of 64 GB, and Intel(R) Xeon(R) CPU E5-2609 processors with a speed of 1.70 GHz.

For the PSO, the number of iterations and the population size are set to 50, the ($c1$) and ($c2$) constants are set to 2, and the maximum and minimum values of the inertia weight are $\in [0.6, 0.9]$. Because the PSO searches for optimal values of three parameters (number of features, minimum document frequency, and cost), the size of the search space of the feature subsets is the maximum number words (102,994), while for the minimum document frequency it is in {1,2,3,4}. The range of the search for the cost parameter is [0,1.28]. Furthermore, all experiments were repeated (30) times to ensure the stability of results.

### *8.2. Effect of fitness function parameters*

This subsection investigates the influence of the weighting parameters ($\alpha$,$\beta$) of the fitness equation on the performance of the proposed ($BPSO_{TF}-SVM_{OVR}$).

Different values of $\alpha$ and $\beta$ are used, ranging from 0.0 to 0.2 for the $\beta$ coefficient and from 0.8 to 1.0 for the $\alpha$ parameter. Table 2 provides details of the performance of ($BPSO_{TF} - SVM_{OVR}$) for different values of $\alpha$ and $\beta$ regarding its accuracy, the number of features, and the FRR. In terms of accuracy, its performance is shows stability across the different values of $\alpha$ and $\beta$. When ($\alpha = 0.85$ and $\beta = 0.15$), the accuracy reached its minimal value of 82%. Meanwhile, when $\alpha = 0.9999$ and $\beta = 0.0001$, the accuracy achieved its highest value (85.7%). Nonetheless, regarding the number of features, having ($0.99 < \alpha < 1.0$) results in greater number of features and lower FRR. In contrast, when ($0.80 \leq \alpha < 0.99$), a lower $\alpha$ value yields a higher FRR. Hence, balancing between minimizing the number of features and maximizing the accuracy results can be achieved at ($\alpha = 0.99$ and $\beta = 0.01$.)

### *8.3. Effect of transfer functions*

The binarization mechanisms play a vital role in enhancing the performance of metaheuristic optimizers and hindering them from converging

Table 2: The effect of ($\alpha$) and ($\beta$) on the accuracy, average number of selected features, and the features reduction rate.

| $\alpha$ | $\beta$ | **Accuracy** | **No. of Features** | **FRR** |
|---|---|---|---|---|
| 1.0000 | 0.0000 | 0.854 | 37231 | 0.639 |
| 0.9999 | 0.0001 | 0.857 | 36954 | 0.641 |
| 0.9990 | 0.0010 | 0.855 | 10889 | 0.894 |
| **0.9900** | **0.0100** | **0.854** | **4686** | **0.955** |
| 0.9500 | 0.0500 | 0.839 | 1690 | 0.984 |
| 0.9000 | 0.1000 | 0.836 | 1657 | 0.984 |
| 0.8500 | 0.1500 | 0.820 | 982 | 0.990 |
| 0.8000 | 0.2000 | 0.822 | 1027 | 0.990 |

quickly and getting trapped in local regions. Therefore, two variants of transfer functions for binarizing the search space of PSO are utilized, namely, are the S-shaped and V-shaped families.

Table 3 shows the performance of the proposed ($BPSO_{TF} - SVM_{OVR}$) approach using four S-shaped and four V-shaped functions, in terms of its accuracy, number of features, and FRR. Generally, it is clear that the S-shaped functions can guide the algorithm toward better performance when considering the accuracy metric. For instance, at all ($S1$,$S2$,$S3$,$S4$), the algorithm approximately obtained 85% accuracy. In contrast, even though ($V1$ and $V4$) achieved nearly 85.5% accuracy, both ($V2$ and $V3$) failed to achieve more than 84.4%. Further, when considering the number of features, $V1$ performed the worst (22,656 features out of 102,994). Similarly, $V3$ had 13,631 features. Conversely, all S-shaped functions minimized the number of features remarkably, which were expressed as FRR values of more than 95%. ($S4$) showed competitive results in terms of accuracy, number of features, and FRR (85.3%, 4129, and 95.9%, respectively). Even though ($V2$) dramatically reduced the number of features to 3,835, it was significantly less accurate than other variants.

Since the best performance was achieved by the algorithm at ($S4$), the rest of the experiments are conducted based on the ($S4$) binarization mechanism.

