

## MAXIMUM 2-SATISFIABILITY IN RADIAL BASIS FUNCTION NEURAL NETWORK

(Kebolehpuasan-2 Maksimum dalam Rangkaian Neural Fungsi Asas Jejari)

SHEHAB ABDULHABIB ALZAEEMI<sup>1</sup>, SARATHA SATHASIVAM<sup>1,\*</sup>, MOHD SHAREDUWAN  
MOHD KASHIMUDDIN<sup>1</sup> & MOHD. ASYRAF MANSOR<sup>2</sup>

### ABSTRACT

Maximum  $k$ -Satisfiability (MAX- $k$ SAT) is the logic to determine the maximum number of satisfied clauses. Correctly, this logic plays a prominent role in numerous applications as a combinatorial optimization logic. MAX2SAT is a case of MAX- $k$ SAT and is written in Conjunctive Normal Form (CNF) with two variables in each clause. This paper presents a new paradigm in using MAX2SAT by implementing in Radial Basis Function Neural Network (RBFNN). Hence, we restrict the analysis to MAX2SAT clauses. We utilize Dev C++ as the platform of training and testing our proposed algorithm. In this study, the effectiveness of RBFNN-MAX2SAT can be estimated by evaluating the proposed models with testing data sets. The results obtained are analysed using the ratio of satisfied clause (RSC), the root means square error (RMSE), and CPU time. The simulated results suggest that the proposed algorithm is effective in doing MAX2SAT logic programming by analysing the performance by obtaining lower Root Mean Square Error, high ratio of satisfied clauses and lesser CPU time.

*Keywords:* MAX- $k$ SAT; Conjunctive Normal Form; ratio of satisfied clause; combinatorial optimisation

### ABSTRAK

Kebolehpuasan- $k$  maksimum (MAX- $k$ SAT) adalah logik untuk menentukan jumlah maksimum bagi klausa yang dipenuhi. Secara tepatnya, logik ini memainkan peranan penting dalam pelbagai aplikasi sebagai logik pengoptimuman gabungan. MAX2SAT adalah kes yang khusus dengan MAX- $k$ SAT ditulis dalam bentuk normal konjungsi (CNF) dengan dua pemboleh ubah dalam setiap klausa. Dalam makalah ini dibentangkan paradigma penggunaan MAX2SAT dengan pelaksanaannya dalam rangkaian neural fungsi asas jejari (RBFNN). Selain itu, analisis dihadkan sehingga klausa MAX2SAT. Dev C++ digunakan sebagai platform untuk latihan dan ujian al-Khwarizmi yang dicadangkan. Dalam kajian ini, keberkesanan RBFNN-MAX2SAT boleh dianggarkan dengan menilai model data ujian yang dicadangkan menggunakan nisbah klausa yang dipenuhi (RSC), punca min ralat kuasa dua (RMSE) dan masa CPU. Hasil simulasi menunjukkan bahawa al-Khwarizmi yang dicadangkan dapat digunakan untuk melakukan pengaturcaraan logik MAX2SAT dengan analisis ke atas prestasi yang diperolehi berdasarkan punca min ralat kuasa dua yang lebih rendah, nisbah klausa dipenuhi yang lebih tinggi dan masa CPU yang lebih rendah.

*Kata kunci:* MAX- $k$ SAT; bentuk normal konjungsi; nisbah klausa dipenuhi; pengoptimuman gabungan

## 1. Introduction

Recently, Satisfiability (SAT) is an NP-complete that undergo extensive research (Mansor *et al.* 2019). In general, the Boolean Formula of the logic programming is represented in Conjunctive Normal Form (CNF) and the task is to find assignments (consist of 0 and 1) that satisfy a particular formula (Mansor *et al.* 2019). Inspired by the counterpart of SAT, Maximum Satisfiability or MAXSAT has been prominent in representing negative outcome. MAXSAT can be defined as a logical rule that contain the maximum number of satisfied clauses. Other MAXSAT instances, which were used in several researches, comprise mainly unweighted MAXSAT, uniform MAX- $k$ SAT instances, where each clause contains exactly  $k$  variables (Raman *et al.* 1998). In our research, we focus on  $k=2$  to obtain MAX2SAT.

