MJFAS
Malaysian Journal of Fundamental and Applied Sciences

RESEARCH ARTICLE

Randomised Alpha-Cut Fuzzy Logic Hybrid Model in Solving 3-Satisfiability Hopfield Neural Network

Farah Liyana Azizan[a,b], Saratha Sathasivam[a,*]

[a]School of Mathematical Sciences, Universiti Sains Malaysia, Penang, 11800, USM, Malaysia; [b]Centre for Pre-University Studies, Universiti Malaysia Sarawak, 94300, Kota Samarahan, Sarawak, Malaysia

Abstract This paper proposes an innovative approach to improve the performance of 3SAT logic programming in the Hopfield neural network. The merged structures of the 3SAT and Hopfield network have specific weaknesses, one of which is that, at times, the system attained local minimum solutions rather than global minimum solutions. A new model of integration randomised alpha-cut fuzzy logic with 3SAT in the Hopfield network is built to convey information more effectively. 3SAT and fuzzy logic can work together to solve Hopfield networks' combinatorial optimisation issues. Procedures of fuzzifying and defuzzifying the neurons might ease the computational burden of determining the correct neuron states. Until the proper neuron state is established, unsatisfied neuron clauses will be modified using a randomised alpha-cut approach in the defuzzifier step. An incorporated design built a random approach to select the alpha-cut values of 0.1, 0.25, and 0.5. At this point, a fuzzy value switches into a crisp output back through the defuzzifier process. Based on the results obtained, the proposed hybrid strategy effectively improves the indicators of RMSE, SSE, MAE, MAPE, global minima and total computational time. A computer-generated data set was used to measure how well the hybridised techniques performed. The performance of the proposed network was trained and validated using Matlab 2020b. The results are significant because this model significantly affects how successfully Hopfield networks merged with fuzzy logic can tackle the 3SAT challenges. The obtained data and ideas will help to create novel approaches to data collection for upcoming logic programming exploration.

Keywords: 3-Satisfiability, Alpha-cut, Fuzzy logic, Hopfield Network, Logic programming.

***For correspondence:**
saratha@usm.my

Received: 1 August 2022
Accepted: 28 Dec. 2022

Introduction

A notable advance in artificial intelligence (AI) technology can be seen through merging neural networks, logic programming models, and the SAT issue into one unified hybrid system. The artificial neural network (ANN) is an advanced analytical managing prototype that has already been extensively researched and applied by professionals and researchers due to its capacity to control and represent complex issues. The Hopfield network (HNN), invented by Hopfield and Tank [1], is one of many types of neural networks. The HNN is a recurrent network that works like the human brain and can successfully solve a variety of mathematical complexities. Wan Abdullah made one of the characterised logic rules that created a symbolised representational rule that controls the information stored in the HNN [2], [3]. The proposed HNN model was significantly enhanced by Sathasivam, which has a low storage capacity, requires lengthy computations, and is commonly caught in local minima [4]. The networks frequently have restrictions on the data model, limiting the potential to take into account information collecting. The SAT problem is among the complexity of the algorithm exploration problems based on theory. Numerous approaches can be used to solve the SAT issue. In terms of logic programming, the SAT problem is like

a combinatorial optimisation problem. In other words, a SAT problem is a task that requires limitations, such as circuit layout and pattern reconstruction [5]. Since it may be seen as a combinatorial optimisation problem, SAT logic programming can be used on a neural network to produce the desired outcome. It is intended to reflect information effectively by combining SAT issues with HNN. The expansion of the existing network, fuzzy logic is presented in this project which introduces the hybridised network of 3SAT and randomised alpha-cut fuzzy logic to make the logic programming in HNN perform more effectively significantly. Fuzzy logic minimises the processing burden and enables the suggested strategy to accommodate higher accurate neurons since it uses a fuzzification and defuzzification mechanism to build the right neurons. In addition, until the ideal neuron state is identified, any unsatisfactory neuron clauses will be modified using the randomised alpha-cut method during the defuzzifying procedure. The link between fuzzy and crisp sets highly depends on the alpha-cut concept. An alpha-cut is a set with membership grades greater than or equal to the alpha value $[0, 1]$. In order to connect fuzzy and crisp sets, the alpha-cut concept is essential. Depending on the alpha value, sharpening generates a clean set. The alpha-cut method will modify neuron clauses during defuzzification until the proper neuron state is attained [6]. The breakdowns of the paper sections are as follows: In the first part, we briefly discussed the introduction of HNN, satisfiability problem and fuzzy logic. Whereby in Part 2, we reviewed the 3SAT logic in HNN. Next is the explanation of randomised alpha-cut fuzzy logic techniques. Then, the algorithm and workflow of executing 3SAT with randomised alpha-cut fuzzy logic in HNN are presented. Followed by results and discussion, and lastly, the research findings are concluded.

Materials and Methods

3-Satisfiability in Hopfield Neural Network

A mathematical logical rule called satisfiability (SAT) is made up of sentences with variables. In this study, we will concentrate on 3-satisfiability, also known as 3SAT. The 3SAT problem investigates whether a specific 3SAT rule has a value that assesses the formula to be true. The three components of basic SAT problems are as follows.