Fig. 9 shows the convergence curves of ($BPSO_{TF} - SVM_{OVR}$) based on

Table 3: The effect of binarization techniques on the accuracy, average number of selected features, and the features reduction rate.

| TFs | Accuracy | No. of Features | FRR |
|---|---|---|---|
| S1 | 0.852 | 4268 | 0.9586 |
| S2 | 0.854 | 4686 | 0.9545 |
| S3 | 0.853 | 4147 | 0.9597 |
| **S4** | **0.853** | **4129** | **0.9599** |
| V1 | 0.855 | 22656 | 0.7800 |
| V2 | 0.841 | 3835 | 0.9628 |
| V3 | 0.844 | 13631 | 0.8677 |
| V4 | 0.856 | 7962 | 0.9227 |

the S-shaped transfer functions. All S-shaped transfer functions converged relatively closely throughout the course of iterations. However, ($S1$) converged fast, which might lead to trapping in local optima. Meanwhile, ($S4$) converged smoothly.

Fig. 10 depicts the convergence curves of ($BPSO_{TF} - SVM_{OVR}$) based on the V-shaped transfer functions. It is obvious that all ($V1$,$V2$,$V3$,$V4$) readily became stuck in a local solution at the beginning of the iterations. Also, it is clear that ($V3$) performed weakly in comparison with the other variants, while ($S4$) outperformed the V-shaped transfer functions.

Fig. 11 depicts the precision and recall of all the 15 classes of the proposed approach ($BPSO_{S4} - SVM_{OVR}$). It is clear from the plot that the algorithm achieved recall scores of greater than 80% for all classes. Meanwhile, for the precision scores, even that they increased by more than (80%) most of the time, the proposed approach performed slightly lower than (80%) for the "Dermatology" and "Psychiatric diseases" classes.

Further, Fig. 12 shows the confusion matrix of the best obtained model of the proposed approach. The diagonal values correspond for the number of correctly predicted samples, while the other values represent the misclassified samples. Values in the diagonal can be confirmed by the tSNE representation in Fig. 4, which refers to clusters of clearly defined shape. Moreover, it can be noticed that there are seven samples of the ("Ear, Nose & Throat

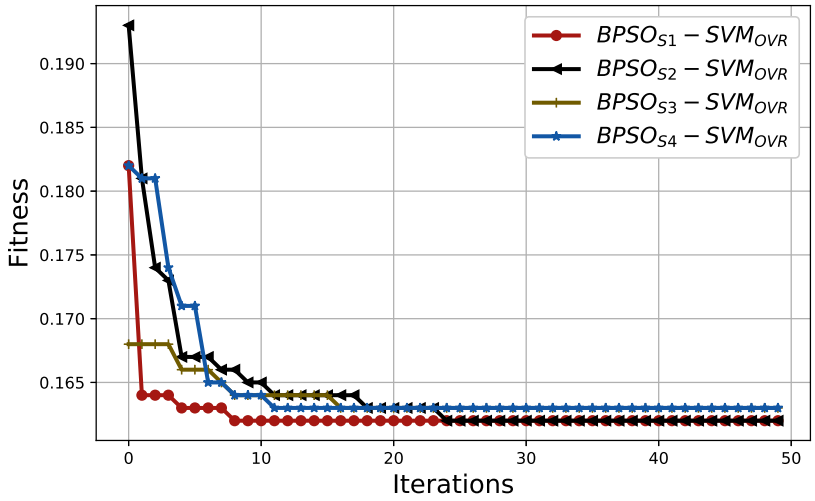



Figure 9: The convergence curves of the proposed ($BPSO_{TF} - SVM_{OVR}$) across the S-shaped transfer functions.

