mathematics

MDPI

Article

An Improved Machine Learning Model with Hybrid Technique in VANET for Robust Communication

Gagan Preet Kour Marwah [1], Anuj Jain [1], Praveen Kumar Malik [1], Manwinder Singh [1], Sudeep Tanwar [2,*], Calin Ovidiu Safirescu [3,*], Traian Candin Mihaltan [4], Ravi Sharma [5] and Ahmed Alkhayyat [6]

[1] School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar 144411, India
[2] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, India
[3] Environment Protection Department, Faculty of Agriculture, University of Agricultural Sciences and Veterinary Medicine Cluj-Napoca, Calea Mănăștur 3-5, 400372 Cluj-Napoca, Romania
[4] Faculty of Building Services, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania
[5] Centre for Inter-Disciplinary Research and Innovation, University of Petroleum and Energy Studies, Dehradun 248001, India
[6] College of Technical Engineering, The Islamic University, Najaf 54001, Iraq
* Correspondence: sudeep.tanwar@nirmauni.ac.in (S.T.); calin.safirescu@usamvcluj.ro (C.O.S.)

Abstract: The vehicular ad hoc network, VANET, is one of the most popular and promising technologies in intelligent transportation today. However, VANET is susceptible to several vulnerabilities that result in an intrusion. This intrusion must be solved before VANET technology can be adopted. In this study, we suggest a unique machine learning technique to improve VANET's effectiveness. The proposed method incorporates two phases. Phase I detects the DDoS attack using a novel machine learning technique called SVM-HHO, which provides information about the vehicle. Phase II mitigates the impact of a DDoS attack and allocates bandwidth using a reliable resources management technique based on the hybrid whale dragonfly optimization algorithm (H-WDFOA). This proposed model could be an effective technique predicting and utilizing reliable information that provides effective results in smart vehicles. The novel machine learning-based technique was implemented through MATLAB and NS2 platforms. Network quality measurements included congestion, transit, collision, and QoS awareness cost. Based on the constraints, a different cost framework was designed. In addition, data preprocessing of the QoS factor and total routing costs were considered. Rider integrated cuckoo search (RI-CS) is a novel optimization algorithm that combines the concepts of the rider optimization algorithm (ROA) and cuckoo search (CS) to determine the optimal route with the lowest routing cost. The enhanced hybrid ant colony optimization routing protocol (EHACORP) is a networking technology that increases efficiency by utilizing the shortest route. The shortest path of the proposed protocol had the lowest communication overhead and the fewest number of hops between sending and receiving vehicles. The EHACORP involved two stages. To find the distance between cars in phase 1, EHACORP employed a method for calculating distance. Using starting point ant colony optimization, the ants were guided in phase 2 to develop the shortest route with the least number of connections to send information. The relatively short approach increases protocol efficiency in every way. The pairing of DCM and SBACO at H-WDFOA-VANET accelerated packet processing, reduced ant search time, eliminated blind broadcasting, and prevented stagnation issues. The delivery ratio and throughput of the H-WDFOA-packet VANET benefitted from its use of the shortest channel without stagnation, its rapid packet processing, and its rapid convergence speed. In conclusion, the proposed hybrid whale dragonfly optimization approach (H-WDFOA-VANET) was compared with industry standard models, such as rider integrated cuckoo search (RI-CS) and enhanced hybrid ant colony optimization routing protocol (EHACORP). With the proposed method, throughput could be increased. The proposed system had energy consumption values of 2.00000 mJ, latency values of 15.61668 s, and a drop at node 60 of 0.15759. Additionally, a higher throughput was achieved with the new method. With the suggested method, it is possible to meet the energy consumption targets, delay value, and drop value at node 60. The proposed method reduces the drop value at node 80 to 0.15504, delay time to 15.64318 s, and energy consumption to 2.00000 mJ. These

check for updates

Citation: Marwah, G.P.K.; Jain, A.; Malik, P.K.; Singh, M.; Tanwar, S.; Safirescu, C.O.; Mihaltan, T.C.; Sharma, R.; Alkhayyat, A. An Improved Machine Learning Model with Hybrid Technique in VANET for Robust Communication. *Mathematics* **2022**, *10*, 4030. https://doi.org/10.3390/math10214030

Academic Editors: Adrian Sergiu Darabant and Diana-Laura Borza

Received: 11 September 2022
Accepted: 25 October 2022
Published: 30 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

outcomes demonstrate the effectiveness of our proposed method. Thus, the proposed system is more efficient than existing systems.

Keywords: VANET; support vector machine; rider optimization algorithm; cuckoo search; whale optimization algorithm; dragonfly; DDoS attack

MSC: 68M10

1. Introduction

In recent years, the vehicular ad-hoc network (VANET) has gained prominence in the networking world. It is a mobile ad hoc network (MANET) primarily utilized for vehicle-to-vehicle and vehicle-to-roadside-unit communication. Data is shared via vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) communications. VANET applies to various areas such as traffic control, traffic congestion monitoring, and driver safety programs [1,2]. However, VANET has its disadvantages, such as in dynamic behavior. This issue is rectified by an effective and efficient routing protocol that broadcasts the data. As cars may not be able to communicate information without an efficient routing method, they will lose all the benefits of sophisticated VANET technology [3].

Recently, many efforts have been made to adapt software-defined network (SDN) ideas to vehicular networks to benefit from the SDN's flexibility and programmability. The ultimate objective of this adaption is to improve VANET performance and make it more suitable for certain services and applications [4]. The software-defined network (SDN) is one of the emerging revolutionary network concepts utilized in data centers and operator networks to satisfy the growing demands of future networks [5]. Network programmability, centralized management control, and interface openness are some features of SDN. It also substantially simplifies network administration and improves networkability [6,7]. SDN is divided into three layers: data, control, and application. Routers, switches, vehicles, and wireless gadgets are all part of the data layer. According to the controller, which would be the SDN's brain and resides in the control plane, devices in the data layer are responsible for packet transmission. It determines how data layer devices send and receive data and how the flow is managed. The network layer consists of software initiatives that manage the network's programming. SDN and VANET techniques are not without their flaws. Despite their numerous advantages and excellent performance, they are subject to attack risk, namely from DoS and DDoS, just as in traditional wireless networks. However, SDN features make it simpler to identify and respond to DoS attacks, and the openness of the data plane creates favorable conditions for additional DoS attacks [8]. A distributed DoS (DDoS) assault occurs when a DoS attack is launched from several vehicles dispersed throughout the network. The targeted system cannot handle any incoming requests due to the breach, and the entire system crashes [9]. In such assaults, the hacker compromises thousands of hosts, including PCs, servers, and IoT devices, and exploits them all at once to overwhelm the victim's resources with traffic. This traffic suspends network users and defending against it is quite difficult [10]. A system administrator can detect a potential threat but cannot handle the simultaneous attacks in real-time. As a result, the SDN controller must be subjected to particular security constraints. Additionally, automatic attack detection systems are required to ensure an optimum level of service quality and to detect suspicious intrusions [11].

The detection of DDoS attacks includes various challenges, including identifying the characteristics of attack traffic; lack of cooperation among the nodes of a coherent network; modification in attack tools; usage of a fake IP address; and short attack duration and limited response time [12]. To deal with these issues, we need a system that can detect intrusions and store data in communication while ensuring data integrity. The ultimate purpose of the intrusion detection method is to identify intrusion before it causes damage

to a targeted server and to stop attacks as close to their source as possible [13]. Machine learning (ML) techniques provide a framework frequently employed in VANETs to evaluate large amounts of data and generate efficient rules for attack detection, classification, and prediction [14]. The ML approaches apply not only to intrusion detection but also in fields such as speech recognition, recommendation systems, medical diagnosis, and traffic prediction. It also offers a desirable approach for detecting intrusions in VANETs ahead of time with acceptable speed and accuracy [15].

There are four different forms of communication in VANETs.

- In-vehicle communication: the in-vehicle domain is becoming increasingly crucial in VANET research. Driver and public safety are both improved when in-car communication systems can detect issues such as driver fatigue and drowsiness.
- To better assist drivers, V2V communication can facilitate the exchange of information and warning messages between vehicles.
- Virtual-to-individual (V2I) communication in VANETs is an emerging research area. Through vehicle-to-infrastructure (V2I) communication, drivers can access real-time traffic and weather updates and participate in environmental sensing and monitoring.
- Wireless broadband technology, such as 3G/4G, may be used for vehicle-to-vehicle (V2V) communication. With more traffic data, monitoring information, and entertainment options available in the internet cloud, this type of connectivity will be beneficial for active driver assistance and vehicle tracking.

The primary goal of public safety technology is to improve the overall safety of transportation systems. Real-time traffic conditions, post-crash alerts, traffic monitoring, collision mitigation, and other safety applications are available. Non-public security applications aim to improve driver comfort by offering value-added information, such as nearby petrol stations, hotels, workshops, and restrooms, among other things. It is often advisable to use public safety technology than non-public safety technology. Numerous applications are used in many nations, including Germany, Europe, and Switzerland, to make roads and automobiles safer. Convenience applications are mainly used to handle traffic congestion scenarios, such as available parking spaces, route diversions, and electronic toll collection, among other things. New elements to the services mentioned above that focus on converting traffic jam time into constructive work, such as reading on the internet while waiting in a vehicle for someone, are known as productive activities. Additionally, fuel is saved when the average toll booth wait time is decreased to 1–2 min [16].

The following are the main contributions of the current study:

- The primary objective of this paper was to use a machine learning-hybrid optimization strategy to detect DDoS attacks and allocate bandwidth within the VANET.
- Algorithms and models based on machine learning can support robust communication analysis and resource management, as well as the increasing communication and computation demands of emerging networking applications.
- Security-based modified SVM with HHO technique was utilized to identify DDoS attacks. The SVM-HHO technique demonstrated the most outstanding feature selection and parameter optimization capabilities compared with other well-known metaheuristic algorithms. Using H-WDFOA, the impact of a DDoS attack on the VANET was diminished (H-WDFOA-VANET).
- The combination of DF optimization and WOA increased bandwidth. The condition was repeated until all available bandwidth had been allocated. The H-WDFOA was utilized to achieve effective bandwidth allocation and optimal vehicle routing.
- For the stability of the vehicular network, swift and efficient communication was established once the predicted information had been validated.
- Utilizing performance indicators such as energy consumption, throughput, latency, and packet delivery ratio, the performance of the proposed method was evaluated. The implementation was accomplished using MATLAB software.

2. Literature Review

Poongodi et al. [17] proposed a trust-based methodology for determining the distributed denial of service (DDoS) attack in VANET. The residual energy, trust policy, data factor, statistics, such as frequency value, and true hypothesis were used to create the trust elements. These trust elements were utilized to generate the trust evaluation matrix. The deterrence design of the trust evaluation approach was incorporated with a clustering method to identify the attacker in a proficient manner.

For identifying DDoS assaults on VANET, Adhikary et al. [18] used a hybrid technique. These hybrid approaches were developed using ANOVA Dot and RBF Dot SVM kernel methodology. Features such as collision, jitter, and packet drop were used to replicate a real-time network communication scenario. As a result, the hybrid algorithm-based model was superior at detecting DDoS attacks.

To identify DDoS assaults on VANET, Kolandaisamy et al. [19] devised a multivariant-stream analysis (MVSA) method. The network was protected from distributed denial of service attacks by keeping multiple levels active. This tactic's success hinges on the ability of individual cars to talk to one another through the Road Side Unit.

Using big data technology, Marwah et al. [20] provided a method for detecting DDoS attacks. The real-time network traffic-collecting module and traffic-detection module are two components of the described technique. In addition, spark and HFSA-VANET were also used to speed up data processing and store significant suspicious attacks.

Hind Bangui et al. [21] suggested a machine learning approach that combined traditional methods with more modern ones (IDS) to boost the effectiveness of intrusion detection systems. The IDS was a potential tool for thwarting VANET attacks in advance. Consequently, the coreset was utilized to enhance detection accuracy, and the random forest (RF) was implemented to boost IDS functionality. For VANET, Tang et al. [22] implemented a centralized routing mechanism with mobility prediction (CRS-MP), using a software-defined network (SDN) controller. The Road Side Unit (RSU) or Base Station (BS) determined the average latency and transmission probability of each request from the vehicle based on the mobility forecast. Information was gathered by the SDN controller via RSU, whereas the BS was viewed as mere switches.

Zhang and Zhu [23] presented VANET's PML-CIDS. In this study, the authors used the alternating direction multiplier method (ADMM) to a set of empirical risk minimization (ERM) problems. Differential privacy was used to acquire the PML-CIDS privacy notation, and the NSL-KDD dataset was used for the evaluation. An improved VANET routing model was developed by Gnanasekar and Samiappan [24], which accounted for congestion, travel, collision, and QoS awareness cost. An updated pricing scheme was developed in light of the restrictions. When calculating the optimal route, the authors considered the total cost and quality of service (QoS) component, which was fuzzy-calculated. The optimal routing in VANET was achieved at the lowest possible routing cost by developing a novel RI-CS optimization approach, which was built by fusing the ideas of ROA and CS.

To improve the effectiveness of the routing procedure in VANETs, Ramamoorthyet al.proposed a new routing protocol, EHACORP, which was presented in [25]. At EHACORP, the combination of DCM and SBACO shortened the time it took for ants to find their way around, increased the rate at which they converged, eliminated the need for blind broadcasting packets, expedited packet processing, and prevented bottlenecks. EHACORP increased its packet delivery ratio (PDR) and throughput by avoiding bottlenecks, rapidly processing packets, and quickly converging.

In [26], the authors presented a study on the split delivery vehicle routing problem that arises in the distribution of fresh agricultural products. A mathematical model was developed to find an optimal solution for the problem by considering travel costs, fixed costs, service costs, refrigeration costs, carbon emission costs, and customer satisfaction. This paper added time-varying road network constraints to the model, and multiple time windows were set for each customer. A variable neighborhood search, combined with the non-dominated sorting genetic algorithm II (VNS-NSGA-II), and techniques for order

preference by similarity to an ideal solution (TOPSIS) was proposed and applied to solve this problem.

In [27], the authors proposed a scale-adaptive mathematical morphology spectrum entropy (AMMSE) to improve the scale selection. To support the proposed method, two properties of the mathematical morphology spectrum (MMS), namely non-negativity and monotonic decreasing, were shown. It could be concluded from the two properties that feature loss of MMS decreased with the increase in scale. Based on this conclusion, two adaptive scale selection strategies were proposed to automatically determine the scale by reducing the feature loss of MMS. The presented method was applied to identify fault degree on a CWRU-bearing data set and evaluate performance degradation onan IMS-bearing data set. The experiment results showed that AMMSE had better results in both experiments with the same parameters.