A group of l variables in SAT formula where all variables are related with function OR $(\vee)$.

$$v_1, v_2, \dots, v_l, \text{ with every } v_i \in \{1, -1\} \tag{1}$$

The literals involved are either a variable (as a positive variable) or its negation (as a negative variable). A group of n different clauses joined by logical AND $(\wedge)$.

$$S_1, S_2, \dots, S_n \tag{2}$$

3SAT is a clause that contains three literals per clause. As a result, the following are the general attributes of 3SAT clauses.

$$G = \wedge_{n=1}^{NC} S_n \tag{3}$$

where S_n implies the clauses and n indicates the number of clauses and NC is the overall number of clauses. The goal G must be justified in this investigation with $\{1, -1\}$ are the Boolean numbers.
The 3SAT formula with strictly three literals per clause is shown in Equation 4:

$$G_{3SAT} = (V_1 \vee \neg V_2 \vee \neg V_3) \wedge (V_4 \vee V_5 \vee \neg V_6) \wedge (V_7 \vee V_8 \vee V_9) \tag{4}$$

The logic programme's prime purpose is to identify the pattern's significance that fulfils the entire clause. Therefore, the fundamental challenge of converting Equation 4 into the negation of the formula G to identify the logical contradictions.

$$\neg G_{3SAT} = (\neg V_1 \wedge V_2 \wedge V_3) \vee (\neg V_4 \wedge \neg V_5 \wedge V_6) \vee (\neg V_7 \wedge \neg V_8 \wedge \neg V_9) \tag{5}$$

Meanwhile, the cost function for 3SAT is generated by the following formula:

$$E_{G_{3SAT}} = \frac{1}{2^3} \sum_{n=1} \left(\prod_{i=1}^{3} V_i \right) \tag{6}$$

with the corresponding formula

$$V_i = \begin{cases} (1 - NS) & \text{if } \neg V_i \\ (1 + NS) & \text{if } V_i \end{cases} \tag{7}$$

where NS is the state value of $V_1, V_2, V_3, \dots V_i$.

Most of the study agree that HNN has desirable characteristics, including exceptional stability and parallel execution for speedy calculation. The logical conceptualisation of 3SAT can improve HNN's capabilities with extraordinary storage given that its structure is non-symbolic. The mathematical formula for the neuron's activation is stated in Equation 8 below with W_{ij} signifies the synaptic weight for j to i and ξ_i indicates the limit value.

$$S_i = \begin{cases} 1, & \sum_j W_{ij} S_j > \xi_i \\ -1, & otherwise \end{cases} \tag{8}$$

The 3SAT algorithm used in this research, known as HNN-3SAT, only utilises three number of neurons for each sentence. Before designing the last state, the local field essentially controlled the production that was developed.

$$h_i = \sum_j W_{ij} S_j + W_i \tag{9}$$

Finally, the energy function of Lyapunov is exhibit in Equation 10.

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k W_{ijk} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j W_{ij} S_i S_j - \sum_i W_i \tag{10}$$

Randomised Alpha-Cut Fuzzy Logic

Fuzzy sets and logic have proven to be effective strategies for coping with the approximate basis of human thinking and decreasing the design complexity of "humanistic" systems. The groundbreaking ideas work in a decision-making environment characterised by ambiguities, imprecision, and doubt [7]. A fuzzy set expands a classical set as in Boolean form expansion. Fuzzy logic has been proven superior to classical logic on numerous occasions [8]. Standard clauses that use fuzzy logic allow for more truth solutions within its membership function than just a true or false in each variable. It is a type of logic where the truth values can span between zero and one. It is used to convey the concept of partial truth, for which the truth value may range from 100 per cent precisely to 100 per cent erroneous. Therefore, fuzzy logic offers a wider option to make adjustments to the statement, enabling the consideration of contradictions and ambiguity [9]. In the meantime, the Boolean logic form's truth values only offer zero or one [10].

Consider a function of a fuzzy set A, that fits into the interval of $A \in [0,1]$, and is obliged to suit into a membership function of

$$\mu_A : X \to [0,1] \tag{11}$$

The equation below signifies the relationship of the membership function of x in A, $0 \leq \mu_A(x) \leq 1$. The membership functions that can be used are triangular, trapezoidal and many more.

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{12}$$

Furthermore, the notation below is utilised for a discrete universe X which $\sum$ indicates the union of all $x \in X$. The degree of membership function of x in A is called $\mu_A(x)$.

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_n)}{x_n} = \sum_{x \in X} \frac{\mu_A(x)}{x} \tag{13}$$

Mathematical operators like conjunction, disjunction, implication, and negation typically correspond to connectives in classical logic in fuzzy logic [11]. Disjunction (∨) is a connective that is viewed as a truth function corresponding to a common interpretation of logic. Its output is true if at least one of the input sentences (disjuncts) is true and false if it is not. For example, given two statements in a single sentence, $A_i \vee A_j$. The statement is true if one of the inputs is true. The disjunction property for fuzzy logic is as follows:

$$F_{\vee}(A_i, A_j) = max\{A_i, A_j\} \tag{14}$$

When two assertions are combined using the connection AND, the outcome is a conjunction. The conjunction symbol is ∧ which can be interpreted as AND. The truth-functional operator of logical conjunction in logic, mathematics, and linguistics is typically used to represent this operator's logical connective. For example, the conjunction is used to represent the joining of two statements, $A_i \wedge A_j$, in a single sentence. This statement will be true if both combined statements are true; else, it is false. The conjunction property for fuzzy logic is exhibited below.

$$F_{\wedge}(A_i, A_j) = min\{A_i, A_j\} \quad (15)$$

The implication is a logical procedure, usually referred to as logical consequence, implies. When one or more statements follow from one another logically, the relationship between them is what is true. The implication property for fuzzy logic is displayed as follows.

$$F_{\neg}(A_i, A_j) = min\{1,1 - A_i, A_j\} \quad (16)$$

A negation in logic is an operation that converts a statement A_i into a proposition $\neg A_i$, also known as the logical complement. The negation property for fuzzy logic is stated below.

$$F_{\neg}(A_i) = 1 - \mu_{A_i} \quad (17)$$

Several concepts have been developed to improve the efficiency of fuzzy logic systems and provide systematic approaches for converting expert information into fuzzy inference systems. A fuzzy logic system's fundamental components are a fuzzifier, rule evaluator, and defuzzifier. In the fuzzifier stage, it transforms crisp inputs into fuzzy sets. Max and Min rules for OR and AND are evaluated using fuzzy set operations. A fuzzy set is transformed into a clear output by the defuzzifier, a mapping stage in a fuzzy system. The alpha-cut representation of a fuzzy set used in the defuzzification process generates crisp production. Since the alpha-cut method may extract the crisp value from a fuzzy set, the concept of alpha-cut is crucial for connecting fuzzy and crisp sets [12].

