BAREKENG: Journal of Mathematics and Its Applications
September 2026 Volume 20 Issue 3 Page 2413-2426
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss3pp2413-2426

ROBUSTNESS EVALUATION OF THE 3-SATISFIABILITY REVERSE ANALYSIS METHOD WITH DISCRETE HOPFIELD NEURAL NETWORK AND GENETIC ALGORITHM FOR TRAFFIC FLOW DATASET

Amierah Abdul Malik[1*], **Mohd. Asyraf Mansor**[2], **Nur Ezlin Zamri**[3], **Nurul Atiqah Romli**[4]

[1,2] *School of Distance Education, Universiti Sains Malaysia*
Penang, 11800 USM, Malaysia.

[3] *Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia*
Serdang, Selangor, 43400 UPM, Malaysia.

[4] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA*
Perlis Branch, Arau, 02600 Perlis, Malaysia.

*Corresponding author's e-mail: * **amierah@student.usm.my***

Article Info

Article History:

Received: 28th August 2025
Revised: 12th December 2025
Accepted: 17th March 2026
Published:8th April 2026

Keywords:

3-Satisfiability Reverse Analysis;
3-Satisfiability;
Discrete Hopfield Neural Network;
Genetic Algorithm;
Traffic flow.

ABSTRACT

Traffic flow congestion is a pervasive global phenomenon. Nonetheless, the systematic analysis and identification of traffic flow patterns remain a challenge as the volume of traffic data increases. Consequently, robust data extraction methods are required to uncover underlying data patterns. This paper proposes a 3-Satisfiability logic mining approach using a Discrete Hopfield Neural Network, develops the 3-Satisfiability Reverse Analysis method by integrating the Discrete Hopfield Neural Network with a Genetic Algorithm, and implements this method on traffic flow datasets, comparing its accuracy with existing approaches. The 3-Satisfiability Reverse Analysis method employs 3-Satisfiability for logical representation and integrates a Discrete Hopfield Neural Network with a Genetic Algorithm as its learning system. A simulation was conducted using the Urban Traffic dataset for São Paulo, Brazil. The robustness of the method in extracting relationships within traffic flow data was evaluated using selected performance metrics. The results indicated that the proposed 3-Satisfiability Reverse Analysis method, which integrates the Discrete Hopfield Neural Network and Genetic Algorithm, achieved promising performance with an accuracy rate of 80%, outperforming existing methods.

How to cite this article:

A. A. Malik, M. A. Mansor, N. E. Zamri and N. A. Romli., "ROBUSTNESS EVALUATION OF THE 3-SATISFIABILITY REVERSE ANALYSIS METHOD WITH DISCRETE HOPFIELD NEURAL NETWORK AND GENETIC ALGORITHM FOR TRAFFIC FLOW DATASET", *BAREKENG: J. Math. & App.*, vol. 20, no. 3, pp. 2413-2426, Sep, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*
Research Article · Open Access

1. INTRODUCTION

Urbanization and rapid industrial development have transformed cities into complex ecosystems where mobility is both an enabler of growth and a pressing challenge. Rapid economic expansion and population growth have intensified urban traffic congestion, further compounded by infrastructure development in industrialized regions [1], [2]. As cities expand, the demand for efficient transportation systems becomes increasingly urgent, not only to improve mobility but also to reduce congestion, lower emissions, and enhance the overall quality of urban life [3], [4]. Yet, despite significant advances in intelligent transportation systems, long-term traffic forecasting remains challenging due to the complexity of metropolitan systems [5]. Accurate prediction requires the ability to capture patterns of congestion that are often obscured by variability in human behavior, environmental factors, and infrastructure conditions. Identifying these recurring patterns of congestion is vital for improving efficiency and quality of life [6]. Therefore, providing systematic, interpretable methods will be beneficial for extracting and identifying traffic flow patterns.

Artificial Neural Networks (ANNs) are widely used in classification, clustering, and optimization tasks [7], [8]. The Discrete Hopfield Neural Network (DHNN), a recurrent associative memory model, exhibits strong learning capabilities but has not yet been applied to traffic data extraction [9], [10]. To strengthen its training, Genetic Algorithm (GA) is introduced as a metaheuristic optimizer, offering efficient exploration and incremental solution refinement [11], [12]. GA not only reduces computational load but also improves the robustness of DHNN.

The satisfiability representation (SAT) is a flexible formalism for constraint satisfaction problems across mathematics, computer science, and physics [11], [13]. The 3-SAT form, expressed in Conjunctive Normal Form (CNF) with three literals per clause, enables systematic transformation of datasets [9], [12]. Prior work has demonstrated *k*-Satisfiability Reverse Analysis (*k*-SATRA) in domains such as game analytics [14]. This work successfully identifies the best logical representation that determines how game objectives affect the game's outcome. However, the work by [14] focused mainly on 2-Satisfiability Reverse Analysis to extract outcomes from game datasets. The use of 2-Satisfiability limits each clause to two literals. Thus, this will limit exploration of the dataset because fewer literals are used to represent it. In addition, work by [14] focuses only on game datasets and is not tested on traffic datasets.

This paper addresses the challenge of identifying traffic behavior patterns by proposing the 3-Satisfiability Reverse Analysis Method with a Discrete Hopfield Neural Network and a Genetic Algorithm, referred to as DHHN-3SATGA. The proposed method integrates 3-SAT as the logical representation, DHNN as the computational intelligence, and GA as the optimizer for the DHNN's learning phase. The integration of DHNN-3SATGA will be employed to induce the best logical representation that can extract and capture broader traffic flow behavior.

This paper is organized in the following structure. Section 2 discusses detailed information on the 3-Satisfiability, Discrete Hopfield Neural Network, Genetic Algorithm, performance metrics used, and experimental design employed in this study. Section 3 presents the tabulated results, while Section 5 concludes the paper with its concluding remarks.