The development MAX2SAT logical rule in neural network has been a pioneer work for Kasihmuddin *et al.* (2018). In this paper, MAX2SAT became a logical rule in Hopfield Neural Network. This is the first attempt to do logic programming that contain non-satisfiable logical rule. This paper also pioneered a few researches in MAX2SAT by creating efficient learning method during learning phase of HNN. The first effort in implement logic is carried out by Abdullah (1992) where he proposed a method (Direct Method) to implement logic in a Hopfield neural network (HNN). Furthermore, Sathasivam (2008) capitalized Horn logical rule as a symbolic system in HNN. Another study conducted by Hölldobler *et al.* (1999) where they utilized logic programming by computing the fixed point of the meaning function for feed-forward neural networks with three layers. Radial Basis Function (RBFNN) is a feedforward ANN that contains hidden layers which requires the calculation of centre and width. The advantages of RBFNN includes easy design, good generalization and strong tolerance to input noise (Yu *et al.*, 2011). Logic programming in RBFNN has been proposed by Hamadneh *et al.* (2012a). In this paper, satisfiable logic which is general SAT has been embedded into RBFNN. In order to create an efficient training mechanism, the usage of metaheuristics algorithms to obtain the final output of the RBFNN (Hamadneh *et al.* 2012b) has been recommended. The more systematic logical rule in RBFNN was proposed by Alzaeemi *et al.* (2020). In this paper, 2SAT became a logical rule for the RBFNN where the complexity of the hidden layers dramatically reduced.

Unfortunately, there is no recent work that integrate MAX2SAT into RBFNN due to redundancy issues related to the logical rules. According to our results, it has been showed that RBFNN-MAX2SATFT perform the best performance compared to conventional RBFNN-MAX2SATNT.

## 2. Maximum 2-Satisfiability (MAX2SAT)

MAX2SAT is one of the logical rules in the complexity of computational in one of the branches of computer science. MAX-SAT generalizes the Boolean Satisfiability logic which the decision state considered in the theory of complexity (Kasihmuddin *et al.* 2017). MAX2SAT is special case of MAX- $k$ SAT logical rule where the Boolean expressions are written in CNF with two variable or literal per clause. The definition of MAX2SAT can be demarcated as CNF formula with  $m$  clauses and at most two literals (variables) in each to find the satisfying assignment in the maximum number of clauses in  $P_{MAX\ 2SAT}$  (Kasihmuddin *et al.* 2018). The core impetus of MAX2SAT is to determine how many clauses are true. The example of MAX2SAT formula in the following Eq. (1) (Kasihmuddin *et al.* 2018):

$$P_{MAX\ 2SAT} = (A \vee B)(\neg A \vee B)(A \vee \neg B)(\neg A \vee \neg B) \wedge (\neg C \vee D) \wedge (E \vee \neg F) \quad (1)$$

where  $\wedge$  is the Conjunction (AND),  $\neg$  refers to the negation of the variables,  $\vee$  is the Disjunction (OR),  $(A, B, C, D, E, F) \rightarrow \{0,1\}$ .  $P_{MAX2SAT}$  is a classical optimize logic with decision version was proved NP-complete by Minoux and Mavrocordatos (2001).

