Analysis of inter-concept dependencies in disease diagnostic cognitive maps using recurrent neural network and genetic algorithms in time series clinical data for targeted treatment

Aditya Duneja[1] · Thendral Puyalnithi[1] · Madhu Viswanatham Vankadara[1] · Naveen Chilamkurti[2]

Abstract

The knowledge of dependency of a particular factor on another is a very important aspect in the healthcare field. If we have a rough idea about the effect a prescribed drug has on the cure of a disease or sufficient information about certain symptoms of a disease being linked to each other, we can make an informed decision about its treatment over a period of time. This paper proposes a method which makes use of a special kind of recurrent neural network (RNN) known as long short term memory network (LSTM) to make predictions for time series data. Genetic algorithms are also incorporated to identify the most important concepts affecting a patient over that period of time. The output of the LSTM network is in the form of binary strings and is utilized to generate a fuzzy cognitive map (FCM) for the same and a novel method is proposed to find the values of the interdependencies between various concepts, an approach which can be applied in clinical decision support systems. This method makes use of the weight matrices obtained after training the neural network. It is shown to be an improvement over the previous work done in this domain. The proposed method was tested with various clinical datasets and results were obtained for the same.

1 Introduction

1.1 Cognitive maps

Cognitive maps are the representations of physical concepts formed in our minds. Humans as well as some animals (for example: rats) use these maps to visualize certain locations, situations which are present in their environment.

✉ Naveen Chilamkurti
N.Chilamkurti@latrobe.edu.au

Aditya Duneja
adityaduneja27@gmail.com

Thendral Puyalnithi
thendral.p@vit.ac.in

Madhu Viswanatham Vankadara
vmadhuviswanatham@vit.ac.in

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

[2] La Trobe University, Melbourne, Australia

Cognitive maps actually are a way to represent knowledge about a certain field in a conceptual manner. Cognitive maps are used in psychology, healthcare, astronomy, energy and various other fields. There are other ways to model knowledge about a field, mainly artificial neural networks (ANN). They are mainly used to find a relation between the inputs and outputs using the hidden nodes as well. In this case the neuron between the input and output nodes is like a black box. It computes the function between the input and output but there is no real physical meaning behind it. That is the main difference between cognitive maps and artificial neural networks. Cognitive maps are also used to model causal relationships between certain concepts. For example, if some factors affect precipitation like evaporation, transpiration etc., this can be illustrated through a cognitive map. Now the value which determines the magnitude of the cause of one thing or another can be binary or numerical. For example, if a factor does not affect precipitation it can be assigned with 0 to it could be assigned a suitable number, say 0.8. So, the types of maps where fuzzy values are used are known as fuzzy cognitive maps (FCM).

1.1.1 Concepts

Concepts in fuzzy cognitive maps are the important features out of many present in a dataset (medical dataset in this case). For example, a particular hormone's secretion depends on many other organs functioning correctly or the amount of red blood cells (RBC) in the body depends on many other factors and the magnitude of their dependency varies from patient to patient. For a particular patient, the reasons for getting a particular disease would be dependent on different factors than another patient. The reason for that maybe genetic in nature but has not been discovered yet. In this paper, some work is done to devise a method to identify the important concepts from a clinical data.

1.2 Time series data

In the medical field, lots of data which is present is mostly time series. Say a patient is admitted in the ICU due to some illness. Now his or health will constantly be monitored over a period of time and each record of its health at an instance (heart rate, hemoglobin, RBCs etc.) will depend on the values for the previous instances. So, if one has to make the prediction of the patient's state at a particular time, he has to have the knowledge of his health from the start of its treatment.

1.3 Recurrent neural networks and LSTM

That is why recurrent neural networks are used instead of simple artificial neural networks to predict time series data. In recurrent neural networks (RNNs), some part of the previous input is supplied along with the current input to process the output at that given moment. There are 3 sets of weights (U, V and W) for each time step and the hidden layers between the 2 time steps are connected to each other as shown in Fig. 1. The U matrix of weights is between the input and the hidden layer, V is between the successive hidden layers and W is between the hidden layer and the output node. This helps in taking care of data which spans over a certain time period and the values of an instance at a point of time depends on the values of the previous ones. But there is a drawback in recurrent neural networks which prevents it from learning dependencies over a long period of time. Wider the time gap between the current input and its dependency, harder it is for the RNN to learn it.

