Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406
International Journal of Advance Engineering and Research Development
Volume 5, Issue 07, July -2018

OPINION ANALYSIS OF TREATMENTS FROM ONLINE HEALTHCARE FORUMS FOR THE DISEASE PSORIASIS USING NLP AND ARTIFICIAL NEURAL NETWORK

Mamatha Balipa
Associate Professor
Department of Master of Computer Applications
NMAM Institute of Technology, Nitte

Dr. Balasubramani R
Professor
Department of Information Science and Engineering
NMAM Institute of Technology, Nitte

Abstract—*In this work, the comments gathered from online healthcare forums regarding the disease Psoriasis are classified as either giving information about the treatments that have worked and treatments that have not worked or comments that pose more questions. To do this NLP and Artificial Neural Network is utilized and the algorithm is found to be giving better accuracies than SVM and Nave Bayes.*

Index Terms—Keywords*: Artificial Neural Network, Synapses, Psoriasis, Online healthcare forums.*

I. Introduction

The disease Psoriasis is an auto immune disease. The solution for the disease is very rare. Many people have tried various medications and treatments. The treatments people have undergone are Allopathic, Homeopathy, Ayurveda, etc., People who have undergone different types of treatments have shared their experiences online on the web on various health-care forums. People who are interested in finding different types of treatments for the disease and their success rates online, have to wade through different types of comments. This work extracts the messages about successful treatments posted on online health. This work by no means recommends any medication or treatment. The work only sifts through the different types of treatments undergone by people for the disease and gives a consolidated output containing only the comments depicting the treatments that have worked or whose success rate is high. The messages or comments posted on healthcare forums are in English, hence to analyse the text, Natural Language Processing techniques are applied.

To classify the comments as solutions or non-solutions, ma-chine learning algorithms like Naive Bayes, Decision Trees, Support Vector Machine, and Logistic Regression were ex-plored. Nave Bayes provided an accuracy of 94/%, Decision Tree provided an accuracy of 88%, SVM provided an accuracy of 88%, and Logistic Regression provided an accuracy of almost 100%. In the current work carried out, Artificial Neural Network has been used to classify the comments and a mean score of 99% has been obtained.

II. Literature

The process of text classification involves the following steps. [2]

- Text pre-processing.
- Feature extraction.
- Developing a model by using various classification tech-niques.
- Training the model using training data.
- Testing the model using test data.

The common algorithms used for text classification are

- Naive Bayes
- Decision Trees
- K-Nearest Neighbour
- SVM

K-Nearest Neighbour and SVM algorithms were used by Nihar Ranjan et al[11] to classify text as belonging to the categories politics, sport and art. The authors investigate the utilization of Support Vector Machines (SVMs) for analyzing text data and recognize the suitability of using SVMs for the job.

Quoc Le et al, proposed a framework, Paragraph Vector which is an unsupervised framework for learning continuos and distributed vectors that represent chunks of text.

Max Entropy algoritm was used by Kaufmann [5] to identify similar sentences between any two languages using limited training data.

SVMs were used by Li and Li [8] to classify sentiment polarity. They proposed that opinion subjectivity and expresser credibility should also be considered for sentiment polarity analysis.

Moraes and Valiati [10] provided a comparison between SVM and Artificial Neural Networks for sentiment analysis performed on entire documents. They have also performed various experiments to evaluate the performanve of various supervised methods and concluded that ANN provided better results.

Yan and Bing [14] explored a graph-based approach incorporating a propagation approach to utilize the inside and outside features of a sentence.

Ko and Seo [6] proposed a method that utilizes categorized keywords and sentence similarity measures to categorize sentences in a document.

Xianghua and Guo [13] also used unsupervised methods to automatically identify sentiments and aspects discussed in reviews posted on Chinese social media. They used LDA model for topic detection in social reviews and applied sentiment analysis using a sliding window approach over the review text. The data set they used contained reviews extracted from blogs(2000-SINA) and a lexicon (300-SINA Hownet) was aslo used.

Victoria Bobicev [1] applied machine learning techniques for classifying sentiments in forum texts that are labelled with different sentiment labels and achieved an F-measure of 0.805.