3. Proposed Methodology for Robust Communication through Machine Learning Algorithm

The VANET's dynamic condition was developed in this section to select the suitable routing of each node. Furthermore, allocating the bandwidth of each node was obtained for accurate routing. In VANET, the optimization approach SVM-HHO-VANET was developed to select a route based on clustering to provide precise bandwidth allocation for proper routing. Here, the proposed methodology was halved into two phases. Security-based modified SVM was used to find DDoS attacks early on. In the next stage, the HHO optimization approach for the VANET dynamic scenario was used to achieve effective bandwidth allocation. This research included cluster selection, accurate cluster node communication, detecting DDoS attacks, route prediction, bandwidth allocation between the cluster nodes, distance, velocity, speed, bandwidth, number of nearby vehicles, and throughput parameters for QoS analysis. Additionally, evaluation metrics such as energy consumption, delay, and drop were considered for QoS performance analysis. In this work, several nodes in the vehicle were analyzed based on distance, speed, velocity, and bandwidth. The location of the vehicle was determined in each iteration, according to fitness. When the node's position was less than 0.5 and the nearby vehicle was greater than or equal to 1, the best location was updated. If the present iteration was higher than the maximum iteration, the target vehicle's position was considered the best location. To achieve a maximum optimal solution, the DF approach was integrated with WOA.

Furthermore, the dragonfly solution was initialized and determined the bandwidth by QoS value according to the objective function. The maximum iteration was utilized until the process continued to enhance the bandwidth. Finally, energy consumption, delay, and drop were considered based on the QoS parameter for accurate routing. The proposed approach was accurate in allocating bandwidth compared with other existing techniques. This study is robust and fast in predicting DDoS attacks and allocating the bandwidth using the H-WDFOA technique, in the best position after validating for stability in vehicular networks.

4. Background of HHO, SVM, and DDoS Attack

4.1. Harris Hawks Optimization (HHO)

Harris hawks optimization (HHO) is an optimization method that uses a swarm intelligence approach. The primary focus of HHO is on simulating cooperative hunting and successful prey escape strategies employed by hawks in the wild to solve the problem of a single target [28]. Hawks are search agents in HHO, and the best position is that of the prey. The demonstration of the HHO is described in full detail below and depicted in Figure 1.

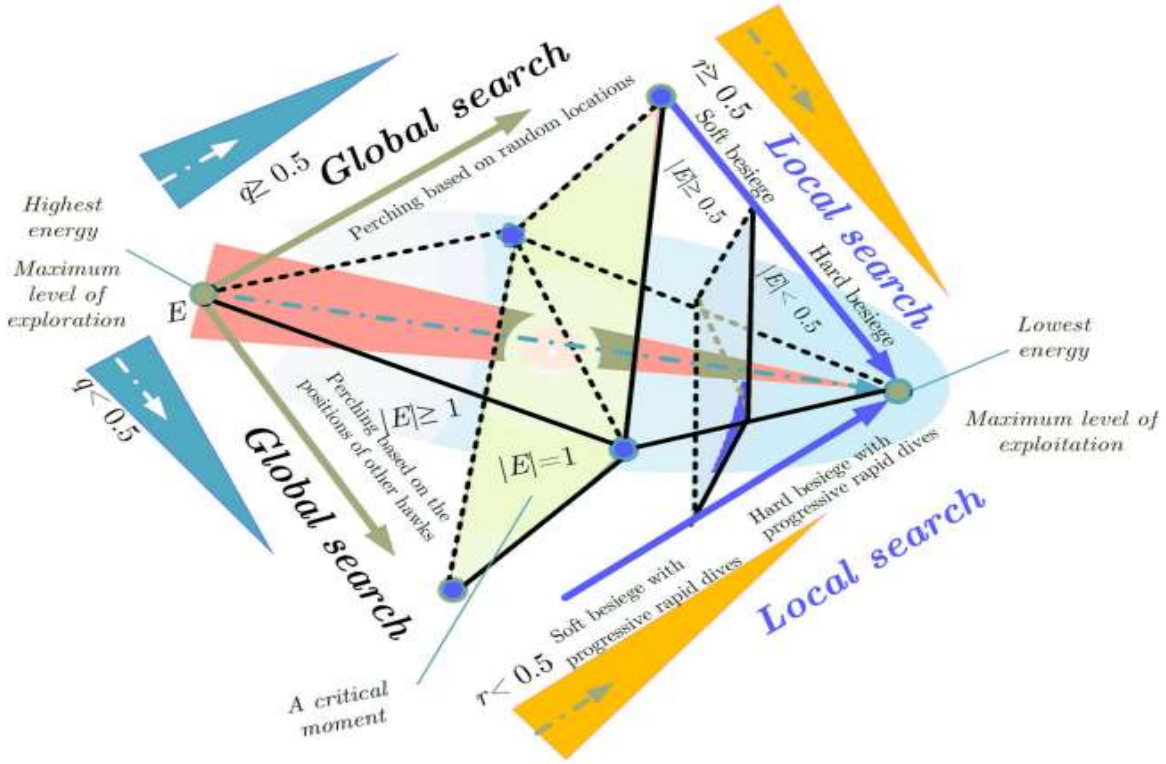

Figure 1. HHO phase [1].

In this approach, hawks locate their prey depending on the positions of the genuine members (the total number of hawks defined by N and represented by Equation (1).

$$X_i(t+1) = \{(X_{prey}(t) - X_m(t) - Y),\ q < 0.5)(X_{rand}(t) - |X_{rand}(t) - 2r_2\ X(t)|, q \geq 0.5) \quad (1)$$

The next iteration shows that the revised position of hawks, t, is represented by $X_i(t+1)$; $X_{rand}(t)$ represents the current location of the hawks; and $r1$, $r2$, $r3$, $r4$, and q denote random numbers inside the collection of hawks (0,1). The position of prey is represented by $X_{prey}(t)$. $X_m(t)$ defines the total average positions of all hawks, which is calculated using Equation (2).

$$X_m(t) = \frac{\sum_{i=1}^{N} X_i(t)}{N} \quad (2)$$

$Y = r_3(LB + r_4\ (UB - LB)$ stands for the gap between the maximum and minimum values of a variable.

4.2. Support Vector Machine (SVM)

The SVM has been validated as a powerful paradigm for classification. SVM provides a crucial mathematical model for use in classification and regression. Twin SVM, Lagrangian SVM, least-square SVM, decision tree SVM, directed acyclic graph SVM, and multi-kernel SVM are only some of the SVM variants that have emerged during the past few decades [29]. In support vector machines (SVM), multi-class categories are typically translated into a large number of binary variants using either OvO (one vs. one) or OvR (one vs. many or one vs. rest) [30]. SVM classifies data by translating it into multidimensional space and creating a higher dimension of hyperplanes. This hyperplane refers to the decision planes in Figure 2.

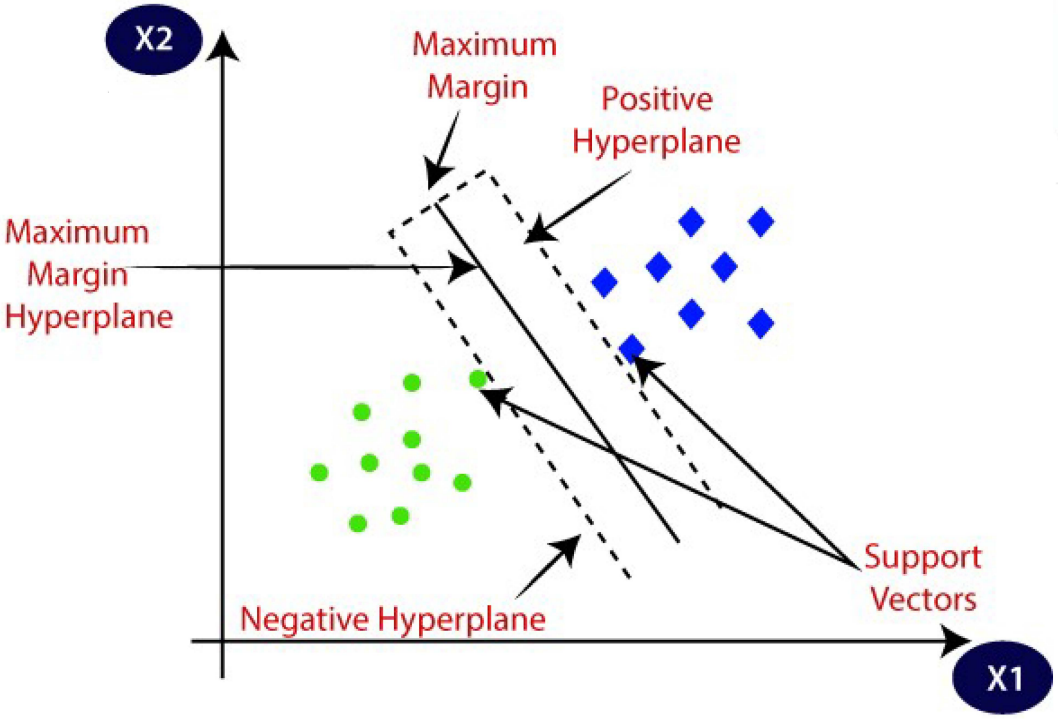

Figure 2. Hyperplane supporter in SVM plan.

4.3. DDoS Attack

As per Figure 3, DDoS attacks are network security threats that aim to overload target networks with malicious traffic. Though several statistical methods exist for DDoS attack detection, the development of a real-time detector that requires minimal processing resources remains to be a fundamental challenge [31]. This paper discusses recent developments in cloud-based DDoS mitigation technologies. In particular, we provide a detailed analysis of the characterization, prevention, detection, and mitigation strategies used in these attacks.

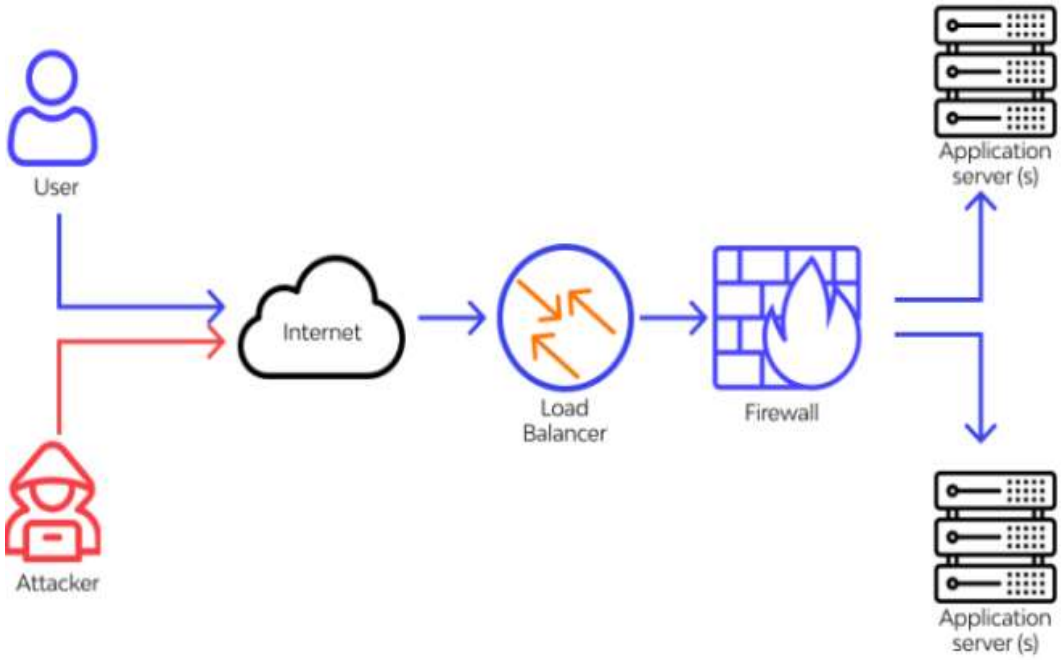

Figure 3. Attack by DDoS.

The flowchart of the proposed methodology is illustrated in Figure 4. First, we initialized the dataset, where 70% of data were utilized in the training phase and 30% in the testing phase. However, the DDoS attack was analyzed and detected by modified SVM. Due to a DDoS attack, node performance was reduced, and bandwidth attracted more attenuation. In addition, based on the number of nodes, hybrid WDFOA was initialized for allocating bandwidth. For allocation, the QoS value was separately analyzed based on bandwidth, distance, speed, and vehicle velocity. The capacity of the node to transfer data was established by running condition assessment if the node's bandwidth is less than a predetermined threshold. To enhance the bandwidth, DF optimization was integrated with WOA. This condition was repeated until the termination of bandwidth was allocated.