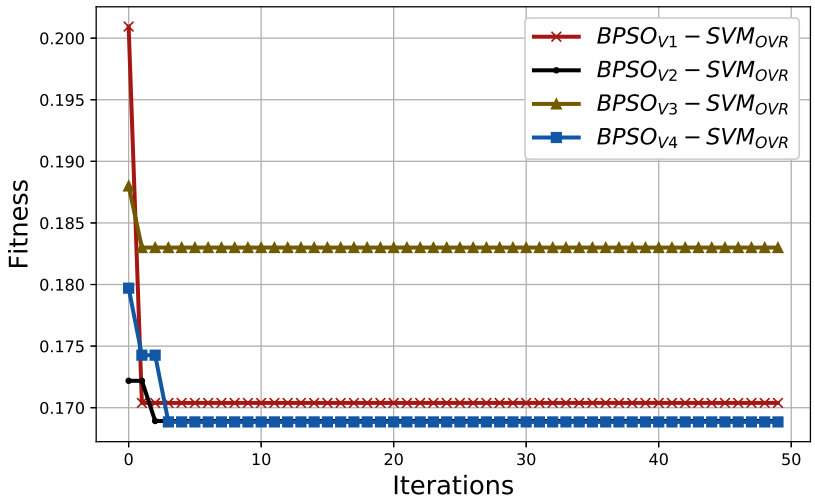



Figure 10: The convergence curves of the proposed ($BPSO_{TF} - SVM_{OVR}$) across the V-shaped transfer functions.

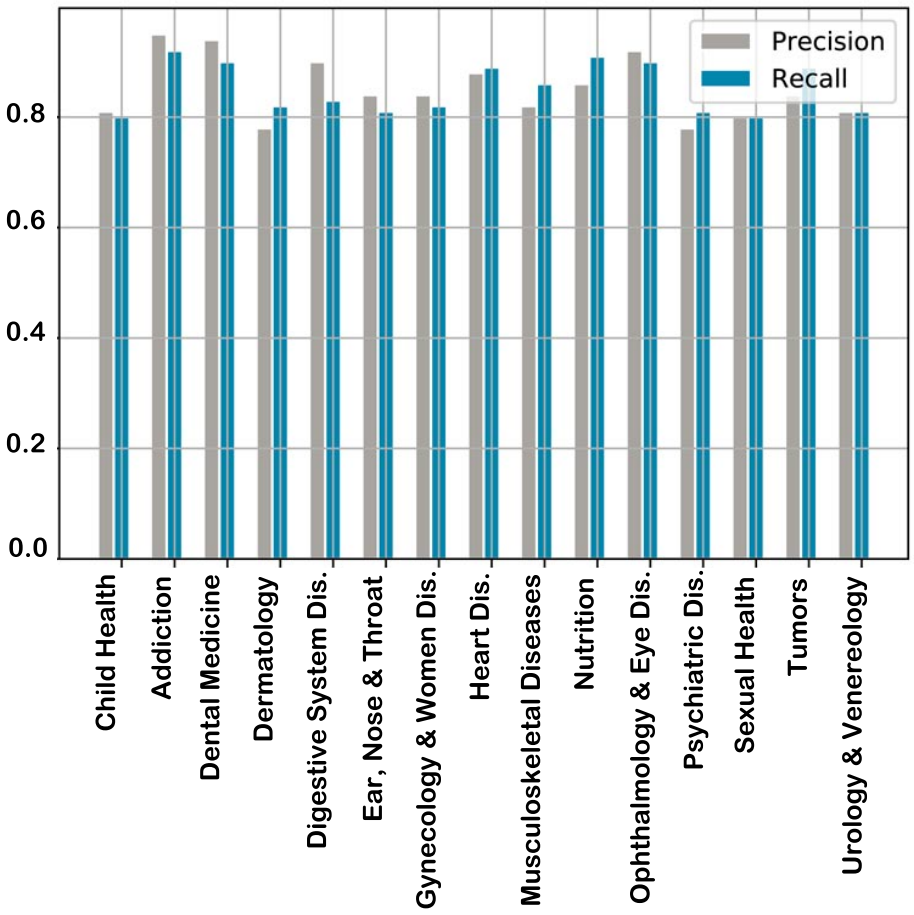



Figure 11: An illustration of the distribution of the recall and precision measures over all classes.

Problems") class were misclassified as ("Dental Medicine"). Also, one sample of the ("Dermatology") class was incorrectly classified as ("Dental Medicine"). Furthermore, it is obvious that some questions of the ("Sexual Health") class were misclassified as ("Urology & Venereology") and ("Gynecology & Women Diseases"), which is reasonable due to the similar nature of these classes.