### 3. Radial Basis Function Neural Network (RBFNN)

RBFNN is the neural network as the feed-forward network which was first used by Moody and Darken (1989). Compared with other networks, RBFNN has more integrated topology and faster learning speed. In terms of structure, RBFNN contains three neuron layers for computation purposes. In input layer,  $m$  neurons represent the input data that was transferred to the system. During training phase, the parameters (centre and width) will be calculated in the hidden layer. The parameter obtained will be used to calculate the output weight in the output layer. In order to reduce the dimensionality from the input to output layer, Gaussian activation function has been introduced. Gaussian activation function  $\varphi_i(x)$  of the hidden neuron in RBFNN is as given in Eq. (2) (Bouhmala 2016):

$$\varphi_i(x) = e^{-\frac{\left\| \sum_{j=1}^n w_{ji} x_j - c_i \right\|^2}{2\sigma_i^2}} \quad (2)$$

where  $w_{ji}$  is the inputs weights between the input neuron  $j$  in the input layer and the hidden neuron  $i$  in hidden layer. Structurally,  $c_i$ , is the centre and  $\sigma_i$  is the width of the hidden neuron. In this case,  $x_j$  is a binary input values for  $N$  input neurons and Euclidean norm  $\| \cdot \|$  from neuron  $i$  to  $j$ . The final output of RBFNN  $f_{MAX2SAT}(w_i)$  is given in Eq. (3):

$$f_{MAX2SAT}(w_i) = \sum_{i=1}^j w_i \varphi_i(x_i) \quad (3)$$

where  $f_{MAX2SAT}(w_i)$  called the output value of RBFNN and the output weight is  $w_i$ . The aim of our proposed network in doing logic MAX2SAT is to find the maximum number of the truth clauses. We also want to find the true values of the input data in the input layer of RBFNN for each the clauses.

### 4. Methodology

The objective of conducting the logic programming  $P_{MAX2SAT}$  is to obtain the true values for the input data in the input layer of RBFNN for each of the clauses. General form of  $P_{MAX2SAT}$  logical is given by the following Eq. (4) (Hölldobler *et al.* 1999):

$$P_{MAX2SAT} = \bigvee_{i=1}^k C_i \bigvee_{j=1}^n D_j, \quad \text{where } k, n \text{ are real numbers.} \quad (4)$$

where  $C_i$  and  $D_j$  are atoms. The logic  $P_{MAX2SAT}$  in RBFNN is embedded to determine the training dataset for each clause. The general form for calculating the training data (input data) in input layer for each clause in the  $P_{MAX2SAT}$  is as per the following Eq. (5) (Hamadneh *et al.* 2012a):

$$x_i = \sum_{i=1}^n I(C_i) + \sum_{j=1}^n I(D_j) \quad (5)$$

whereas the state of  $I(C_i), I(D_j)$  in the following “Eq. (6)”,

$$I(C_i) \text{ or } I(D_j) = \begin{cases} 1, & \text{when } C_i \text{ or } D_j \text{ is true.} \\ 0, & \text{when } C_i \text{ or } D_j \text{ is false.} \end{cases} \quad (6)$$

where  $\{0, 1\}$  indicates the Boolean values. Here, 1 and 0 refers to true and false respectively.

The  $P_{MAX2SAT}$  clause in the Eq. (1) is later embedded in RBFNN by using the following algorithm:

#### Step 1

Convert the  $P_{MAX2SAT}$  clauses in Eq. (1) into conjunctive normal form (CNF).

#### Step 2

$P_{MAX2SAT}$  based RBFNN will be representing the clauses in the Eq. (1). It consists the variables as the input neurons for the input layer of RBFNN-MAX2SAT (Hamadneh *et al.* 2012a). The clauses also consist the variables for output neurons of RBFNN-MAX2SAT (Mohammadi *et al.* 2017).

#### Step 3

Select the training pattern to train RBFNN-MAX2SAT that represents the clauses in the Eq. (1). Firstly, calculate the parameters of hidden neuron as the centre and the width in each hidden layer. After that, determine the number of hidden neurons between the number of variables (number of the input neurons) and the number of the clauses (number of the output neurons). This is to accelerate the network (Mohammadi *et al.* 2017).

#### Step 4

Choose a training pattern to train the  $P_{MAX2SAT}$  based on RBFNN parameters (output weight in the output layer). This is to increase the performance of proposed network RBFNN-MAX2SAT.