The defuzzifier stage is important, since it can build a defuzzifying system that incorporates defuzzification, for example, a randomised alpha-cut method, into the whole setup of the system components. Several researchers have stated the benefits that could be achieved if a match between defuzzifying and the other components of the fuzzy system is improved [13]–[15]. In addition, the researchers have developed defuzzification strategies to aid in optimising or improving system performance. Fuzzy systems use the notion of fuzzy sets to represent and work with uncertain and imprecise data. Most of these fuzzy systems include a defuzzification procedure as a final step that converts a fuzzy set into a crisp value. The defuzzifying process is generally covered in far less detail in fuzzy technology than in the other phases. The system manipulates imprecise and/or uncertain information using fuzzy sets to address the drawbacks of a traditional crisp system. As a result, it makes sense that the core of a fuzzy system will produce a fuzzy set that includes a representation of uncertainty and/or imprecision. Since defuzzification reduces the fuzzy set to a single crisp value, eliminating all of this. Defuzzification, in turn, is merely the last stage.

The idea of alpha-cut is crucial in the relation across fuzzy with crisp sets. It is recognised that the group of all of a fuzzy set's alpha-cuts can be used to describe it uniquely. Every alpha-cut A of a fuzzy set A provides a crisp estimate of A at the level $[0, 1]$. The preceding sharpening of coefficients used to achieve the estimation are as follows:

$A(x)$: if $A(x) \geq \alpha$ then $A(x) = 1$ (sharpening up) (18)

$A(x) < \alpha$ then $A(x) = 0$ (sharpening down) (19)

The sharpening output, the crisp set *A*, is obviously dependent on α. As a result, the phrase "alpha-sharpening" is appropriate. The alpha-sharpening procedure lowers the vagueness of a fuzzy in relation to α. Defuzzification is only necessary to interact with sharp models of the world that are incapable of handling uncertainty or imprecision. Defuzzification is a synthesis process, to sum up. As a result, the entire idea of defuzzification runs directly counter to fuzzy set theory's primary goal, which is the extension of clear notions and theories. A researcher might prefer to replace the alpha value with any other alpha ∈ $(0,1)$ [12]. For instance, one could select alpha=0.25 if only factors $A(x) > 0.25$ must be regarded as "high value" from a practical standpoint. The amount of X components that are included in the alpha-cut of A will decrease as alpha increases in value.

Implementation

Using the Wan Abdullah method during the training phase, we developed a network and improvised it using randomised alpha-cut fuzzy logic technique called HNN-3SATRandFuzzy. By applying the process of fuzzifying and defuzzifying stages, the HNN-3SATRandFuzzy algorithm is constructed until it reaches its final state. The final condition's stability was examined to find a global minima solution.

In this trial design, Matlab 2020b software is used to train and execute the basic version of HNN-3SAT and a new system of the HNN-3SATRandFuzzy network. The HNN designs in this work produced 3SAT clauses with varying degrees of difficulty using simulated datasets. 9 to 180 neurons (NN) are used in the computations of this investigation. The trial will be terminated if the computation time used to generate data surpasses 24 hours [16]. In addition, we choose the hyperbolic activation function (HTAF)

as an activation function due to its stability. HTAF is seen to be an excellent example of an activation function that has been built in HNN. The hybridised network continues to operate even when the activation function is not employed. The final energy performance constraints were set at 0.001 in order to minimised statistical errors more effectively [17]. The effectiveness of this design will be defined by contrasting the efficacy and accuracy of the models: HNN-3SAT and various alpha-cut values for HNN-3SATRandFuzzy. The key components to formulate and implement HNN-3SATRandFuzzy are all listed in Table 1 below.

Table 1. The key components of HNN-3SATRandFuzzy

Number of neurons, NN	$9 \leq NN \leq 180$
Total of combinations	100
Tolerance measurement	0.001
CPU time threshold	24 hours
Activation function	HTAF
Randomised alpha-cut value	$0 \leq$ alpha-cut ≤ 1

Figure 1 depicts the methodology algorithm for the HNN-3SATRandFuzzy. The initial step is to build a 3SAT logic program that must show its incoherence in order to prove a specific goal. Following that, the logic structure is transformed into a Boolean algebraic structure and its negation. Fuzzification is a technique used in the advanced model that connects a neuron's identity to its membership function. We can assess rules using fuzzy logic attributes by converting the Boolean model into a fuzzy structure. The randomised alpha-cut defuzzification approach is then used. The link between fuzzy and crisp sets depends on the alpha-cut principle. At this point, the value of alpha-cut is determined by randomising alpha $\in (0,1)$. The chosen values for alpha-cut are $\beta = 0.1, \gamma = 0.25, \delta = 0.5$. Then the synaptic weights must then be determined using the cost function. Later, the estimations of synaptic weights are obtained by analysing the cost function along with the energy function. Let the neural networks develop until they achieve a state with the least energy.