2. RESEARCH METHODS

2.1 3-Satisfiability

According to [7], Boolean satisfiability, often known as SAT logic, is a logical rule composed of sentences comprising literals or variables. This work used 3-SAT Boolean logic, which involves 3 literals per clause and bipolar values of 1 or -1. Numerous imperative optimization issues involving non-Boolean values can be interpreted as satisfiability problems [15].

Other than that, 3-SAT can be characterized as an equation in CNF in which every clause is constrained to a total of three literals or entirely three literals [13]. According to [15], any real dataset can be represented as a 3-SAT formula, which can be used to model 3-dimensional decision problems. Therefore, 3-SAT representation is chosen for this work. According to [11], the generalized form 3-SAT formula is shown in Eq. (1):

$$P_{3-SAT} = \Lambda_{i=1}^{n} T_i, \quad (1)$$

whereby T represents the clause and i indicates the total number of clauses involved in the 3-SAT formula. Furthermore, each of the clauses is connected by AND operators, and each literal in the clauses is connected by OR operators. Hence, the 3-SAT logical rule can be expressed in the following manner as an example:

$$P_{3-SAT} = (A \vee \neg M \vee I) \wedge (\neg E \vee R \vee \neg H) \wedge (L \vee \neg K \vee Z). \quad (2)$$

Eq. (2) shows the CNF form of the 3-SAT, with strictly 3 literals per clause. To be more specific, this paper limits the 3-SAT structure to only non-redundant literals in each clause. There are four fundamental aspects of 3-SAT for the CNF, which are listed as follows [11]:

1. Inside each of the clauses in the SAT formula is a group of h variables $x_1, x_2, \dots, x_n$. For this specific study, 3-SAT, $h = 3$ is used.
2. A collection of m different clauses included a Boolean formula,$\exists m: P = c_1 \wedge c_2 \wedge, \dots, c_m$.
3. A group of $l_{k,i}$ literals. Only 3 literals were considered for each clause in the 3-SAT equation. Each clause c_k is just composed of literals joined together by the OR operator, where $\forall l \leq k \leq m: c_k = (l_{k,1} \vee l_{k,2} \vee l_{k,3})$.
4. The literals can only be either the variable itself or the variable's negation, $\forall l \leq k \leq m, 1 \leq i \leq 3: l_{k,i} = x_p$ or $l_{k,i} = \neg x_p$ for $1 \leq p \leq n$.

2.2 Discrete Hopfield Neural Network

DHNN is a recurrent neural network variant capable of solving a wide variety of optimization problems [16]. DHNN is composed of neurons linked to each other so that all the neurons' outputs are relayed back to their respective inputs [17]. According to [18], DHNN has impressive features such as a recurrent network and parallelism, resulting in lower computational time, better memory, and stability. The ability of DHNN to learn and store information is represented by the link between neurons, which also represents the synaptic weights.

According to [19], DHNN makes use of the Content Addressable Memory (CAM) mechanism, which has the capacity to store an unlimited number of patterns to store the synaptic weights. Stored synaptic weights can be accessed as needed. The Hopfield nets are binary threshold units [20], which can take bipolar values such as 1 and -1. Thus, the fundamental overview of the activated neurons' state is shown in Eq. (3).

$$S_i = \begin{cases} 1, if \sum_j^N W_{ij}^{(2)} S_j \geq \theta_i \\ -1, otherwise \end{cases}, \quad (3)$$

where N is the number of neurons, $W_{ij}^{(2)}$ is the synaptic weight from i to j, S_i is the state of unit i and θ_i is the threshold of unit i. Reference [20] discovered that in the Hopfield net, the connections are typically symmetric or bidirectional and do not have any connection with itself, indicating that $W_{i,j} = 0$. $S_i \in \{-1,1\}$ are governed by the dynamics of $S_i \rightarrow sgn[h_i]$, the h_i functioning as the local field. Higher-order connections can be added to the connection model, which can be generalized as follows:

$$h_i = \sum_{j=1, i\neq j\neq k}^{N} \sum_{k=1, i\neq j\neq k}^{N} W_{ijk}^{(3)} S_j S_k + \sum_{j=1, i\neq j\neq k}^{N} W_{jk}^{(2)} S_j + W_i^{(1)} \quad (4)$$

where $W_{ijk}^{(3)}$ is the third-order synaptic weight, the second-order synaptic weight is $W_{jk}^{(2)}$ whereas $W_i^{(1)}$ is the first-order synaptic weight. The synaptic weight in DHNN is to ensure that Hopfield always maintains the symmetrical property and prevents a zero diagonal. Therefore, the revised rule is going to be:

$$S_i(t+1) = sgn[h_i(t)] \quad (5)$$

The function is referred to as the signum function, commonly denoted as sgn. The current rule decreases according to the dynamics, as per Eq. (5). In the conventional DHNN model, the framework's state progresses from any starting point to the final state, which is known as the local minimum of the Lyapunov energy function [13]. Lyapunov energy could also cater for the higher complexity of neurons as formulated in the following equation.

$$H_{P_{3-SAT}} = -\frac{1}{3}\sum_{i=1,i\neq j\neq k}^{N}\sum_{j=1,i\neq j\neq k}^{N}\sum_{k=1,i\neq j\neq k}^{N} W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} \quad (6)$$

Additionally, in determining the stability of DHNN, the Lyapunov energy equation is very effective. Additionally, this equation plays a vital role in determining DHNN's convergence. Thus, Eq. (6) is designed specifically for an adequate 3-SAT problem in DHNN. This paper incorporates the hyperbolic tangent activation function (HTAF). HTAF acts as the post-optimization technique to accelerate the proposed algorithm. According to [9], HTAF was used because it is the most robust for 3-SAT logic programming. Moreover, the logical inconsistency between the logic rule and the DHNN implementation will be minimized. HTAF is required to systematically update the neurons and successfully stabilize the final state of neurons. Reference [21] introduced HTAF as in Eq. (7).

$$g(h_i) = \begin{cases} 1, if\ \tanh(h_i) \geq 0 \\ -1, otherwise \end{cases}, \quad (7)$$

whereby

$$\tanh(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}}, \quad (8)$$

where h_i is the local field of the neurons.