#### Step 5

Calculating the initial output weight of RBFNN-MAX2SAT between the hidden layer of the RBFNN-MAX2SAT and the output layer of the network by using the Eq. (3).

**Step 6**

Calculate the corresponding RMSE, RSC and CPU time for the network performance RBFNN-MAX2SAT. Models will be evaluated with different numbers of neurons in terms of Root Mean Square Error (RMSE). RMSE is a standard error estimator that widely been used in predictions and classifications (Chai & Draxler 2014). The value for ratio of satisfied clauses (RSC) has good agreement with analytical study done by Paul *et al.* (2018) and CPU time also has been utilized in several papers (Kasihmuddin *et al.* 2018; Kasihmuddin *et al.* 2016) for examining the complexity of the proposed MAX-*k*SAT model.

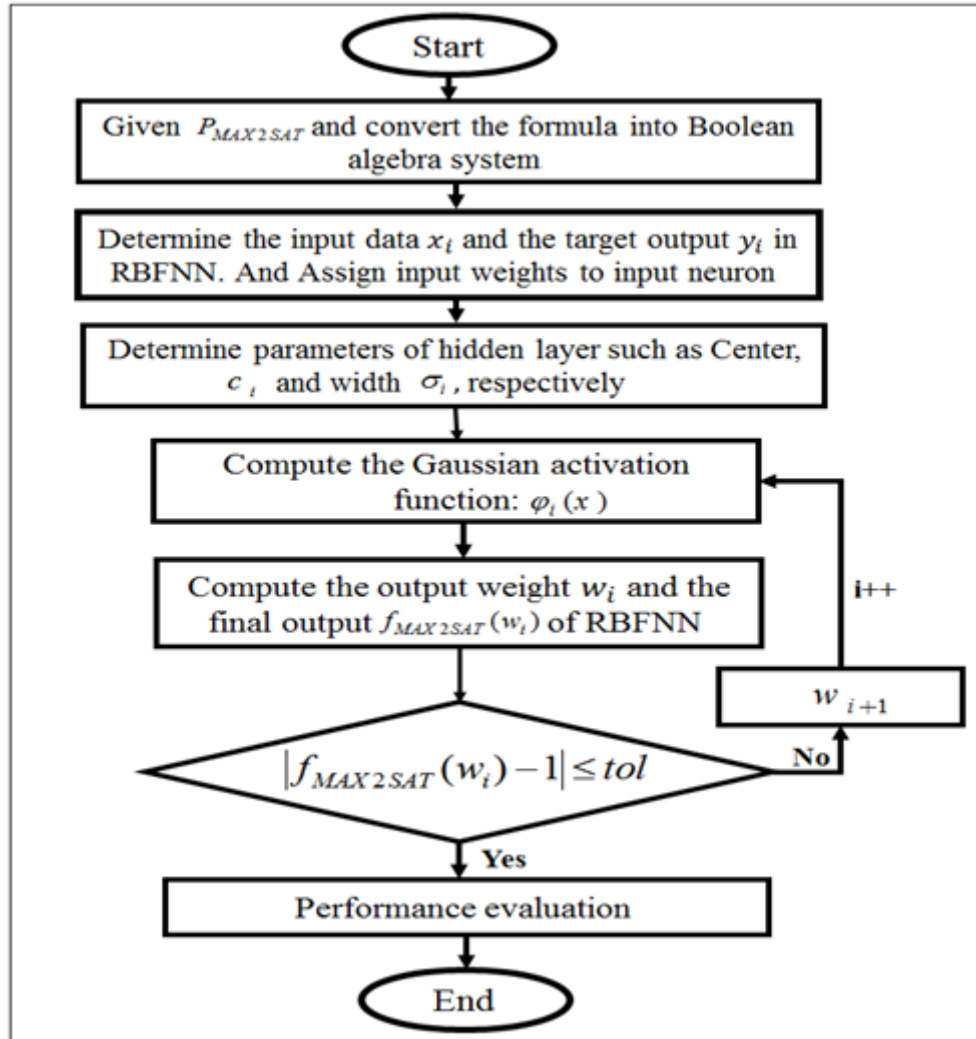


Figure 1: Flowchart of RBFNN to do  $P_{MAX2SAT}$

This is the first attempt to embed MAX2SAT logical rule (knowledge) to the feed-forward neural network. In this study, the MAX2SAT logical rule has been embedded in RBFNN systematically by obtaining the optimal value of parameters (centre and width). 2SAT logical rule is expected to optimize the structure of the RBFNN by fixing the number of hidden neurons involved in both models of RBFNN-MAX2SAT. This approach is called as traditional method Radial Basis Function Neural Network Maximum 2-Satisfiability with no training technique (RBFNN-MAX2SATNT). On the other hand, the hybrid method called Radial Basis Function

Neural Network Maximum 2-Satisfiability with full training technique (RBFNN-MAX2SATFT).

### 5. Experimental Results

This section presents the results of experimental performance of the Radial Basis Function Neural Network with Maximum 2-Satisfiability in the aspect of ratio of satisfied clause (RSC), the root mean square error (RMSE), and CPU time. Results are obtained by comparing the traditional method called Radial Basis Function Neural Network Maximum 2-Satisfiability with no training technique (RBFNN-MAX2SATNT) for the hybrid method called as Radial Basis Function Neural Network Maximum 2-Satisfiability with full training technique (RBFNN-MAX2SATFT). All the proposed RBFNN-MAX2SAT models will be implemented in Microsoft Visual Dev C++ as the platform of training and testing in Microsoft Window 7 (64-bit, with 500 GB hard drive specifications, 4096 MB RAM, and 3.40 GHz processor).