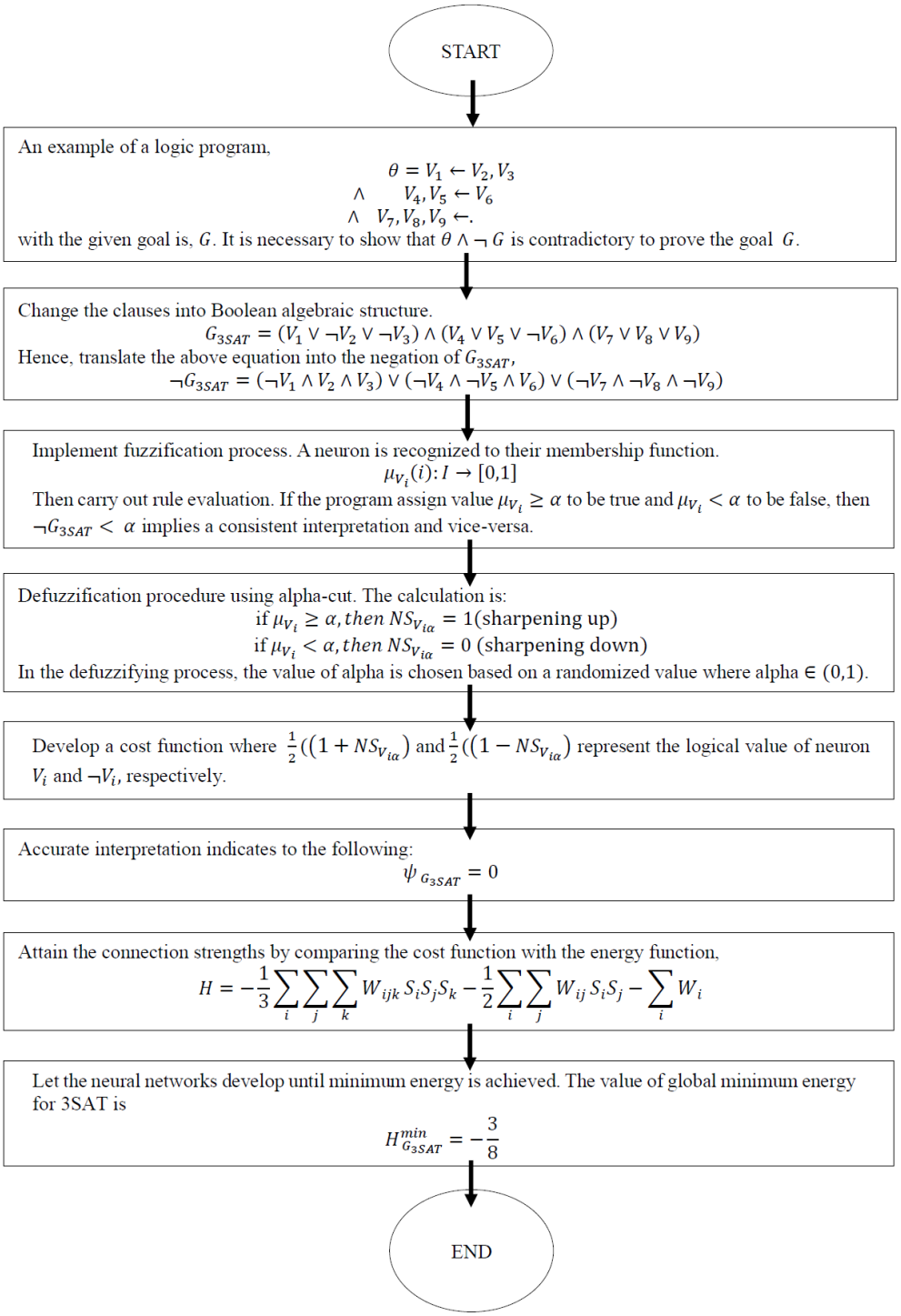

Figure 1. The methodology of HNN-3SATRandFuzzy

We use the Wan Abdullah approach for the HNN-3SAT and HNN-3SATRandFuzzy with alpha-cut values, $\beta = 0.1, \gamma = 0.25, \delta = 0.5$ during the training period. The improved hybrid HNN-3SATRandFuzzy is incorporated with the fuzzy logic method with a randomised value for alpha-cut. The alpha-cut value is chosen by a random process built into the integrated development system. The chosen alpha-cut values are $\beta = 0.1, \gamma = 0.25, \delta = 0.5$. An integrated framework was created by applying the fuzzification and defuzzification process until it reached its final form. It was determined whether the final situation was stable. In addition, the HNN-3SATRandFuzzy algorithm will use the alpha-cut approach to modify the unsatisfied neuron clauses during the defuzzification process until the proper neuron state is identified. A stable state is one that was obtained and consistent across both algorithms. In each

simulation, the process is repeated 100 times with 100 different neuronal sequences. In order to reduce statistical errors, the execution conditions for the final energy were adjusted to 0.001 [4]. We compared the error analysis for RMSE, MAE, SSE and MAPE and energy analysis of global minima ratio as well as CPU time. The newly proposed hybrid method, which combines 3SAT and fuzzy logic in the HNN, would extract data from a whole logical rule and assess clause fulfilment during the training phase more effectively than the basic version of HNN-3SAT. Table 2 below contains a list of the subsection's performance indication metrics. These metrics measure how well the HNN-3SATRandFuzzy performed compared to HNN-3SAT.