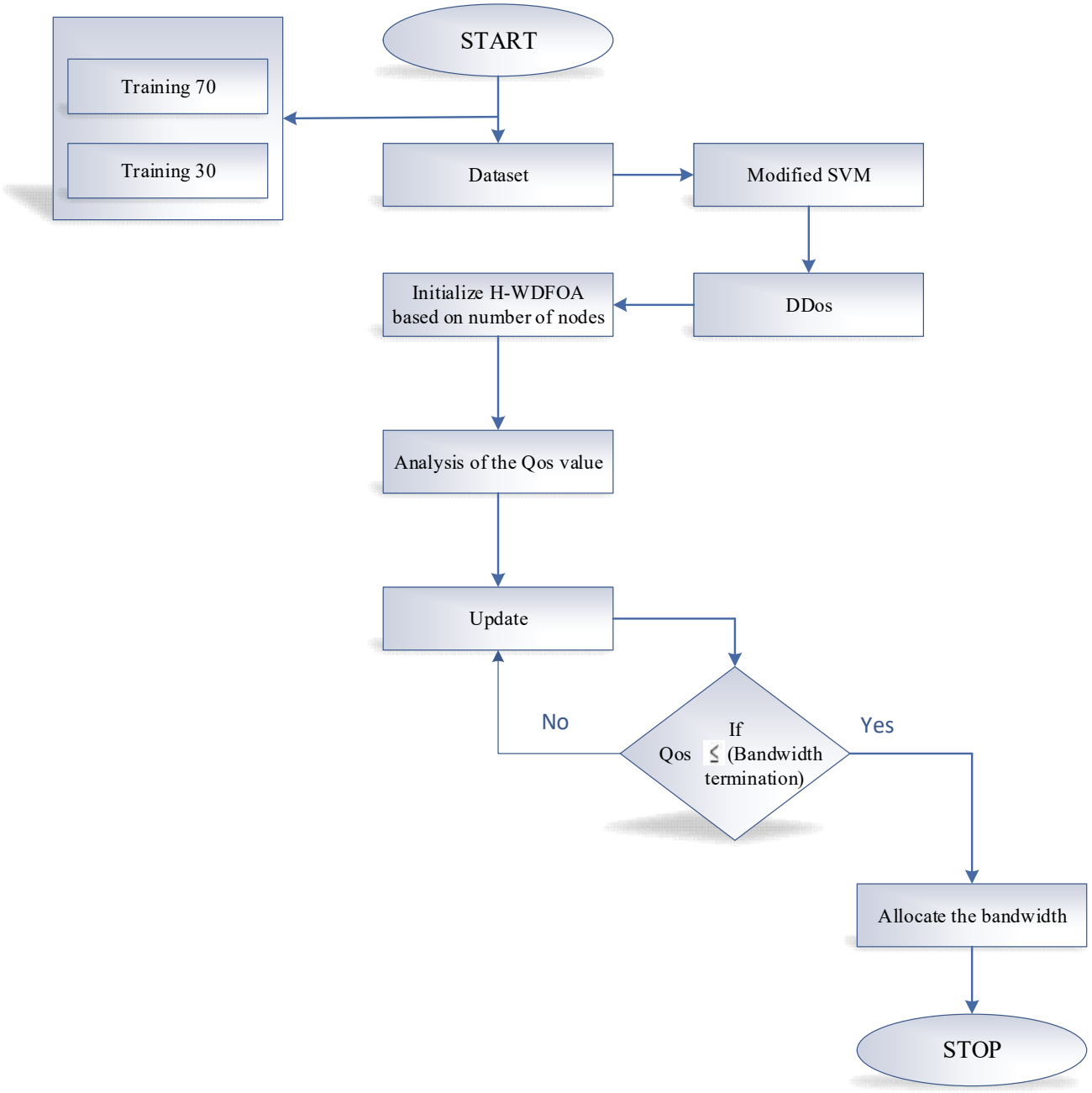

Figure 4. Flowchart for proposed methodology.

5. Simulated Steps for Proposed Methodology

At a fixed location, utilizing 1000 vehicles, VANET was randomly implemented and placed in 1000 × 1000 m. The placement of vehicles is tabulated in Table 1. In a defined area, VANET was randomly placed.

Table 1. Vehicle placement in the VANET.

 # # At Random Places, Vehicles Are Places to Create VANET
➢ $In\ Total - Vehicles,\ For\ each\ x$
➢ $Va(x) = 1000 * run\ if\ (1)$
➢ $Vb(x) = 1000 * run\ if\ (1)$
➢ $Fix\ (V\ a(x),\ V\ b(x))$
➢ $End\ For$

a and *b* are the coordinatesof the vehicle, denoted by *Va* and *Vb*. To randomly position a car at positions a and b, R-studio was used to incorporate the random function run if (1).

5.1. Dataset Production

The data set called the "DDoS attack-based SDN Network Dataset" used deep learning and machine learning techniques that were publicly accessible and used by many researchers. There were 104,345 traffic flows in 23 features in this dataset. The usual attack traffic class label used a dataset comprising UDP, ICMP, and TCP traffic.

5.2. Phase: 1—Security-Based Modified SVM

This study used the modified support vector machine (MSVM) to identify the attack on SDN-based VANETs by DDoS. Here, SVM utilized kernel function to transfer non-linear data to a high-dimensional space based on Harris hawks optimization (HHO). In the SVM classifier, the optimal kernel was selected based on HHO, known as modified SVM. Thus, the kernel-based MSVM was suggested in this paper to identify DDoS attacks.

The architecture of the suggested hybrid detection is shown in Figure 5. The VANET setup consisted of 1000 vehicles in the first module. The second module produced the information needed to develop a training and testing procedure. Finally, a hybrid method was used to make predictions based on data collected during training and testing. Predictions consisted of binary values"0 or 1" for indicating efficient behavior or DDoS attack.

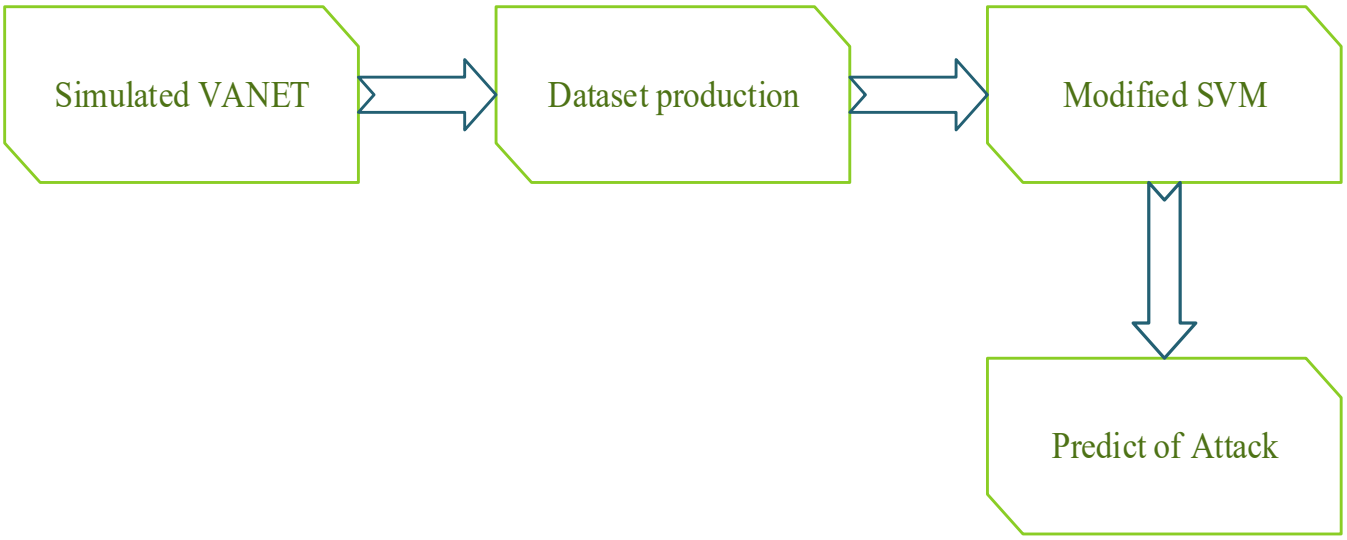

Figure 5. Basic architecture for detection method.

5.2.1. Modified Support-Vector Machine

SVM generalization is the most significant margin classifier and is said to be the best method for categorization difficulties. When comparing two data sets, the margin

is the line closest to either set. A flat affine subspace in an x-dimension space forms an x−1 dimension [32] in a hyperplane. In a hyperplane, the following equation determines two dimensions.

$$\alpha_0\alpha_1 y_1 + \alpha_2 y_2 = 0 \tag{3}$$

Here, in the hyperplane, $Y = (Y_1,\ Y_2)^T$ is a point and $\alpha_0,\ \alpha_1,$ and α_2 are the parameters. Equation (1) can be extended, for n-dimensional space.

$$\alpha_0 + \alpha_1 y_1 + \alpha_2 y_2 + \ldots + \alpha_n y_n = 0 \tag{4}$$

If $Y = (Y_1,\ Y_2,\ \ldots Y_n)^T$, Y lies on the hyper plane. The left side of the hyperplane is represented by x in the Equation (3), and is more than zero, whereas if it is on the other side, it is less than zero. A hyperplane reveals which side a point is on in this way.

The SVC can determine which side of the hyperplane an observation from the test space lies on and classify it accordingly. As shown in Figure 6, some data points fell outside of the hyperplane's boundaries. The equation expresses the SVM's working principle,

$$Maximize_{\alpha_0,\alpha_1.....\alpha_n,\epsilon_1.....\epsilon_n} \tag{5}$$

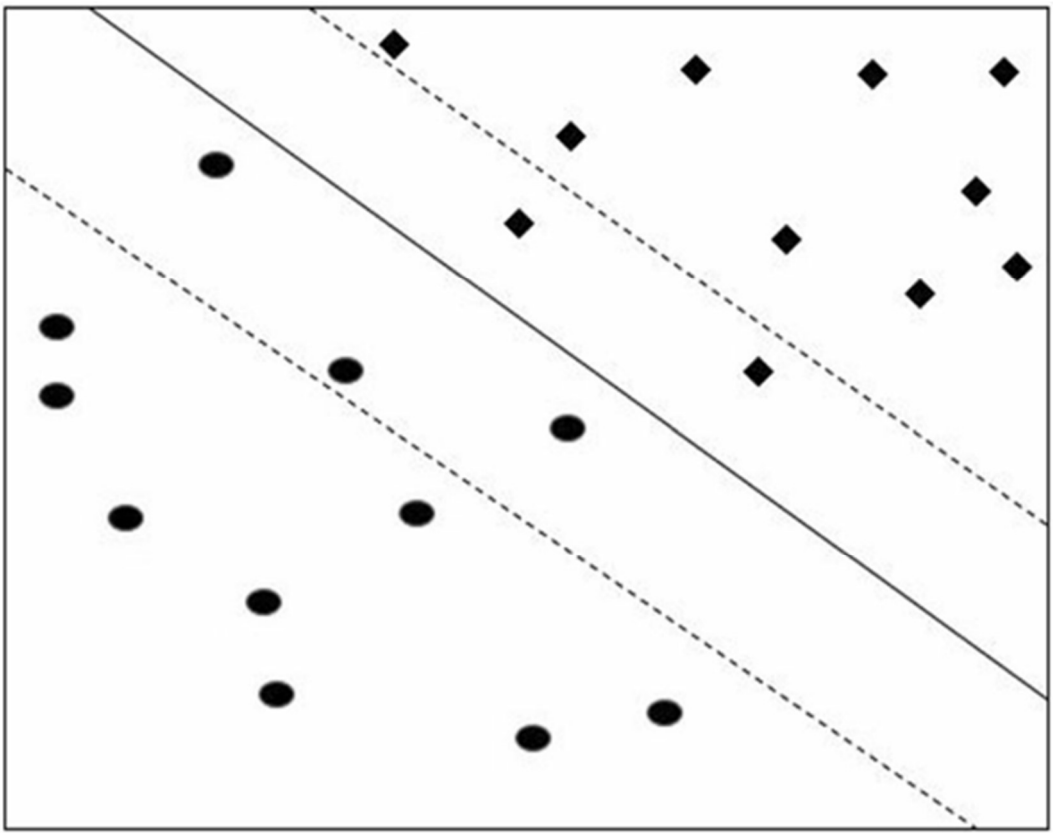

Figure 6. Concept of support vector machine (SVM).

A circumstance where

$$\sum_{i=1}^{n} a_i^2 = 1 \tag{6}$$

$$f_x(\alpha_0 + \alpha_1 y_1 + \alpha_2 y_2 \ldots \ldots \ldots + \alpha_n y_n) \ \geq W(1 - \epsilon_x) \tag{7}$$

$$\epsilon_x \geq 0,\ \sum_{x=1}^{n} \in_x \leq T \tag{8}$$

In this case, the positive tuning parameter is denoted by T. The width of the margin is represented by W. A slack variable is ϵ_x, this enables for an observation on the margin's wrong side. The sign of $\alpha_0 + \alpha_1 y_1 + \alpha_2 y_2 \ldots\ldots\ldots + \alpha_n y_n$ is classified as an observation.

To transfer non-linear data, SVM used kernel function based on HHO, known as MSVM. To detect DDoS attacks, MSVM is employed. Hence, the optimal kernel-HHO is detailed in the following section.

5.2.2. Selection of Optimal Kernel Based on Harris Hawks Optimization (HHO)

HHO is defined by a meta-heuristic technique based on a natural population. Harris hawks cooperative hunting behavior is for prey such as rabbits. Exploration, transformation, and exploitation are the three steps included in the HHO optimization technique.

- Initialization Phase

The initial hawk population initializes the solution for this phase. The number of nearest vehicles ($n_1, n_2, \ldots, n_N$) encompasses the population. Here, the number of populations are denoted as N. The execution of the technique was done by cluster vehicle (n_{cl}). The cluster and nearby vehicles are denoted as "hawks", whereas the destination vehicle is referred to as prey (rabbit). For all cars, the fitness function was determined in this phase. The fitness of jth vehicle was calculated as follows,

$$f(n_j) = \{\sum D,\ M,\ \delta,\ K\} \tag{9}$$

Here, the fitness function is developed as mobility (M), direction (D), vehicle state (δ), and availability of spectrum (K). Binary values '0' and '1' are described as vehicles' direction. The fitness of the vehicle is enhanced if target and cluster vehicles are in the same direction, so that D = 1 or 0. Furthermore, to enrich the computation fitness, the mobility will be mapped as 0 and 1. When M = 0 or 1, mobility of n_j is higher than average mobility. For data transmission K = 0 or 1, if n_j has an available channel. The function of evaluation metrics is represented as follows:

$$\delta = \frac{P + DR}{D} \tag{10}$$

The throughput function of δ is n_j (P) for n_j, delay (D), and data rate (DR). For data transmission, the optimal path can be achieved by the deciding factor δ of computation fitness with effective channel estimation. In the population, for all vehicles, $f(n_j)$ was computed.

Exploration, transformation, and exploitation are the three phases of HHO. These three phases are described as follows:

- Exploration Phase

During this phase, the process of waiting, searching, and detecting prey is carried out. The following is an example of a hawk's position: The V represents the current iteration and V + 1 represents the next iteration. Hawks can be found at the locations γ_1,γ_2, γ_3, and γ_4, with the range [0, 1] expressing the random integer y. n_{j_a} is the average location of the hawk.

$$n_j\,(V+1) = \left\{ n_{j_{rand}}(V) - \gamma_1 \left| n_{j_{rand}}(V) - 2\, n_j \gamma_2(V) \right| \; if\; m \;\geq 0.5 n_{j_q}\,(V) n_{j_a}\,(V) - \gamma_3\,(l_1 + \gamma_4(v_b - l_1))\; if\; m < 0.5 \right. \tag{11}$$

$$n_{j_a}\,(V) = \frac{1}{N} \sum_{i=1}^{N} n_j(V) \tag{12}$$

- Transformation Phase

In the exploitation phase, the exploration phase is transformed into the exploitation phase. The energy of the prey degrades from the evading behaviour. Equation (17) defined the transformation phase.

$$\omega_{prey} = 2\omega_0\,(1 - \frac{V}{V_{max}}) \tag{13}$$

where ω_0 shows the initial energy state of prey, V represents the current iteration numbers, and V_{max} represents the maximum number of iterations.