2.3 Genetic Algorithms in Discrete Hopfield Neural Networks

Genetic algorithms (GA) are a widely acknowledged algorithm that derives inspiration from the biological process of evolution. This approach is based on the idea that the best solution will be found by the fittest models [12]. According to [22], Darwin described that an organism's survival can be maintained through the process of reproduction, crossover, and mutation. According to [8], the GA model is a renowned metaheuristic approach that effectively minimizes the computational complexity of optimization problems without the need for intricate mathematical formulae. Besides, GA incorporates several crucial operators that enhance the solution obtained.

To find a suitable solution to a particular problem, GA proceeds through a sequence of phases [23]. It begins by encoding the problem and calculating the fitness. Next, it selects the parent and mother using a roulette method. The algorithm then generates children with high fitness through crossover and mutation. Reference [22] pioneering work has demonstrated the effectiveness of GA in identifying feasible solutions to computationally challenging problems with significant complexity. Therefore, GA is used to facilitate the training phase of the 3-SATRA model. In this work, GA serves as a training algorithm to facilitate the clause-satisfaction checking process of the 3-SATRA model.

In this paper, the parameter tuning for all the operators involved is limited as shown in Table 4. Determining the appropriate value for each operator is important. Therefore, the parameter followed the parameter used in the work of [12]. The implementation of GA during the training phase of DHNN-3SAT is defined as DHNN-3SATGA. The process of GA involves 5 main stages [15]. Therefore, the stages involved in DHNN-3SATGA are classified as follows:

1. Chromosome Initialization
   a. Generate 100 chromosomes at random to represent bipolar interpretations.
2. Fitness Evaluation
   a. According to [12], the optimal assignment is determined by the highest number of satisfied clauses. The fitness evaluation is as follows:

$$f_{k-SAT} = c_1(x) + c_2(x) + c_3(x) + \cdots + c_{totalNC}(x) \quad (9)$$

   where $c_1(x), c_2(x), c_3(x), \ldots, c_{totalNC}(x)$ is the number of satisfied clauses.

   b. The number of satisfied clauses will lead to the cost function being zero. Eq. (10) shows the cost function:

$$E_{P_{3-SAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{NV} L_{ij} \tag{10}$$

where NC and NV denote as the number of clauses and variables, respectively. L_{ij} is given as follows:

$$L_{ij} = \begin{cases} \frac{1}{2}(1 - S_x), if\ \neg x \\ \frac{1}{2}(1 + S_x), if\ otherwise \end{cases}, \tag{11}$$

where S_x represents the truth values of x.

c. The main interaction of GA in DHNN happens in this stage. Table 1 shows examples of how to check the clause satisfaction by referring to the logical rule in Eq. (2).

Table 1. Clause Satisfaction of P_{3-SAT}

S_A	S_M	S_I	S_E	S_R	S_H	S_L	S_K	S_Z	c_1	c_2	c_3	$E_{P_{3-SAT}}$
-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	0
1	-1	-1	-1	-1	1	-1	1	1	1	1	1	0
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	0
-1	1	-1	-1	-1	1	-1	1	-1	-1	1	-1	2
1	-1	1	-1	-1	1	-1	1	-1	1	1	-1	1

3. Selection Stage

a. According to [15], the top 10 candidates of chromosomes with the highest fitness out of a total of 100 will be selected to improve the subsequent generation and stage in GA.

b. Subsequently, the selected chromosomes will be subjected to the crossover process to improve fitness and genetic diversity.

4. Crossover Operator

a. The premier genetic alteration process in GA involves the crossover process. The genetic data exchange between two chromosomal substructures happens here. The crossover position in this stage will be selected randomly.

b. Crossover ordinarily results in an increase in the number of fulfilled clauses within the new chromosome sets. This is because the primary goal of the crossover process is to broaden the possible fitness of offspring (new chromosome pairs). According to [15], the crossover process is demonstrated in Fig. 1.

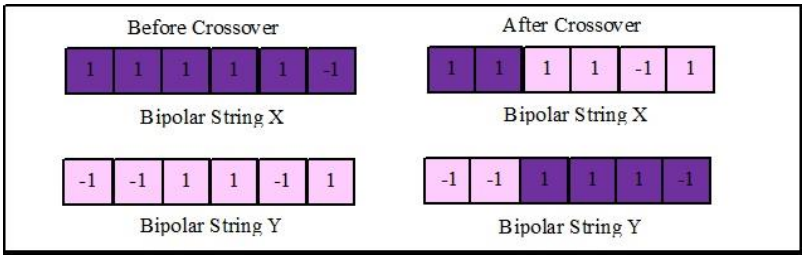

Figure 1. The Crossover Process

5. Mutation

a. In GA, the mutation entails changing the current state of the bit string from a value of 1 to -1 or vice versa [24]. Hence, the mutation's random chromosomal position is set. The best chromosomes will also be created following the mutation operator.

b. Newly created chromosomes' fitness will be assessed and quantified. Finally, whenever the fitness values have not yet reached the highest fitness level, the primary phase shall be rehashed. Table 5 lists the parameters in DHNN-3SATGA.