Table 2. List of the metrics and their formula used in all performance evaluation measures

Metrics	Formula
Root Mean Square Error (RMSE)	$RMSE = \sum_{i=1}^{n} \sqrt{\frac{1}{n}\left(I_{highest} - I_x\right)^2}$
Mean Absolute Error (MAE)	$MAE = \sum_{i=1}^{n} \frac{1}{n}\left\|I_{highest} - I_x\right\|$
Sum Squared Error (SSE)	$SSE = \sum_{i=1}^{n} \left(I_{highest} - I_x\right)^2$
Mean Absolute Percentage Error (MAPE)	$MAPE = \sum_{i=1}^{n} \frac{100}{n} \frac{\left\|I_{highest} - I_x\right\|}{\left\|I_x\right\|}$
Global Minima Ratio (Zm)	$Global\ Minima = \frac{1}{NT.COMBMAX} \sum_{i=1}^{n} N$
CPU (central processing unit) time	Processing time = Training phase time + Retrieval Phase time

Results and Discussion

The results in this section were obtained from the computational process of the simulated data sets performed on the two core models known as HNN-3SAT and HNN-3SATRandFuzzy. The performance of HNN-3SAT and HNN-3SATRandFuzzy with alpha-cut values, $\beta = 0.1, \gamma = 0.25, \delta = 0.5$ are depicted in Figure 2 to Figure 7 using the metrics of RMSE, MAE, SSE, MAPE, global minima and processing time respectively. According to the experimental findings, the best model is HNN-3SATRandFuzzy with $\beta = 0.1$, which can classify data using the 3SAT logical rule while utilising the lowest possible values of RMSE, SSE, MAE, MAPE as well as the ideal value of global minima and fastest CPU time. The HNN-3SATRandFuzzy model with $\beta = 0.1$ had the best error performance as the number of hidden neurons was increased, as illustrated in Figures 2 to 7. Although HNN-3SATRandFuzzy with $\delta = 0.5$ results are higher than 0.1 or 0.25, it is regarded as superior to the HNN-3SAT network's basic version since it has a lower learning error and good quality of energy analysis results compared to HNN-3SAT. The model's performance capacity was impacted by the important model characteristics and fixing the unsatisfied neuron using randomised alpha-cut fuzzy logic.