### *8.4. Comparison results*

The best obtained model of the swarm intelligent based SVM was the ($S4$) binarization mechanism, with an average FRR of 95.7% (out of 30 runs), and the most frequent value of the minimum document frequency of

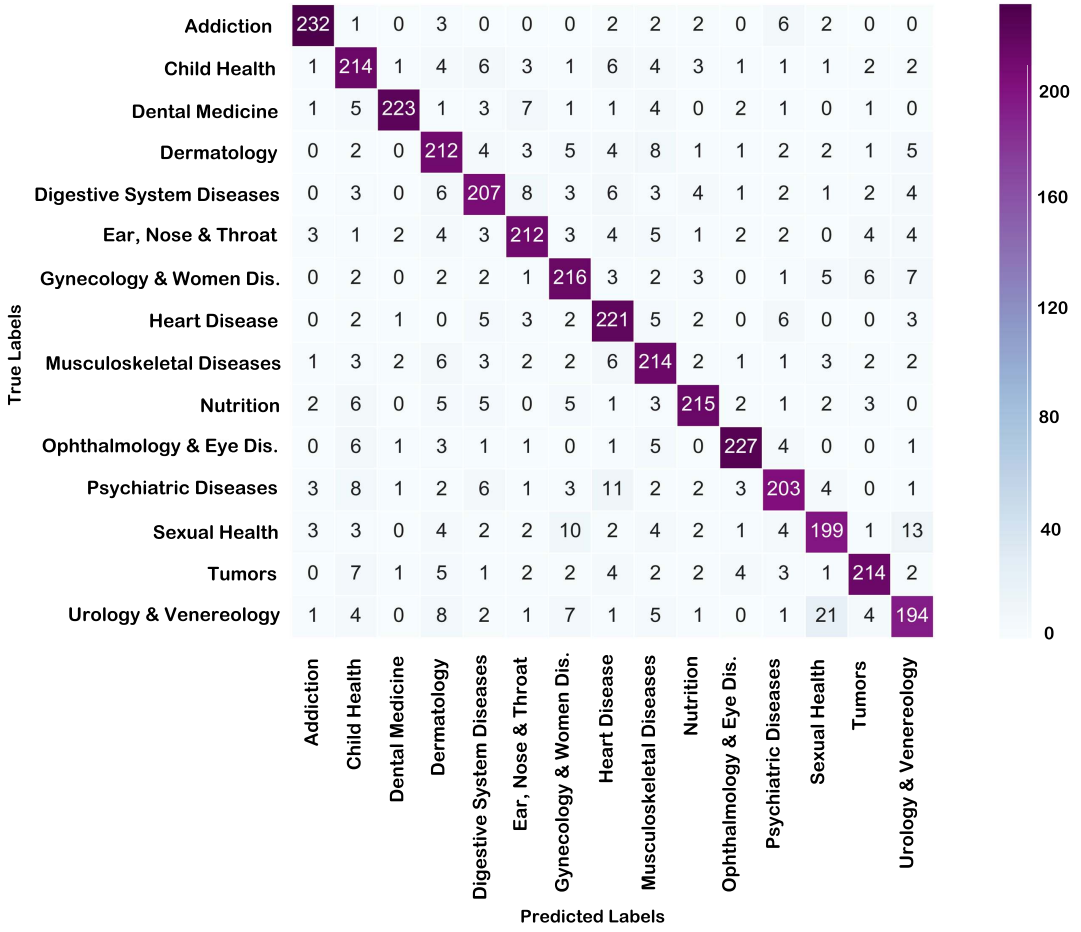



Figure 12: A heatmap presentation of the confusion matrix of the best model of the proposed ($BPSO_{S4} - SVM_{OVR}$).

(4). Hence, the best model ($BPSO_{S4} - SVM_{OVR}$) is compared with various common machine learning algorithms (i.e. linear SVMs, Random Forest [41], Logistic Regression [42], Multinomial Naïve Bayes [43], Complement Naïve Bayes [44], Bernoulli Naïve Bayes [43], Stochastic Gradient Descent classifier (SGDClassifier) [45], SVMs based on the non-linear (RBF) kernel, XGBoost classifier [46], Adaboost [47], and the K-Nearest Neighbor (KNN) [48] algorithm). The performance of the ($BPSO_{S4} - SVM_{OVR}$) is compared with the above-mentioned algorithms in three phases. The first is when the TF-IDF vectorizer used the total number of features. The second is when

the TF-IDF vectorizer used 50% of the features. The third is when it used (25%) of the features. All three experiments were repeated (30) times to ensure the stability of the results. All experiments were evaluated in terms of their accuracy, macro F1-score, macro recall, and macro precision. For linear SVMs, several initial settings of the ($C$) were tested, mainly at costs of 0.2, 0.5, 1.0, and 1.5. The best model was determined as that with ($C$) value of 0.5.

