Pertanika J. Sci. & Technol. 31 (4): 1695 - 1716 (2023)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Hybridised Intelligent Dynamic Model of 3-Satisfiability Fuzzy Logic Hopfield Neural Network

Farah Liyana Azizan[1,2], Saratha Sathasivam[1]* and Majid Khan Majahar Ali[1]

[1]*School of Mathematical Sciences, Universiti Sains Malaysia, Penang, 11800, USM, Malaysia*
[2]*Centre for Pre-University Studies, Universiti Malaysia Sarawak, 94300 UNIMAS, Kota Samarahan, Sarawak, Malaysia*

ABSTRACT

This study presents a new way of increasing 3SAT logic programming's efficiency in the Hopfield network. A new model of merging fuzzy logic with 3SAT in the Hopfield network is presented called HNN-3SATFuzzy. The hybridised dynamic model can avoid locally minimal solutions and lessen the computing burden by utilising fuzzification and defuzzification techniques in fuzzy logic. In addressing the 3SAT issue, the proposed hybrid approach can select neuron states between zero and one. Aside from that, unsatisfied neuron clauses will be changed using the alpha-cut method as a defuzzifier step until the correct neuron state is determined. The defuzzification process is a mapping stage that converts a fuzzy value into a crisp output. The corrected neuron state using alpha-cut in the defuzzification stage is either sharpening up to one or sharpening down to zero. A simulated data collection was utilised to evaluate the hybrid techniques' performance. In the training phase, the network for HNN-3SATFuzzy was weighed using RMSE, SSE, MAE and MAPE metrics. The energy analysis also considers the ratio of global minima and processing period to assess its robustness. The findings are significant because this model considerably impacts Hopfield networks' capacity to handle 3SAT problems with less complexity and speed. The new information and ideas will aid in developing innovative ways to gather knowledge for future research in logic programming. Furthermore, the breakthrough in dynamic learning is considered a significant step forward in neuro-symbolic integration.

Keywords: 3SAT, alpha-cut, defuzzification, fuzzification, fuzzy logic, Hopfield network

ARTICLE INFO

Article history:
Received: 16 June 2022
Accepted: 13 September 2022
Published: 24 May 2023

DOI: https://doi.org/10.47836/pjst.31.4.06

E-mail addresses:
afliyana@unimas.my (Farah Liyana Azizan)
saratha@usm.my (Saratha Sathasivam)
majidkhanmajaharali@usm.my (Majid Khan Majahar Ali)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

Artificial Intelligence (AI) is the impetus for today's technological advancement. Thus, it leads to the advanced development of machine learning techniques to solve those problems. Artificial Neural Networks (ANNs) can be categorised as a sub-domain of AI widely used to improve decision-making in various disciplines. An ANN comprises interconnected neurons with discrete input and output layers inspired by the biological neuron model. The system is an aligned computing system created by simulating the human's instinctive thinking while investigating the biological brain's network in terms of biological neurons (Garcez & Zaverucha, 1999). The Hopfield Network (HNN) is a single-level recursive neural network (RNN) in which every single neuron output is linked to every other neuron response (Hopfield & Tank, 1985). HNN uses a particular symbolic learning model to efficiently coordinate the propagation of the input and output neurons in solving problems. The capacity of the HNN to resolve to the closest minimal solution determines the neuron state's dynamic behaviour. Abdullah (1992) proposes a method for logic programming on the HNN.

After defining the connection strengths, or mostly called the synaptic weight with logic programming, that is, by comparing cost and energy functions, the network performed a logical inconsistency reduction in programming. Abdullah (1993) introduces the learning phase in the HNN directly. The logic paradigm of Abdullah has become the most prominent and has lately been employed (Mansor & Sathasivam, 2021; Sathasivam et al., 2020). A mathematical framework can describe various scientific and technological challenges in daily life. However, one must first create methods for resolving some mathematical issues to do so. Many crucial problems, such as categorising or finding an ordered list, can be solved with realistic solutions. Nevertheless, a mathematical problem is the Satisfiability Problem (SAT). Unravelling these difficulties is possible with the aid of a computer. The Satisfiability Problem, or SAT, is one of the most well-known issues. It is described as an approach for achieving the best task utilising Boolean quantities to verify that the 3SAT formula is met. A large number of NP issues can be simplified via SAT.

In earlier research, the HNN model and 3SAT logic programming were combined to characterise the innovation as a singular data mining method. This model has been tested with a real-life dataset to assess its efficiency of the model. The method assesses various data sets related to cardiovascular disorders (Mansor et al., 2018). More logic mining strategies, including 3SAT in HNN, have been presented using real-life datasets such as the Bach Choral Harmony and German Credit (Zamri et al., 2020). However, the existing work's 3SAT problem in the Hopfield network only considers zero and one neuron values. Hence, to resolve this problem, this model is further improved by incorporating fuzzy logic techniques to create a hybridised intelligent dynamic model that can choose between zero and one neuron states. Traditional logic, as well as logic programming languages, are incapable of dealing with uncertainty. Crisp relations are nonfuzzy relations that use the basic two-valued Boolean logic connectives to define their operations, a mathematical system based on true and false statements. Fuzzy logic connectives are extensions that substitute two-valued Boolean logic connectives with many-valued logic connectives.

The Boolean relations and sets that are crisp and nonfuzzy are essentially particular examples of fuzzy relational structures, thanks to a unified approach to relations.

MATERIALS AND METHODS

Satisfiability Problem

The challenge of establishing the exposition of an assignment using a specified Boolean formula that assesses it as true or false is known as Boolean Satisfiability (SAT). Every variable is denoted by $X_1, X_2, \ldots, X_n$ for any $n \in \mathbb{N}$ in a propositional formula, θ. Each value from the set {0,1}, signifying false and true, can be assigned to these variables. If a variable has not yet been assigned a truth value, it is a free variable (Maandag, 2012). A propositional Boolean formula can also include the Boolean connectives of AND $(\wedge)$, OR $(\vee)$, NOT $(\neg)$ as well as parentheses to denote precedence.

If a propositional expression contains one or more clauses, it is understood as conjunctive normal form (CNF). A clause is a disjunction with one or many literals from the set *L*, including all literals. Each *L* represents whether it is a variable ω_i or its negation, $\neg\omega_i$. The following is an example of a propositional formula, θ. An *i*-CNF formula is a CNF formula in which each sentence has at most *i* different literals. For example, Equation 1 below is a 2-CNF formula.

$$\theta = (\omega_1 \vee \omega_2) \wedge (\neg\omega_1 \vee \omega_3) \wedge (\neg\omega_2 \vee \neg\omega_3) \qquad [1]$$

This work will emphasise 3-CNF satisfiability or 3SAT in abbreviated form. The 3SAT problem examines whether a 3-CNF formula has a valuation that evaluates the formula as true or if a particular 3-CNF formula is satisfactory.

The structure of SAT will be described below:

1. The *i* variables in the Boolean SAT formula as given in Equation 2:

   $$\omega_1, \omega_2, \ldots, \omega_i \text{ for every } \omega \in \{-1,1\} \qquad [2]$$

   Every variable of the clause is related to function OR $(\vee)$. Because this study will focus on 3SAT, it will consist of 3 literals per clause.
2. In a 3SAT formula, a set of a clause, β_m joined by AND $(\wedge)$ as given in Equation 3

$$\beta_1 \wedge \beta_2 \wedge \ldots \wedge \beta_m \tag{3}$$

where if $m = 3$, the Boolean SAT will have three clauses.