- Phase of Exploitation

The hawk's attack in the previous step serves as a barometer for the quality of the picked prey. In this stage, we tested four distinct approaches: soft besiege, hard besiege, soft besiege with advancing fast dives, and hard besiege with advanced rapid dives. The occurrence of hard and soft relies on the energy level, where $|\omega_{prey}| \geq 0.5$ and $|\omega_{prey}| < 0.5$, respectively.

Attack on a Soft Target:

The attack on a soft target known as a soft besiege and this method may be chosen if $|\omega_{prey}| \geq 0.5$ and $\gamma \geq 0.5$. It is described as:

$$n_j\ (V+1) = \Delta n_j\ (V) - \omega_{prey}\ |\zeta n_j(V - n_j(V))| \tag{14}$$

In this case, the evading method for jump intensity (ζ) is established by $\zeta = 2(1-\gamma_5)$ and $\Delta n_j\ (V) = n_{prey}(V) - n_j(V)$.

Besiege for Hard Target:

If $\gamma \geq 0.5$ and $|\omega_{prey}| < 0.5$, the hard besiege method can be chosen. is the formula is indicated below:

$$n_j\ (V+1) = n_{prey}(V) - \omega_{prey}\ |\ \Delta n_j \cdot\ (V)\ | \tag{15}$$

Besiege for Soft with High Progressive Dives:

If $\gamma < 0.5$ and $|\omega_{prey}| \geq 0.5$, the soft besiege approach can be chosen. It is determined as follows:

$$\varrho = n_j(M) - \omega_{prey}\ |\ \zeta n_{prey}(V - n_j(V))| \tag{16}$$

The next moving stride (ϱ) of the hawks is calculated. In addition, when hawks dive, the following equation is used to attack the prey.

$$D_n = \varrho + R * LF\ (d) \tag{17}$$

As the dimension d of levy flight is specified as $LF\ (d)$, the random vector is also written as R. The updated location is as follows:

$$n_j\ (V+1) = \{\varrho iff(\varrho) < f(n_j(V)) D_n iff(D_n) < f\ (n_j(V)) \tag{18}$$

Besiege for Hard with High Progressive Dives:

If $\gamma < 0.5$ and $|\omega_{prey}| < 0.5$, the mild besiege strategy may be chosen. It is determined by the updated location:

$$n_j\ (V+1) = \{\varrho iff(\varrho) < f(n_j(V)) D_n iff(D_n) < f\ (n_j(V)) \tag{19}$$

As a result of this, the following equation is derived by ϱ,

$$\varrho = y_{prey}(V) - \omega_{prey}|| \zeta n_{prey}(V - n_{j_a}(V))|| \tag{20}$$

The population based on fitness is updated on the location of prey, based on the HHO approach. Thus, the detection of DDoS attacks is effectively classified utilizing MSVM.

5.3. Phase: 2—Reliable Resource Management Using H-WDFOA-VANET

This paper uses hybrid whale–dragonfly optimization-based VANET (H-WDFOA-VANET) for reliable resource management to allocate bandwidth. The proposed H-WDFOA approach is explained, then the background of WOA and DF optimization techniques are individually discussed. In a dynamic VANET scenario, a combination of whale and dragonfly optimization algorithms is represented as H-WDFOA. To achieve bandwidth allocation and obtain routing of the vehicle, the H-WDFOA was employed. Protocol routing also initiated latency and bandwidth utilization to get the designation path. It was difficult during a safety-related and critical time. As a result, to solve this issue, the efficient H-WDFOA technique was utilized to allocate bandwidth and routing protocol.

In a dynamic VANET environment, to see the effect of QoS value, meta-heuristic-based WOA and DF techniques are coupled in this proposed methodology. The hunting behaviour of humpback whales was considered for the whale optimization algorithm

(WOA) generation. The humpback whale's current best strategy is to check its prey's location and encircle it, whereas WOA thinks its prey is a target. After reviewing the current best solution, the best search agent is considered. The different vehicle parameters, such as throughput, number of nearby vehicles, speed, velocity, distance, and speed, were used as the initial population during the start-up phase. The mathematical version of this operation is represented as follows.

$$M = |V \cdot P^*(t) - X(t)| \tag{21}$$

$$P(t+1) = P^*(t) - B \cdot M \tag{22}$$

where t represents iteration, B and V are coefficient vectors, P^* is the position vector of best solution, and P is the position vector

When determining the best solution, P^* of every iteration is periodically updated,

$$B = 2x * r - x \tag{23}$$

$$V = 2 \cdot x \tag{24}$$

Here, r is a random number which range from 0,1.

During the exploitation and exploration phases, the range of the variable is reduced from 2 to 0. At the interval $[-1, 1]$, the phase showed major differences. It was possible for the vehicle to move from its original position to a new position. In the 2D space of the interval $[-1, 1]$, X and Y to X* and Y* were the possible achieved positions. Furthermore, the position was accomplished by updating the procedure, and mathematically represented as:

$$M = |V \cdot P_{rand} - P| \tag{25}$$

$$P(t+1) = P_{rand} - B \cdot M \tag{26}$$

The spiral updating technique can be constructed to analyze the distance of vehicles with one other. This process is determined by the following equation:

$$P(t+1) = M' \cdot e^{sr} \cdot cos\ cos(2\pi r) + P^*(t) \tag{27}$$

Here, M' is the distance of one vehicle $P(t)$ towards another vehicle $P^*(t)$, t is the iteration, s denotes the shape of the logarithmic spiral, and r represents a random number in range $[-1, 1]$.

$$P(t+1) = \left\{ P^*(t) - B \cdot V\ if\ p < 0.5\ M' \cdot e^{sr} \cdot cos\ cos(2\pi r) + P^*(t)\ if\ p > 0.5 \right\} \tag{28}$$

To update vehicle position, 50% probability was chosen for the shrinking encircling mechanism or spiral model. Here, r denoted a random number between the range [0, 1].

To get the maximum optimal solution, the dragonfly optimization approach was hybridized with the whale optimization algorithm to allocate the bandwidth.

The dragonfly's [33] population was initialized in terms of vehicle nodes of $n_j{}^*$.

$$n_j{}^* = n_1{}^*,\ n_2{}^*,\ \ldots,\ n_n{}^* \text{Here, j} = 1, 2, \ldots, \text{n.} \tag{29}$$

In an inquiry space and development reproduction, the dragonfly's artificial position was updated, i.e., position (X) and step (ΔV). The progression vector demonstrates the development of the dragonfly and characters.

$$\Delta X_{t+1} = (wW_i + mA_i + nC_i + pF_i + qE_i) + \overline{w}\ \Delta X_t \tag{30}$$

Here, weight separation is w, weight aliment is m, and n is the cohesion weight, food factor is p, the enemy factor is q, inertia weight is $\overline{w}$, and count iteration is t.

Position vector could be determined as follows:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{31}$$

In the enhancement phase, various explorative and exploitative processes are accomplished. By Cauchy mutation probability, the dragonfly's situation is refreshed at the point where there is no neighbouring solution. The position vector X is determined by,

$$X_{t+1} = X_t + N_p X_t \tag{32}$$

In the entire swarm at any moment, food source and enemies are determined by the entire arrangement. Thus, the accurate vehicle node is selected, and bandwidth allocation is achieved. A brief flowchart is visualized below.

In Figure 7, the flowchart of the hybrid WDFOA-based VANET is visualized. Initially, the whale optimization parameters were initialized based on the search agent. In this work, several nodes in the vehicle were analyzed based on distance, speed, velocity, and bandwidth. The fitness parameter in each iteration determined the current position of the vehicle. If the node's position was less than 0.5, and the nearby vehicle was equal to or greater than 1, the best location was updated. If the current iteration was higher than the maximum iteration, the target vehicle's position was best. DF was coupled with WOA to obtain optimum optimality.

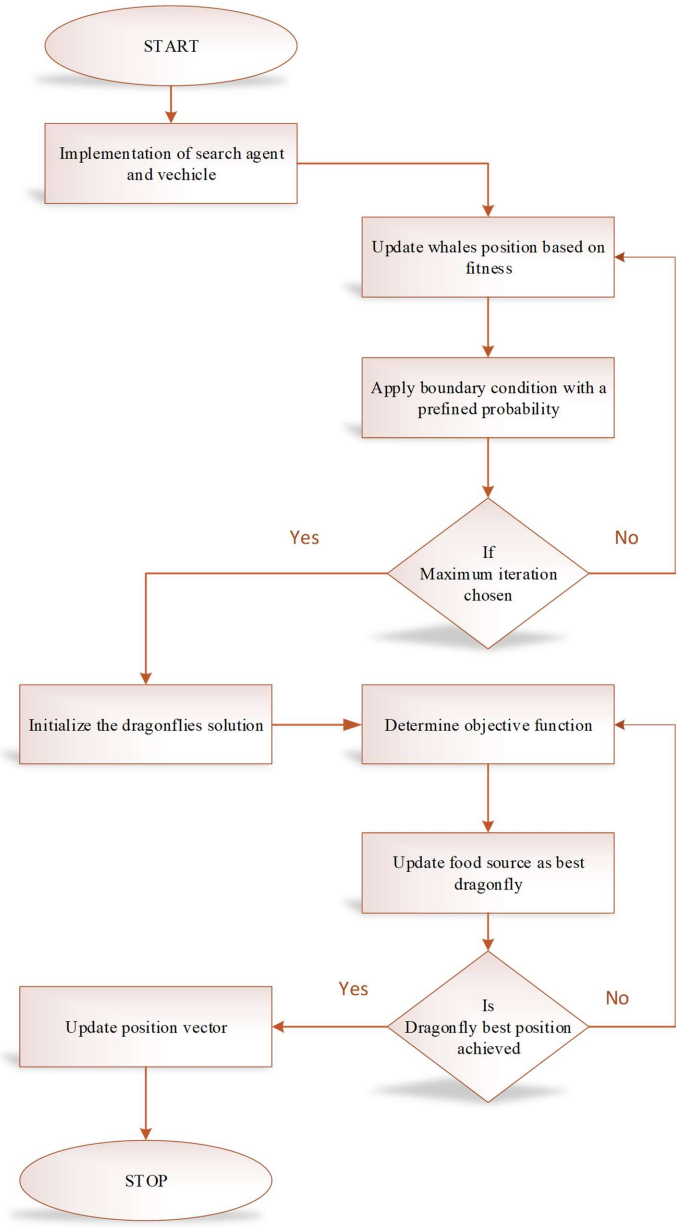

Figure 7. Flowchart for H-WDFOA-VANET.

Furthermore, the dragonfly solution was initialized, and the QoS value determined the bandwidth according to the objective function. The maximum iteration was used until the process is completed and the bandwidth was upgraded. Finally, energy consumption, delay, and drop were considered based on QoS parameters for accurate routing. The proposed approach was valid in allocating bandwidth compared with other existing techniques. This study was robust and fast in predicting DDoS attacks using MSVM and giving the bandwidth using the H-WDFOA technique in the best position after validation for stability in vehicular networks.

6. Results and Discussions

6.1. Dataset Description and Evaluation Metrics

The SDN dataset was created using the Mininet simulator, which was used to classify traffic using machine and deep learning methods [34]. This study began with ten topologies in the Mininet emulator, each of which connected these switches to a single Ryu controller. The network simulators used ICMP, UDP, and TCP traffic, with hostile traffic consisting of a combination of UDP Flood, TCP Sync, and ICMP attacks. The dataset contained a total of 23 features, some of which were collected from switches and others that were estimated. The retrieved features (duration in nan seconds) included Byte count, switch ID, duration, packet count, and duration (in n sec). The sum of duration-sec and duration-nsec was equal to the entire duration. The total number of bytes commuted from the switch port was determined by the destination IP, source IP, port number, and Tx-bytes. Rx-bytes was the total number of bytes received over the switch port. The time and date were transformed into numbers in the dt field, and flow was monitored at 30 s intervals. Furthermore, calculated features such as packet per-flow and byte per-flow counted packets as a single flow. The packet rate was calculated by dividing the total packets per flow by the total number of packets sent per second. Tx-kbps and Rx-kbps were data reception and transmission rates, respectively, and the sums of port bandwidth were Tx-kbps and Rx-kbps. The class labels present in the columns refer to whether the type of traffic was malicious or benign. However, labels 0 and 1 respectively indicate benign and malicious traffic. The duration of the network simulation was 250 min and the row of data possessed was 104,345.

Along with the detailed interval, the simulation of data was authenticated. The dataset was split into two halves for testing and training: 30 and 70%, respectively. The full training process took 42 s to complete [35–37].

The proposed platform was MATLAB 2020a, along with system requirements, which are listed in Table 2.

Table 2. Configuration of Simulation.

System	Requirements
Tool	NS2, MATLAB 2020a
Computer	Windows 10 PRO
Processor	Intel core with high end configuration
RAM	16 GB

6.1.1. Performance Metrics (Ref. [2])

- Efficiency

The efficiency (throughput) parameter defines the total number of packets generated by the sending node in a certain time interval in relation to the quantity of data established via the receiving node in packets. It is represented as:

$$Efficiency(throughput) = number\ of\ received\ data\ packet \times 8/number\ of\ data\ packet\ transmission\ period$$

- Packet Delivery Ratio

This is a method for analyzing incoming and outgoing network data packets in the most efficient way possible. It is calculated based on the ratio of packets generated by the source to data received by the vehicle. The formula utilized to calculate the PDR is:

$$PDR = \frac{ReceivedPackets}{GeneratedPackets} * 100$$

- Drop of packet

During a denial-of-service attack, this parameter tracks the number of packets that were lost because of the malicious behaviour of a single node.

$$Drop\ Ratio = \frac{number\ of\ send\ packet - number\ of\ received\ packet}{number\ of\ send\ packet}$$

- Latency (delay)

It takes time for a package to get from its origin to its destination.

$$delay = \frac{length}{bandwidth}$$

6.1.2. Performance Analysis

The performance of proposed approaches is described in this part using measures such as energy consumption, drop, delay, throughput, packet delivery ratio, and fairness index.

(1) **PHASE 1:**

Phase I detected the DDoS attack using a novel machine learning technique called SVM-HHO, which provided vehicle information. This section describes the analysis, and the proposed performance is detailed below.