III. About The Work

In this work, messages extracted from health forums are extracted and classified as treatments that have worked and comments that do not mention successful treatments. So the issue is to group a remark as an answer for the sickness Psoriasis or as not an answer. In this work, Artificial Neural Network algorithm is applied for classification. Two layers of neurons (1 concealed layer) and a "pack of words" way are utilized to deal with arranging the training data. Classification of text can be achieved using various techniques. WalaaMedhat et al. Some of them are:

- Pattern matching
- Algorithms
- Neural nets.

Though the algorithmic approach like Naive Bayes, SVM and Logistic Regression are effective in classifying text, they have their drawbacks. The algorithms create a score as opposed to probability. We need a probability to overlook predictions under a certain threshold. The algorithms get trained about comments belonging to a class. But they do not learn about the text that do not belong to the class. Sometimes it is important to identify text that do not belong to the class.

The algorithms are sometimes forced to adjust the scores rel-ative to the class size due to incorrect classification scores got by large training sets that belong to a particular class. Similarly as with Naive Bayes algorithm, the classifier developed isn't endeavouring to comprehend the meaning of a sentence, it's attempting to classify it.

Steps involved in the work carried out:

1) Information retrieval
2) Prepare the training data
3) Develop the ANN Algorithm
4) Test the results
5) Tune the model
6) Iterate
7) Abstract

A. Information retrieval:

Information retrieval was performed using crawlers developed by the authors to extract messages from sources like psoriasis-association.org.uk, healingwell.com, MedHelp.org [4] and HealthBoards.com [4]. The search engine to do the same was developed using JSoup API[3], a Java HTML parser library and Apache Lucene[9]. About 2000 posts were collected from psoriasis-association.org.uk, healingwell.com, MedHelp.org [4] and HealthBoards.com [4]. The Text processing part of the system was used to extract the text messages from the document collection. The text extracted was pre-processed and then transformed into a form which can then be used by the information extraction part of the system. The information extraction part of the system uses a pipeline of NLP techniques to process the text.

Since the text is in English language, the concepts of Natural language processing is used in developing the classifier. Python and NLTK APIs have been used in the work. The comments are first divided into training and test text. The training text are manually labelled as belonging to pos and neg classes. The comments labelled as pos are comments that specify a treatment that has worked. Comments labelled as neg are those comments that may either be another query regarding the disease or a discussion about a treatment or medicine that has not worked. Healingwell.com, MedHelp.org [4] and HealthBoards.com.com [4] are health forums having multiple threads discussing issues regarding Psoriasis. Users discuss treatments they have undergone that have not worked, treatments that have worked, post questions, food that aggravate the symptoms or are the cause for the disease, food that give relief from the symptoms and all the issues pertaining to the disease. A message may consist of single sentence or multiple sentences. Since the messages are free flow of text in English, the text needs to be transformed into a form which can be processed as well as since the messages are in natural language, NLP techniques need to be applied to extract information from the text.

Since the data used is extracted from online healthcare forums, the system utilizes the features of Big Data. Healthcare message boards available online provide huge volume of latest and raw data which can be used to mine useful information. In the first step, that is information retrieval, a search engine was developed that will search and download all the pages from the web pertaining to the disease Psoriasis. To check the relevance of the page, a threshold value for the count of the number of times the word Psoriasis occurs in the page is maintained. The search engine was developed using Apache Lucene and Jsoup API.

Using JSoup API, individual comments from the online users in the page are extracted and a corpus of text containing the comments is created.

Topic detection: It is ensured that the topic of discussion in the page is about Psoriasis by using Latent Dirichlet Allocation (LDA) model. For this, first the text is normalized by eliminating stop words, punctuation symbols and lemmatizing the text. A term dictionary of the text and Document Term Matrix is created. Finally the topic of discussion in the text is arrived at by applying the LDA model on the document term matrix.

Feature extraction: The comments in the corpus are categorized into solutions and non solutions. An algorithm is developed and implemented to extract features that identifies a particular text as solution or treatment for the disease Psoriasis.

The algorithm works as follows.