3. The literal's status can then be either the negative or the positive of the variables.

The logical formula is derived from the randomised 3SAT formula in this study. The propositional logic formula, including the 3SAT formula, can be translated into logic programming notations (Abdullah, 1992; Kowalski & Sergot, 1986). The ideal performance measures define and evaluate the 3SAT problem in the HNN. The example of 3SAT logic programming is shown in Equation 4.

$$\begin{aligned} \theta &= \omega_1 \leftarrow \omega_2, \omega_3 \\ \wedge &\quad \omega_4, \omega_5 \leftarrow \omega_6 \\ \wedge &\quad \omega_7, \omega_8, \omega_9 \leftarrow . \end{aligned} \tag{4}$$

Given the goal as Equation 5,

$$\leftarrow \vartheta \tag{5}$$

where $\omega_1, \omega_2 \ \ldots.. \omega_n$ for any $n \in \mathbb{N}$ refers to the literals in the clauses and ← describes the implication and the given goal is ϑ.

The general formula for 3SAT is expressed in Equation 6.

$$\theta = \wedge_{i=1}^{n} \beta_m \tag{6}$$

where β_m signifies a set of a clause and i indicates the number of the clause.

The 3SAT logical representation in Boolean algebraic form with strictly three literals per sentence is known as discrete logic representation, as demonstrated in Equation 7.

$$\theta_{3SAT} = (\omega_1 \vee \neg\omega_2 \vee \neg\omega_3) \wedge (\omega_4 \vee \omega_5 \vee \neg\omega_6) \wedge (\omega_7 \vee \omega_8 \vee \omega_9) \tag{7}$$

where θ_{3SAT} will be fulfilled if $\theta_{3SAT} = 1$, the capacity to store information in Bipolar states, in which each state represents a significant structure for the dataset, is one of the critical reasons for encoding the variable in the form Equation 6. The logic program's primary goal is to find an interpretation of the structure that satisfies the whole clause. The hybridised dynamic model of HNN will have the 3SAT logical rule encoded in it. As a result, finding the logical inconsistencies transforms Equation 7 into the negation of θ_{3SAT} as presented in Equation 8.

$$\neg\theta_{3SAT} = (\neg\omega_1 \wedge \omega_2 \wedge \omega_3) \vee (\neg\omega_4 \wedge \neg\omega_5 \wedge \omega_6) \vee (\neg\omega_7 \wedge \neg\omega_8 \wedge \neg\omega_9) \tag{8}$$

Logic Programming in Hopfield Neural Network

HNN maintains the discrete nature of the difficulty and resolves it by minimising the energy function related to the outcome. HNNs are particularly good at pattern recognition (Fung et al., 2019) and defect identification (Pan et al., 2020). According to Little (1974), the dynamics of this model are asynchronous, with each neuron changing its state deterministically. According to most studies, HNN is regarded to have good properties, such as parallel execution for quick calculation and outstanding stability. Considering the structure of HNN is non-symbolic, the logical concept of 3SAT can enhance its ability with exceptional storage. The neuron's activation can be mathematically formulated as Equation 9:

$$\omega_i = \begin{cases} 1 & if \sum_j \varphi_{ij}\,\omega_i > \gamma \\ -1 & else \end{cases} \qquad [9]$$

where φ_{ij} denotes the weight for part j to i, along with γ implies the threshold value. This paper executes 3SAT in HNN called HNN-3SAT, where we only incorporate three neurons for each clause. The local field efficiently suppressed the obtained output before producing the final state. Equation 10 shows the formulation for the local field with $m = 3$.

$$h_i = \sum_k \varphi_{ijk}\,\delta_{\omega_j}\delta_{\omega_k} + \sum_j \varphi_{ij}\,\delta_{\omega_j} + \varphi_i \quad \text{, for } m = 3 \qquad [10]$$

These local fields will determine the functionality as well as the flexibility of the last states. As a result, the last interpretation will decide whether the result is overfitted. The updating rule remains as Equation 11.

$$\delta_{\omega_i}(t+1) = sgn[h_i(t)] \qquad [11]$$

The neuron connection is symmetric and zeroes diagonal. Such cases are as given in Equation 12:

$$\varphi_{ii}^{(2)} = \varphi_{jj}^{(2)} = \varphi_{kk}^{(2)} = \varphi_{iii}^{(3)} = \varphi_{jjj}^{(3)} = \varphi_{kkk}^{(3)} = 0 \qquad [12]$$

The structure of the generalised Lyapunov final energy of each variation of HNN-3SAT is in Equation 13:

$$E = -\frac{1}{3}\sum_i\sum_j\sum_k \varphi_{ijk}\,\delta_{\omega_i}\delta_{\omega_j}\delta_{\omega_k} - \frac{1}{2}\sum_i\sum_j \varphi_{ij}\,\delta_{\omega_i}\delta_{\omega_j} - \sum_i \varphi_i\,\delta_{\omega_i} \qquad [13]$$

The Lyapunov energy function is always minimised when HNN is used. The HNN energy landscape comprises a high-level-dimensional formation with hills and valleys (Lee & Gyvez, 1996).

Fuzzy Logic

The fuzzy logic development is built on fuzzy set theory, the development of classical set theory. A fuzzy clause is when each variable is true to a certain degree, which can be any real integer between zero and one. Fuzzy logic is a multivalued logic that allows for intermediate numbers within typical classification evaluations like correct or incorrect, yes or no, high or low, and so on (Badawi et al., 2022). As a result, when dealing with inconsistency and vagueness, fuzzy logic allows us to be more adaptable in our argument (Halaby & Abdalla, 2016). The truth numbers in Boolean can only be the binary numbers of zero or one, $x \in \{0,1\}$ (Novák et al., 1999). Meanwhile, fuzzy logic offers truth numbers between zero and $\mu \in (0,1)$.

A fuzzy set A is a function on universe X that matches into the range [0,1] and is probably bound to fit into a group such $\mu_A: X \rightarrow [0,1]$. As seen in Equation 14, the membership function of A is symbolised by the symbol μ_A:

$$A = \{(x, \mu_A(x)) | x \in X\} \qquad [14]$$

in which $0 \leq \mu_A(x) \leq 1$. The notation can be used for discrete universe X, as stated in Equation 15, to illustrate the set (Zadeh, 1973).

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_n)}{x_n} = \sum_{x \in X} \frac{\mu_A(x)}{x} \qquad [15]$$

where Σ implies union across all $x \in X$. The degree of membership of x in A is known as the value of $\mu_A(x)$.

Traditional formal logic has been effectively applied for computations problems such as in Horn clause form (Sathasivam & Abdullah, 2008) and has shown to be a powerful reasoning technique. While this type of logic is solid, it is also restricted and lacks expression. It is especially true when there is much ambiguity. Fuzzy logic provides an intriguing result to this dilemma because it allows for manipulating propositions containing ambiguity (Zadeh, 1974, 1979). Therefore, its clause needs the equivalent common structure as a standard clause, excluding its vagueness (Rhodes & Menani, 1992). Because of the complexity of many computation problems and the difficulty of coping with uncertainty, researchers have turned to fuzzy logic theory to solve optimisation problems (Nasir et al., 2021). Furthermore, the distinctive and valuable characteristics of fuzzy logic and the Hopfield network have caused each of these ways to reinforce itself by leveraging the strengths of both methods.

In fuzzy logic, connectives from classical logic are often linked to operators like conjunction, disjunction, implication, and negation (Brys et al., 2012). Table 1 shows the operands of fuzzy logic.

Table 1
Operators in fuzzy logic

Operators	Formula
Negation, NOT (¬)	$F_{\neg}(\omega_i) = 1 - \mu_{\omega_i}$
Disjunction, OR (∨)	$F_{\vee}(\omega_i, \omega_j) = max\{\omega_i, \omega_j\}$
Conjunction, AND (∧)	$F_{\wedge}(\omega_i, \omega_j) = min\{\omega_i, \omega_j\}$
Implication	$F_{\rightarrow}(\omega_i, \omega_j) = min\{1,1 - \omega_i, \omega_j\}$

A fuzzy logic system's basic design consists of a fuzzifier, rule evaluator, and defuzzifier. In the fuzzifier step, it converts crisp inputs into fuzzy sets. For OR and AND, fuzzy set operations assess Max and Min rules, respectively. The defuzzifier is a mapping stage in a fuzzy system that converts a fuzzy set into a crisp output. In the defuzzification process, the crisp production is generated using the alpha-cut form of a fuzzy collection. Because the alpha-cut method may extract the crisp value of a fuzzy set (Bodjanova, 2002), the theory of alpha-cut is crucial in combining fuzzy sets and crisp sets. Figure 1 shows the fundamental model of a fuzzy logic structure.