Table 4 presents the performance results of ($BPSO_{S4} - SVM_{OVR}$) with 11 machine learning algorithms concerning four evaluation measures and the total number of features. The ($BPSO_{S4} - SVM_{OVR}$) significantly outperformed all other algorithms in terms of average accuracy, $f1 - score_m$, $recall_m$, and $precision_m$, which reinforced by the reasonable values of standard deviation accompanied by the $\pm$ symbol. The model yielded values (0.852, 0.851, 0.851, and 0.852) for accuracy, $f1 - score_m$, $recall_m$, and $precision_m$, respectively. In terms of accuracy, ($BPSO_{S4} - SVM_{OVR}$) performed the best. The LinearSVM, Logistic Regression, Multinomial NB, Bernoulli NB, and SGDClassifier achieved greater than 80% accuracy. As with the rest of the measures, the SVMs (RBF) performed the worst with an accuracy of (25%). Further, Wilcoxon Ranksum test [49] was employed to determine whether the differences between the obtained results were statistically significant. This test was based on the accuracy metric with a significance level ($\alpha$) is (0.05). As shown in Table 4, the difference between the results of the proposed approach and the other algorithms is significant.

Tables (5 and 6) show the performance of ($BPSO_{S4} - SVM_{OVR}$) in comparison to other machine learning algorithms when the considered number of features are half and a quarter of the original number of features. However, from the obtained results, it is noticeable that changing the number of features did not affect significantly on the performance of the utilized machine learning algorithms.

## 9. Conclusion

Text classification is a crucial part of a successful QA system. However, recent technological advances and the continuous emergence of health requirements have made it crucial to develop a medical QA system to promote the biomedical informatics sector. This article takes Altibbi company as a case study. This company provides telemedicine services in the MENA region. The objective of this case study is to develop a text (medical question)

Table 4: A comparison of the average performance of $BPSO_{S4} - SVM_{OVR}$ with other machine learning algorithms considering the number of all features. P-Values are based on ($\alpha = 0.05$), where the significant results are with underline typeface.

| **Algorithm** | **Accuracy** | $F1 - score_m$ | $Recall_m$ | $Precision_m$ |
|---|---|---|---|---|
| $BPSO_{S4} - SVM_{OVR}$ | **0.852 ± 0.001** | **0.851 ± 0.001** | **0.851 ± 0.001** | **0.852 ± 0.001** |
| linearSVM(C=0.5) | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.845 ± 0.000 |
| | 2.8719E-11 | | | |
| Random Forest | 0.736 ± 0.006 | 0.735 ± 0.007 | 0.736 ± 0.006 | 0.739 ± 0.007 |
| | 2.8719E-11 | | | |
| Logistic Regression | 0.835 ± 0.003 | 0.836 ± 0.003 | 0.839 ± 0.003 | 0.840 ± 0.003 |
| | 2.8719E-11 | | | |
| MultinomialNB | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.814 ± 0.035 |
| | 2.8719E-11 | | | |
| ComplementNB | 0.791 ± 0.023 | 0.789 ± 0.023 | 0.791 ± 0.023 | 0.791 ± 0.023 |
| | 2.8719E-11 | | | |
| BernoulliNB | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.811 ± 0.032 |
| | 2.8719E-11 | | | |
| SGDClassifier | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 |
| | 2.8719E-11 | | | |
| SVMs (RBF) | 0.250 ± 0.280 | 0.230 ± 0.300 | 0.250 ± 0.280 | 0.750 ± 0.060 |
| | 2.8719E-11 | | | |
| XGBoost | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.785 ± 0.010 |
| | 2.8719E-11 | | | |
| Adaboost | 0.615 ± 0.005 | 0.626 ± 0.008 | 0.615 ± 0.005 | 0.686 ± 0.010 |
| | 2.8719E-11 | | | |
| KNN | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.765 ± 0.039 |
| | 2.8719E-11 | | | |

speciality classification system. The proposed approach addresses essential challenges related to handling and preprocessing the Arabic language and the large number of classes. The swarm intelligence PSO with SVMs is applied with the one-versus-rest mechanism for feature selection and hyperparameter tuning purposes. Meanwhile, the proposed approach is used to handle multi-class classification tasks based on multiple binarization mechanisms.