First the corpus of comments is read. The corpus has comments that suggest solutions as well as comments that are not solutions. The comments present in the corpus are manually categorized as solutions and non-solutions. The comments are organized as:

1) Information retrieval
2) Prepare the training data
3) Develop the ANN Algorithm
4) Test the results
5) Tune the model
6) Iterate
7) Abstract

Each comment in the training set is word tokenized and added to documents list labelled with its class name. The stem or root word[12] of each word in the document is found and is converted to lowercase. Duplicate words are eliminated. List of classes are retrieved from the documents and unique classes from the class list is prepared. The training data is finally converted into a bag of words. Each comment in the training set is converted into an array of 0s and 1s against the array of unique words in the corpus. A comment text can belong to multiple classes or none. A two layered Neural Network is developed with the following functions.

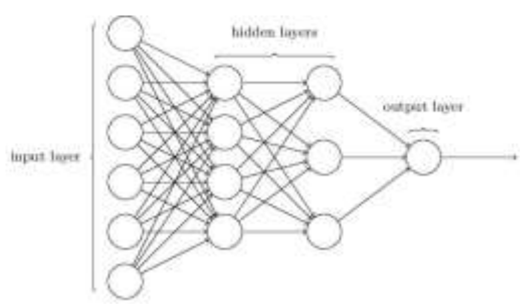

Fig. 1. A Neural Network [7]

A sigmoid function is utilized to normalize values and its subsidiary to gauge the error rate. Iterating and modifying until the point that the error rate is acceptably low. Also a bag-of-words function is implemented to transform an input sentence into an array of 0s and 1s. It matches precisely with the transformation for training data, which is crucial.

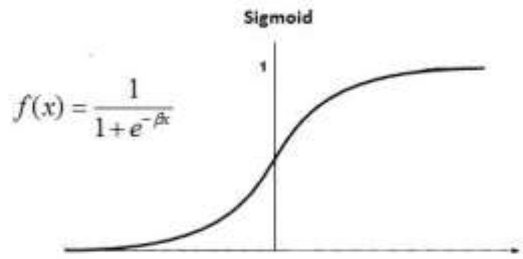

Fig. 2. Sigmoid Function [7]

A bag-of-words function is implemented to transform a comment into an array of 0s and 1s. This is also to transform the training data.

First the sigmoid nonlinearity is computed. Then the output of the sigmoid function is converted to its derivative. The bag-of-words array for each comment or message is derived. That is, 0 or 1 for each word in the bag that exists for in the comment. The input layer is the bag-of-words. The input layer and the hidden layer is multiplied and the output layer is got.

A neural network training function is developed to create synaptic weights. In this function ten hidden neurons are created. Gradient descent(alpha) value is assigned 1, epochs are 50000, dropout is False, Dropout percent is 0.5. Seed is 1 and last mean error is assigned 1. The weights are randomly initialized with mean 0. The synaptic weights are found, each time checking the error rates for every 10000 iterations.

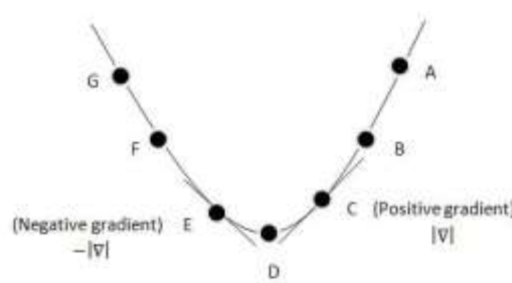

Fig. 3. Gradient Descent [7]

Once the synaptic weights are ready the neural network model can be built. The Gradient descent parameter, alpha helps in finding the lowest error rate.

$$synapse_0 + = alpha * synapse_0 weight_u pdate \qquad (1)$$

Twenty neurons are used in the hidden layer which can be adjusted. The parameters will vary depending on the dimensions and shape of the training data. They are tuned down to ˜10ˆ-3 as a reasonable error rate. The synaptic weights are saved into a file which is the model created.

Once the synapse weights have been calculated, the function

to classify the comments as solutions or non-solutions is created. The probability of a comments belonging to either of the classes can now be generated. The error threshold is assigned as 0.2. The synapses stored in the file is loaded and used for classification in the function.
The neural net learns from non-matching word also. A low-probability classification is easily shown by providing a sentence where a (common word).