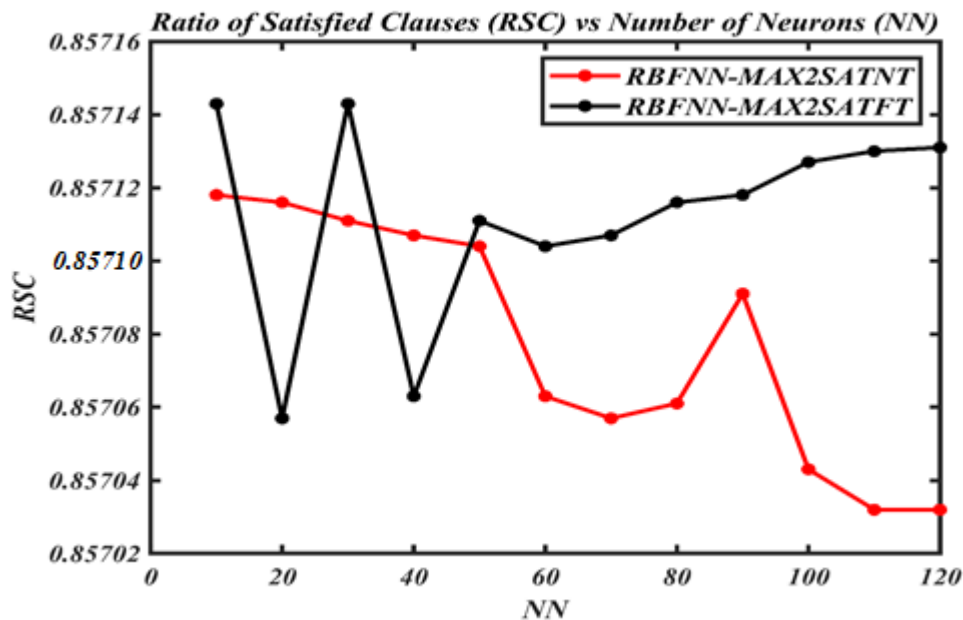


Figure 2: RSC for all RBFNN-MAX2SAT models

Figure 2 shows the RSC values of RBFNN-MAX2SAT for both models RBFNN-MAX2SATNT and RBFNN-MAX2SATFT. RSC values are reported to experience fluctuations. Note that RSC is given as the ratio of satisfied clauses in MAX2SAT logical. More satisfied clauses are observed for both models, RBFNN-MAX2SATFT compared with the traditional RBFNN-MAX2SATNT. Above all, the network RBFNN-MAX2SATFT is able to achieve the desirable maximum number of satisfying clauses while increasing the number of neurons. From theory, the suggested best value of RSC in RBFNN-MAX2SATFT is 0.857130 that has a good agreement with Kasihmuddin *et al.* (2018). In his study, it was mentioned that the best RSC for any of MAX-*k*SAT is between 0.7 and 1. In general, both models of RBFNN-MAX2SAT are able to achieve the desired maximum number of satisfied clauses.

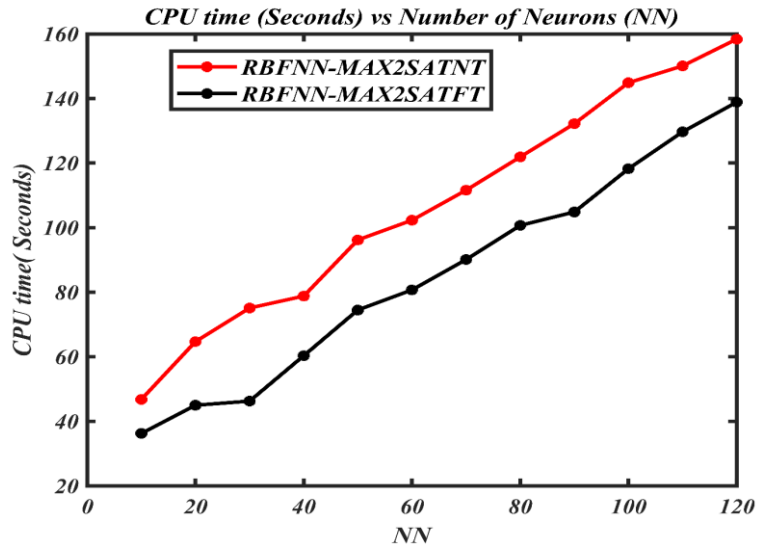


Figure 3: CPU time for all RBFNN-MAX2SAT models

Figure 3 shows the CPU time value for RBFNN-MAX2SAT in both models RBFNN-MAX2SATNT and RBFNN-MAX2SATFT. Computation time described as the time quantity required for the network to reach the best output (final solution). As the network gets bigger by increasing the number of neurons (the input neuron in the input layer of the network), the network required more CPU time to reach the final solution. The reason is the network will be more complex when the numbers of neurons increased. Figure 3 shows RBFNN-MAX2SATFT (the hybrid model) requires minimal computational time compare to RBFNN-MAX2SATNT. Both models of RBFNN-MAX2SAT are observed to find the most consistent  $P_{MAX2SAT}$  interpretation in an acceptable time, compared with the previous model as Minoux and Mavrocordatos (2001), with computational effort, CPU = 7849.23 seconds.