This is because of the vanishing/exploding gradient problem. While computing the error gradient of an input which is far away in time as compared to the current input, taking the gradient for all the subsequent inputs reduces the value to almost 0(vanishing) or infinity (exploding). The LSTM (long short term memory) solves this problem by employing a more complex structure of its node (repeating structure)
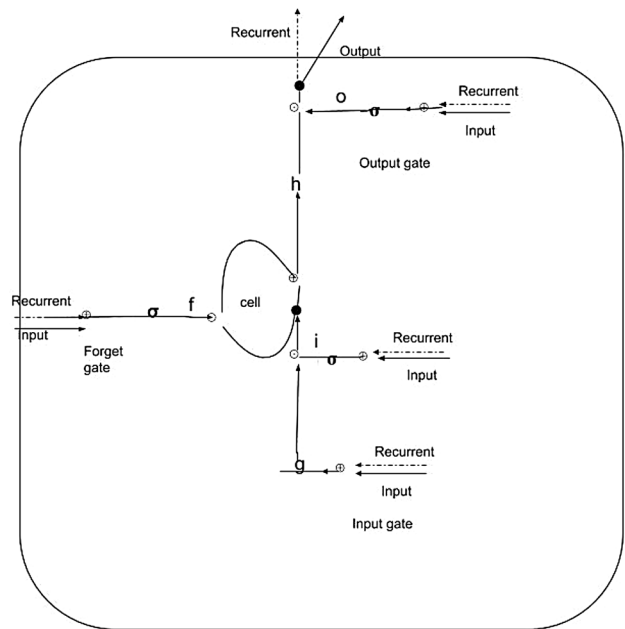

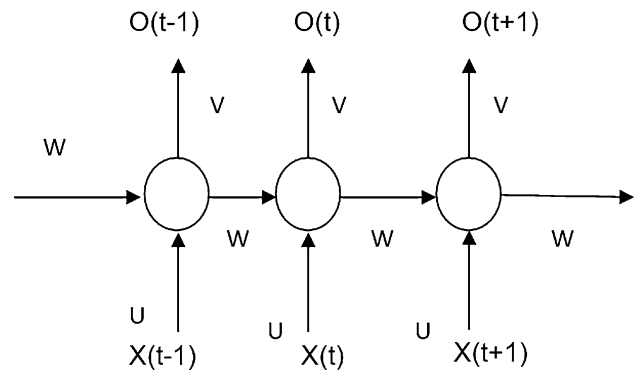**Fig. 1** Recurrent neural networks

Fig. 2 LSTM block

unlike RNN (whose node is just like that of a simple neural network). The structure of LSTM is as follows: LSTM contains a cell state (as shown in Fig. 2) which refers to the long term memory. Small operations are done on this memory like adding or removing information with the help of gates. Gates help in selective flow of information to the long term memory (cell state). It has a neural net layer and a multiplication operator. This selective flow of information is decided by the activation function, namely the sigmoid function which restricts the value between 0 and 1, 0 being to stop all information from going through and 1 to let all the information being added to the cell state. There are 3 such gates in LSTM: forget gate, input gate and output gate. The *forget* gate is responsible for the information which is to be removed from the cell state. The sigmoid activation

function decides how much of it remains in the cell state where 1 means to leave all the information there and 0 means to remove all of it. The *input* gate on the other hand, decides what information we have to add to the cell state. This takes place in 2 steps. In the first step, a vector of candidate values is created using *tan h* activation function, which decides the values which will be added to the cell state. The second step has a sigmoid function which decides which values already present in that state will get updated. The gate which is responsible to decide the output values is known as the *output* gate. Output values are computed with the help of tanh activation function which keeps the values between − 1 and 1. These values, in turn are multiplied with the sigmoid function which decide which values should actually be output. There are other variants of LSTM as well. Given below is the architecture of the long short term memory (LSTM).

1.4 Genetic algorithms

Genetic algorithms are used for optimization of problems. There are a few common steps involved in the implementation of genetic algorithms and have been used in this paper. The first step is the initialization of a population of candidate solutions which is done randomly. Then, every candidate solution (element in the population) is evaluated by a fitness function. This step is succeeded by the selection process where only a certain number of candidate solutions are selected on the basis of their performance when passed through the fitness function. The third step involves mating which involves the exchange of some set of bits which results in some new off-springs with desirable qualities through processes like crossover. For some variation in the off-springs, processes like mutation take place which take a random bit and flip it (from 0 to 1 or vice-versa). One of the drawbacks of genetic algorithms is the efficiency, which is really low because very candidate solution is evaluated separately.