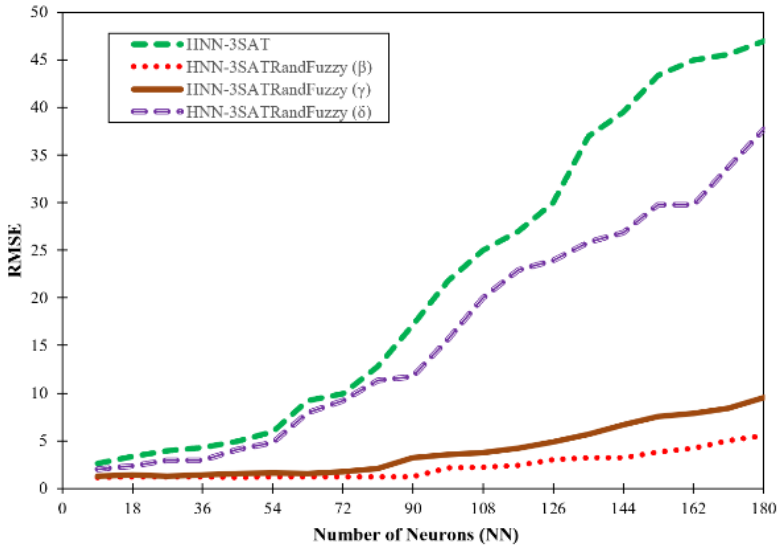

Figure 2. Figure of RMSE

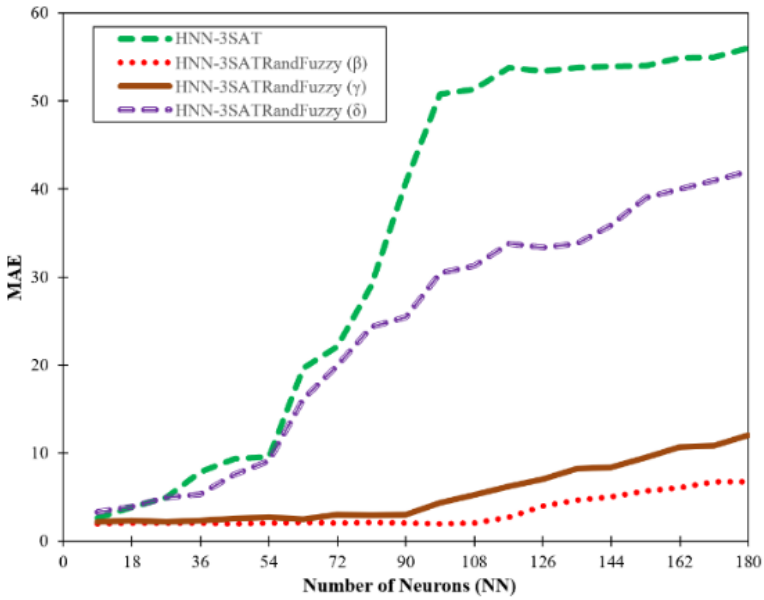

Figure 3. Figure of MAE

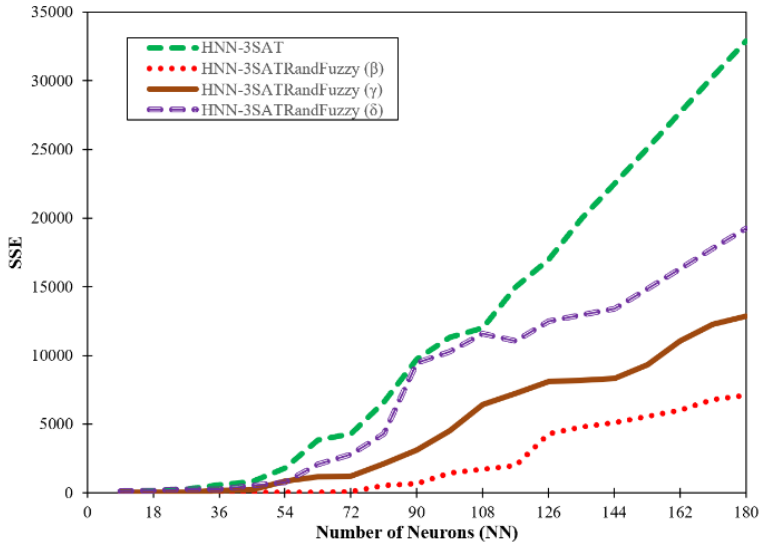

Figure 4. Figure of SSE

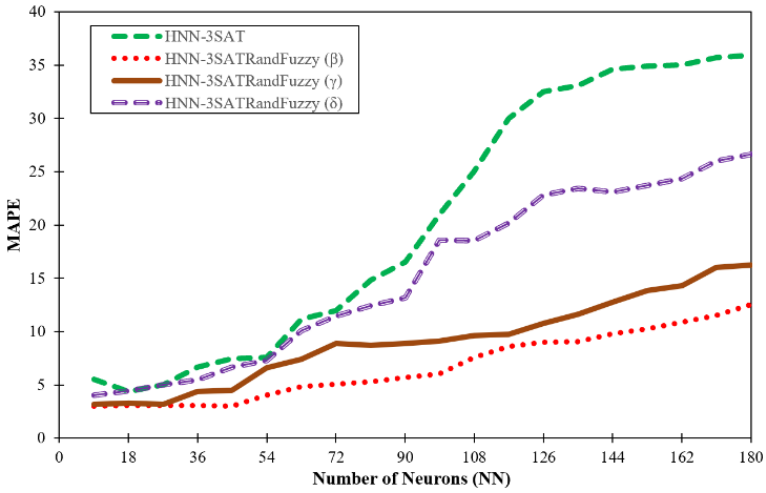

Figure 5. Figure of MAPE

Figure 2 compares the learning error value for root mean square error (RMSE) obtained by the basic version of the HNN-3SAT model and the new hybrid model of HNN-3SATRandFuzzy with randomised alpha-cut value; $\beta = 0.1, \gamma = 0.25, \delta = 0.5$. RMSE performance for HNN-3SAT and HNN-3SATRandFuzzy are shown in Figure 2. By comparing to all performances, HNN-3SATRandFuzzy with $\beta = 0.1$ approach performs remarkably good. This is because the neurons are being fuzzified and defuzzified by the robust operators with the correct condition of randomised alpha-cut value. Figure 2 has a consistently lowered value of RMSE, which suggests fewer repetitions were required throughout the learning phase. This shows that HNN-3SATRandFuzzy with $\beta = 0.1$ is accurate in minimising the 3SAT logical rule's cost function in a limited number of iterations. The value of MAE in Figure 3 made it clear that HNN-3SATRandFuzzy with $\beta = 0.1$ consistently recorded MAE less value for all hidden neuron counts. On the other hand, up until the final runs, the MAE for HNN-3SAT, HNN-3SATRandFuzzy with $\delta = 0.5$ grew noticeably. Even though HNN-3SATRandFuzzy with $\delta = 0.5$ is elevated than $\beta = 0.1, \gamma = 0.25$, it is considered better since it has a lower MAE value than the basic version of HNN-3SAT network. The reduced value of MAE has offered compelling evidence of fuzzy logic's effectiveness in conjunction with HNN-3SAT. Fuzzification and defuzzification processes both exhibit updates in the neuron states, increasing the likelihood of finding the optimal solution in $\psi_{G_{3SAT}} = 0$. Additionally, continual updates of the neurons through randomised alpha-cut defuzzification process creates less learning iterations and guarantees the effective randomisation within defuzzification steps. In contrast, despite the convergence guarantee, HNN-3SAT performs poorly based on the MAE because the trial

and enumerate approach fully minimise the cost function of 3SAT. In Figure 4, the SSE values were compared to HNN-3SAT, HNN-3SATRandFuzzy with $\beta = 0.1, \gamma = 0.25, \delta = 0.5$. SSE is referred to as an actual error and is frequently utilised by researchers to examine whether the answer is accurate. According to Figure 4, the HNN-3SATRandFuzzy network with $\beta = 0.1$ has a stronger ability to train the data set than the other network because of the lower SSE values. Figure 4 explains how sensitive the hybrid models are to potential errors during the learning process. The model with the lowest SSE value is HNN-3SATRandFuzzy with $\beta = 0.1$ which shows that it is less susceptible to incoming errors. As a result, HNN-3SATRandFuzzy has a lesser sensitivity than HNN-3SAT. The fuzzifying operators have ensured that the solutions will be updated, and the final selection will be computed iteratively during defuzzification in order to achieve the optimal solution with the correct neurons. The significant results increase in SSE for HNN-3SAT, which denotes an ineffective learning phase, is the most intriguing pattern in Figure 4. To obtain $\psi_{G_{3SAT}} = 0$, the trial-and-error process in HNN-3SAT needs more learning iterations. Figure 5 displays the MAPE findings for each HNN-3SAT model using the experimental data. The MAPE value for the hybridised models slowly rises as the number of neurons increases. Since the models can reach global convergence in fewer iterations, HNN-3SATRandFuzzy with $\beta = 0.1$ creates less MAPE. Compared to all networks, HNN-3SATRandFuzzy with $\beta = 0.1$ has a significantly better learning MAPE. According to Figure 5, the MAPE value for HNN-3SATRandFuzzy with $\beta = 0.1$ is significantly lower than that of the other models. The learning phase does not fully satisfy a significant portion of the 3SAT clause found in HNN-3SAT and the rest of HNN-3SATRandFuzzy. HNN-3SATRandFuzzy with $\beta = 0.1$, on the other hand, has a more consistent clause from the total clause because it has a lower MAPE value. The fuzzification operator has improved the solution to become global without requiring a trial-and-error stage. It became evident that the MAPE solution for HNN-3SATRandFuzzy with $\beta = 0.1$ shows the least values when compared to the standard HNN-3SAT model. This is the outcome of the fuzzification and defuzzification developments in updating neurons until it reached the best solution. Because of the better performance in learning errors RMSE, MAE, SSE, and MAPE, HNN-3SATRandFuzzy is a good option for 3SAT logic programming at various levels of complexity with randomised alpha-cut value $\beta = 0.1$.