Performance measurements were primarily used to assess the model's ability to identify DDoS attacks and allocate bandwidth in VANET situations. From Table 3, it is shown that the proposed modified machine learning-based hybrid optimization technique is more effective compared with existing approaches. In node 60, the proposed methodology attained energy consumption of 0.35 J, delay ratio of 50, and drop ratio of 14.6785. In addition, the suggested approach attained an efficiency of 1710 in comparison with the existing method that attained efficiency of 794.

Table 3. Performance of proposed approach.

Node	Energy Consumption (Joules)	Drop	Delay	Throughput (kilo Bytes/sec)	Network Lifetime	Packet Delivery Ratio (PDR)	Fairness Index (FI)
Proposed Method							
20	0.25	1.6531	97	1278	339.52	20	4.2477
40	0.32	9.9854	37	1920	664.32	33	2.4125
60	0.35	14.6785	50	1710	524.00	48	1.7961
80	0.43	50.5456	95	1345	370.81	59	1.5752
100	0.51	57.1534	68	1209	307.71	64	1.3947

- Consumption of Energy

The total energy consumption of the proposed approach is shown in Figure 8, based on the data in Table 3. The rate of energy consumption of each node is noted. At node 100, the energy consumption of 0.5 is habited. The increase in nodes is accompanied with an increase in energy consumption.

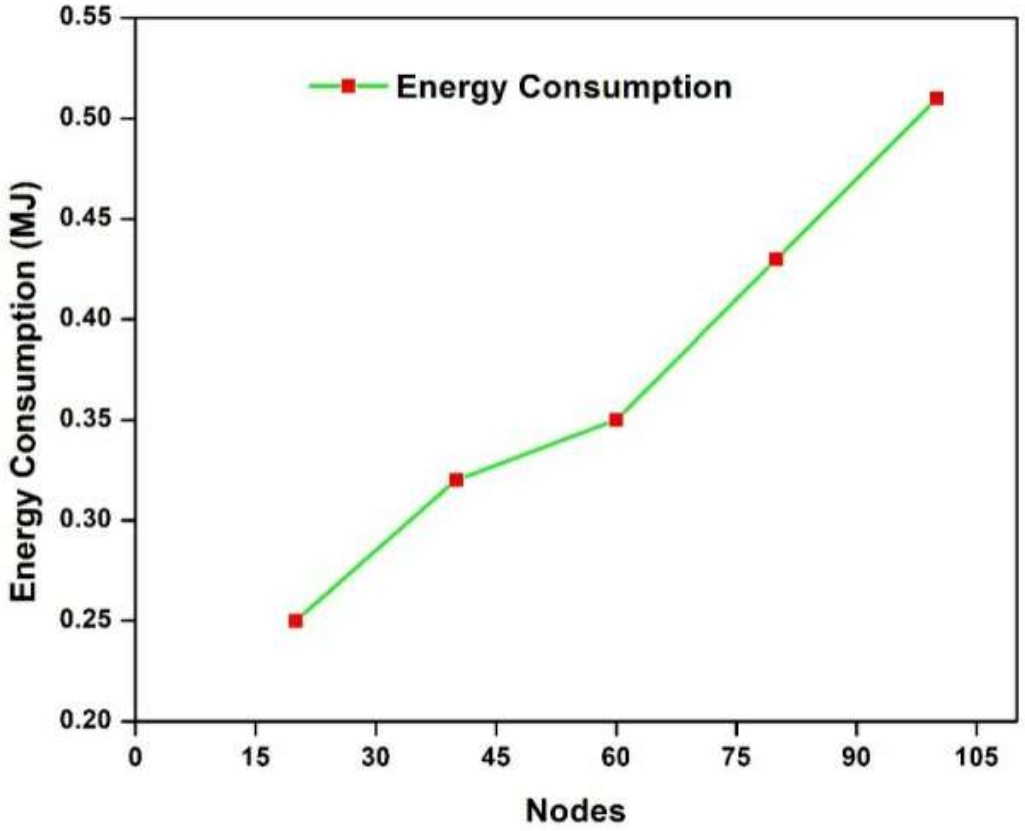

Figure 8. Energy consumption of proposed method.

- Data Rate-Based Energy Consumption

From Table 4, the energy consumption of the proposed method is tabulated at each node with the data set. Nodes with data set values are marked in Table 4.

Table 4. Energy consumption based on data rate.

Node	Data Rate at 4	Data Rate at 6	Data Rate at 8	Date Rate at 10	Data Rate at 12	Data Rate at 14
20	0.8	1.2	1.6	2.1	2.5	2.9
40	0.4	0.6	0.8	1.1	1.2	1.5
60	0.5	0.8	1.0	1.2	1.6	1.8
80	0.7	1.1	1.5	1.9	2.3	2.6
100	0.9	1.3	1.8	2.3	2.7	3.2

Figure 9 shows the data rate-based energy consumption ratio of the suggested approach. If the data packet rates are high, the energy is at a peak; if not, the energy produces lower data rates. From Table 4, we can see that at node 20, energy consumption is 0.8, obtained at a data rate of 4, 1.2 is attained at a data rate of 6, 1.6 is acquired in a data rate of 8, and 2.1 is obtained at a data rate of 10. At a data rate of 12, 2.5 is gained, and at a data rate of 14, energy consumption is 2.9. Similarly, in node 40, energy consumption is 0.4 at a data rate of 4, 0.6 at a data rate of 6, 0.8 at a data rate of 8, 1.1 at a data rate of 10, 1.2 at a data rate of 12, and 1.5 at a data rate of 14. Moreover, at node 60, energy consumption is 0.5 at a data rate of 4, 0.8 at a data rate of 6, 1.0 at a data rate of 8, 1.2 at a data rate of 10, 1.6 at a data rate of 12, and 1.8 at a data rate of 14. In addition, at node 80, energy consumption is 0.7 at a data rate of 4, 1.1 at a data rate of 6, 1.5 at a data rate of 8, 1.9 at a data rate of 10, 2.3 at a data rate of 12, and 2.6 at a data rate of 14. Furthermore, at node 100, energy

consumption is 0.9 at a data rate of 4, 1.3 at a data rate of 6, 1.8 at a data rate of 8, 2.3 at a data rate of 10, 2.7 at a data rate of 12, and 3.2 at a data rate of 14.

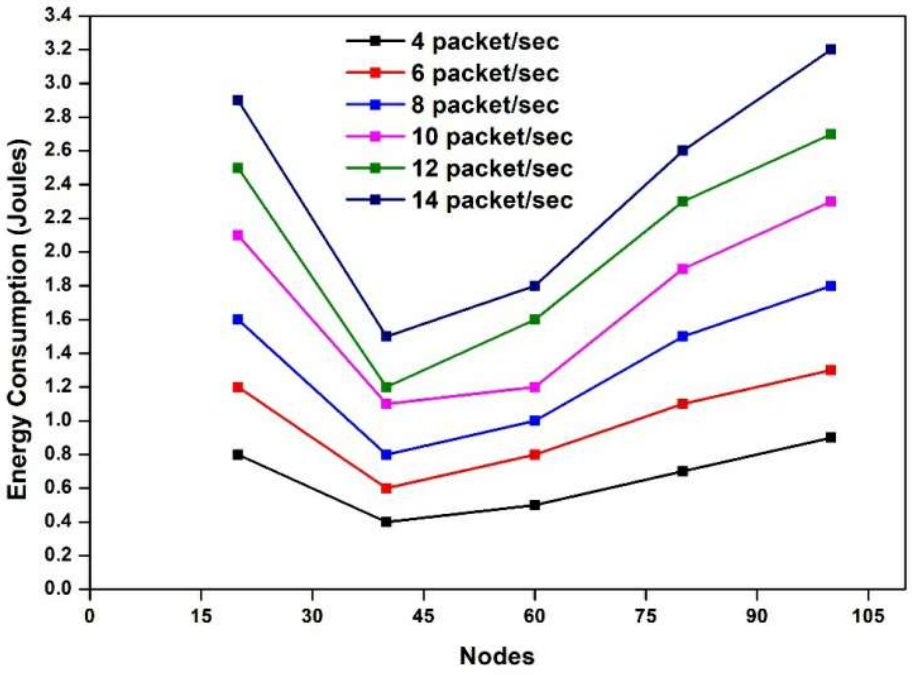

Figure 9. Consumption of energy based on data set.

- Lifetime of the Network

The overall network lifetime of the suggested method from Table 3 is plotted and detailed in Figure 10.

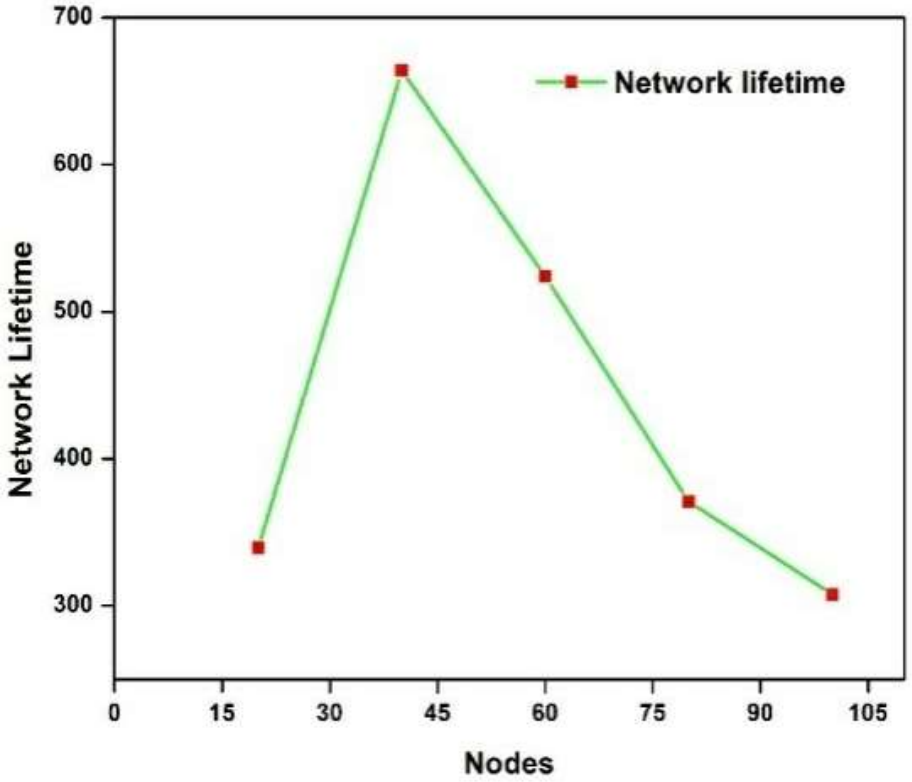

Figure 10. Network lifetime of proposed method.

The lifetime of the network of the suggested approach is shown in Figure 10. The lifetime of the network for each node was noted. At node 100, the network lifetime of 307.71 was habited. The increase in nodes was accompanied with an increase in network lifetime.

- Data Rate-Based Network Lifetime

From Table 5, the network lifetime of the proposed method was tabulated at each node with the data set. Nodes with data set values are marked in the Table 5.

Table 5. Lifetime of network.

Node	Date Rate at 4	Date Rate at 6	Date Rate at 8	Date Rate at 10	Date Rate at 12	Date Rate at 14
20	1339.5	896.1	675.3	500.2	392.2	388
40	2250	1575	1230.1	963	650	600
60	1854.2	1246.5	916.3	669.8	550	500
80	1370.2	932.6	619.1	500.9	400.2	395.4
100	1100	759	530.8	438.5	318	300

Figure 11 demonstrates the proposed system's network lifetime. If the data packet rate is low, the network lifetime appears high. From Table 5, we can see that at node 20, the network lifetime of 1339.5 is obtained at a data rate of 4, 896.1 is attained at a data rate of 6, 675.3 is acquired at a data rate of 8, and 500.2 is accomplished at a data rate of 10. At a data rate of 12, 392.2 is gained, and at a data rate of 14, network lifetime is 388. Similarly, in node 40, the network lifetime is 2250 at a data rate of 4, 1575 at a data rate of 6, 1230.1 at a data rate of 8, 963 at a data rate of 10, 650 at a data rate of 12, and 600 at a data rate of 14. Moreover, at node 60, network lifetime is 1854.2 at a data rate of 4, 1246.5 at a data rate of 6, 916.3 at a data rate of 8, 669.8 at a data rate of 10, 550 at a data rate of 12, and 500 at a data rate of 14. In addition, at node 80, network lifetime is 1370.2 at a data rate of 4, 932.6 at a data rate of 6, 619.1 at a data rate of 8, 500.9 at a data rate of 10, 400.2 at a data rate of 12, and 395.4 at a data rate of 14. Furthermore, at node 100, the network lifetime is 1100 at a data rate of 4, 759 at a data rate of 6, 530.8 at a data rate of 8, 438.5 at a data rate of 10, 318 at a data rate of 12, and 300 at a data rate of 14.

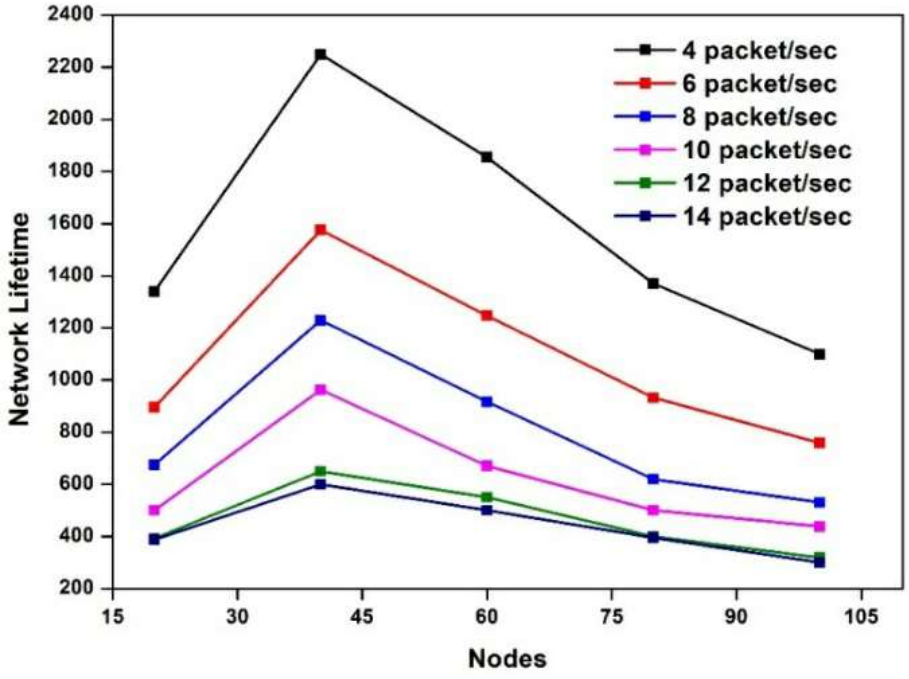

Figure 11. Network lifetime.