2 Previous works

LSTM network is a type of recurrent neural network which is generally used for time series or sequential predictions was proposed in the late 1990s by two German professors as an improvement to the original RNNs (recurrent neural networks). The architecture which was earlier proposed could store short amounts of memory for short periods of time. The method explained above was improved by using LSTM which was formally documented (Hochreiter and Schmidhuber 1997), as it is ideal for making predictions for time series data. The reason behind using the LSTM network is for long term predictions as it eliminates the exploding/vanishing gradient problem. LSTM, can store short amounts of memory for longer periods of time by enforcing constant error flow through internal states. LSTM has been used in various fields by researchers, mainly in the speech and text recognition one. LSTM is applied in the energy field (Qing and Niu 2018). Prediction of solar irradiance is essential for minimizing energy costs. LSTM was used for the same by using weather forecasting data taking into account the dependence between consecutive hours of the same day. Combination of LSTMs, convolutional neural networks (CNN) and the deep neural networks (DNN) which are used for multilingual speech recognition for the prediction and correction (PAC) architecture. Vanishing gradient problem in deep neural networks by combining max-out neurons along with LSTMs and CNN. This, in turn has been used to improve the performance of the speech recognition model (Cai and Liu 2016; Wang and Wang 2017). LSTM is applied in the field of Natural Language Processing by modeling a word generator (He et al. 2017). The work enhanced the domain of image captioning by using parts of speech guidance to improve its performance.

The field of image processing is combined with that of natural language processing with the help of LSTM (Chen et al. 2017). The work converted car images to vectors of some definite size and also their corresponding descriptions into same size vectors. These image and description vector pairs are then fed to the LSTM network for training. Simple RNNs and LSTM have been used as dual encoders in the field of natural language processing to deep learn an Indonesian conversation (Chowanda and Chowanda 2017). A novel attribute reduction algorithm is proposed which is based on artificial fish swarm algorithm (AFSA) (Luan et al. 2016). Normal and Cauchy distribution functions, crossover and mutation operators are used to overcome the problem of slow convergence rate of the original algorithm. An attribute reduction algorithm, namely MCMAR (multi agent consensus MapReduce based attribute reduction algorithm) is proposed to involve a modified version of the particle swarm optimization (PSO) and is tested with a lot of medical datasets (Ding et al. 2018). As explained in the above section, the main objective is to illustrate the cause effect relationship of various factors in the form of a fuzzy cognitive map. Fuzzy cognitive maps is a soft computing methodology used for modeling cause-effect relationships between various factors illustrated (Stach et al. 2008; Froelich et al. 2012). The main difference between a simple neural network and a Fuzzy cognitive map is that of semantics. In the previously referred work, the fuzzy cognitive map is designed based on prior knowledge of the concepts. It may happen that we do not know the most important factors which can determine the cause and effect relationship. To rectify this, all the attributes are initially used for predictions. Then evolutionary algorithms are used to optimize the number of attributes that would be required to roughly get the same output (Luan et al.

2016; Ding et al. 2018). Another drawback of this work is the weight predictions method. The weights (direct dependency) are computed from scratch using real coded genetic algorithms (Froelich et al. 2012). The drawback for this method is the dependency of the solution (the weight vector) on the initial population selected. There might be a better solution, but due to the initial population selected, that solution might be unreachable from there. To rectify this problem, a novel method is proposed to find the weights (direct dependency).

FCMs have been modeled to build an energy management system which optimizes the performance of the autonomous polygeneration micrograms (APM) which are used to supply power, water and fuel for transportation, especially in remote areas (Kyriakarakos et al. 2012). FCM is used in the field of healthcare by modeling the psychosocial determinants of obesity through their work, experts could find values by employing different techniques, which helped in giving an exact measure of the people who were affected by obesity (Giabbanelli et al. 2012). FCM is applied in the field of astronomy as well (Furfaro et al. 2012). The proposed work uses evolutionary fuzzy cognitive map (EFCM) to find sites on Venus and Titan with the maximum chance of scientific discovery. The states of the EFCM keep changing in response to the real time data during descent till they find the best solution. FCMs have also found large applicability in many other diverse scientific areas (Froelich and Wakuliczdeja 2010; Glykas 2010; Papageorgiou et al. 2010). A novel version of LSTM for question classification is proposed and in that the authors have highlighted the shortcoming of the recurrent neural networks (RNNs) which is the inability to learn long term dependencies (Xia et al. 2018). The same problem is highlighted in this paper. LSTM have been applied to recognize the moods based on the data acquired by a wearable device which tracks the environment of the person at all times (Son 2017). LSTM is also proposed in an algorithm which predict accidents at nuclear power plants (Yang and Kim 2018). The work show that that the best performance of a neural network on tie series data was that of LSTM. This has been shown in this paper as well. Further LSTMs have been used in artificial neural networks to predict the abundance of malaria in some places over a period of time (Thakur and Dharavath 2018).The prediction could have been improved if the recurrent neural networks were used instead because data of a particular year depends on the previous one. Deep convolutional networks are used to classify mass lesions (Chougrad et al. 2018). This shows the application of deep neural networks in the medical field, a concept illustrated in this paper as well. LSTM is used to help to improve the gait of a person which may be affected by neuro degenerative diseases (Zhao et al. 2018). The work shows the prediction mechanism of LSTM exceeds that of others. Long term dependencies between concepts are predicted describe a using a novel way using different variants of the recurrent neural networks, namely multi recurrent network (MRN) which makes use of the sluggish state memory for learning long term dependencies (Tepper et al. 2016). The authors maintain in the end that LSTM method was still the preferred method to their own. In one of the researches, the authors have devised a way to recognize irregular entities in biomedical text by using bi directional LSTMs (Li et al. 2017). Work on identifying entity recognition in the medical domain and have shown that the bidirectional LSTM model has outperformed other machine learning algorithms like support vector machines and conditional random fields (Unanue et al. 2017). All the above mentioned works showed that many variants of the LSTM are being used in the medical domain. So, the use of LSTM to remember the pattern unfolding especially in time series data is very much justified. Since most the time series analysis works did not use feed-back model of analysis and so this work which is using LSTM model stands out prominently.