1) Results:
  a) Synapses::

Synapses:
{
    "classes": [
        "pos",
        "neg"
    ],
    "datetime": "2018-02-05 20:11",
    "synapse0": [
        [
            -0.1533879806767439,
            0.4508848253200487,
            -1.0791910906571138,
            -0.3804974258163726,
            -0.7296908152980215,
            -0.7166050585739692,
            -0.8146301842971215,
            -0.2287242136622027,
            -0.23267863331487035,
            0.1086202069374877,
            0.04946424122190201,
            0.3340422000720627,
            -0.5475810032715567,
            0.7118980230440306,
            -0.9545622528726294,
            0.3965402495945006,
            0.0021039154494169074,
            0.1487213983568592,
            -0.7553923271802865,
            -0.6092988474772373
        ],
        [
            0.6243567844870782,
            0.9391315661231261,
            -0.3733547875508656,
            0.33519462246947546,
            0.7321914883222622,
            0.7930640555252656,
            -0.8654567039180125,
            -0.9179673441020602,
            -0.6652903648225592,
            0.7567964086619711,
            -0.8032821625293427,
            -0.16653833223704517,
            0.9159748794186824,
            0.10692155103234775,
            0.37229379679863767,
            -0.36777847953189907,
            0.39720990953212537,
            0.6122100559560409,
            -0.9604171372338728,
            0.48968774746767263
        ],
so on
[
            0.7944451980725279,
            -0.7013638529565499
        ],
        [
            2.6237432216892884,
            -2.82548750249371
        ],
        [
            -0.9847604473406809,
            1.0417863160100742
        ],
        [
            1.0552545180942436,
            -1.031175716056142
        ],
        [
            1.762650587650318,
            -1.7805827741390097
        ],
        [
            -3.48668840963923,
            3.4379453057638965
        ],
        [
            0.6372158664044987,
            -0.6019991460120345
        ],
        [
            -1.9099577599275783,
            1.7587393569050014
        ]
    ],

  b) The Unique Stemmed Words::

"words": [
        "her",
        "treat/",
        "hvac",
        "caus",
        "greek",
        "prom",
        "an",
        "through",
        "lip",

"dab",
"really-everyone",
"want",
"pres",
"although",
"popato",
"wat",
"get",
"it..",
"flannel",
"lax",
"tingl",
"button",
"be",
"prescrib",
"learn",
":",
"min",
"inst",
"gold",
"sticky",
"frenzy",
"biggest",
"thos",
"fight",
"hel",
"med",
"chair",
"acid",
"food",
"avocado",
"addit",
"sel",
"money",
"proof",
"irrevers",
"elm",
"job",
"without",
"appetit",
"bout",
"prev",
"concern",
"holland",
"decid",
"etc..",
"concret",
"subtl",
"pak",
"walmart",
"unit",
"ser",
"intestin",
"milit",
"easy",
"shy",
"might",
",",
"n’t",
"go",
"countless",
"impair",
"becom",
"shut",
"feet.i",
"nystatin",
"powd",
"champix",
"for",
"bandaid",
"sid",
"pung",
"today",
"light",
"avoid",
"task",
"marvel",
"u.",
"altogeth",
"send",
"larg",
"broth",
"rash",
"soc",
"granul",
"veg",
"branch",
"p.",
"inflam",
"rath",
"forev",
"coff",
"4-8",
"cos",
"was",
"bath",
"impact",
"greet",
"overact",
"success",
"or",
"swe",
"energet",
"find",
"speak",
"homeopath",
"becaus",
"incid",
"new",
"norm",
"yes-",
"goe",

"season",
"legum",
"spicy",
"gut",
"mod",
"built-up-toxins",
".very",
"erupt",
"numb..",
"feet",
"contract",
"set",
"brown",
"voer",
"test",
"tisl",
"allerg",
"counsel",
"switch",
"would",
"recovery",
"splash",
"suspect",
"happy",
"fought",
"tiny",
"discuss",
"hor",
"herbel",
"anyway",
"a",
"abnorm",
"combin",
"respond",
"expery",
"drop",
"ure",
"quest",
"ey",
"blow",
"stretch",
"somewh",
"drink",

and so on
]
}

c) Words in each comment converted to an array of 0 and 1::

[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,], and so on,]

d) Output:: [[1, 0], [0, 1], [0, 1]]

e) Training with 20 neurons, alpha:0.1, dropout:False :

delta after 10000 iterations:0.0019165701 delta after 20000 iterations:0.0013236611 delta after 30000 iterations:0.0010646909 delta after 40000 iterations:0.0009164618 delta after 50000 iterations:0.0008166112 delta after 60000 iterations:0.0007433369 delta after 70000 iterations:0.0006865893 delta after 80000 iterations:0.0006409592 delta after 90000 iterations:0.0006032341 delta after 100000 iterations:0.000571368

saved synapses to: synapses.json

processing time: 319.7013988494873 seconds

f) Classification of some of the samples ::

I flared up out of nowhere on my legs a few weeks ago. It's also on my back, across my chest, arms, neck and even forehead :(My back is the worst.