- Throughput

The overall efficiency(throughput) of the suggested approach is shown in Figure 12. The rate of throughput of each node is noted. At node 100, a throughput of 1209 was habited. An increase in nodes is accompanied with an increase in throughput.

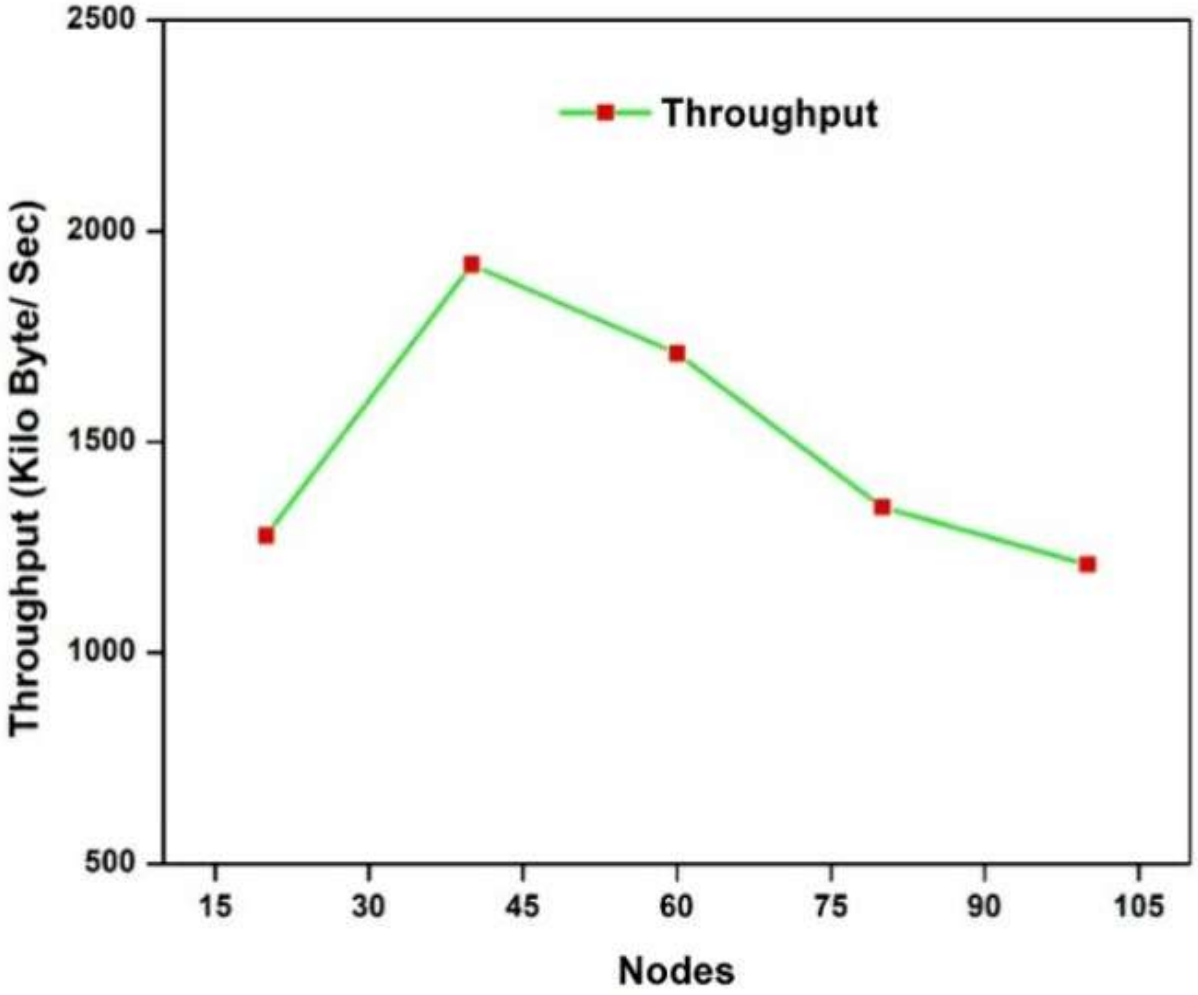

Figure 12. Throughput of the proposed method.

- Data Rate-Based Throughput

In Table 6, the throughput of the suggested approach is tabulated at each node with the data set. Nodes with data set values are marked in Table 6.

Table 6. Throughput based data set.

Node	Data Rate at 4	Data Rate at 6	Data Rate at 8	Data Rate at 10	Data Rate at 12	Data Rate at 14
20	12,700	19,180	25,570	31,960	38,350	44,740
40	19,200	28,800	38,400	48,000	57,600	67,280
60	17,100	25,650	34,200	42,750	51,300	61,700
80	13,450	20,170	26,890	33,620	40,340	47,060
100	12,090	18,140	24,180	30,230	36,280	42,320

Figure 13 demonstrates the throughput ratio of the proposed system. If the data packet rates were low, then the throughput ratio also decreased. The throughput of the proposed technique was reached by 42,320 in node 100 at a data rate of 14 packets/sec. At node

20, 12,700 throughputs were obtained at a data rate of 4. At node 20, 19,180 throughputs were attained at a data rate of 6. At node 20, 25,570 throughputs were acquired at a data rate of 8. At node 20, 31,960 throughputs were accomplished at a data rate of 10. At node 20, 38,350 throughputs were gained at a data rate of 12. At node 20, 44,740 throughputs were fetched at a data rate of 14. Similarly, in node 40, there were 19,200 throughputs at a data rate of 4, 28,800 throughputs at a data rate of 6, 38,400 throughputs at a data rate of 8, 48,000 throughputs at a data rate of 10, 57,600 throughputs at a data rate of 12, and 67,280 throughputs at a data rate of 14. At node 60, there were 17,100 throughputs at a data rate of 4, 25,650 throughputs at a data rate of 6, 34,200 throughputs at a data rate of 8, 42,750 throughputs at a data rate of 10, 51,300 throughputs at a data rate of 12, and 61,700 throughputs at a data rate of 14. In addition, at node 80, there were 13,450 throughputs at a data rate of 4, 20,170 throughputs at a data rate of 6, 26,890 throughputs at a data rate of 8, 33,620 throughputs at a data rate of 10, 40,340 throughputs at a data rate of 12, and 47,060 throughputs at a data rate of 14. Furthermore, at node 100, there were 12,090 throughputs at a data rate of 4, 18,140 throughputs at a data rate of 6, 24,180 throughputs at a data rate of 8, 30,230 throughputs at a data rate of 10, 36,280 throughputs at a data rate of 12, and 42,320 throughputs at a data rate of 14.

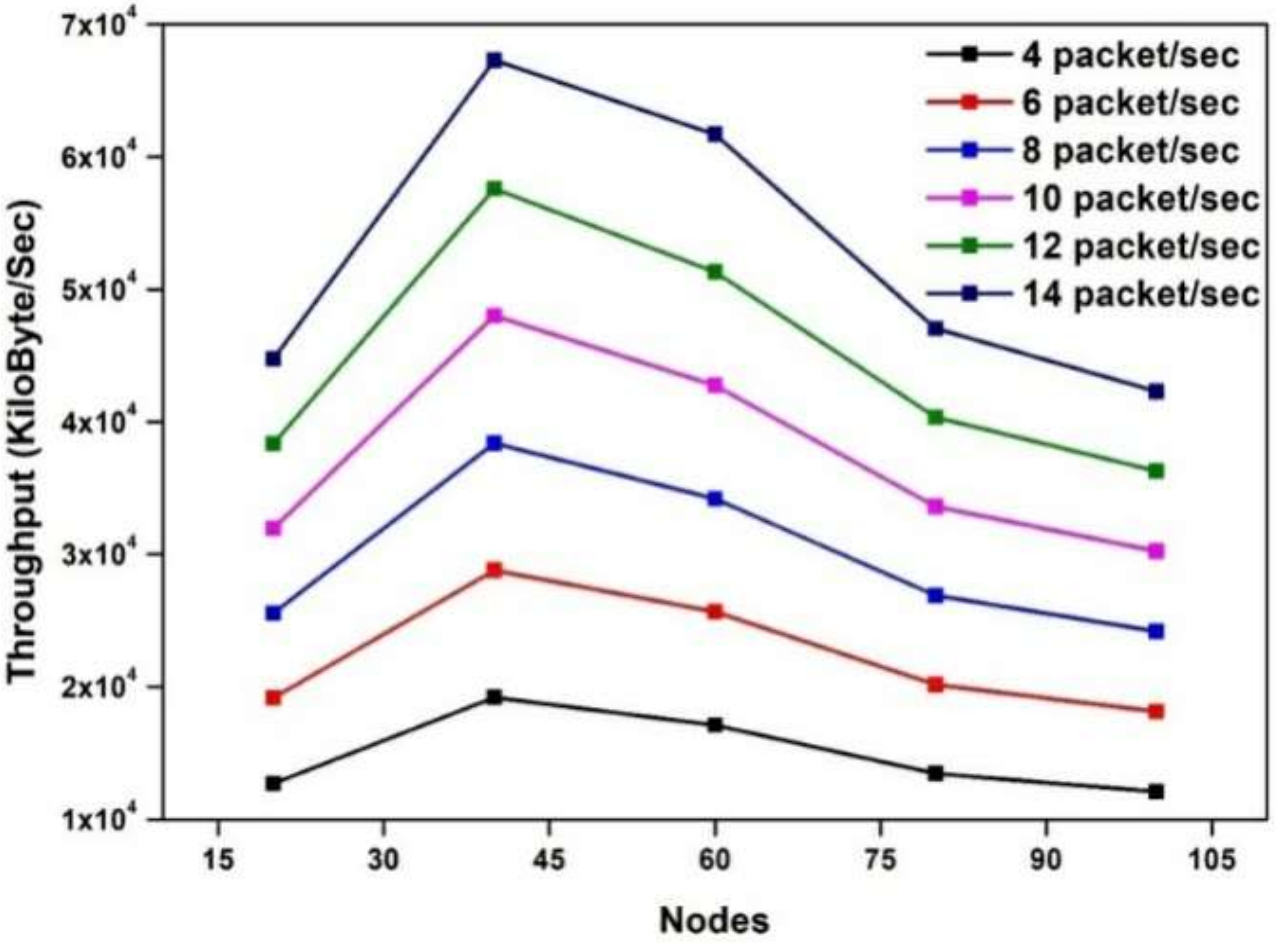

Figure 13. Plot for throughput.

- Packet Delivery Ratio (PDR)

Figure 14 illustrates the packet delivery ratio. To optimize incoming and outgoing network packets, the PDR was evaluated to assess the process. In the VANET scenario, the packet deliver ratio represents the greater presentation of the system.

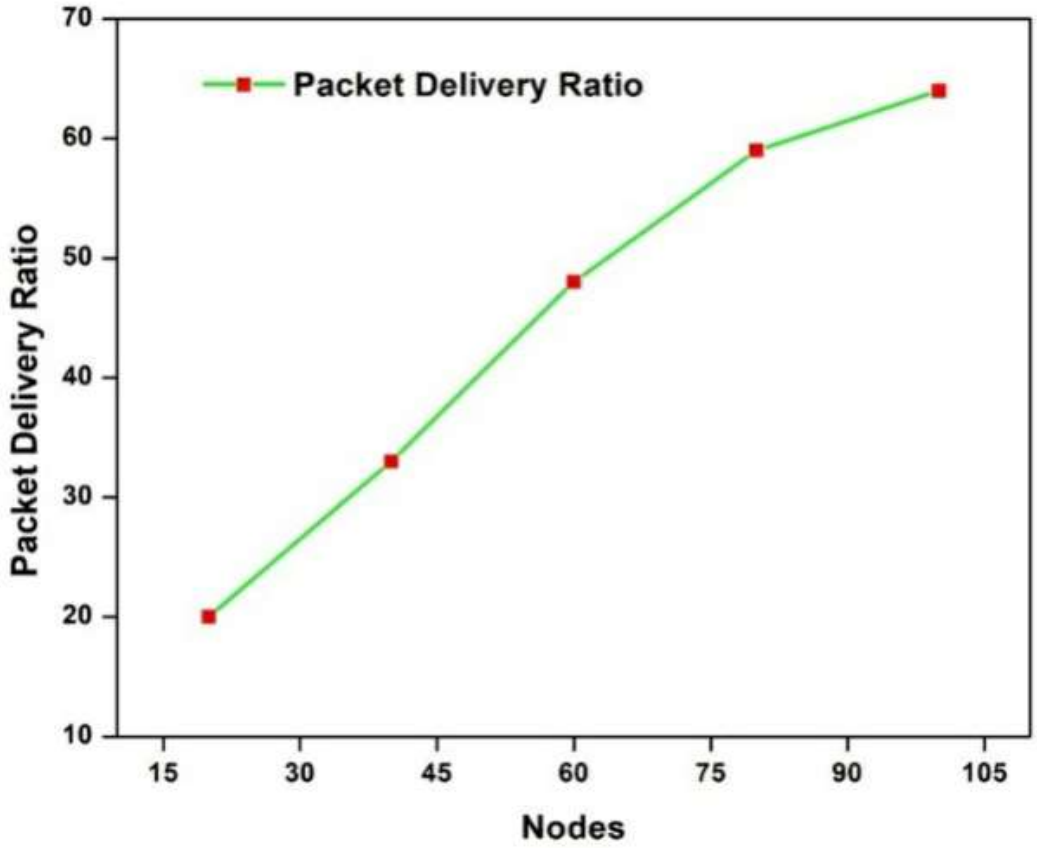

Figure 14. Ratio of packet delivery.

(2) **PHASE 2:**

To better anticipate mobility in VANET, a second phase presented the outcomes of the suggested and implemented approaches for enhancing machine learning with a hybrid optimization strategy. The project's implementation (H-WDFOA-VANET) was analyzed and compared with modern standards (RI-CS, EHACORP). Measured values for the delay, energy consumption, drop, throughput, and fairness index was computed and compared with both proposed (RI-CS) and existing (EHACORP) approaches. In addition, the suggested technique was tested on both the NS2 and MATLAB platforms.