2.4 Performance Evaluation Metrics

In this research, five performance evaluation metrics will be used to assess the capability of the DHNN model. The DHNN model will be selected based on RMSE, MAE, SSE, MAPE, and network accuracy. RMSE, MAE, SSE, and MAPE are effective error measures for evaluating our model, as they directly address the residuals it produces. The CPU Time is utilized to evaluate the durability of the models. Root Mean Square Error (RMSE) measures the discrepancy between the expected and calculated values [25]. In this research, we have modified RMSE to comply with our problem and to imitate the DHNN model, as follows:

$$RMSE = \sum_{i=1}^{n} \sqrt{\frac{1}{n}(f_{max} - f_i)^2}\,. \tag{12}$$

Mean Absolute Error (MAE) is widely used for its ability to estimate error. Reference [14] calculated the MAE by considering the absolute value of the difference between the predicted values and real values over the number of iterations. Eq. (13) shows the formula for MAE:

$$MAE = \sum_{i=1}^{n} \frac{1}{n} |(f_{max} - f_i)|\,. \tag{13}$$

In regression analysis, the Sum of Squared Errors (SSE) is a measure of the dispersion of the data points. The standard SSE by [26] has been recrafted as Eq. (14).

$$SSE = \sum_{i=1}^{n} (f_i - f_{max})^2\,. \tag{14}$$

The Mean Absolute Percentage Error (MAPE) metric has been employed to assess the discrepancy between the actual and global solutions [25]. The evaluation of MAPE can be expressed as

$$MAPE = \sum_{i=1}^{n} \frac{1}{n} \frac{|f_{max} - f_i|}{f_i}. \tag{15}$$

Computational Time (CPU Time) denotes the duration necessary for the DHNN-3SATGA model to complete a single execution. The DHNN-3SATGA model's stability and competence can be inferred from its CPU Time. The CPU Time is calculated as follows.

$$CPU\ TIME = TrainingTime(s) + RetrievalTime(s). \tag{16}$$

CPU Time is important in model execution because it indicates network complexity across simulation iterations. A faster-executing model will benefit the network. The accuracy (Q) is used to determine the correct solutions obtained. Moreover, accuracy represents the degree to which the solutions generated by the induced logic align with successful outcomes for a real dataset. The accuracy of the network is described in Eq. (17).

$$Q = \frac{Correct\ P_{induce}}{Total\ P_{test}}. \tag{17}$$

Table 2. List of Parameters in Performance Evaluation Metrics

Parameter	Parameter Name
f_i	Fitness measured
f_{max}	Maximum fitness
n	Number of iterations before $f_i = f_{max}$
P_{induce}	Induced logic
P_{test}	Testing data

2.5 Experimental Setup

2.5.1 Datasets Information

The UCI Machine Learning Repository was used to get the Urban Traffic of the City of Sao Paulo in Brazil (UTCSPB) traffic flow dataset that is used for the simulations. This study used 9 attributes, and the real dataset used multivariate data. The goal of this experiment is to compare the accuracy of DHNN-3SATGA with that of other models, such as E-2SATRA and 3-SATRA, and with that of researchers who used the same dataset. Table 3 and Table 4 provide details about the UTCSPB dataset.

Table 3. Method Used in the Existing Model

Dataset	Method
UTCSPB	Rough-Neuro Fuzzy Network (RNFN) – [27]
	Energy-Based Logic Mining (E-2SATRA) – [28]
	3-Satisfiability Reverse Analysis (3-SATRA) – [15]

Table 4. List Of Attributes for Each Dataset

Dataset	Details of Each Attribute			Output P_{3-SAT}
UTCSPB	A	:	Immobilized Bus	To identify the causes of slowness in road traffic flow
	B	:	Broken Truck	
	C	:	Hour	
	D	:	Occurrence Involving Dangerous Freight	
	E	:	Fire Vehicles	
	F	:	Tree on the road	
	G	:	Lack of electricity	
	H	:	Point of flooding	
	I	:	Vehicles Excess	

2.5.2 Baseline Method

The efficacy of the proposed method can be verified by comparing the performance of the DHNN-3SATGA model with several established contemporary works. The chosen comparison method varies in whether it uses the concept of induced logic. DHNN-3SATGA, E-2SATRA [28] and 3-SATRA [15] used the concept of induced logic, while RNFN [27]did not use the concept of induced logic.

The E-2SATRA approach, used in reference [28], has been effectively utilized as a feasible substitute for extracting significant correlations among attributes, resulting in reduced error and improved accuracy. The E-2SATRA approach employs energy-based logic mining to guarantee that the induced logic variable consistently adheres to the Lyapunov function's dynamics.

3-SATRA adheres to the methodology proposed by [15], which employs a randomized attribute selection approach to identify attributes that effectively represent 3-SAT clauses.

Reference [27] uses Rough-Fuzzy Sets to describe inference structures, enabling the behavior to be captured in a structured rule base without the help of a human expert. Rough Sets Theory identifies the most important characteristics and suggests adding fuzzy relations to a Rough Neuro-Fuzzy Network (RNFN) of type Multilayer Perceptron (MLP) to achieve the best results. The primary benefit of using the RNFN is its ability to reduce reliance on human experts for selecting and developing inference rule mechanisms.

2.5.3 Parameter Settings

DHNN-3SATGA were executed on DEV C++ Version 5.11. Dev. C++ is chosen as the platform for this work because it is open source and, more importantly, it is a user-friendly interface. The hybrid networks are simulated by using the same processor to avoid any unfairness during the experiment. Output from the DHNN-3SATGA model that exceeds the CPU Time limit of 24 hours will be omitted [18]. Table 5, Table 6,

and Table 7 provide a summary of the parameter assignments for the DHNN-3SATGA, 3-SATRA, and E-2SATRA models.

Table 5. List of Parameters in DHNN-3SATGA

Parameter	Parameter Value
Neuron Combination	100
Number of Chromosomes	100
Selection Rate	0.1
Mutation Rate	0.01
Crossover Rate	0.9
Generation	1000
Type of Crossover Point	1

Table 6. List of Parameters in 3-SATRA, [12]

Parameter	Parameter Value
Neuron Combination	100
Number of Trials	100
Tolerance value	0.001
Attribute Selection	Random
No of attributes	9
No of clause	3

Table 7. List of Parameters in E-2SATRA, [28]

Parameter	Parameter Value
Neuron Combination	100
Number of Trials	100
Tolerance value	0.001
Attribute Selection	Random
No of attributes	6
No of clause	3