They started tiny and got a little bigger since then. I saw my primary care and am being referred to a dermatologist for it. I've had some spots on my apper back and would get dry scalp for years, but always kept under control with over the counter hydrocortisone and sal acid shampoo. It's never been this bad for me.

My Dad had psoriasis in the past, as does my older half sister,
I'm just waiting on the dermatologist for the official diagnosis.

classification: [['neg', 0.99956527046916643]]

I keep getting red patches without flaky skin on my forehead and around eyebrows, nose Really starting to annoy me Also got very short hair and it on the side of my head and very noticeable Any one recommend and thing to use on the face??

classification: [['neg', 0.9989025807386066]]

Agree with you about Sorion, can't believe the quick results. Wish it were a bit cheaper though.

classification: [['pos', 0.99870265704942318]]

Hi Alex, I get exactly the same thing. The only thing I've really found that clears it quickly is Elocon. I know it's a fairly potent steroid and my GP does warn me about it, but I'm always careful never to use it more than a day or two at a time and then

won't use it again for at least a week. Used like that it doesn't seem to cause any problems.

classification: [['pos', 0.99921327091222456]]

Sorry Anna
but your second dr was right. The steroid in Dovobet is far too strong to use on the face - sadly that's probably why it's been working.

It is ok to use mild steroids on the face for short periods so it might be worth asking about an alternative.

classification: [['neg', 0.99935327495487347]]

The mean of the accuracies is 0.99.

IV. Performance Comparison

First, we compare the performance of the proposed technique with other standard techniques like Naive Bayes, SVM, Decision Tree, etc.,

A. Performance Analysis of Classification Techniques

1) Naive Bayes classifier: Naive Bayes classifier provides an accuracy of 94.1176470588

TABLE I
Naive Bayes most informative features

contains(clear) = True	pos : neg = 7.4 : 1.0
contains(quickly) = False	neg : pos = 4.8 : 1.0
contains(believe) = False	neg : pos = 4.8 : 1.0
contains(appeared) = False	neg : pos = 4.8 : 1.0
contains(soothing) = False	neg : pos = 4.8 : 1.0
contains(great) = False	neg : pos = 4.8 : 1.0
contains(purely) = False	neg : pos = 4.8 : 1.0
contains(clear) = False	neg : pos = 4.8 : 1.0
contains(cannot believe)= False	neg : pos = 4.8 : 1.0
contains(grateful) = False	neg : pos = 4.8 : 1.0

TABLE II
Confusion Matrix for Naive Bayes

	Neg	Pos
Neg	4	1
Pos	0	12

TABLE III
Precision, Recall, F Measure for Naive Bayes

Label	Precision	Recall	F Measure
Neg	1.0	0.8	0.888888888889
Pos	0.923076923077	1.0	0.96

2) Decision Tree classifier: Decision Tree classifier provides an accuracy of 88.2352941176

TABLE IV
Confusion Matrix for Decision Tree

	Neg	Pos
Neg	1.0	2
Pos	0	14

TABLE V
Precision, Recall, F Measure for Decision Tree

Label	Precision	Recall	F Measure
Neg	1.0	0.333333333333	0.5
Pos	0.875	1.0	0.933333333333

3) SVM classifier: SVM classifier provides an accuracy of 88.2352941176

TABLE VI
Confusion Matrix for SVM

	Neg	Pos
Neg	10	1
Pos	1	5

TABLE VII
Precision, Recall, F Measure for SVM

Label	Precision	Recall	F Measure
Neg	0.909090909091	0.909090909091	0.909090909091
Pos	0.833333333333	0.833333333333	0.833333333333