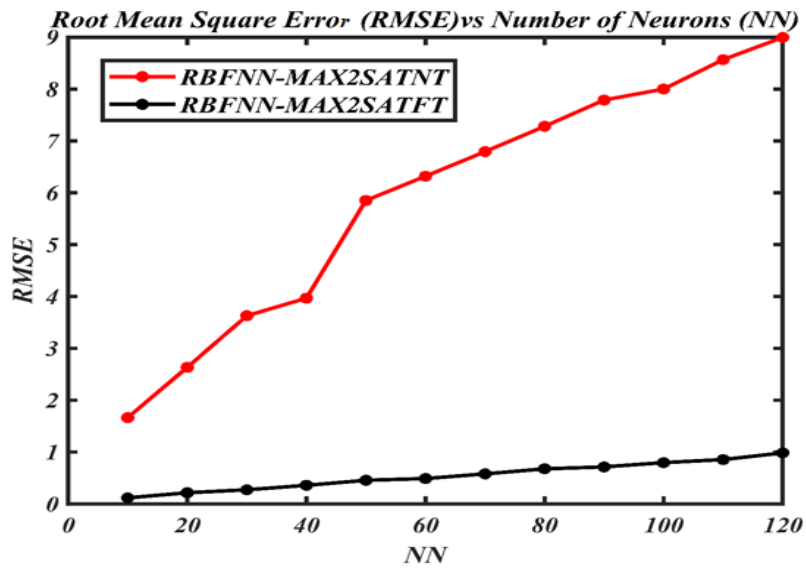


Figure 4: Root Mean Square Error for all RBFNN-MAX2SAT models

Figure 4 shows the RMSE value in RBFNN-MAX2SAT in both models RBFNN-MAX2SATNT and RBFNN-MAX2SATFT. Note that the RMSE values increase when the number of neurons (NN) in both models of RBFNN-MAX2SAT increased. However, the network RBFNN-MAX2SATFT can achieve the minimum value of RMSE, although the number of neurons is increased. The values of RMSE in RBFNN-MAX2SATFT is more acceptable compared to other models. In general, both models of RBFNN-MAX2SAT can achieve the desired maximum number of satisfied clauses but with the different values of error. The apparent reason of why RBFNN-MAX2SATNT needed more time compared to RBFNN-MAX2SATFT is because the RBFNN-MAX2SATNT mechanism leads to a property of entire bit strings of logical rule collapsing when any of the clauses is not satisfied. Thus, more iterations are required to produce a plausible solution, and this generate larger value of RMSE. However, RBFNN-MAX2SATFT can minimize iterations in the completion of learning process due to its optimization operator (Mansor *et al.* 2020).

## 6. Conclusion

In this study an overview of the model Radial Basis Function Neural Network approaches to tackle the  $P_{MAX2SAT}$  logic rule has been proposed. There are two methods proposed and compared to tackle the  $P_{MAX2SAT}$  logical rule in order to obtain an optimal solution. The two methods are traditional technique Radial Basis Function Neural Network Maximum 2-Satisfiability no training (RBFNN-MAX2SATNT) and hybrid technique Radial Basis Function Neural Network Maximum 2-Satisfiability with full training technique (RBFNN-MAX2SATFT). To get high-quality solutions for  $P_{MAX2SAT}$ , the current state-of-the-art for the  $P_{MAX2SAT}$  logic rule suggests that model called RBFNN-MAX2SATFT is the currently most efficient solution technique. This finding is supported by the best performance of RBFNN-MAX2SATFT measured with RMSE (Root Mean Square Error), RSC (Ratio of Satisfied Clause) and CPU time (Computation Time). In future work, a hybrid technique called as Radial Basis Function Neural Network Maximum 2-Satisfiability with full training technique (RBFNN-MAX2SATFT) can be used to evaluate different types of SAT logical rules such as XORSAT, MINSAT and MAJSAT.

## Acknowledgments

This research is supported by Fundamental Research Grant Scheme (FRGS) (203/PMATHS/6711689) by Ministry of Higher Education Malaysia and Universiti Sains Malaysia.