2.5.4 Simulation Design

The DHNN-3SATGA utilizes the 3-SATRA to produce an upgraded induced logical rule from the traffic flow dataset. The current scenario involves translating unprocessed data into 3-SAT. The unprocessed data value will be transformed into bipolar form. $S_i = \{-1,1\}$ using k-mean clustering [25]. Furthermore, the simulation employs the train-split technique in which the ratio of the testing and training data is 6:4, respectively [25]. DHNN-3SATGA starts with dividing the dataset into training and testing phases, with the training phase focused on generating the best logical rule P_{best}. Then, the generated P_{best} then will be used to derive the cost function. The Genetic Algorithm is then applied to optimize the P_{best}. Once optimized, the corresponding synaptic weights are computed and stored in the CAM. The testing phase starts with applying the energy relaxation process to minimize the energy function and reach a stable neuron state, which results in the generation of induced logic P_{induce}. The P_{induce} is then compared to the actual test data P_{test}. The performance of DHNN-3SATGA during the training phase is then assessed using RMSE, MAE, SSE, and CPU time. In addition, the accuracy of DHNN-3SATGA is compared with that of existing work. Fig. 2 shows the overall flow of the DHNN-3SATGA.

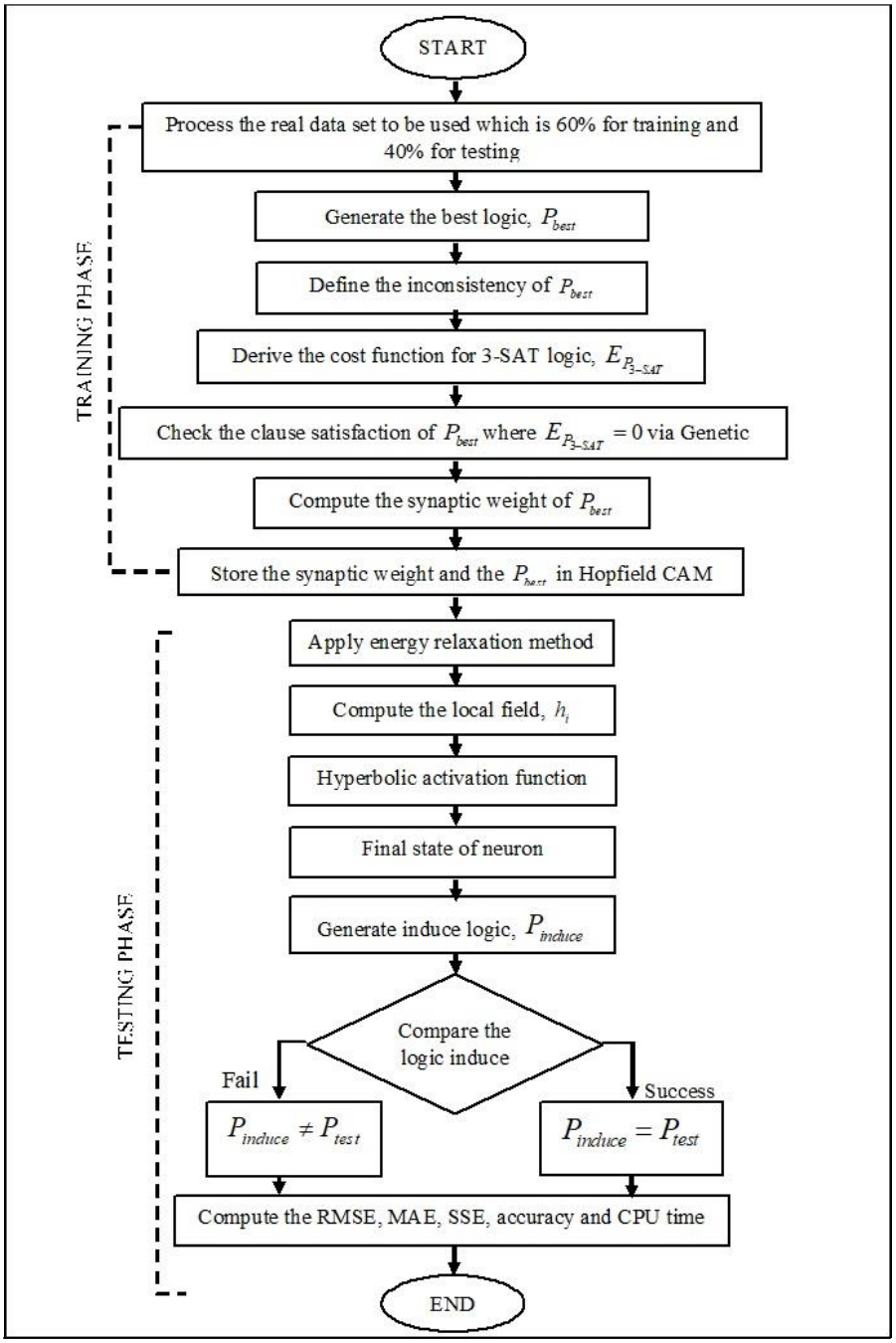

Figure 2. The Overall Flow of the DHNN-3SATGA

3. RESULT AND DISCUSSION

This part will explain the results DHNN-3SATGA. During the training phase, DHNN-3SATGA will involve a distinct number of clauses, NC. Note that, 60.00% of the overall data points will be utilized for the training phase. Meanwhile during the testing phase, the remaining 40.00% of data points will be used.

This UTCSPB dataset records the behavior of the Urban Traffic for Sao Paulo, Brazil from Monday to Friday starting from 7:00 am to 20:00 pm. The behavior of the dataset is recorded every 30 minutes. The dataset comprised 135 instances with 9 attributes. The goal of this research was to create and assess a prediction model that can be used to determine which attributes contribute to the slowness of road traffic flow. The DHNN-3SATGA algorithm is employed to induce the optimal logical rule to identify the underlying factors contributing to the slowness of road traffic flow. This research incorporates a total of 83 data points within the DHNN-3SATGA for the purpose of learning, and another 54 data points for testing. Table 8 shows the training error and CPU Time of DHNN-3SATGA.