Table 5: A comparison of the average performance of $BPSO_{S4} - SVM_{OVR}$ with other machine learning algorithms considering 50% of the features.

| **Algorithm** | **Accuracy** | $F1 - score_m$ | $Recall_m$ | $Precision_m$ |
|---|---|---|---|---|
| $BPSO_{S4} - SVM_{OVR}$ | **0.852 ± 0.001** | **0.851 ± 0.001** | **0.851 ± 0.001** | **0.852 ± 0.001** |
| linearSVM(C=0.5) | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.845 ± 0.000 |
| Random Forest | 0.736 ± 0.006 | 0.735 ± 0.007 | 0.736 ± 0.006 | 0.739 ± 0.007 |
| Logistic Regression | 0.839 ± 0.003 | 0.839 ± 0.003 | 0.839 ± 0.003 | 0.840 ± 0.003 |
| MultinomialNB | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.814 ± 0.035 |
| ComplementNB | 0.791 ± 0.023 | 0.789 ± 0.023 | 0.791 ± 0.023 | 0.791 ± 0.023 |
| BernoulliNB | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.811 ± 0.032 |
| SGDClassifier | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 |
| SVMs (RBF) | 0.253 ± 0.285 | 0.228 ± 0.305 | 0.253 ± 0.285 | 0.755 ± 0.061 |
| XGBoost | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.785 ± 0.010 |
| Adaboost | 0.615 ± 0.005 | 0.626 ± 0.008 | 0.615 ± 0.005 | 0.686 ± 0.010 |
| KNN | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.765 ± 0.039 |

Table 6: A comparison of the average performance of $BPSO_{S4} - SVM_{OVR}$ with other machine learning algorithms considering 25% of the features.

| **Algorithm** | **Accuracy** | $F1 - score_m$ | $Recall_m$ | $Precision_m$ |
|---|---|---|---|---|
| $BPSO_{S4} - SVM_{OVR}$ | **0.852 ± 0.001** | **0.851 ± 0.001** | **0.851 ± 0.001** | **0.852 ± 0.001** |
| linearSVM(C=0.5) | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.844 ± 0.000 | 0.845 ± 0.000 |
| Random Forest | 0.736 ± 0.006 | 0.735 ± 0.007 | 0.736 ± 0.006 | 0.739 ± 0.007 |
| Logistic Regression | 0.839 ± 0.003 | 0.839 ± 0.003 | 0.839 ± 0.003 | 0.840 ± 0.003 |
| MultinomialNB | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.812 ± 0.036 | 0.814 ± 0.035 |
| ComplementNB | 0.791 ± 0.023 | 0.789 ± 0.023 | 0.791 ± 0.023 | 0.791 ± 0.023 |
| BernoulliNB | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.809 ± 0.033 | 0.811 ± 0.032 |
| SGDClassifier | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 | 0.842 ± 0.001 |
| SVMs (RBF) | 0.250 ± 0.280 | 0.230 ± 0.300 | 0.250 ± 0.280 | 0.750 ± 0.060 |
| XGBoost | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.783 ± 0.010 | 0.785 ± 0.010 |
| Adaboost | 0.615 ± 0.005 | 0.626 ± 0.008 | 0.615 ± 0.005 | 0.686 ± 0.010 |
| KNN | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.761 ± 0.042 | 0.765 ± 0.039 |

Remarkably, the proposed ($BPSO_{S4} - SVM_{OVR}$) achieved promising results in comparison with previously developed approaches. Furthermore, it outperformed 11 well-known machine learning algorithms in terms of the accuracy, macro f1-score, macro recall, macro precision, and features reduction rate.

An important implication that emerges from this study is the ability to automate the classification of patients' questions more accurately and in real-time. Where this undoubtedly saves the resources and time, as well as, lessens doctors' and practitioners' efforts in directing the consultations to their correct specialities. Indeed, automatic question classification is a stepping stone to automatic question answering, particularly, for artificial-oriented natural language systems. However, the scope of the current work is limited in several ways. First, the number of considered questions was restricted to 15,000, while more data can be obtained and curated, which can widen the horizon for further experiments and findings. Second, the spatial representation of texts by the TF-IDF vectorizer is context-free, which means lacking the ability to capture the semantics of the questions. Incorporating vectorization techniques that are capable of expressing the hidden semantics are a breakthrough nowadays, that is interesting for further research work.

This research can be expanded to research niches in more depth. For example, future studies can investigate the adoption of advanced feature representations like word embedding not merely to understand the syntax of words, but also to capture the semantics of hidden information. In addition, is the utilization of transfer learning, including the pre-trained word embedding, or the pre-trained deep learning models. These research recommendations can also be followed using large datasets of medical consultations.