4) SVC classifier: SVC classifier provides an accuracy of 52.94117647058824

TABLE VIII
Confusion Matrix for SVC

	Neg	Pos
Neg	3	4
Pos	4	6

TABLE IX
Precision, Recall, F Measure for SVC

Label	Precision	Recall	F Measure
Neg	0.429	0.429	0.429
Pos	0.6	0.6	0.6

5) LinearSVC classifier: LinearSVC classifier provides an accuracy of 82.35294117647058

TABLE X
Confusion Matrix for LinearSVC

	Neg	Pos
Neg	9	0
Pos	3	5

TABLE XI
Precision, Recall, F Measure for LinearSVC

Label	Precision	Recall	F Measure
Neg	0.75	1.0	0.8571428571428571
Pos	1.0	0.625	0.7692307692307693

6) NuSVC classifier: NuSVC classifier provides an accuracy of 70.58823529411765

TABLE XII
CONFUSION MATRIX FOR NUSVC

	Neg	Pos
Neg	6	2
Pos	3	6

TABLE XIII
PRECISION, RECALL, F MEASURE FOR NUSVC

Label	Precision	Recall	F Measure
Neg	0.67	0.75	0.71
Pos	0.75	0.67	0.71

The above empirical study shows that among the techniques used to classify messages as opinions stating successful treatments or other types of messages, the artificial neural network with an accuracy of 99% provides the better accuracy.

V. LIMITATIONS

The limitations of this approach is that, since the messages are posted by average users, there may be noise, inaccurate and exaggerated information with spelling mistakes. Mining such information may lead to false positives. But the volume of the data may help in solving this problem. Repeatedly occurring treatments can be considered as true positives. Some of the messages that are posted may be to promote certain drugs or products. So further investigation by medical experts may be required.

VI. CONCLUSION

In this paper the author has extracted messages from health forums and classified them as treatments that have worked and treatments that have not worked for the disease Psoriasis. For classification Artificial Neural Network(ANN) algorithm was applied. An accuracy of 0.99 was achieved. ANN is fater than other algorithms like SVM, Naive Bayes, Decision tree and accuracy is also higher than the other algorithms.

VII. CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

[1] Victoria Bobicev. Text classification: the case of multiple labels. In Communications (COMM), 2016 International Conference on, pages 39–42. IEEE, 2016.
[2] Mita K Dalal and Mukesh A Zaveri. Automatic text classification: a technical review. International Journal of Computer Applications, 28(2):37–40, 2011.
[3] Jonathan Hedley et al. jsoup: Java html parser, 2015. Website (https://jsoup. org/).
[4] Hyeju Jang, Sa-Kwang Song, and Sung-Hyon Myaeng. Text mining for medical documents using a hidden markov model. In AIRS, pages 553–559. Springer, 2006.
[5] Max Kaufmann. Jmaxalign: A maximum entropy parallel sentence alignment tool. Proceedings of COLING 2012: Demonstration Papers, pages 277–288, 2012.
[6] Youngjoong Ko and Jungyun Seo. Automatic text categorization by unsupervised learning. In Proceedings of the 18th conference on Computational linguistics-Volume 1, pages 453–459. Association for Computational Linguistics, 2000.
[7] Machine Learnings. Text classification using neural networks, 1999.
[8] Yung-Ming Li and Tsung-Ying Li. Deriving market intelligence from microblogs. Decision Support Systems, 55(1):206–217, 2013.
[9] Apache Lucene. Apache lucene core, 2013.
[10] Rodrigo Moraes, JoaO˜ Francisco Valiati, and Wilson P GaviaO˜ Neto. Document-level sentiment classification: An empirical comparison be-tween svm and ann. Expert Systems with Applications, 40(2):621–633, 2013.
[11] Nihar Ranjan, Abhishek Gupta, Ishwari Dhumale, Payal Gogawale, and Rugved Gramopadhye. Full length review article. 2015.
[12] V. G Vinod Vydiswaran, Qiaozhu Mei, David A. Hanauer, and Kai Zheng. Mining consumer health vocabulary from community-generated text. AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium, 2014:1150–1159, 2014.
[13] Fu Xianghua, Liu Guo, Guo Yanyan, and Wang Zhiqiang. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. Knowledge-Based Systems, 37:186–195, 2013.
[14] Zhao Yan-Yan, Qin Bing, and Liu Ting. Integrating intra-and inter-document evidences for improving sentence sentiment classification. Acta Automatica Sinica, 36(10):1417–1425, 2010.