Table 8. Training Error and CPU Time for UTCSPB

NC	RMSE	MAE	SSE	MAPE	CPU Time
1	1.12	0.75	5.00	25.00	0.88
2	2.29	1.88	327.71	31.30	3.78
3	3.45	2.84	1300.17	31.56	8.97
4	4.33	3.44	3188.27	28.68	22.01
5	5.31	4.21	7027.91	28.07	41.21
6	5.94	4.56	12797.45	25.33	70.12
7	6.77	5.15	22095.00	24.54	113.50

Note that the aim of the training phase is to find satisfied clauses. Different NCs are being incorporated to test DHNN-3SATGA's ability to handle larger NC values. Based on Table 8, the trend of training errors is increasing as the network complexity is also increasing with NC. The RMSE and MAE recorded from $\mathrm{NC} = 1$ to $\mathrm{NC} = 7$ is not significantly massive. This is due to the operators such as crossover and mutation in GA, which have improved the solutions. The crossover and mutation operators in GA are being used to find satisfied clauses. Thus, lower values of RMSE and MAE indicate that the crossover and mutation operators have successfully found satisfied clauses. In addition, at $\mathrm{NC} = 2$, the value of SSE is 327.71 which signifies the sensitivity error of the network. Besides that, $\mathrm{NC} = 3$ recorded the highest value of MAPE which means about 31. 56% of the iterations in DHNN-3SATGA have a cost function for 3-SAT that is not zero. The highest value of MAPE recorded is less than 50%. This result again shows the prominent role of the crossover and mutation operators in finding satisfied clauses.

Additionally, CPU time increases as NC increases. The CPU Time recorded for $\mathrm{NC} = 1$ until $NC = 6$ is considerably still fast, under 100 seconds. The crossover and mutation operators in GA have reduced the number of iterations required to achieve convergence during the training phase. Thus, the CPU Time will be faster for these reasons.

The performance of DHNN-3SATGA in the testing phase is measured by the accuracy (Q) achieved. The Q obtained by DHNN-3SATGA is 80.00%. During the testing phase, DHNN-3SATGA will induce a logical rule that represents the pattern of the UTCSPB dataset. Eq. (18) shows the induced logical rule generated by DHNN-3SATGA:

$$P_{induce} = (A \vee \neg B \vee C) \wedge (\neg D \vee E \vee F) \wedge (\neg G \vee H \vee \neg I). \quad (18)$$

Eq. (18) represents the overall pattern of the UTCSPB dataset. Equation (18) will be compared to 40.00% of the testing dataset. If Eq. (18) matches the test dataset, then DHNN-3SATGA has successfully extracted the pattern of the UTCSPB dataset. In addition, DHNN-3SATGA performed quite well because DHNN-3SATGA managed to achieve 80.00% of Q. The DHNN-3SATGA demonstrates its ability to extract the most effective logical rule that can describe the relationship between attributes. This is because the DHNN-3SATGA can achieve a Q value of over 50.00% on the given dataset. Table 9 presents the attribute-range classification for the UTCSPB dataset.

Table 9. The Attributes Range Classification for Urban Traffic for Sao Paulo, Brazil

Literal	Attribute	State	Range
A	Immobilized Bus	1	≥1.31
		-1	<1.31
B	Broken Truck	1	≥1.66
		-1	<1.66
C	Hour	1	≥20.50
		-1	<20.50
D	Occurrence Involving Dangerous Freight	1	≥1.00
		-1	<1.00

Literal	Attribute	State	Range
E	Fire Vehicles	1	≥1.00
		-1	<1.00
F	Tree on the Road	1	≥1.00
		-1	<1.00
G	Lack Of Electricity	1	≥1.00
		-1	<1.00
H	Point of Flooding	1	≥0.02
		-1	<0.02
I	Vehicles Excess	1	≥1.00
		-1	<1.00

Each attribute is assigned a different literal. The range in Table 9 will indicate whether the attribute contributes to traffic slowness. The range is categorized by k-means clustering [25]. Literals can only be either the variable itself or the variable's negation. Eq. (18) consists of five variables by itself and four variables as negation. The induced logical rule generated by DHNN-3SATGA can be explained by referring to both Eq. (18) and Table 9. If the city has at least one excess vehicle, a tree on the road, a lack of electricity, fire vehicles, or occurrences involving dangerous freight, then these attributes will cause traffic to slow. The slowness of the road traffic flow will also occur when the flooding point is greater than or equal to 0.02. Besides that, the range for hour attributes is 20.50 or higher, indicating the traffic flow will start to slow down around 17.00 pm. Note that the traffic flow behavior is recorded every 30 minutes; thus, the range of 20.50 is around 17.00 pm in a 24-hour system. Finally, immobilized buses and broken trucks will slow road traffic if the range is greater than or equal to 1.31 and 1.66, respectively. Therefore, the result shows that immobilized buses, peak hour, fire vehicles, trees on the road, and points of flooding are the attributes that affect the slowness in road traffic flow, while broken trucks, occurrences involving dangerous freight, and lack of electricity show less effect on the slowness in road traffic flow.

Table 10 compares the Q of DHNN-3SATGA with that of the existing model. The existing models are RNFN, E-2SATRA, and 3-SATRA. RNFN is proposed by [27]. Rough set theory, a characteristic of the RNFN model, provides the pertinent features that should be added to RNFN to get the best performance [27]. The RNFN model used 16 attributes to achieve 74.00% accuracy. E-2SATRA was introduced by [28]. This method emphasizes energy analysis throughout the DHNN retrieval stage. E-2SATRA uses the fewest attributes. E-2SATRA uses only 6 attributes, and its accuracy cannot exceed 50.00%. On the other hand, 3-SATRA is a logic mining model that is similar to the proposed DHNN-3SATGA. However, 3-SATRA achieves only 37% accuracy. This shows that the significance of a metaheuristic algorithm like GA is that it facilitates the training algorithm.

The proposed model DHNN-3SATGA enhances its accuracy by implementing a GA metaheuristic during training. GA incorporates essential operators, including crossover and mutation, that enhance the clause-satisfaction checking process during the training phase of the DHNN-3SATGA model. The crossover and mutation stages in the GA are the most vital, as the best logical representation is produced during them. The performance of DHNN-3SATGA is incomparable to that of other existing work.