## Author contributions section

The authors' contributions were as follows—Hossam Faris and Maria Habib: designed the study, performed the analysis and wrote the paper; Maria Habib and Mohammad Faris: conducting the experiments and plot the figures; Manal Alomari: Data description; Alaa Alomari: data collection and project administration; and all authors: read and approved the final manuscript.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work

reported in this paper.

## Acknowledgement

We would like to express our gratitude to Altibbi Company for their support and for providing us with the information and datasets that significantly facilitated this research work.

## References

[1] X. Li, D. Roth, Learning question classifiers, in: Proceedings of the 19th international conference on Computational linguistics-Volume 1, Association for Computational Linguistics, 2002, pp. 1–7.

[2] I. Statista, The most common languages on the internet, https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet (July, 2019).

[3] G. Badaro, R. Baly, H. Hajj, W. El-Hajj, K. B. Shaban, N. Habash, A. Al-Sallab, A. Hamdi, A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations, ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18 (2019) 27.

[4] J. Hegde, B. Rokseth, Applications of machine learning methods for engineering risk assessment–a review, Safety science 122 (2020) 104492.

[5] L. Gan, H. Wang, Z. Yang, Machine learning solutions to challenges in finance: An application to the pricing of financial products, Technological Forecasting and Social Change 153 (2020) 119928.

[6] G. B. Kim, W. J. Kim, H. U. Kim, S. Y. Lee, Machine learning applications in systems metabolic engineering, Current Opinion in Biotechnology 64 (2020) 1–9.

[7] D. Zhang, W. S. Lee, Question classification using support vector machines, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 26–32.

[8] D. Metzler, W. B. Croft, Analysis of statistical question classification for fact-based questions, Information Retrieval 8 (2005) 481–504.

[9] X. Li, X.-J. Huang, L. Wu, Question classification using multiple classifiers, in: Proceedings of the Fifth Workshop on Asian Language Resources (ALR-05) and First Symposium on Asian Language Resources Network (ALRN), 2005.

[10] Z. Huang, M. Thint, A. Celikyilmaz, Investigation of question classifier in question answering, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2, Association for Computational Linguistics, 2009, pp. 543–550.

[11] Z. Yu, L. Su, L. Li, Q. Zhao, C. Mao, J. Guo, Question classification based on co-training style semi-supervised learning, Pattern Recognition Letters 31 (2010) 1975–1980.

[12] L. Liu, Z. Yu, J. Guo, C. Mao, X. Hong, Chinese question classification based on question property kernel, International Journal of Machine Learning and Cybernetics 5 (2014) 713–720.

[13] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, H. Yu, Askhermes: An online question answering system for complex clinical questions, Journal of biomedical informatics 44 (2011) 277–288.

[14] P. Le-Hong, X.-H. Phan, T.-D. Nguyen, Using dependency analysis to improve question classification, in: Knowledge and Systems Engineering, Springer, 2015, pp. 653–665.

[15] A. Mohasseb, M. Bader-El-Den, M. Cocea, H. Liu, Improving imbalanced question classification using structured smote based approach, in: 2018 International Conference on Machine Learning and Cybernetics (ICMLC), volume 2, IEEE, 2018, pp. 593–597.

[16] M. Sarrouti, S. O. El Alaoui, A machine learning-based method for question type classification in biomedical question answering, Methods of information in medicine 56 (2017) 209–216.

[17] A. Mohasseb, M. Bader-El-Den, M. Cocea, Question categorization and classification using grammar based approach, Information Processing & Management 54 (2018) 1228–1243.

[18] H. Abdelnasser, M. Ragab, R. Mohamed, A. Mohamed, B. Farouk, N. M. El-Makky, M. Torki, Al-bayan: an arabic question answering system for the holy quran, in: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), 2014, pp. 57–64.

[19] A. Waheeb, A. Babu, Classification of arabic questions using multinomial naïve bayes and support vector machines, International Journal of Latest Trends In Engineering And Technology (2016) 82–86.

[20] A. Hamza, N. En-Nahnahi, K. A. Zidani, S. E. A. Ouatik, An arabic question classification method based on new taxonomy and continuous distributed representation of words, Journal of King Saud University-Computer and Information Sciences (2019 Jan 14). doi:https://doi.org/10.1016/j.jksuci.2019.01.001.