3 Proposed method

3.1 Flow chart of the proposed method

The flowchart in Fig. 3 is explained in this section. Assume that a dataset with m tuples and n attributes was taken. We had to identify the cause and effects with respect to the features present in the dataset. To achieve this goal, we took every attribute as an output (target) vector (effect) and the rest of them as causes and trained the LSTM network. For every attribute considered as a target, optimal number of attributes were found out using genetic algorithms. The process of genetic algorithms involved choosing a set of $n-1$ sized binary strings (where n are the total number of attributes). So, after training the LSTM with all the attributes included, the network was tested with different combinations of the attributes based on the population of binary strings. Binary strings denote the inclusion or exclusion of a certain attribute and the fitness function into which these strings are evaluated is inversely proportional to the error generated by the LSTM. In the end, we have n sets of binary strings denoting which attributes are the causes and which of them are effects. This map can be called the fuzzy cognitive map as the values will not be purely binary. These dependency values were found out using a novel approach. The approach which was chosen to find the dependency values was by matrix multiplication of the input-hidden layer weight matrix and hidden layer-output weight matrix. The reason behind this is that multiplication of the 2 or more matrices will lead to a measure of the upper bound for the change in output with respect to the input. Hence, if the contribution is very small, it can be said that the input is not important. The step by step process is explained in the following algorithm clearly.

Fig. 3 Flowchart of the proposed method
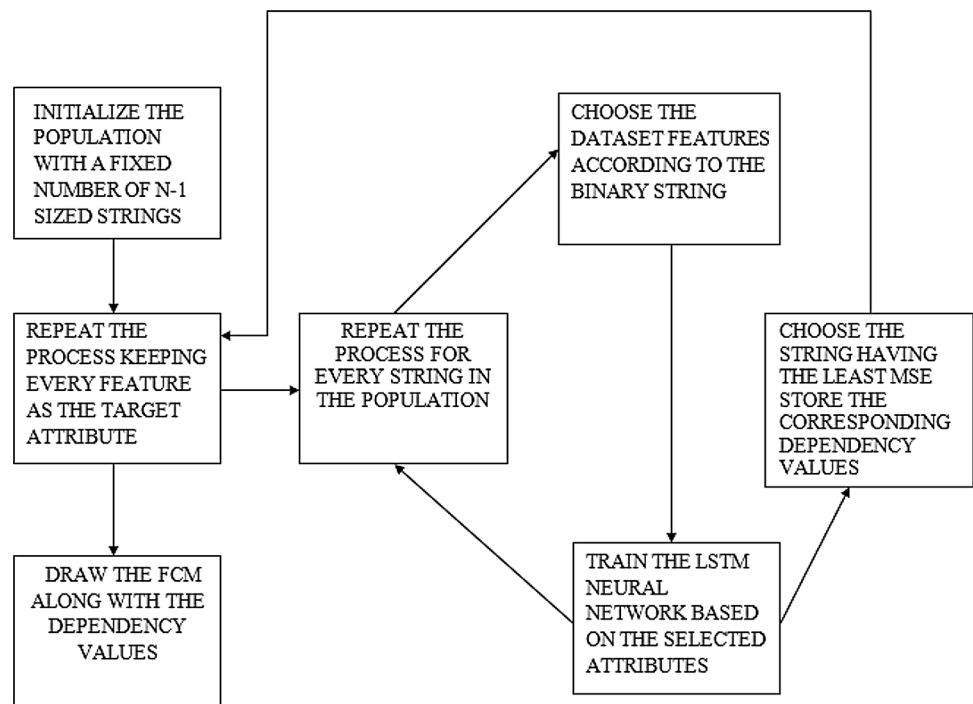

```
INPUT:  train_data=training set, test_data=test set, n=total number of features
OUTPUT: Set of strings and dependency values=dvec

1. Import datasets, python modules
2. Dataset=Normalize the dataset values
3. Population =  Initialize the population of strings
4.  train_data, test_data = Poplulation1, Population2. Divide the population into training and test sets.
5. for each attribute as the target feature:
        a. for each string in population:
                i) Train the LSTM neural network by adding layers
                ii) dvec =  Input weight vector ∗ hidden weight vector ∗ output weight vectors
                iii)  Pop the layers
        b. Store the strings and corresponding errors as key value pairs.
        c. Choose the string with the least error.
6. Draw the map along with the dependency values.
```