By using 9 attributes, DHNN-3SATGA achieves up to 80.00% accuracy. According to Table 10, the models that compete with Q are RNFN, E-2SATRA, and 3-SATRA. Therefore, the proposed model effectively extracts information on the traffic flow of the dataset.

Table 10. Accuracy of DHNN-3SATGA with other Existing Models.

Model	Accuracy (Q)
DHNN-3SATGA	80.00%
RNFN [27]	74.00%
E-2SATRA [28]	44.00%
3-SATRA [15]	37.00%

4. CONCLUSION

The DHNN-3SATGA model was successfully implemented with the traffic flow datasets. The results of the traffic flow dataset indicate that DHNN-3SATGA is proficient at extracting optimal logic that thoroughly explains the pattern of traffic flow data. DHNN-3SATGA showed promising results across RMSE, MAE, SSE, MAPE, accuracy, and CPU Time. The proposed DHNN-3SATGA outperforms the existing model, indicating that the dataset performs well with the model. In conclusion, we have successfully identified the factors contributing to traffic flow congestion. The extracted factors would benefit transportation authorities and planners. The utilization of this model enables the prediction of traffic conditions and the enhancement of traffic flow management. Since DHNN-3SATGA outperformed the existing method, extracting medical datasets or agriculture datasets using the DHNN-3SATGA model as follow-up research will be interesting.

Author Contributions

Amierah Abdul Malik: Formal Analysis, Writing-Original Draft. Mohd. Asyraf Mansor: Validation, Funding Acquisition and Supervision. Nur Ezlin Zamri: Conceptualization and Review. Nurul Atiqah Romli: Methodology and Editing. All authors have read and agreed to the published version of the manuscript.

Funding Statement

This research is financially supported by a Bridging Grant, Universiti Sains Malaysia, with Project Code: R501-LR-RND003-0000002087-0000.

Acknowledgment

The research team would like to express our deepest gratitude to the individuals and institutions that have made important contributions to the success of our research. The support, guidance, and collaboration provided have been invaluable to the smooth progress and success of this research.

Declarations

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Declaration of Generative AI and AI-assisted technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement (grammar, spelling, and clarity). The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. The authors reviewed and approved all final text.

REFERENCES

[1] G. Li, H. Deng, and H. Yang, "TRAFFIC FLOW PREDICTION MODEL BASED ON IMPROVED VARIATIONAL MODE DECOMPOSITION AND ERROR CORRECTION," *Alexandria Engineering Journal*, vol. 76, pp. 361–389, 2023, doi: https://doi.org/10.1016/j.aej.2023.06.008.

[2] S. Wang, Y. Zhang, Y. Hu, and B. Yin, "KNOWLEDGE FUSION ENHANCED GRAPH NEURAL NETWORK FOR TRAFFIC FLOW PREDICTION," *Physica A: Statistical Mechanics and its Applications*, vol. 623, p. 128842, 2023, doi: https://doi.org/10.1016/j.physa.2023.128842.

[3] Q. Chen, W. Wang, X. Huang, and H. N. Liang, "ATTENTION-BASED RECURRENT NEURAL NETWORK FOR TRAFFIC FLOW PREDICTION," *Journal of Internet Technology*, vol. 21, no. 3, pp. 831–839, 2020.

[4] T. Jia and P. Yan, "PREDICTING CITYWIDE ROAD TRAFFIC FLOW USING DEEP SPATIOTEMPORAL NEURAL NETWORKS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3101–3111, 2021, doi: https://doi.org/10.1109/TITS.2020.2979634.

[5] S. Sun, H. Wu, and L. Xiang, "CITY-WIDE TRAFFIC FLOW FORECASTING USING A DEEP CONVOLUTIONAL NEURAL NETWORK," *Sensors*, vol. 20, no. 2, p. 421, 2020, doi: https://doi.org/10.3390/s20020421.

[6] J. Chen, L. Zheng, Y. Hu, W. Wang, H. Zhang, and X. Hu, "TRAFFIC FLOW MATRIX-BASED GRAPH NEURAL NETWORK WITH ATTENTION MECHANISM FOR TRAFFIC FLOW PREDICTION," *Information Fusion*, vol. 104, p. 102146, 2024, doi: https://doi.org/10.1016/j.inffus.2023.102146