## References

- Abdullah W.A.T.W. 1992. Logic programming on a neural network. *International journal of intelligent systems* **7**(6): 513-519.
- Alzaeemi S., Mansor M.A., Kasihmuddin M.S.M., Sathasivam S. & Mamat M. 2020. Radial basis function neural network for 2 satisfiability programming. *Indonesian Journal of Electrical Engineering and Computer Science* **18**(1): 459-469.
- Bouhmala N. 2016. A variable neighbourhood search structure based-genetic algorithm for combinatorial optimization problems. *International Journal of Intelligent Systems Technologies and Applications* **15**(2): 127-146.
- Chai T. & Draxler R.R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? –Arguments against avoiding RMSE in the literature. *Geoscientific model development* **7**(3): 1247-1250.
- Hamadneh N., Sathasivam S. & Choon O.H. 2012a. Higher order logic programming in radial basis function neural network. *Appl Math Sci.* **6**(3): 115-127.
- Hamadneh N., Sathasivam S., Tilahun S. L. & Choon O. H. 2012b. Learning logic programming in radial basis function network via genetic algorithm. *Journal of Applied Sciences (Faisalabad)* **12**(9): 840-847.



- Hölldobler S., Kalinke Y. & Störr H.P. 1999. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence* **11**(1): 45-58.
- Kasihmuddin M.S.M., Mansor M.A. & Sathasivam S. 2018. Discrete Hopfield neural network in restricted maximum  $k$ -satisfiability logic programming. *Sains Malaysiana* **47**(6): 1327-1335.
- Kasihmuddin M.S.M., Sathasivam S., & Mansor M.A. 2017. Hybrid genetic algorithm in the Hopfield network for maximum 2-satisfiability problem. *AIP Conference Proceedings (Proceedings of the 24<sup>th</sup> National Symposium on Mathematical Sciences)* 1870 (050001).
- Kasihmuddin M.S.B.M., Mansor M.A.B. & Sathasivam S. 2016. Genetic algorithm for restricted maximum  $k$ -satisfiability in the Hopfield network. *IJIMAI* **4**(2): 52-60.
- Mansor M., Mohd Jamaludin S.Z., Mohd Kasihmuddin M.S., Alzaeemi S.A., Md Basir M.F. & Sathasivam S. 2020. Systematic Boolean satisfiability programming in radial basis function neural network. *Processes* **8**(2): 214-229.
- Mansor M.A., Kasihmuddin M.S.M. & Sathasivam S. 2019. Modified artificial immune system algorithm with Elliot Hopfield neural network for 3-satisfiability programming. *J. Inform. Math. Sci.* **11**: 81-98.
- Minoux M. & Mavrocordatos P. 2001. Relaxations and upper bounds for maximum constraint satisfaction. *Application to Frequency Assignment* **1**(7): 1325-1335.
- Mohammadi M., Krishna A., Nalesh S. & Nandy S.K. 2017. A hardware architecture for radial basis function neural network classifier. *IEEE Transactions on Parallel and Distributed Systems* **29**(3): 481-495.
- Moody J. & Darken C.J. 1989. Fast learning in networks of locally tuned processing units. *Neural computation* **1**(2): 281-294.
- Paul A., Poloczek M. & Williamson D.P. 2018. Simple approximation algorithms for balanced MAX 2SAT. *Algorithmic* **80**(3): 995-1012.
- Raman V., Ravikumar B. & Rao S.S. 1998. A simplified NP-complete MAXSAT problem. *Information Processing Letters* **65**(1): 1-6.
- Sathasivam S. 2008. Learning in the recurrent Hopfield network. In. *Proceedings of the 2008 Fifth International Conference on Computer Graphics, Imaging and Visualization*, pp. 323-328.
- Yu H., Xin T., Kaczynski S. & Wilamowski B.M. 2011. Advantages of radial basis function networks for dynamic system design. *IEEE Transactions on Industrial Electronics* **58**(12): 5438-5450.

<sup>1</sup>*School of Mathematical Sciences*

*Universiti Sains Malaysia*

*11800 USM*

*Penang, MALAYSIA*

*E-mail: shehab\_alzaeemi@yahoo.com, saratha@usm.my\*, shareduwan@usm.my*

<sup>2</sup>*School of Distance Education*

*Universiti Sains Malaysia*

*11800 USM*

*Penang, MALAYSIA*

*E-mail: asyrafman@usm.my*

Received: 9 March 2020

Accepted: 1 July 2020

---

\*Corresponding author