4 Implementation and result analysis

4.1 Dataset details

Results of the work are given and analyzed in this section. In Table 1, the details of the datasets used are given, in Table 2 the implementation parameters are given and finally the results obtained are given in Table 3. The datasets are taken from the University of California Irvine (UCI) machine learning repository and they are used as benchmark for verification of algorithms by machine learning research community.

Table 1 Dataset details

S. no.	Dataset	Instances	Attributes	Data type
1	Prostate cancer	8	6	Continuous
2	Epileptic seizure recognition	11,500	5	Continuous
3	EG eye state	14,980	8	Continuous

Table 2 General implementation parameters

S. no.	Implementation parameters	Values
1	Programming language used	Python
2	Important packages imported	Keras, numpy, random
3	Number of hidden layers	1
4	Number of hidden neurons	50
5	Number of output neurons	1
6	Number of input dimensions	Dependent on the number of features

Table 3 Strings and dependency values of the prostate cancer dataset

Dataset	Strings	Dependency values
Prostate cancer	11011	[0.706,1.431,0.354,0.319]
	00111	[0.249,0.336,0.798]
	00110	[0.731,0.152]
	11010	[0.208,0.349,−0.159]
	10000	[−0.303]
	10011	[0.414,0.562,0.238]

Table 4 Strings and dependency values of the epileptic seizure recognition dataset

Dataset	Strings	Dependency values
Epileptic seizure recognition	0111	[−0.479,0.937,0.722]
	0001	[0.997]
	0101	[0.418,0.827]
	0110	[−0.4228,0.303]
	0111	[0.545,−0.599,0.112]

Table 5 Strings and dependency values of the EEG eye state dataset

Dataset	Strings	Dependency values
EEG eye state	1111001	[−0.366,−0.584,0.439,0.049,0.762]
	0010110	[0.692,0.738,0.232]
	1101010	[1.107,0.73,−0.45,0.027]
	1101111	[−0.734,0.189,−1.03,0.449,1.31,0.59]
	0110101	[−0.59,−0.425,−0.597,0.271]
	1101010	[0.103,−0.991,−0.538,−0.067]
	0101001	[−0.303,−1.084,−0.474]
	0111010	[0.1,0.225,0.355,−0.528]

4.1.1 Prostate cancer dataset

This dataset was previously referred in some papers. It consists of *8 tuples*, indicating one tuple as a period of 1 quarter. It has *6 attributes*, all continuous and medically relevant. These attributes are Hct, WBC PSA free, PSA total, PSAf/PSAt and PAP respectively.

4.1.2 Epileptic seizure recognition dataset

This dataset is a pre-processed and re-structured/reshaped version of a very commonly used dataset featuring epileptic seizure detection. In the reshaped version, all the *11,500 tuples* have been considered but for the sake of simple illustration of the work carried out, only *5 attributes* have been considered out of the original number. The behavior of the algorithm proposed in the work would not differ for any number of attributes and this is the reason reshaping has been done to the data. All the attributes included for analysis are *continuous*.

4.1.3 EEG eye state dataset

This dataset is a pre-processed and reshaped version of the original dataset. In the reshaped version, the integer type attributes have been left out and only the *continuous* type have been included. All *14,980 tuples* have been considered and *8 attributes* out of the original number have been taken for analysis.

4.2 General implementation parameters

The general implementation parameters include the programming language used, important packages/modules imported, the type of neural network, number of hidden layers and neurons and the input and output dimensions. Details about these parameters are given below in the form of a table.

4.3 Results analysis and discussion

For every dataset, a certain number of strings (equal to the number of attributes of the dataset) are the output. Let n be the total number of attributes and let there be k nodes in the fuzzy cognitive map. So, the total number of strings would be n and size of each string should be n − 1. Number of nodes in the map would be equal to the number of strings, which is n. If we consider the collection of strings as a vector, then the vector at the ith (i being the index, 0 being the base) position is interpreted as follows: if i = 1, then the output 11001 would mean that the 1st attribute is the output and the rest are considered as input namely the 0th, 2nd, 3rd, 4th and 5th respectively. Now the positions which are '1' imply that those attributes are chosen and the rest are ignored. So, this implies that 0, 2 and 5th attributes respectively are taken into consideration.