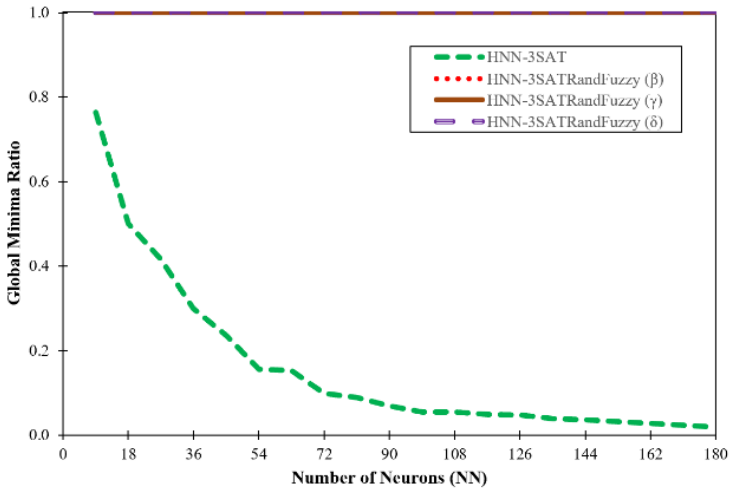

Figure 6. Global Minima Ratio (zM)

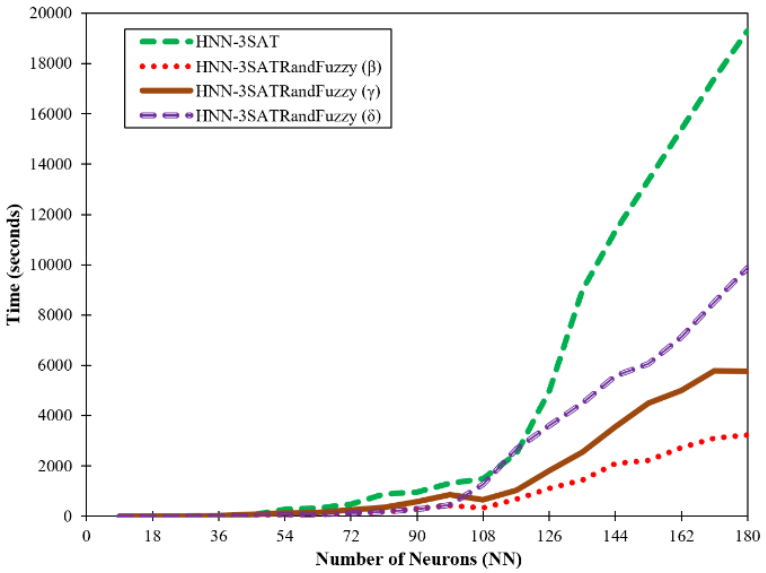

Figure 7. Processing Time

The percentage of the global solutions generated using the basic version HNN-3SAT and HNN-3SATRAndFuzzy for various numbers of literals per clause is shown in Figure 6 above. The effectiveness of all networks will now be evaluated by taking into account their global minima ratio for various network complexity variations. A correlation between the ratio of global minima and the energy attained as a result of the computation was found by Sathasivam [4]. Based on global minima plots, the best optimal solution is one that is close to 1 [18]. Based on Figure 6, HNN-3SATRandFuzzy with $\beta = 0.1$ provides more precise and accurate states in comparison to the other approach. The randomised alpha-cut fuzzy logic solution for $\beta = 0.1$ has achieved the best global minimum energy, which is very close to 1. The recommended method can support more neurons since fuzzy logic uses the state of the neurons to fuzzify and defuzzify to determine the proper states, which reduces the computational load. Additionally, unsatisfied neuron clauses will be improved using the randomised alpha-cut method during the defuzzification process until the proper neuron state is discovered. Despite the increased complexity, Figure 6 demonstrates that fuzzy logic combined with 3SAT in HNN can handle the network better than the HNN-3SAT. Even as the number of neurons and sentences increased, the suggested approach reached global minimum solutions with more reliable outcomes. However, when the system becomes larger or more complex, the HNN-3SAT method oscillates in local rather than global solutions. The randomised alpha-cut fuzzy logic 3SAT in HNN generates global minimum solutions to converge to an optimal solution with superior outcomes when compared to the HNN-3SAT approach. The network's computational processing times for the basic HNN-3SAT network and the randomised alpha-cut fuzzy logic 3SAT are shown in Figure 7 above. The computational time for the learning and retrieval stages of HNN is tracked to evaluate the models' resilience in logic mining. Figure 7 shows that when there is fewer NC deployed, it takes less CPU time to complete one execution of learning and testing. The learning for the HNN-3SAT models takes a substantial amount of time when the complexity is higher. Overall, the HNN continues to be proficient at reducing the 3SAT inconsistencies and computing the overall solution in a reasonable amount of CPU time. Due to the need to handle more instances during the HNN learning and testing phase, the CPU time for HNN-3SAT is constantly greater than HNN-3SATRandFuzzy. Based on the data gathered, it is evident from Figure 7 that the suggested HNN-3SATRandFuzzy performs significantly better than the basic version of HNN-3SAT. The HNN-3SAT approach takes longer to perform than the HNN-3SATRandFuzzy. As the number of neurons increased, the possibility that the same neuron would participate in more than one phrase increased [19]. The likelihood that the network would become stopped in local minima and consume more processing time increased as the network grew larger and more complex. The HNN-3SATRandFuzzy with $\beta = 0.1$ developed the fastest time spending as the number of sentences for each number of neurons increased, despite the fact that the time spent for both networks is slightly different for lower numbers of clauses. For networks with more than 100 neurons, the basic version of HNN-3SAT network gets trapped in a less-than-ideal state, taking a lot longer to find the answer. In order to reach the global minima, the neurons also had to overcome a significant energy barrier. Separately, the training process for the HNN-3SAT approach often requires more computing time due to the iterative processes required to get an acceptable interpretation. When fuzzy logic was employed, on the other hand, the calculation time was