[7] N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, A. Alway, S. Z. M. Jamaludin, and S. A. Alzaeemi, "AMAZON EMPLOYEES' RESOURCES ACCESS DATA EXTRACTION VIA CLONAL SELECTION ALGORITHM AND LOGIC MINING APPROACH," *Entropy*, vol. 22, no. 6, p. 596, 2020, doi: https://doi.org/10.3390/e22060596
[8] S. Sathasivam, M. Mamat, M. S. M. Kasihmuddin, and M. A. Mansor, "METAHEURISTICS APPROACH FOR MAXIMUM K SATISFIABILITY IN RESTRICTED NEURAL SYMBOLIC INTEGRATION," *Pertanika J. Sci. Technol*, vol. 28, no. 2, pp. 545–564, 2020.
[9] M. A. Mansor and S. Sathasivam, "ACCELERATING ACTIVATION FUNCTION FOR 3-SATISFIABILITY LOGIC PROGRAMMING," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 10, pp. 44–50, 2016, doi: https://doi.org/10.5815/ijisa.2016.10.05
[10] M. S. M. Kasihmuddin, M. A. Mansor, M. F. M. Basir, and S. Sathasivam, "DISCRETE MUTATION HOPFIELD NEURAL NETWORK IN PROPOSITIONAL SATISFIABILITY," *Mathematics*, vol. 7, no. 11, p. 1133, 2019, doi: https://doi.org/10.3390/math7111133
[11] M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, "MODIFIED ARTIFICIAL IMMUNE SYSTEM ALGORITHM WITH ELLIOT HOPFIELD NEURAL NETWORK FOR 3-SATISFIABILITY PROGRAMMING," *Journal of Informatics and Mathematical Sciences*, vol. 11, no. 1, pp. 81–98, 2019, doi: https://doi.org/10.26713/jims.v11i1.1062.
[12] M. S. M. Kasihmuddin, M. A. Mansor, M.A., and SSathasivam, "HYBRID GENETIC ALGORITHM IN THE HOPFIELD NETWORK FOR LOGIC SATISFIABILITY PROBLEM," *Pertanika J Sci Technol*, vol. 25, no. 1, pp. 139–152, 2017.
[13] M. M. Bazuhair *et al.*, "NOVEL HOPFIELD NEURAL NETWORK MODEL WITH ELECTION ALGORITHM FOR RANDOM 3 SATISFIABILITY," *Processes*, vol. 9, no. 8, p. 1292, 2021, doi: https://doi.org/10.3390/pr9081292.
[14] L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, "LOGIC MINING IN LEAGUE OF LEGENDS," *Pertanika J Sci Technol*, vol. 28, no. 1, pp. 211–225, 2020.
[15] N. E. Zamri, A. Alway, M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, "MODIFIED IMPERIALISTIC COMPETITIVE ALGORITHM IN HOPFIELD NEURAL NETWORK FOR BOOLEAN THREE SATISFIABILITY LOGIC MINING," *Pertanika J Sci Technol*, vol. 28, no. 3, pp. 983– 1008, 2020.
[16] Y. Gao *et al.*, "GRAN3SAT: CREATING FLEXIBLE HIGHER-ORDER LOGIC SATISFIABILITY IN THE DISCRETE HOPFIELD NEURAL NETWORK," *Mathematics*, vol. 10, no. 11, p. 1899, 2022, doi: https://doi.org/10.3390/math10111899
[17] N. E. Zamri, S. A. Azhar, M. A. Mansor, A. Alway, and M. S. M. Kasihmuddin, "WEIGHTED RANDOM K SATISFIABILITY FOR K= 1, 2 (R2SAT) IN DISCRETE HOPFIELD NEURAL NETWORK," *Appl Soft Comput*, vol. 126, p. 109312, 2022, doi: https://doi.org/10.1016/j.asoc.2022.109312
[18] S. Abdeen, M. S. M. Kasihmuddin, N. E. Zamri, G. Manoharam, M. A. Mansor, and N. Alshehri, "S-TYPE RANDOM K SATISFIABILITY LOGIC IN DISCRETE HOPFIELD NEURAL NETWORK USING PROBABILITY DISTRIBUTION: PERFORMANCE OPTIMIZATION AND ANALYSIS," *Mathematics*, vol. 11, no. 4, p. 984, 2023, doi: https://doi.org/10.3390/math11040984
[19] G. Gosti, V. Folli, M. Leonetti, and G. Ruocco, "BEYOND THE MAXIMUM STORAGE CAPACITY LIMIT IN HOPFIELD RECURRENT NEURAL NETWORKS," *Entropy*, vol. 21, no. 8, p. 726, 2019, doi: https://doi.org/10.3390/e21080726
[20] G. Manoharam *et al.*, "LOG-LINEAR-BASED LOGIC MINING WITH MULTI-DISCRETE HOPFIELD NEURAL NETWORK," *Mathematics*, vol. 11, no. 9, p. 2121, 2023, doi: https://doi.org/10.3390/math11092121
[21] S. A. Karim *et al.*, "RANDOM SATISFIABILITY: A HIGHER-ORDER LOGICAL APPROACH IN DISCRETE HOPFIELD NEURAL NETWORK," *IEEE Access*, vol. 9, pp. 50831–50845, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3068998.
[22] J. H. Holland, *Adaptation in natural and artificial systems"*. University of Michigan Press, Ann Arbor, 1975.
[23] D. Yang, Z. Yu, H. Yuan, and Y. Cui, "AN IMPROVED GENETIC ALGORITHM AND ITS APPLICATION IN NEURAL NETWORK ADVERSARIAL ATTACK," *PLoS One*, vol. 17, no. 5, p. 267970, 2022, doi: https://doi.org/10.1371/journal.pone.0267970.
[24] Y. Xue, H. Zhu, J. Liang, and A. Słowik, "ADAPTIVE CROSSOVER OPERATOR BASED MULTI-OBJECTIVE BINARY GENETIC ALGORITHM FOR FEATURE SELECTION IN CLASSIFICATION," *Knowl Based Syst*, vol. 227, p. 107218, 2021, doi: https://doi.org/10.1016/j.knosys.2021.107218.
[25] S. Z. M. Jamaludin *et al.*, "ARTIFICIAL BEE COLONY FOR LOGIC MINING IN CREDIT SCORING", *Malaysian Journal of Fundamental and Applied Sciences*, vol. 18, no. 6, pp. 654–673, 2022, doi: https://doi.org/10.11113/mjfas.v18n6.2661.
[26] M. A. Mansor and S. Sathasivam, "OPTIMAL PERFORMANCE EVALUATION METRICS FOR SATISFIABILITY LOGIC REPRESENTATION IN DISCRETE HOPFIELD NEURAL NETWORK," *Comput Sci*, vol. 16, no. 3, pp. 963–976, 2021.
[27] C. Affonso, R. J. Sassi, and R. P. Ferreira, "TRAFFIC FLOW BREAKDOWN PREDICTION USING FEATURE REDUCTION THROUGH ROUGH-NEURO FUZZY NETWORKS," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, 2011, pp. 1943–1947, doi: https://doi.org/10.1109/IJCNN.2011.6033462.
[28] S. Z. M. Jamaludin, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, and M. F. M. Basir, "ENERGY-BASED LOGIC MINING ANALYSIS WITH HOPFIELD NEURAL NETWORK FOR RECRUITMENT EVALUATION," *Entropy*, vol. 23, no. 1, p. 40, 2020, doi: https://doi.org/10.3390/e23010040.