These strings are the basis to draw the fuzzy cognitive map. The randomly generated genetic strings and the dependency values arrived at after applying the algorithm
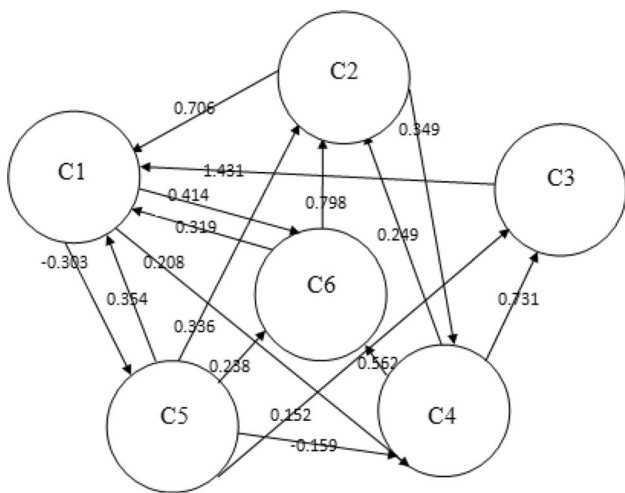
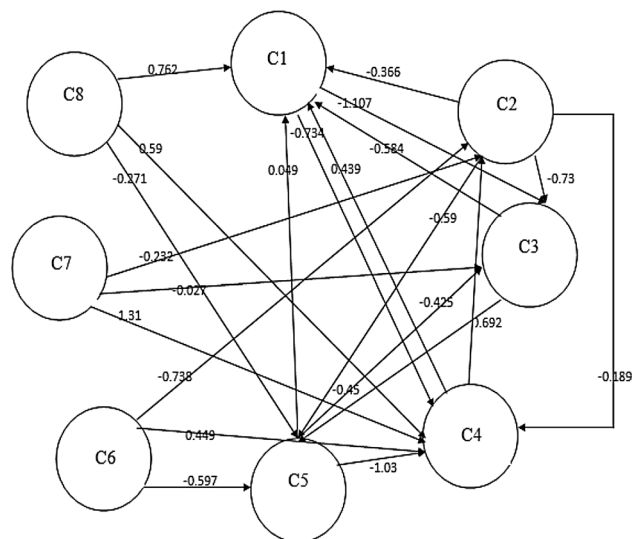**Fig. 4** Dependency relation cognitive map for the prostate cancer dataset

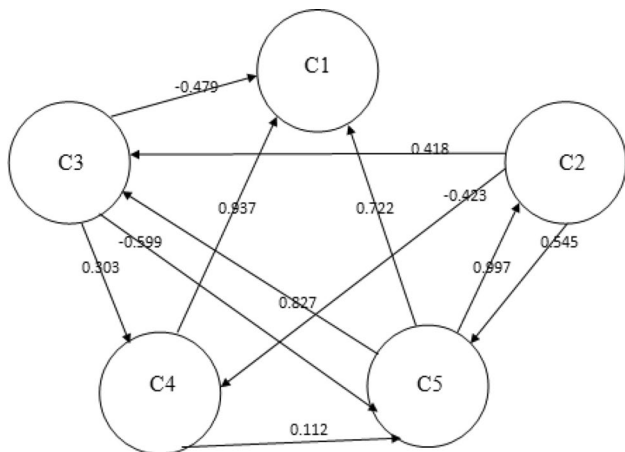**Fig. 6** Dependency relation cognitive map for the EEG eye state dataset

Fig. 5 Dependency relation cognitive map for the epileptic seizure recognition dataset

Table 6 Dependency values and nodes for prostate cancer dataset

Node	Dependency values and nodes
C1	[C2-0.706,C3-1.431,C5-0.354, C6-0.319]
C2	[C4-0.249,C5-0.336,C6-0.798]
C3	[C4-0.731,C5-0.152]
C4	[C1-0.208,C2-0.349,C5- −0.159]
C5	[C1- −0.303]
C6	[C1-0.414,C4-0.562,C5-0.238]