cut in half since the status of the fuzziness neurons was given prior to starting the defuzzification process. The hybridised network gradually transformed the dissatisfied clause into a satisfied clause using randomised alpha-cut method. The findings for 3SATRandFuzzy with $\beta = 0.1$ showed that the suggested network performed better than the rest method in terms of computation time.

Conclusions

This study demonstrated that the HNN-3SATRandFuzzy approach, which combines randomised alpha-cut fuzzy logic and 3SAT in the Hopfield network, is a special way to boost the effectiveness of logic programming. The learning part of the HNN-3SATRandFuzzy network is built on the addition of fuzzifying and defuzzifying procedures. The fuzzification and defuzzification methods with a randomised alpha-cut approach were used to enhance methods for avoiding local minimum solutions and lessen the computational burden required to generate the best outcomes. The insights are important because the new model significantly affects the stability of Hopfield networks to handle problems. The results for global minima ratio, networks' computational time consumption, RMSE, MAE, SSE, and MAPE are all improved in the proposed technique. Despite the fact that complexity increased, the HNN-3SATRandFuzzy network converged to less complicated solutions. On the other hand, the HNN-3SAT was unable to converge on a superior solution as the network grew larger and more intricate. These factors have enhanced the functionality of the Hopfield network, and we may infer that HNN-3SATRandFuzzy enhances the functionality of neural-symbolic integration.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

This research was supported by the Ministry of Higher Education Malaysia (MOHE) through the Fundamental Research Grant Scheme (FRGS), FRGS/1/2022/STG06/USM/02/11 and Universiti Sains Malaysia.

References

[1] Hopfield, J. J., & Tank, D. W.(1985). 'Neural' computation of decisions in optimization problems. *Biol Cybern*, *52*(3), 141-152.
[2] Abdullah, W. A. T. W. (1993). The logic of neural networks. *Phys Lett A*, *176*(3), 202-206.
[3] Abdullah, W. A. T. W. (1992). Logic programming on a neural network. *International Journal of Intelligent Systems*, *7*(6), 513-519.
[4] Sathasivam, S. (2010). Upgrading Logic Programming in Hopfield Network. *Sains Malays*, *39*(1), 115-118.
[5] Peng, C., Xu, Z. & Mei, M. (2020). Applying aspiration in local search for satisfiability. *PLoS One*, *15*(4), 1-16.
[6] Pourabdollah, A., Mendel, J. M. & John, R. I. (2020). Alpha-cut representation used for defuzzification in rule-based systems. *Fuzzy Sets Syst*, *399*, 110-132.
[7] Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, *8*, 338-353.
[8] Agrawal, H., *et al*. (2022). A Fuzzy-Genetic-based integration of renewable energy sources and E-vehicles. *Energies, 15*(9).
[9] Halaby, M. & Abdalla, A. (2016). Fuzzy maximum satisfiability. *ACM International Conference Proceeding Series*, 9-11, 50-55.
[10] Anzilli, L. & Facchinetti, G. (2019). *An alpha-cut evaluation of interval-valued fuzzy sets for application in decision making*, 11291, Springer International Publishing.
[11] Brys, T., Hauwere, Y. M. De., Cock, M. De. and Nowé, A. (2012). Solving satisfiability in fuzzy logics with evolution strategies. *Belgian/Netherlands Artificial Intelligence Conference*.
[12] Bodjanova, S. (2002). A generalized α-cut. *Fuzzy Sets Syst*, *126*(2), 157-176.
[13] Roychowdhury, S. & Wang, B. H. (1996). Cooperative neighbors in defuzzification. *Fuzzy Sets Syst*, *78*(1), 37-49.
[14] Saade, J. J. & Diab, H. B. (2004). Defuzzification methods and new techniques for fuzzy controllers. *Iranian Journal of Electrical and Computer Engineering*, *3*(2), 161-174.
[15] Midaoui, M., Qbadou, M. & Mansouri, K. (2022). A fuzzy-based prediction approach for blood delivery using machine learning and genetic algorithm. *International Journal of Electrical and Computer Engineering*

(IJECE), *12*(1), 1056.
[16] Kho, L. C., Kasihmuddin, M. S. M., Mansor, M. A. & Sathasivam, S. (2020). Logic mining in league of legends. *Pertanika J Sci Technol*, *28*(1), 211-225.
[17] Sathasivam, S., Mansor, M. A., Kasihmuddin, M. S. M. & Abubakar, H. (2020). Election algorithm for random k satisfiability in the hopfield neural network. *Processes*, *8*(5).
[18] Alzaeemi, S. A., Sathasivam, S. & Velavan, M. (2021). Agent-based modeling in doing logic programming in fuzzy hopfield neural network. *International Journal of Modern Education and Computer Science*, 13(2), 23–32.
[19] Sathasivam, S., Mansor, M. A., Ismail, A. I. M. D., Jamaludin, S. Z. M., Kasihmuddin, M. S. M. & Mamat, M. (2020). Novel random k satisfiability for k ≤ 2 in hopfield neural network. *Sains Malays*, *49*(11), 2847-2857.