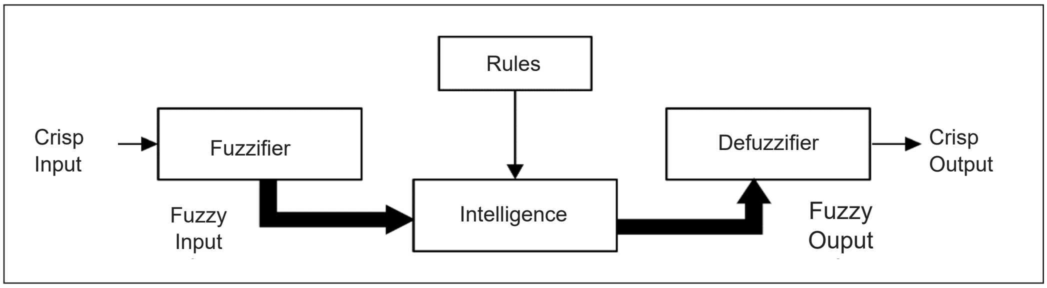

Figure 1. Fuzzy logic structure's fundamental design

3-Satisfiability Fuzzy Logic Hopfield Neural Network Using Abdullah (1992) Method

To assess synaptic weight methodically, we use a solid training strategy. According to Abdullah (1992), logic programming may be integrated into a neural network. When Abdullah's technique seeks the optimum solutions for the logic program's clauses, the subsequent results may shift as new clauses are introduced. As a result, we train the 3SAT using the Abdullah approach. The method was the first in formal synaptic weight derivation of superficial logical contradictions (Abdullah, 1992). Following that, Abdullah (1993) introduced a logic programming paradigm for Horn clauses in a neural network. Sathasivam (2010) expanded the work by introducing neural symbolic integration in HNN. Velavan et al. (2015) published the result on logic programming, which focused on logic programming for higher-order clauses using mean-field theory.

It is an excellent way to train the HNN, especially when accumulating synaptic weight. A set of systematic approaches are used in the training phase of 3SAT logic programming. The selection logic formula is critical for providing the neural network with good instructions. Synaptic weight will be computed using Abdullah (1992). A comprehensive learning approach calculates the corresponding synaptic weight using Boolean logical inconsistencies. The fuzzification and defuzzification algorithms will be connected with the network in the 3SAT programming to link a neuron's membership function to its identity. The 3SAT problem will be treated as an optimisation problem that HNN will solve. The clauses have a cost function that removes logical inconsistencies.

With the third-degree network, we created an HNN-3SAT method using the Abdullah approach during the training phase. The proposed method included a fuzzy logic technique to the network to improve the algorithm and name it HNN-3SATFuzzy. An HNN-3SATFuzzy algorithm is an integrated framework created using the fuzzification and defuzzification procedures until it achieves its final state. The last condition's stability was analysed to achieve a global minimum solution.

Fuzzification. The fuzzification technique connects a neuron's identity to its membership function. The function μ_{ω_X} defines a fuzzy set by pairing each component of the universe of discourse by its membership degree (Equation 16):

$$\mu_{\omega_i}(i): I \rightarrow [0,1] \qquad [16]$$

where $\mu_{\omega_i} = 0$ signifies that a part *i* does not belong to a fuzzy set, and $\mu_{\omega_i} = 1$ denotes that *i* is a component of a fuzzy set (Zadeh, 1974).

Fuzzy Rules. Union, intersection, and complement are all characterised and linked to membership functions for fuzzy sets. Allow the membership function of μ_{ω_i} where $i = 1,2$ ... n for $n = 9$ to express the fuzzy logical sets.

The membership function in Equation 17 is an example of one definition of the fuzzy union.

$$\mu_{\omega_1 \cup \omega_2}(x) = max\left[\mu_{\omega_1}(x), \mu_{\omega_1}(x)\right] \qquad [17]$$

Equation 18 implies the membership function of one definition of fuzzy intersection.

$$\mu_{\omega_1 \cap \omega_2}(x) = min\left[\mu_{\omega_1}(x), \mu_{\omega_1}(x)\right] \qquad [18]$$

Moreover, the membership function in Equation 19 implies a fuzzy complement.

$$\mu_{\overline{\omega_1}}(x) = 1 - \mu_{\omega_1} \qquad [19]$$

We can assess rules using fuzzy logic properties by transforming the 3SAT problem in Equation 5 into a fuzzy logic structure. Then, consider Equation 21 as the negation of Equation 20. The formulas are as follows:

$$\theta_{3SAT} = min\left[max\left(\mu_{\omega_1}, 1-\mu_{\omega_2}, 1-\mu_{\omega_3}\right), max\left(\mu_{\omega_4}, \mu_{\omega_5}, 1-\mu_{\omega_6}\right), max\left(\mu_{\omega_7}, \mu_{\omega_8}, \mu_{\omega_9}\right)\right] \quad [20]$$

$$\neg\theta_{3SAT} = max\left[min\left(1-\mu_{\omega_1}, \mu_{\omega_2}, \mu_{\omega_3}\right), min\left(1-\mu_{\omega_4}, 1-\mu_{\omega_5}, \mu_{\omega_6}\right), min\left(1-\mu_{\omega_7}, 1-\mu_{\omega_8}, 1-\mu_{\omega_9}\right)\right] \quad [21]$$

If the program allocates value $\mu_{\omega_i} \geq \alpha$ to be accurate and $\mu_{\omega_i} < \alpha$ to be false, then $\neg\theta_{3SAT} < \alpha$ indicates a consistent interpretation and $\neg\theta_{3SAT} > \alpha$ implies the clauses in the structure are not fulfilled.

Defuzzification. Aside from that, the HNN-3SATFuzzy algorithm will use the alpha-cut approach in the defuzzification phase to adjust the unsatisfied neuron clauses until the proper neuron state is determined. The defuzzifier is a mapping stage that converts a fuzzy value into a crisp output. It is referred to as a stable state when the state acquired is constant across both algorithms. The alpha-cut defuzzification technique, which is used to make the estimation, is stated in Equations 22 and 23:

$$\text{if } \mu_{\omega_i} \geq \alpha, then\ \delta_{\omega_{i\alpha}} = 1 \quad [22]$$

$$\text{if } \mu_{\omega_i} < \alpha, then\ \delta_{\omega_{i\alpha}} = 0 \quad [23]$$

Alpha-Cut. A subset of the universe with membership grades, μ which are greater than or equal to alpha, α for any $\alpha \in [0,1]$ is called an alpha-cut (Wang, 1996). The idea of alpha-cut is critical in connecting fuzzy and crisp sets. Sharpening produces a clean set, which is dependent on the alpha value. During defuzzification, neuron clauses will be adjusted using the alpha-cut method until the right neuron state is obtained (Pourabdollah et al., 2020). Equation 24 is the modified alpha-cut defuzzification:

$$\text{alpha-cut} = \frac{\sum_i \alpha_i \overline{\left[\mu_{\omega_i}\right]}}{\sum_i \alpha_i} \qquad i = 1, \dots L \quad [24]$$

where $\left[\mu_{\omega_i}\right]$ represents the average of the membership value of neurons, and L is the number of discretisation stages along the vertical axis.