- Results Obtained Through Node

Comparisons were made between the node of the proposed approach and existing techniques in terms of latency, energy consumption, drop, throughput, and fairness index (Figures 15–19). In the table below, we can see the actual values that were measured. Node values for both existing and proposed methods are displayed in Table 7. The major purpose of these performance measures was to evaluate the suggested model's capacity for mobility prediction in VANET. Results from a comparison of the proposed method with state-of-the-art approaches to mobility prediction in VANET are shown in Table 7, which shows that the proposed approach is superior. As such, the performance metrics indicated the proposed model's capacity for mobility prediction in VANET. Table 7 shows that, compared with other methods used to predicting mobility in VANET, the suggested technique, which uses a hybrid optimization strategy to improve machine learning, performs better. Better throughput is achieved using the proposed method. The suggested system achieved 2.00000 mJ of energy consumption, 15.61668 s of latency, and 0.15759 drop in node 60. Furthermore, the throughput attained by the new approach was higher than that of existing methods. There was also a reduction in energy use.

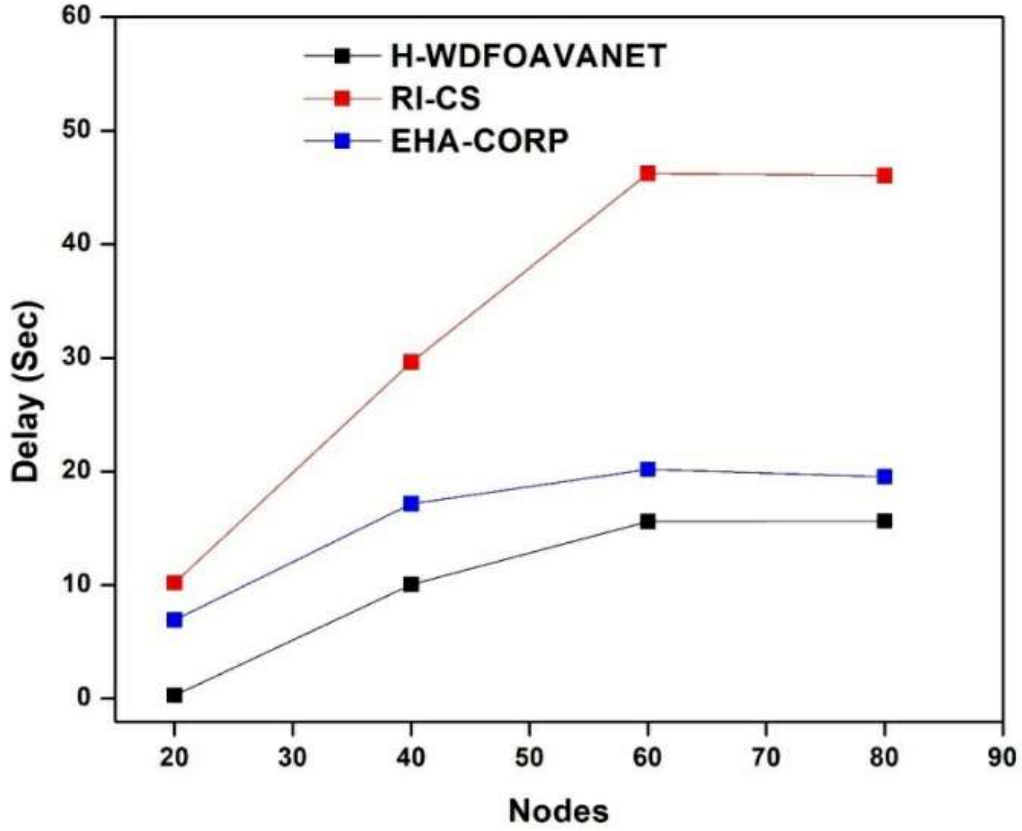

Figure 15. Plots of nodes vs. delays in the suggested approach and existing approaches.

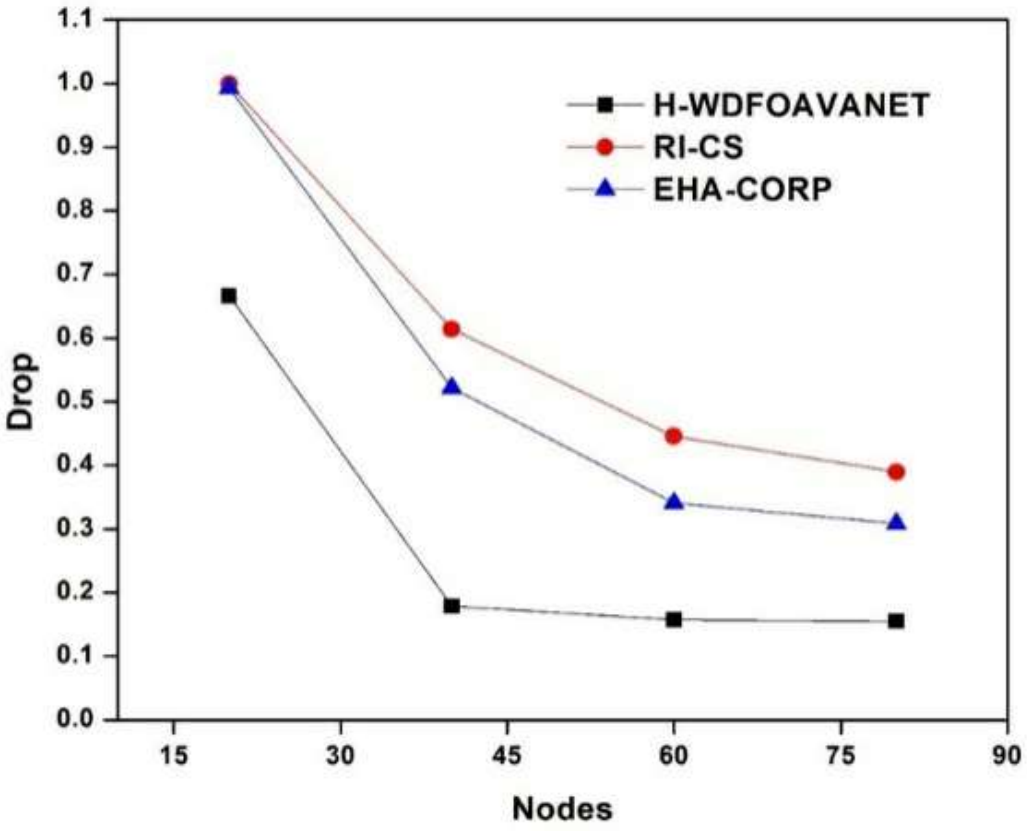

Figure 16. Plots of nodes vs. drop in the suggested approach and existing approaches.

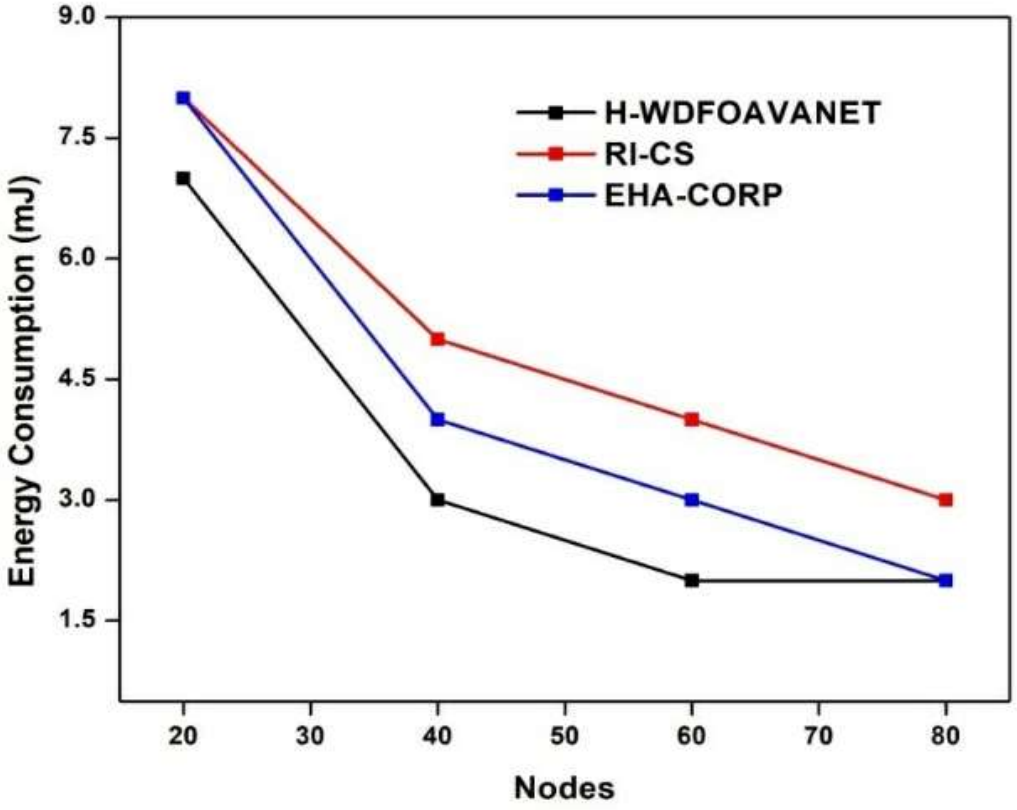

Figure 17. Plots of nodes vs. energy consumption in the suggested approach and existing approaches.

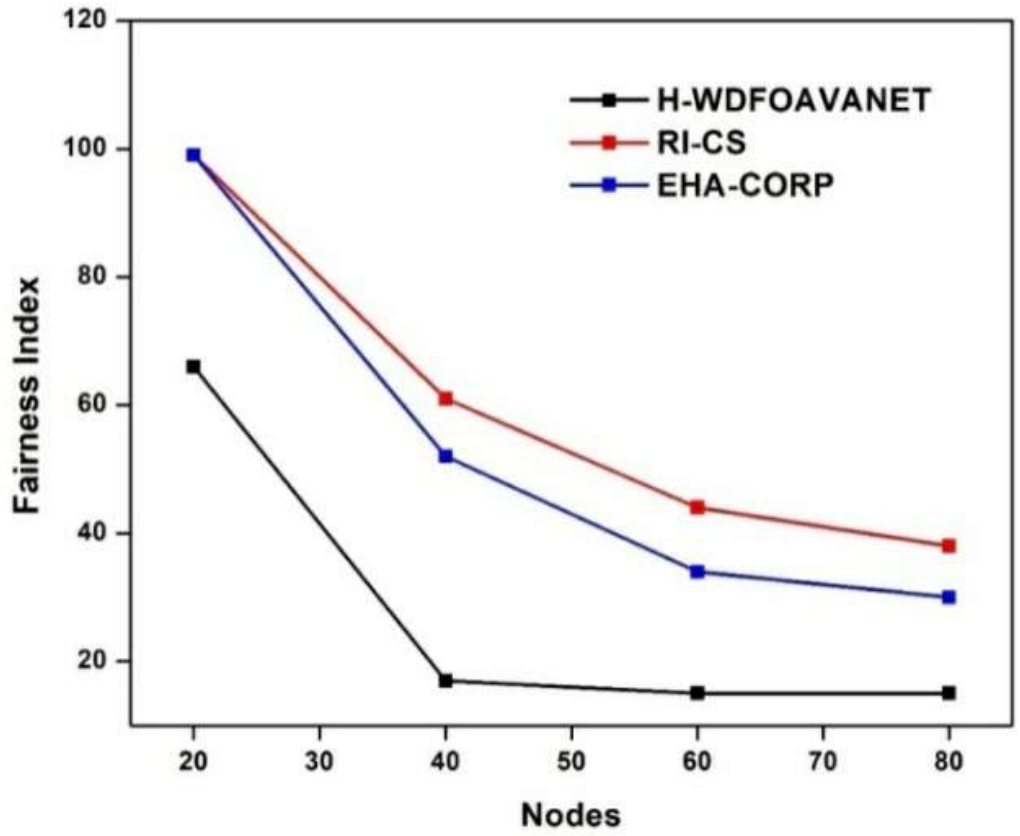

Figure 18. Plots of nodes vs. fairness index in the suggested approach and existing approaches.

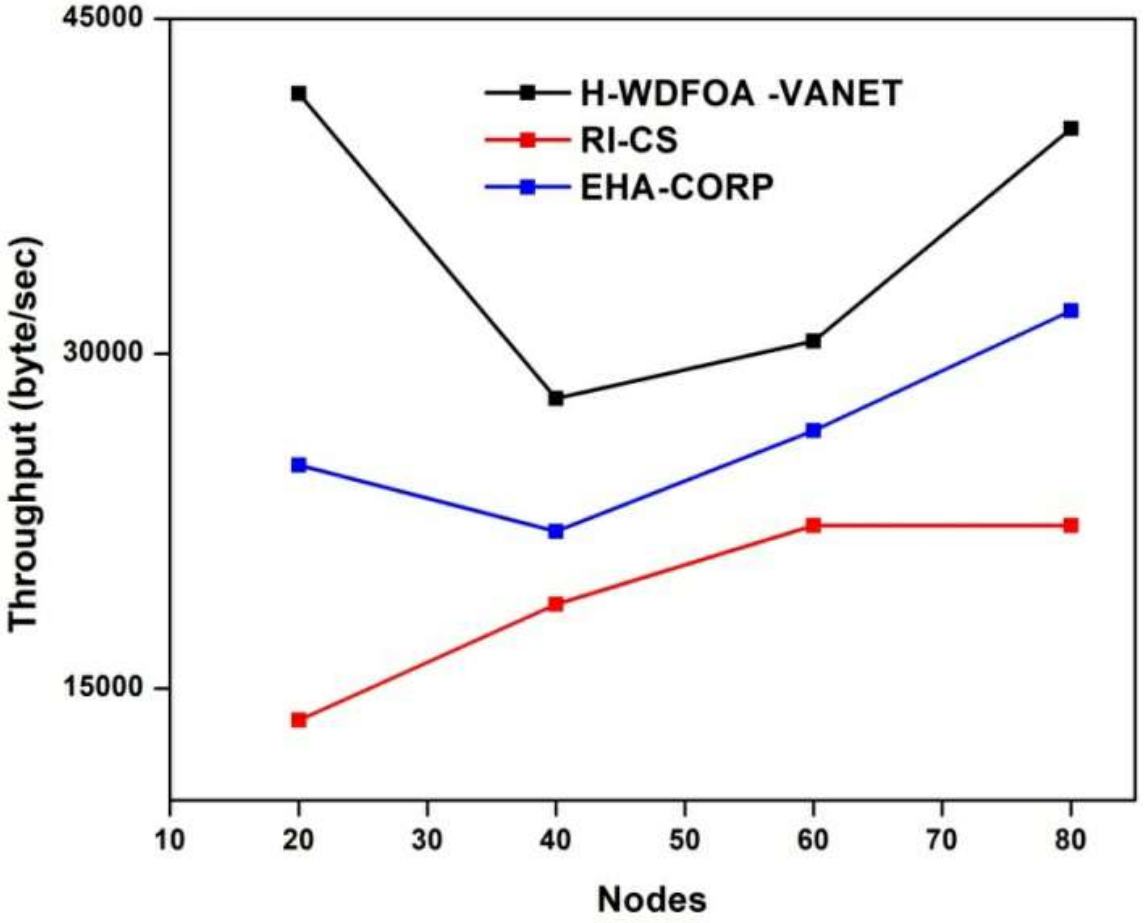

Figure 19. Plots of nodes vs. throughput in the suggested approach and existing approaches.