[21] F. López Seguí, R. Ander Egg Aguilar, G. de Maeztu, A. García-Altés, F. García Cuyàs, S. Walsh, M. Sagarra Castro, J. Vidal-Alaball, Teleconsultations between patients and healthcare professionals in primary care in catalonia: The evaluation of text classification algorithms using supervised machine learning, International Journal of Environmental Research and Public Health 17 (2020) 1093.

[22] M. Wasim, M. N. Asim, M. U. G. Khan, W. Mahmood, Multi-label biomedical question classification for lexical answer type prediction, Journal of biomedical informatics 93 (2019) 103143.

[23] M. Sarrouti, S. O. El Alaoui, Sembionlqa: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions, Artificial Intelligence in Medicine 102 (2020) 101767.

[24] H. Faris, M. A. Hassonah, A.-Z. Ala'M, S. Mirjalili, I. Aljarah, A multiverse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture, Neural Computing and Applications 30 (2018) 2355–2369.

[25] I. Aljarah, A.-Z. Ala'M, H. Faris, M. A. Hassonah, S. Mirjalili, H. Saadeh, Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm, Cognitive Computation 10 (2018) 478–495.

[26] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 144–152.

[27] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE, 1995, pp. 39–43.

[28] M. Elbes, S. Alzubi, T. Kanan, A. Al-Fuqaha, B. Hawashin, A survey on particle swarm optimization with emphasis on engineering and network applications, Evolutionary Intelligence (2019) 1–17.

[29] M. Habib, I. Aljarah, H. Faris, S. Mirjalili, Multi-objective particle swarm optimization: Theory, literature review, and application in feature selection for medical diagnosis, in: Evolutionary Machine Learning Techniques, Springer, 2020, pp. 175–201.

[30] M. Sreedhar, S. A. N. Reddy, S. A. Chakra, T. S. Kumar, S. S. Reddy, B. V. Kumar, A review on advanced optimization algorithms in multidisciplinary applications, in: Recent Trends in Mechanical Engineering, Springer, 2020, pp. 745–755.

[31] J. Kennedy, R. C. Eberhart, A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, volume 5, IEEE, 1997, pp. 4104–4108.

[32] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, Swarm and Evolutionary Computation 9 (2013) 1–14.

[33] E. Loper, S. Bird, Nltk: The natural language toolkit, in: In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics, 2002, pp. 63–70.

[34] V. Kotu, B. Deshpande, Data Science: Concepts and Practice, Morgan Kaufmann, 2018.

[35] D. Kim, D. Seo, S. Cho, P. Kang, Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec, Information Sciences 477 (2019) 15–29.

[36] A. Dhar, N. S. Dash, K. Roy, Categorization of bangla web text documents based on tf-idf-icf text analysis scheme, in: Annual Convention of the Computer Society of India, Springer, 2018, pp. 477–484.

[37] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (2008) 2579–2605.

[38] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of machine learning research 5 (2004) 101–141.

[39] R. T. Marler, J. S. Arora, The weighted sum method for multi-objective optimization: new insights, Structural and multidisciplinary optimization 41 (2010) 853–862.

[40] H. Faris, M. Habib, I. Almomani, M. Eshtay, I. Aljarah, Optimizing extreme learning machines using chains of salps for efficient android ransomware detection, Applied Sciences 10 (2020) 3706.

[41] A. Liaw, M. Wiener, et al., Classification and regression by randomforest, R news 2 (2002) 18–22.

[42] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, M. Klein, Logistic regression, Springer, 2002.

[43] A. McCallum, K. Nigam, et al., A comparison of event models for naive bayes text classification, in: AAAI-98 workshop on learning for text categorization, volume 752, Citeseer, 1998, pp. 41–48.

[44] J. D. Rennie, L. Shih, J. Teevan, D. R. Karger, Tackling the poor assumptions of naive bayes text classifiers, in: Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 616–623.

[45] L. Bottou, Stochastic gradient descent tricks, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 421–436.

[46] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[47] G. Rätsch, T. Onoda, K.-R. Müller, Soft margins for adaboost, Machine learning 42 (2001) 287–320.

[48] S. A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Transactions on Systems, Man, and Cybernetics (1976) 325–327.

[49] M. Neuhäuser, Wilcoxon-Mann-Whitney Test, International Encyclopedia of Statistical Science, Springer Berlin Heidelberg, 2011. URL: https://doi.org/10.1007/978-3-642-04898-2_615.