The learning algorithm of the hybridised intelligent dynamic model of 3SAT using fuzzy logic in HNN is shown in Figure 2. The initial stage is constructing a 3SAT logic programming that must demonstrate inconsistency to prove a certain objective. After that, the process continues to convert the logic structure into a Boolean algebraic structure and negation. The advanced model includes the fuzzification method, which links a neuron's identity to its membership function. We can evaluate rules using fuzzy logic attributes by changing the Boolean model into a fuzzy structure. Next, we employ the alpha-cut defuzzification technique. The principle of alpha-cut is critical in the connection between fuzzy and crisp sets. The cost function is then required to determine the synaptic weights. The estimates of synaptic weights are then obtained by analysing the cost function in conjunction with the energy function. Finally, let the neural networks evolve until they reach a minimum energy state.

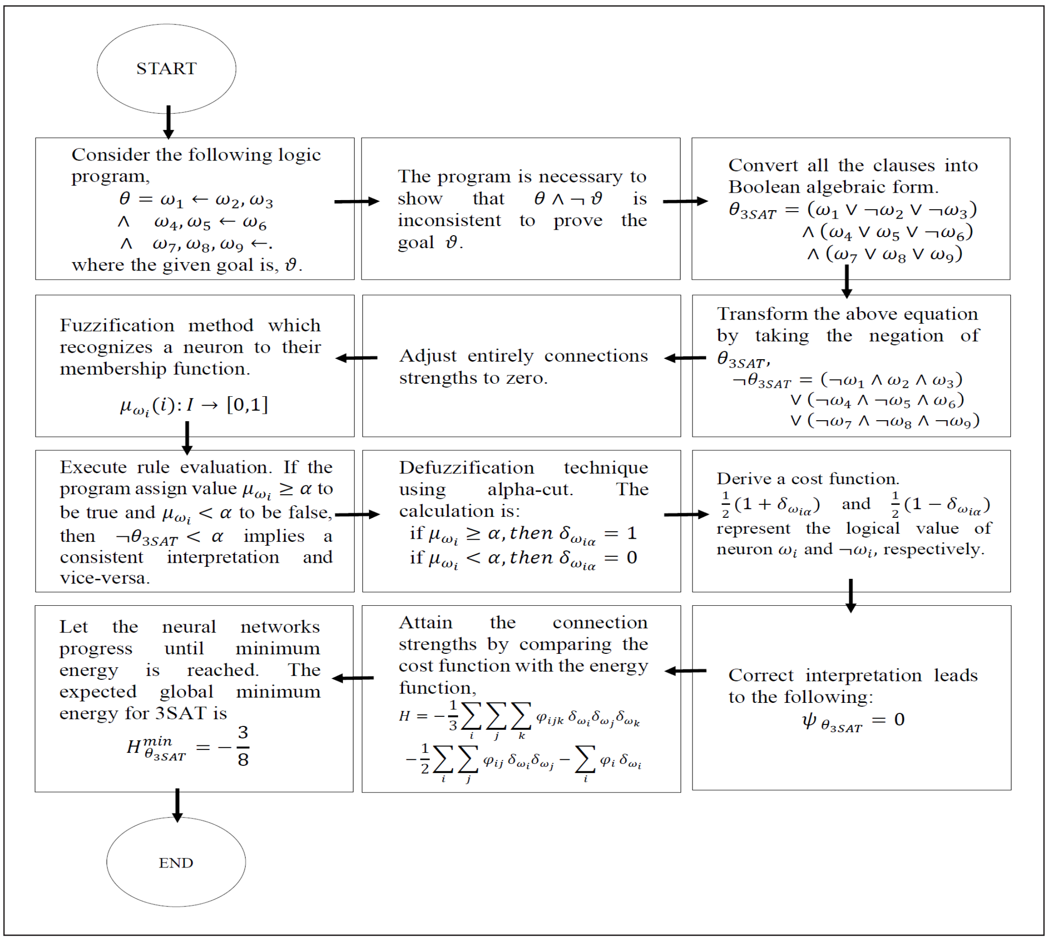

Figure 2. Algorithm in the learning phase for HNN-3SATFuzzy

Implementation and Experimental Setup

The discussion will cover the HNN-3SATFuzzy experimental simulation and algorithm descriptions (Table 2).

Table 2
Listing of related factors utilised in HNN-3SATFuzzy

Parameter	Value
Number of neurons	$9 \leq NN \leq 135$
Total of combinations	100
Tolerance measurement	0.001
CPU time threshold	24H
Activation function	HTAF
The initialisation of fuzzy membership neuron	$\mu_{\omega i} \in [0,1]$
Finalised neuron states	$\delta_{\omega i} \in [-1,1]$

The performance of the fuzzy logic techniques in training HNN to execute 3SAT is carried out using Matlab 2020b software in this experimental simulation. HNN-3SAT models integrated with fuzzy logic algorithm (HNN-3SATFuzzy) are the hybrid HNN models investigated in this work. The HNN models in this study used simulated datasets to generate 3SAT clauses with varying difficulty levels. The simulations of this experiment are carried out with varying numbers of neurons (NN) ranging from 9 to 135. The CPU time cutoff for generating data will be 24 hours (Kho et al., 2020), and if the CPU time exceeds 24 hours, the experiment will be aborted. Aside from that, we employed HTAF in this work because HTAF is regarded as an example of good quality activation functions to be developed in HNN based on its stability. Furthermore, the suggested network operates even if no activation function is used. The final energy execution requirements were set to 0.001 since this reduced statistical errors better (Sathasivam, 2010). The success of this study will be evaluated by comparing the accuracy and efficiency of two models: HNN-3SAT and HNN-3SATFuzzy.

The experiment is divided into three phases to validate the success of the suggested approach: training phase, retrieval phase, and energy analysis. As noted below, each subsection represents a distinct purpose. The list of the three subsections and their metrics may be found in Table 3.

Table 3
List of the phases and metrics used in all performance evaluation measures

Phases	Description	Metrics
Learning phase	to achieve ideal weight management through well-structured training programming.	$RMSE_{Learn}$ MAE_{Learn} SSE_{Learn} $MAPE_{Learn}$
Retrieval phase	to assess the quality of the HNN-3SATFuzzy generated solution	$RMSE_{Retrieve}$ $MAE_{Retrieve}$ $SSE_{Retrieve}$ $MAPE_{Retrieve}$
Energy analysis	to investigate the energy difference obtained by HNN-3SATFuzzy	N_{Local} N_{Global}

Figure 3 shows the summary flowchart of the successful integration of HNN-3SAT with fuzzy logic. Figure 3 shows how the HNN-3SATFuzzy is distributed into the learning and retrieval phases, with the fuzzy logic being implemented in the learning phase. The goal of the proposed network is to achieve the final global states of HNN-3SAT.