Table 7 Dependency values and nodes for epileptic seizure recognition dataset

Node	Dependency values and nodes
C1	[C3- −0.479,C4-0.937,C5-0.722]
C2	[C5-0.997]
C3	[C2-0.418,C5-0.827]
C4	[C2- −0.4228,C3-0.303]
C5	[C2-0.545,C3- −0.599,C4-0.112]

for prostate cancer, epileptic seizure recognition and EEG eye state data sets are shown in Tables 3, 4 and 5 respectively. These dependency values were computed as follows, the approach which was chosen to find the dependency values was by matrix multiplication of the input-hidden layer weight matrix and hidden layer-output weight matrix. The reason behind this is that multiplication of the 2 or more matrices will lead to a measure of the upper bound for the change in output with respect to the input. Hence, if the contribution is very small, it can be said that the input is not important. Let the number of hidden layers be h. Let the number of matrices to be multiplied be m. So, m=h+1. Let the vector containing the dependency values be known as *Dependency_Vec*. So,

$$Dependency\_Vec = Input\_hiddenlayer\_1 \times \prod_{i=1}^{h-1} hidden\_matrix(i) \times Output\_hiddenlayer\_h \qquad (1)$$

Table 8 Dependency values and nodes for EEG eye state dataset

Node	Dependency values and nodes
C1	[C2- − 0.366,C3- − 0.584, C4-0.439 C5-0.049,C8-0.762]
C2	[C4-0.692, C6- − 0.738, C7- − 0.232]
C3	[C1-1.107, C2-0.73,C5- − 0.45, C7-0.027]
C4	[C1- − 0.734, C2-0.189, C5- − 1.03, C6-0.449, C7-1.31,C8-0.59]
C5	[C2- − 0.59,C3- − 0.425 ,C6- − 0.597, C8- − 0.271]
C6	[C1-0.103,C2- − 0.991, C4- − 0.538 ,C7- − 0.067]
C7	[C2- − 0.303,C4- − 1.084,C8- − 0.474]
C8	[C2-0.1, C3-0.225, C4-0.355,C6- − 0.528]

where Input_*hiddenlayer_1* is the weight matrix between the input layer and the first hidden layer of the network, *Output_hiddenlayer_h* is the weight matrix between the last hidden layer and the output layer, *hidden_matrix (i)* is a function which gives the matrix between the hidden layers i and i + 1.

The fuzzy cognitive maps arrived at for prostate cancer, epileptic seizure recognition and EEG eye state data sets are shown Figs. 4, 5 and 6 respectively. The parameter dependency values for prostate cancer, epileptic seizure recognition and EEG eye state data are shown in Tables 6, 7 and 8 respectively.

5 Conclusion

The work is aimed at finding the causal relationship between different features of medical data set. The work aimed at finding dependency of one parameter with other parameters and also to what extent it is dependent, that is the strength of the dependency. The dependency network is represented using cognitive maps and the edge weights represent the strength of dependencies. Neural network has been used as base infrastructure and the weights are optimized using genetic algorithms, which are very good at exploration. Since the time series data is used, recurrent neural network has been taken into consideration. Thus the extent of dependence of a feature on another was found out by using the LSTM neural network. The work previously done in this field involved a lot of reliance on genetic algorithms which does not give as accurate values (weights) as compared to neural networks. A special type of recurrent neural networks which have been used in the field of healthcare previously was also used here to predict the dependency values and the results were tested on 3 medical time series datasets. It gave an improved result as compared to other methods.

This work has future scope in the medical field if merged with IOT. The generation of a fuzzy cognitive map can be merged with IOT to make an impact on the healthcare field. For example, sensors can be attached to the patient whose heart rate, blood pressure etc. is being calculated. If the heart rate goes up or down in an abnormal manner, the appropriate medicine can be injected into the patient's body and in controlled amounts as per the strength of the dependent parameters. The approach is personalized, meaning the strength of parameter dependencies varies from person to person, so if a patient's time series data is available, then the patient centric cognitive map can be arrived at which will be used for further course of treatment. The highlight of the work is providing person centric treatment. In modern medical practices, it has been found out that patients suffering from same disease needs to be treated differently, meaning each patient may show varying response to certain drugs and so patient centric targeted treatment is needed, that in turn needs a patient centric cognitive map. The work is aimed arriving at patient centric cognitive map using LSTM and genetic algorithms.

References

Cai M, Liu J (2016) Maxout neurons for deep convolutional and LSTM neural networks in speech recognition. Speech Commun 77:53–64. https://doi.org/10.1016/j.specom.2015.12.003

Chen L, He Y, Fan L (2017) Let the robot tell: describe car image with natural language via LSTM. Pattern Recogn Lett 98:75–82. https://doi.org/10.1016/j.patrec.2017.09.007

Chougrad H, Zouaki H, Alheyane O (2018) Deep convolutional neural networks for breast cancer screening. Comput Methods Programs Biomed 157:19–30. https://doi.org/10.1016/j.cmpb.2018.01.011

Chowanda A, Chowanda AD (2017) Recurrent neural network to deep learn conversation in Indonesian. Proc Comput Sci 116:579–586. https://doi.org/10.1016/j.procs.2017.10.078