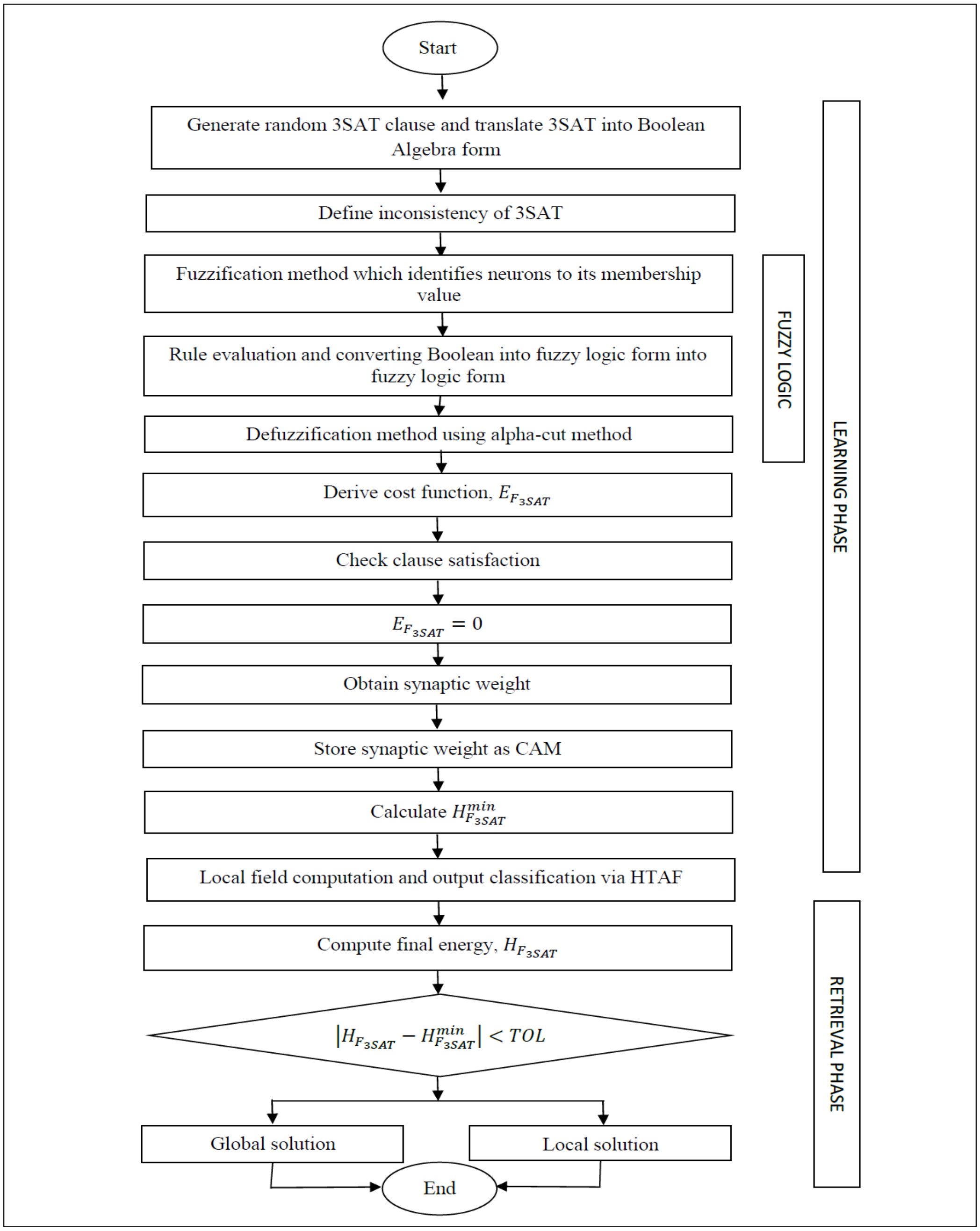

Figure 3. Flowchart of HNN-3SATFuzzy

Performance Evaluation Metrics for HNN-3SATFuzzy

Two measurement methods will be used to assess the competence of HNN-3SATFuzzy models, such as error analysis and energy analysis. The explanations for each metric will be discussed further down in greater detail.

Root Mean Square Error (RMSE). RMSE reports the actual divergence of the anticipated amounts and the computed value (Equation 25) (Willmott & Matsuura, 2005)

$$RMSE = \sum_{i=1}^{n} \sqrt{\frac{1}{n}\left(I_{highest} - I_x\right)^2} \qquad [25]$$

where I_x is the network's computed value, $I_{highest}$ is the network's highest value, and n is the total number of iterations.

Mean Absolute Error (MAE). By computing the disparity of the average gap between the calculated values and the expected values, MAE proves to be a good metric for analysing the model (Equation 26) (Alzaeemi & Sathasivam, 2021).

$$MAE = \sum_{i=1}^{n} \frac{1}{n}\left|I_{highest} - I_x\right| \qquad [26]$$

where I_x stands for the generated values, n for the number of iterations, and $I_{highest}$ for the most significant values by the network, which describes the number of clauses in the kSAT.

Sum Squared Error (SSE). SSE is a statistical technique for calculating how much the data deviates from expected values (Equation 27) (Bilal et al., 2012)

$$SSE = \sum_{i=1}^{n}\left(I_{highest} - I_x\right)^2 \qquad [27]$$

where I_x denotes the computed values, and $I_{highest}$ is the largest number, which relates to the value of the kSAT logic clauses.

Mean Absolute Percentage Error (MAPE). MAPE is a modified form of the MAE in which the results are normalised to a percentage (De Myttenaere et al., 2016). MAPE's formulation is given as Equation 28:

$$MAPE = \sum_{i=1}^{n} \frac{100}{n} \frac{\left|I_{highest} - I_x\right|}{\left|I_x\right|} \qquad [28]$$

where I_x is the network's computed value, $I_{highest}$ is the network's highest value, and n is the total number of iterations.

Global Minima Ratio (Zm). In prior research, global minima are employed to thoroughly investigate energy analysis (Alzaeemi et al., 2021; Mansor & Sathasivam, 2021). The energy process is also an indicator of the program's efficacy (Equation 29).

$$Global\ Minima = \frac{1}{NT.COMBMAX}\sum_{i=1}^{n} N \quad [29]$$

where NT is the total of testing, $COMBMAX$ is the combination of neurons, N is the number of global minima of the network. A network is believed to be strong if the amount of global minima is close to one.

CPU (Central Processing Unit) Time. Processing time generally refers to the total time to finish a simulation. The processing time is used to determine the robustness and stability. This investigation will employ the second SI unit for processing time. When the model's CPU period is reduced, the simulation's productivity is believed to be improved. Equation 30 shows the CPU time calculation:

Processing time = Training phase time + Retrieval phase time [30]

RESULTS AND DISCUSSION

Figures 4 to 9 show the performance of HNN-3SAT and HNN-3SATFuzzy in terms of RMSE, MAE, SSE, MAPE, global minima and CPU time, respectively.

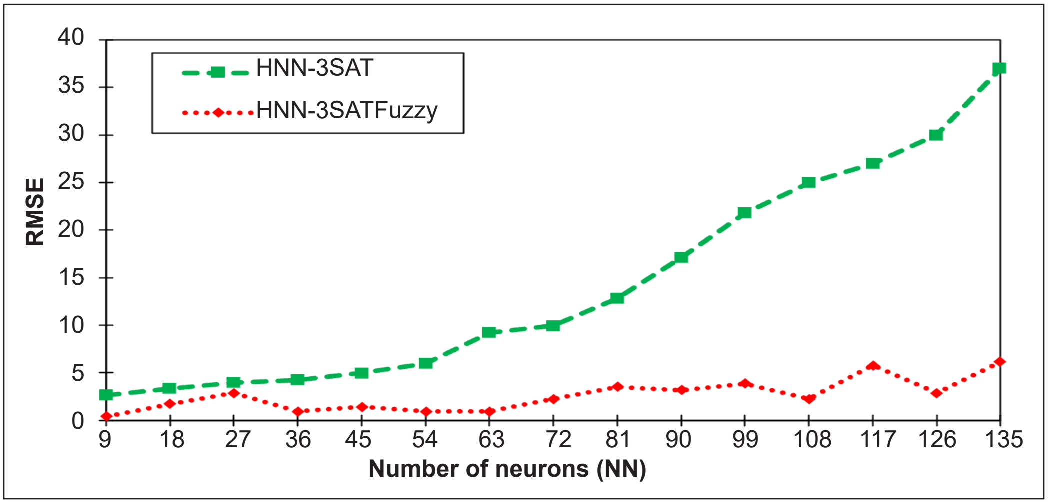

Figure 4. Root Mean Square Error (RMSE)

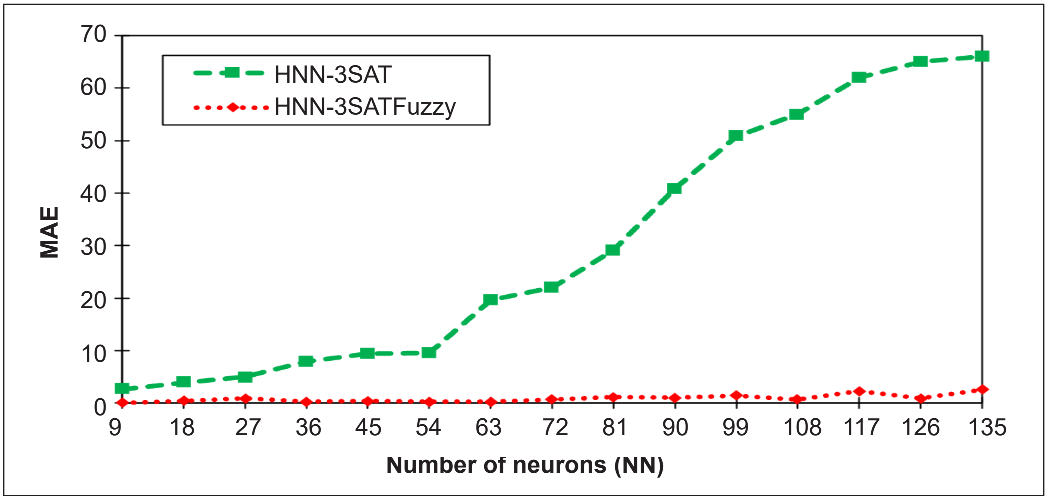

Figure 5. Mean Absolute Error (MAE)

The output of RMSE and MAE in the training stage for HNN-3SAT and HNN-3SATFuzzy are shown in Figures 4 and 5, respectively. During the training phase, HNN-3SATFuzzy beat its equivalent, HNN-3SAT, as per RMSE and MAE indicators. The results show that the RMSE and MAE values for HNN-3SATFuzzy are lower than the HNN-3SAT network, even as the total of neurons (NN) increases. As a result, the HNN-3SATFuzzy solutions diverged less from the potential solutions. At the start of the simulations, the outcomes for both networks appeared to have close results during $9 \leq NN \leq 45$. The performance for RMSE and MAE in HNN-3SAT seemed to rise once it reached $NN = 54$ slowly, and subsequently, the results rose significantly to about 600% and 2000%, respectively, towards the end of the simulations. The suggested technique, HNN-3SATFuzzy, achieves $\psi_{\theta_{3SAT}} = 0$ at lower results than HNN-3SAT, based on RMSE and MAE calculation. The fundamental reason is that 3SAT's fuzzy logic technique partitions solution is better, allowing $\psi_{\theta_{3SAT}} = 0$ to be obtained in fewer rounds. The fuzzy logic algorithm's increased likelihood of exploring for accurate interpretations during training is owing to it. Similarly, the HNN-3SATFuzzy used a systematic strategy using the fuzzification and defuzzification methods throughout the searching neuron stage. Furthermore, HNN-3SATFuzzy could check the correct interpretation efficiently and handle additional limitations compared to the other network.

Figures 6 and 7 show that HNN-3SATFuzzy has a lower SSE and MAPE value than HNN-3SAT. HNN-3SATFuzzy has a more robust capability to train the simulated data set than HNN-3SAT since it has a lower SSE value. It was clear that HNN-3SATFuzzy was found to have good quality results with a lower SSE value for all hidden neuron counts. Although at the beginning of the simulations, when $9 \leq NN \leq 45$ of SSE for both networks seemed to obtain almost similar results, the results for HNN-3SAT dramatically increased

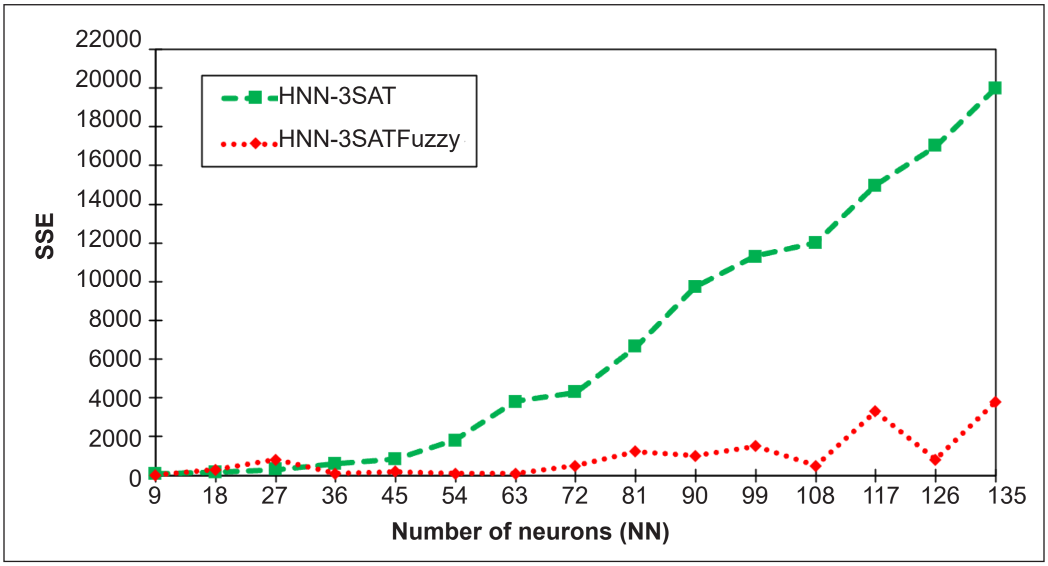

Figure 6. Sum Squared Error (SSE)

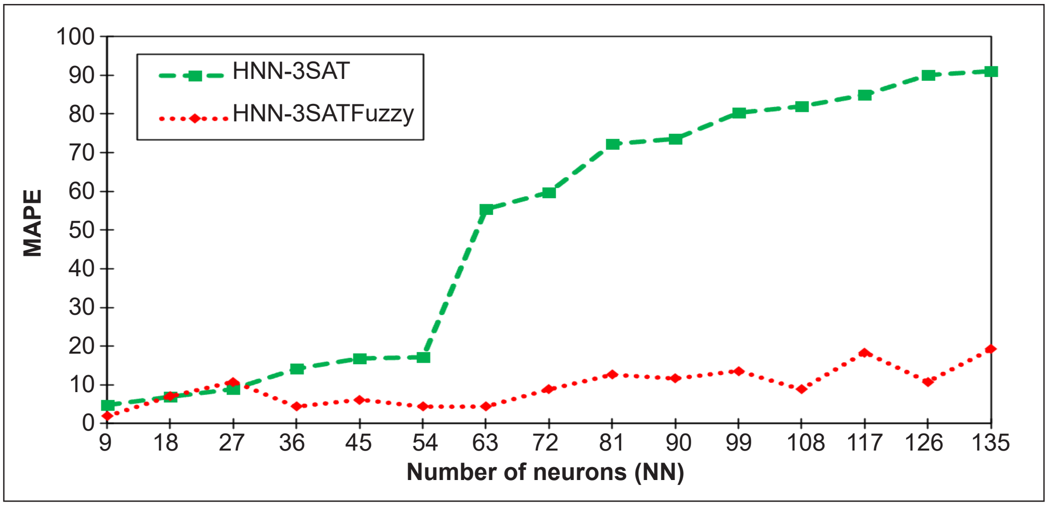

Figure 7. Mean Absolute Percentage Error (MAPE)

when it reached *NN* = 54 till the last simulations. The results of HNN-3SAT increased by about 500% compared to HNN-3SATFuzzy at the final NN, which makes it a poor network. Compared to HNN-3SAT and HNN-3SATFuzzy, a comparable output was obtained for the MAPE values. The MAPE value also has offered strong evidence of fuzzy logic's ability to work well with HNN-3SATFuzzy. The outcomes for HNN-3SAT significantly rose after it hit *NN* = 54. At *NN* = 135, the outcomes of HNN-3SAT converged by roughly 400% compared to HNN-3SATFuzzy. In conclusion, compared to the two results, the HNN-

3SATFuzzy method performs substantially better. It is owing to the efficient operators in the training phase, such as the fuzzification and defuzzification features of fuzzy logic, which increased the compatibility of the solutions. HNN-3SATFuzzy can recover a more accurate end state than HNN-3SAT.