Ding W, Lin CT, Chen S, Zhang X, Hu B (2018) Multiagent-consensus-MapReduce-based attribute reduction using co-evolutionary quantum PSO for big data applications. Neurocomputing 272:136–153. https://doi.org/10.1016/j.neucom.2017.06.059

Froelich W, Wakuliczdeja A (2010) Medical diagnosis support by the application of associational cognitive maps. Control Cybern 39:439–456

Froelich W, Papageorgiou EI, Samarinas M, Skriapas K (2012) Application of evolutionary fuzzy cognitive maps to the long-term prediction of prostate cancer. Appl Soft Comput 12(12):3810–3817. https://doi.org/10.1016/j.asoc.2012.02.005

Furfaro R, Fink W, Kargel JS (2012) Autonomous real-time landing site selection for Venus and Titan using evolutionary fuzzy cognitive maps. Appl Soft Comput 12(12):3825–3839. https://doi.org/10.1016/j.asoc.2012.01.014

Giabbanelli PJ, Torsney-Weir T, Mago VK (2012) A fuzzy cognitive map of the psychosocial determinants of obesity. Appl Soft Comput 12(12):3711–3724. https://doi.org/10.1016/j.asoc.2012.02.006

Glykas M (2010) Fuzzy cognitive maps: advances in theory, methodologies, tools and applications. In: Studies in fuzziness and soft computing, vol 247. Springer, Berlin

He X, Shi B, Bai X, Xia GS, Zhang Z, Dong W (2017) Image caption generation with part of speech guidance. Pattern Recognit Lett. https://doi.org/10.1016/j.patrec.2017.10.018 **(in press)**

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kyriakarakos G, Dounis AI, Arvanitis KG, Papadakis G (2012) A fuzzy cognitive maps–petri nets energy management system for autonomous polygeneration microgrids. Appl Soft Comput 12(12):3785–3797. https://doi.org/10.1016/j.asoc.2012.01.024

Li F, Zhang M, Tian B, Chen B, Fu G, Ji D (2017) Recognizing irregular entities in biomedical text via deep neural networks. Pattern Recogn Lett 105:105–113. https://doi.org/10.1016/j.patrec.2017.06.009

Luan XY, Li ZP, Liu TZ (2016) A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm. Neurocomputing 174:522–529. https://doi.org/10.1016/j.neucom.2015.06.090

Papageorgiou EI, Markinos AT, Gemtos TA (2010) Soft computing technique of fuzzy cognitive maps to connect yield defining parameters with yield in cotton crop production in central Greece as a basis for a decision support system for precision agriculture application. In: Fuzzy cognitive maps. Springer, Berlin, pp 325–362

Qing X, Niu Y (2018) Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. Energy 148:461–468. https://doi.org/10.1016/j.energy.2018.01.177

Son HH (2017) Toward a proposed framework for mood recognition using LSTM recurrent neuron network. Proc Comput Sci 109:1028–1034. https://doi.org/10.1016/j.procs.2017.05.378

Stach W, Kurgan LA, Pedrycz W (2008) Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. IEEE Trans Fuzzy Syst 16(1):61–72

Tepper JA, Shertil MS, Powell HM (2016) On the importance of sluggish state memory for learning long term dependency. Knowl Based Syst 96:104–114. https://doi.org/10.1016/j.knosys.2015.12.024

Thakur S, Dharavath R (2018) Artificial neural network based prediction of malaria abundances using big data: a knowledge capturing approach. Clin Epidemiol Glob Health. https://doi.org/10.1016/j.cegh.2018.03.001 **(in press)**

Unanue IJ, Borzeshi EZ, Piccardi M (2017) Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. J Biomed Inform 76:102–109. https://doi.org/10.1016/j.jbi.2017.11.007

Wang Y, Wang H (2017) Multilingual convolutional, long short-term memory, deep neural networks for low resource speech recognition. Proc Comput Sci 107:842–847. https://doi.org/10.1016/j.procs.2017.03.179

Xia W, Zhu W, Liao B, Chen M, Cai L, Huang L (2018) Novel architecture for long short-term memory used in question classification. Neurocomputing 299:20–31. https://doi.org/10.1016/j.neucom.2018.03.020

Yang J, Kim J (2018) An accident diagnosis algorithm using long short-term memory. Nucl Eng Technol 50(4):582–588. https://doi.org/10.1016/j.net.2018.03.010

Zhao A, Qi L, Dong J, Yu H (2018) Dual channel LSTM based multi-feature extraction in gait for diagnosis of neurodegenerative diseases. Knowl Based Syst 145:91–97. https://doi.org/10.1016/j.knosys.2018.01.004