For varied numbers of neurons, Figure 8 illustrates the global minima ratio recorded by HNN-3SAT and HNN-3SATFuzzy. Sathasivam (2006) discovered a link between the global minima value and the type of energy gained at the last part of the program. Given that the suggested hybrid network's global minima ratio is reaching value one, the results in the system have tentatively achieved minimum global energy except for HNN-3SAT. Compared to HNN- 3SATFuzzy have the potential to provide more exact and correct states. It is because the fuzzy logic algorithm's searching technique is very efficient. The HNN-3SATFuzzy solution has achieved the best minimum global energy of value 1. It is due to HNN's use of the fuzzy logic approach in conjunction with the 3SAT network. The proposed method can accept additional neurons since fuzzy logic reduces computing load by fuzzifying and defuzzifying the state of the neurons to find the appropriate states. Aside from that, during the defuzzification process, unsatisfied neuron clauses will be refined using the alpha-cut method until the correct neuron state is identified. Compared to the other network, this property effectively causes fuzzification and defuzzification techniques to converge to global minima. The network in HNN-3SAT becomes stuck in a suboptimal state when the number of neurons increases. The fuzzy logic algorithm has been shown to reduce the network's complexity, and in comparison to HNN-3SAT, the global minimum solutions of the HNN-3SATFuzzy converged to optimal solutions with beneficial results.

The calculation time is a critical metric or indicator for evaluating the effectiveness of our suggested algorithm. The efficacy of the entire calculation process can be used to

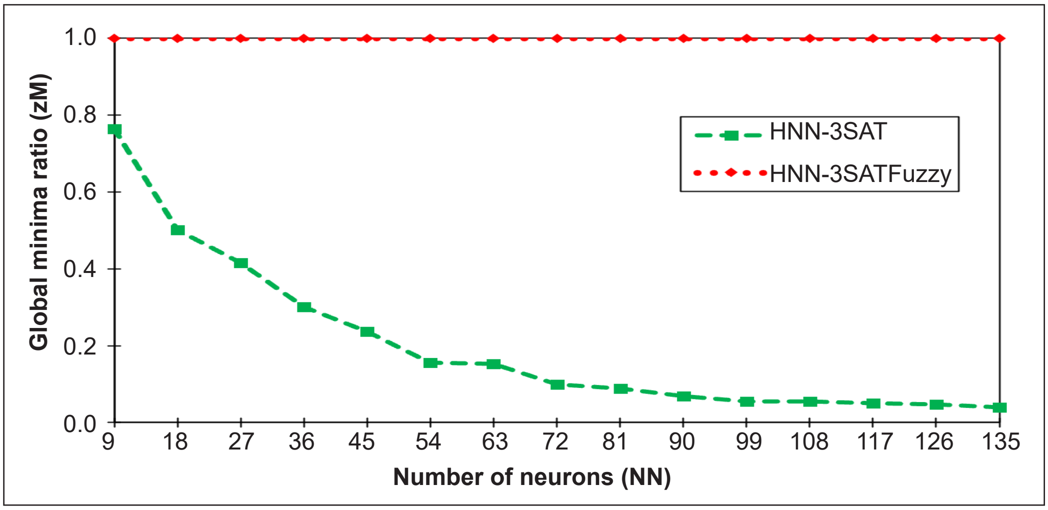

Figure 8. Global Minima Ratio (zM)

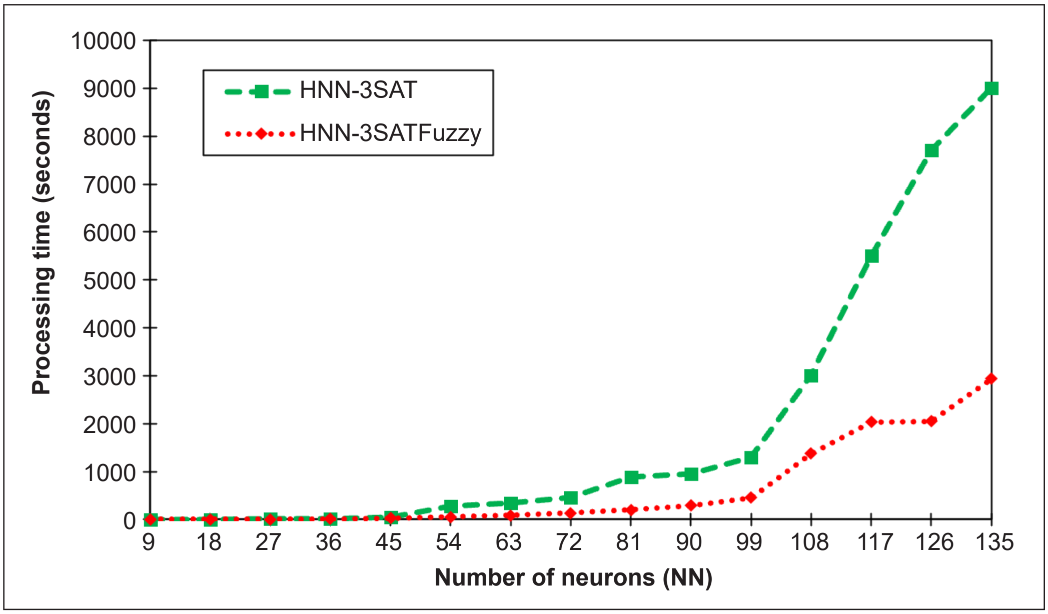

Figure 9. Processing time

indicate our techniques' robustness roughly. The computing time sometimes called the CPU period, can be described as the point it took our system to finish the entire calculation procedure in the investigation (Kubat, 1999). The computation method uses our suggested framework to train and generate the most satisfying phrases. The computing time for the HNN-3SAT and HNN-3SATFuzzy is displayed in Figure 9. As the number of neurons rose, the possibility of the identical neuron being implicated in an additional phrase increased (Sathasivam & Abdullah, 2008). As the network grew more extensive and complex, it was more likely to become stuck in local minima and consume more processing time. For all hybrid networks, the CPU moment rises as the number of neurons increases. Since the logical contradictions have been resolved, the HNN's rigorous search process will examine the appropriate option. As a result, a system that accelerates the training process is required. Furthermore, Figure 9 clearly shows that HNN-3SATFuzzy surpasses its contemporary HNN-3SAT. Even though the time spent by all networks for fewer clauses is not much different, the HNN-3SATFuzzy improved faster than the other network as the number of clauses for each amount of neurons rose. Due to the efficiency of the fuzzification and defuzzification methods, the CPU time was faster when the fuzzy logic technique was used. HNN-3SATFuzzy is slightly quicker than the HNN-3SAT network due to the potential to improve interpretations using fuzzy logic. The computation time was lowered when fuzzy logic was applied because the state of the fuzziness neurons was provided before starting the defuzzification process, which methodically turned the dissatisfied clause into a satisfied clause.

CONCLUSION

The findings proved that HNN-3SATFuzzy is a unique approach to increasing the efficiency of logic programming that integrates fuzzy logic and 3SAT in the Hopfield network. Fuzzification and defuzzification techniques with the alpha-cut approach were applied to improve the strategies of avoiding local minimum solutions and reducing the computer handling load of constructing the best results. When employing the HNN-3SATFuzzy to compute stability, the strength of this technique outperformed HNN-3SAT in terms of the error analysis, as stated in this publication. Furthermore, the suggested paradigm provides a global minima ratio of roughly one. The CPU time of the hybrid method is more rapidly compared to HNN-3SAT. As a result, the HNN-3SATFuzzy has proven to be more potent than the HNN-3SAT in 3SAT logic programming elements, such as better global minima ratio, constant lower error analysis values, and faster CPU time. The discoveries are crucial because the hybrid model considerably impacts Hopfield networks' capacity to solve difficulties with less complexity rapidly. The new knowledge and ideas will aid in developing creative approaches for extracting information in logic programming. Furthermore, dynamic learning advancement is considered a significant breakthrough in the neuro-symbolic field. The results of the suggested model have shown that the network is stable, making it clear that it is a long-lasting hybrid network. In the future, HNN-3SATFuzzy can be further improved to solve satisfiability problems by integrating with a metaheuristic algorithm.

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

Abdullah, W. A. T. W. (1992). Logic programming on a neural network. *International Journal of Intelligent Systems*, *7*(6), 513-519. https://doi.org/10.1002/int.4550070604

Abdullah, W. A. T. W. (1993). The logic of neural networks. *Physics Letters A*, *176*(3-4), 202-206. https://doi.org/10.1016/0375-9601(93)91035-4

Alzaeemi, S. A., & Sathasivam, S. (2021). Examining the forecasting movement of palm oil price using RBFNN-2SATRA metaheuristic algorithms for logic mining. *IEEE Access*, *9*, 22542-22557. https://doi.org/10.1109/ACCESS.2021.3054816

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. https://doi.org/10.5815/IJMECS.2021.02.03

Badawi, M. B., Awad, T. H., & Fahham, I. M. E. (2022). Application of artificial intelligence for the prediction of plain journal bearings performance. *Alexandria Engineering Journal*, *61*(11), 9011-9029. https://doi.org/10.1016/j.aej.2022.02.041

Bilal, M., Masud, S., & Athar, S. (2012). FPGA design for statistics-inspired approximate sum-of-squared-error computation in multimedia applications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *59*(8), 506-510. https://doi.org/10.1109/TCSII.2012.2204841

Bodjanova, S. (2002). A generalized α-cut. *Fuzzy Sets and Systems*, *126*(2), 157-176. https://doi.org/10.1016/S0165-0114(01)00062-8

Brys, T., Hauwere, Y. M. D., Cock, M. D., & Nowé, A. (2012, August 6-8). *Solving satisfiability in fuzzy logics with evolution strategies*. [Paper presentation]. 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), Berkeley, California. https://doi.org/10.1109/NAFIPS.2012.6290998

Fung, C. H., Wong, M. S., & Chan, P. W. (2019). Spatio-temporal data fusion for satellite images using hopfield neural network. *Remote Sensing*, *11*(18), Article 2077. https://doi.org/10.3390/rs11182077

Garcez, A. S. A., & Zaverucha, G. (1999). Connectionist inductive learning and logic programming system. *Applied Intelligence*, *11*(1), 59-77. https://doi.org/10.1023/A:1008328630915

Halaby, M. E., & Abdalla, A. (2016, May 9-11). Fuzzy maximum satisfiability. [Paper presentation]. INFOS '16: The 10th International Conference on Informatics and Systems, Giza, Egypt. https://doi.org/10.1145/2908446.2908476

Kubat, M. (1999). Neural networks: A comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. *The Knowledge Engineering Review, 13*(4), 409-412. https://doi.org/10.1017/s0269888998214044

Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, *52*(3), 141-152. https://doi.org/10.1007/BF00339943

Kho, L. C., Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. (2020). Logic mining in league of legends. *Pertanika Journal of Science and Technology*, *28*(1), 211-225.

Kowalski, R., & Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, *4*(1), 67-95. https://doi.org/10.1007/BF03037383

Lee, C. C., & Gyvez, J. P. (1996). Color image processing in a cellular neural-network environment. *IEEE Transactions on Neural Networks*, *7*(5), 1086-1098. https://doi.org/10.1109/72.536306

Little, W. A. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, *19*(1-2), 101-120. https://doi.org/10.1016/0025-5564(74)90031-5

Maandag, P. (2012). *Solving 3-SAT* [Bachelor dissertation]. Radboud University Nijmegen, Netherlands. https://www.cs.ru.nl/bachelors-theses/2012/Peter_Maandag___3047121___Solving_3-Sat.pdf

Mansor, M. A., & Sathasivam, S. (2021). Optimal performance evaluation metrics for satisfiability logic representation in discrete hopfield neural network. *International Journal of Mathematics and Computer Science*, *16*(3), 963-976.

Mansor, M. A., Sathasivam, S., & Kasihmuddin, M. S. M. (2018). 3-satisfiability logic programming approach for cardiovascular diseases diagnosis. *AIP Conference Proceedings*, *1974*(1), Article 020022. https://doi.org/10.1063/1.5041553

De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean absolute percentage error for regression models. *Neurocomputing*, *192*, 38-48. https://doi.org/10.1016/j.neucom.2015.12.114

Nasir, M., Sadollah, A., Grzegorzewski, P., Yoon, J. H., & Geem, Z. W. (2021). Harmony search algorithm and fuzzy logic theory: An extensive review from theory to applications. *Mathematics*, *9*(21), 1-46. https://doi.org/10.3390/math9212665

Novák, V., Perfilieva, I., & Močkoř, J. (1999). *Mathematical principles of fuzzy logic*. Springer. https://doi.org/10.1007/978-1-4615-5217-8

Pan, J., Pottimurthy, Y., Wang, D., Hwang, S., Patil, S., & Fan, L. S. (2020). Recurrent neural network based detection of faults caused byparticle attrition in chemical looping systems. *Powder Technology*, *367*, 266-276. https://doi.org/10.1016/j.powtec.2020.03.038

Pourabdollah, A., Mendel, J. M., & John, R. I. (2020). Alpha-cut representation used for defuzzification in rule-based systems. *Fuzzy Sets and Systems*, *399*, 110-132. https://doi.org/10.1016/j.fss.2020.05.008

Rhodes, P. C., & Menani, S. M. (1992). Towards a fuzzy-logic programming system: A 1st-order fuzzy logic. *Knowledge-Based Systems*, *5*(2), 106-116. https://doi.org/10.1016/0950-7051(92)90001-V

Sathasivam, S. (2006). *Logic mining in neural networks*. [Unpublished Doctoral Dissertation] Universiti Malaya, Malaysia.

Sathasivam, S. (2010). Upgrading logic programming in hopfield network. *Sains Malaysiana*, *39*(1), 115-118.

Sathasivam, S., & Abdullah, W. A. T. W. (2008). Logic learning in hopfield networks. *Modern Applied Science*, *2*(3), 57-63. https://doi.org/10.5539/mas.v2n3p57

Sathasivam, S., Mamat, M., Kasihmuddin, M. S. M., & Mansor, M. A. (2020). Metaheuristics approach for maximum k satisfiability in restricted neural symbolic integration. *Pertanika Journal of Science and Technology*, *28*(2), 545-564.

Velavan, M., Yahya, R. Z., Halif, M. N. A., & Sathasivam, S. (2015). Mean field theory in doing logic programming using hopfield network. *Modern Applied Science*, *10*(1), 154-160. https://doi.org/10.5539/mas.v10n1p154

Wang, L. X. (1996). *A course in fuzzy systems and control*. Prentice-Hall Inc.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*(1), 79-82. https://doi.org/10.3354/cr030079

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-3*(1), 28-44. https://doi.org/10.1109/TSMC.1973.5408575

Zadeh, L. A. (1974). The concept of a linguistic variable and its application to approximate reasoning. In K. S. Fu & J. T. Tou (Eds.), *Learning Systems and Intelligent Robots* (pp. 1-10). Springer. https://doi.org/10.1007/978-1-4684-2106-4_1

Zadeh, L. A. (1979). A theory of approximation reasoning. In J. E. Hayes, D. Mishie & L. I. Mikulish (Eds.), *Machine Intelligence* (pp. 149-194). Elservier.

Zamri, N. E., Alway, A., Mansor, M. A., Kasihmuddin, M. S. M., & Sathasivam, S. (2020). Modified imperialistic competitive algorithm in hopfield neural network for boolean three satisfiability logic mining. *Pertanika Journal of Science and Technology*, *28*(3), 983-1008.