Table 7. Comparison with existing approaches through nodes.

Performance Parameters	Techniques	Nodes			
		20	**40**	**60**	**80**
Delay (s)	**H-WDFOA-VANET**	0.299063	10.05305	15.61668	15.64318
	RI-CS	10.22709	29.63915	46.23192	46.03685
	EHA-CORP	6.942181	17.167577	20.178139	19.568945
Energy consumption (mJ)	**H-WDFOA-VANET**	7	3	2	2
	RI-CS	8	5	4	3
	EHA-CORP	8	4	3	2
Drop	**H-WDFOA-VANET**	0.66674	0.17871	0.15759	0.15504
	RI-CS	0.99904	0.6142	0.4456	0.3898
	EHA-CORP	0.99293	0.521614	0.340694	0.308867
Throughput (Bps)	**H-WDFOA-VANET**	41,661	27,995	30,560	40,095
	RI-CS	13,582	18,785	202,302	22,299
	EHA-CORP	25,022	22,043	26,556	31,937
Fairness Index (FI)	**H-WDFOA-VANET**	66	17	15	15
	RI-CS	99	61	44	38
	EHA-CORP	99	52	34	30

- Results Obtained by Varying the Speed Parameter

The proposed and existing methods were compared and contrasted in terms of speed with delay, energy consumption, drop, throughput, and fairness index (Figures 20–24). The measured values are displayed in Table 8. The estimated speeds of existing and prospective methods are listed in Table 8. Below is a graphical depiction of the speed of other metrics, such as throughput, drop, energy and fairness index, and delay. An evaluation of the suggested method at speed 20 yielded an energy consumption of 1320 mJ, delay of 90.180295 s, and drop value of 13.334678.

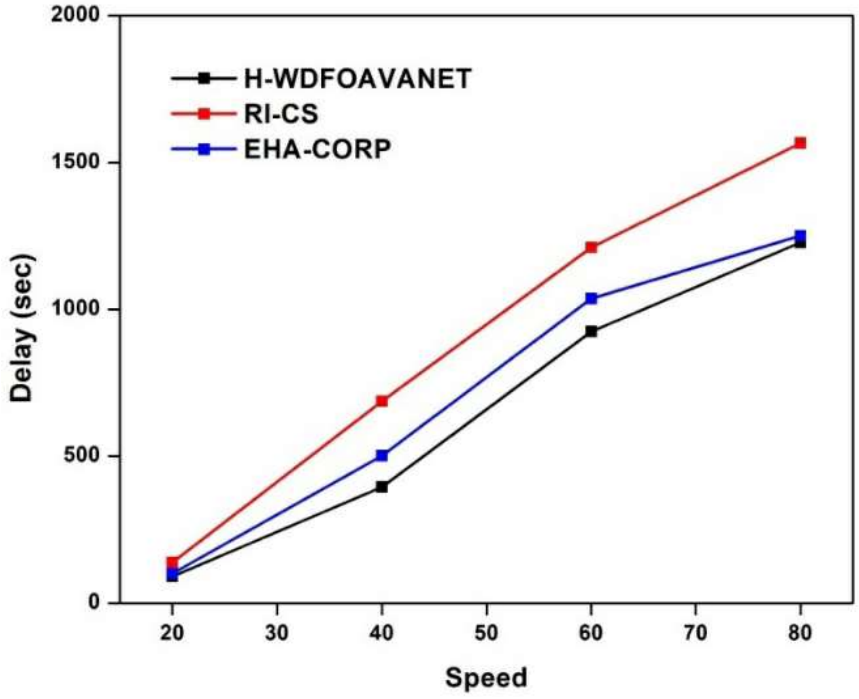

Figure 20. Plots of speed vs. delay in the suggested approach and existing approaches.

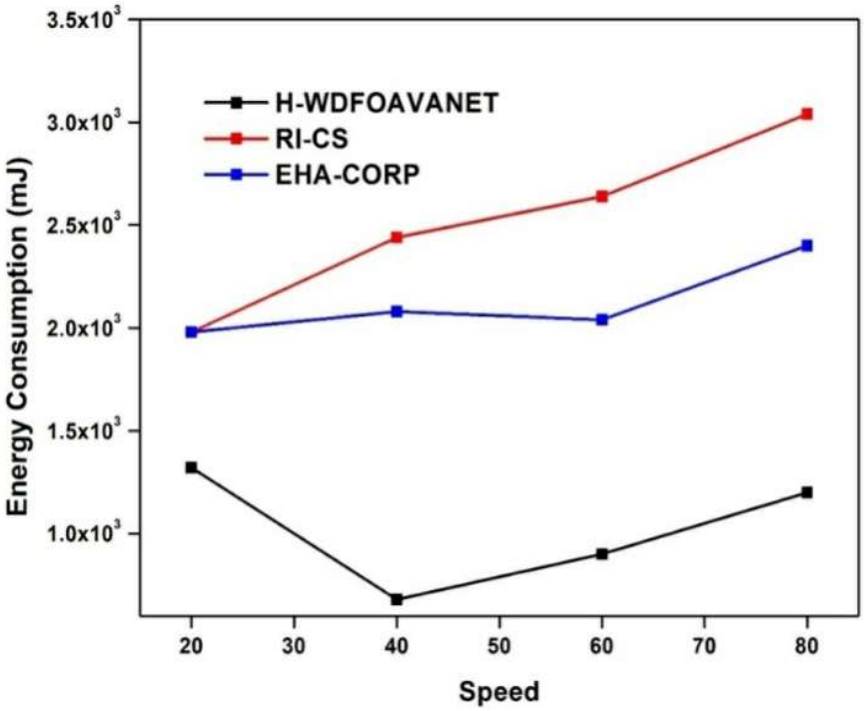

Figure 21. Plots of speed vs. energy consumption in the suggested approach and existing approaches.

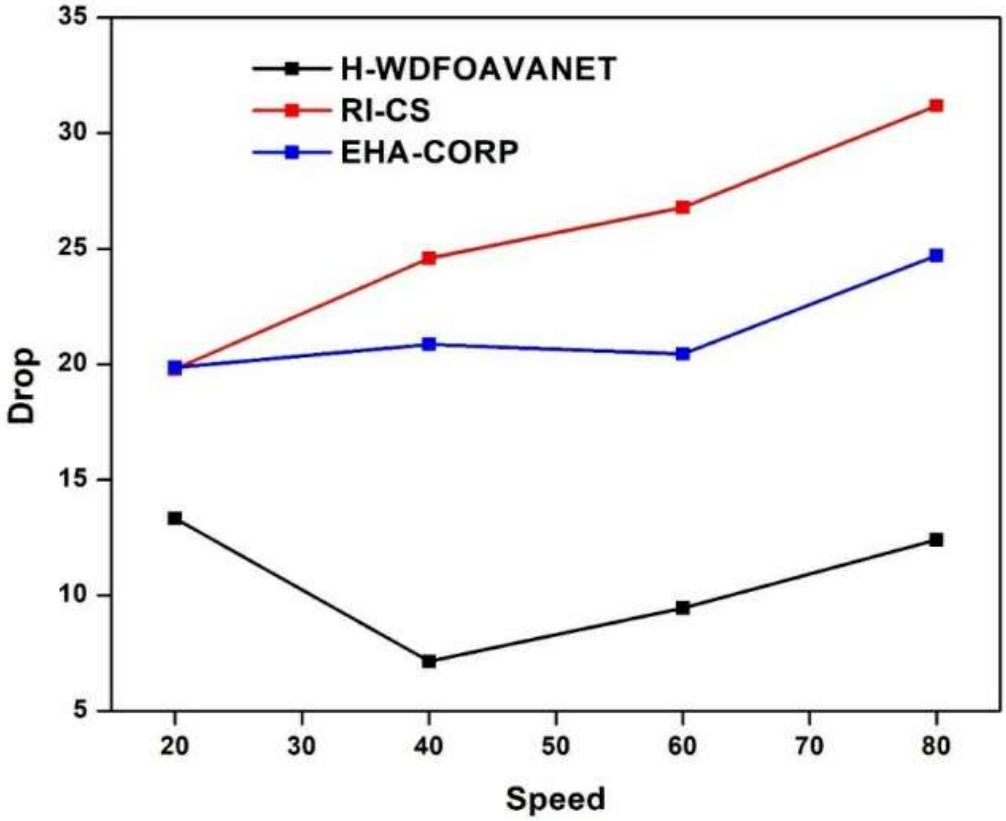

Figure 22. Plots of speed vs. drop in the suggested approach and existing approaches.

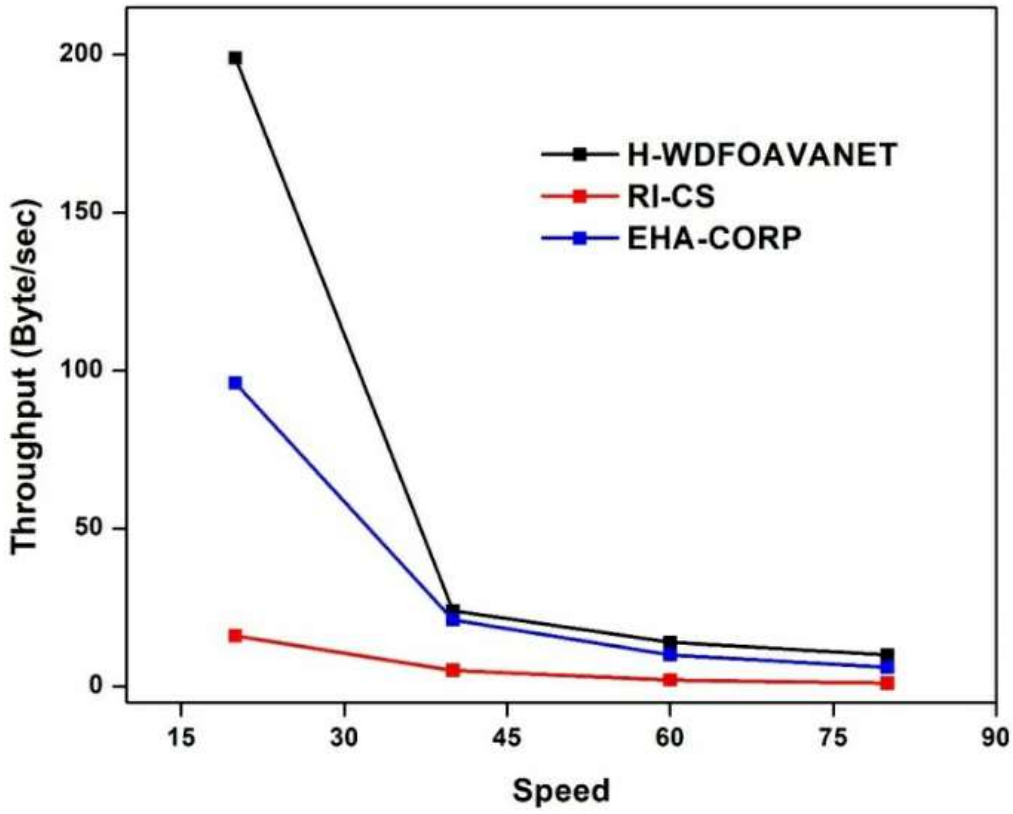

Figure 23. Plots of speed vs. throughput in the suggested approach and existing approaches.

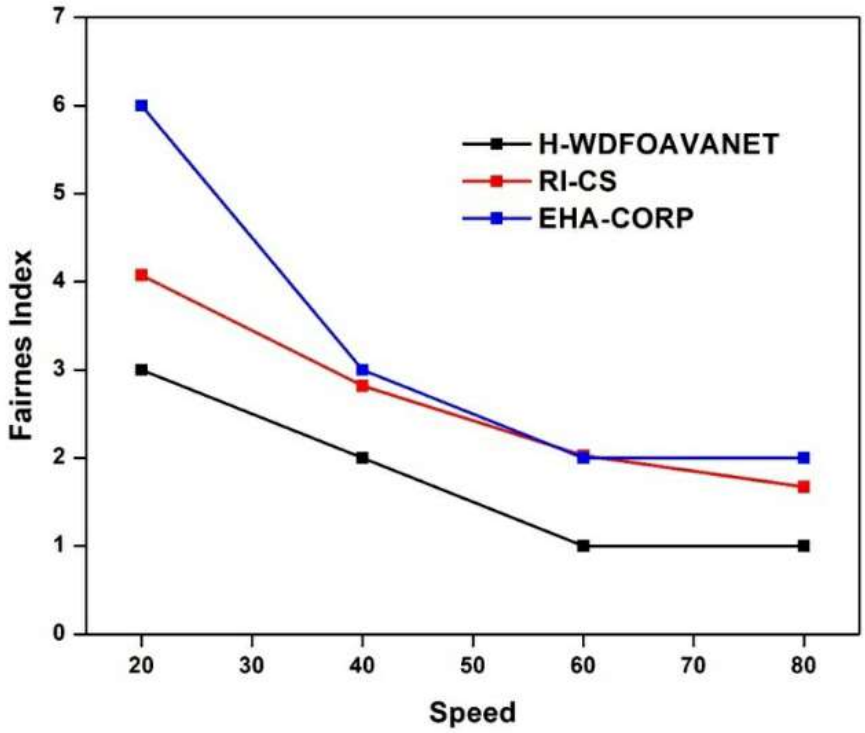

Figure 24. Plots of speed vs. fairness index in the suggested approach and existing approaches.

Table 8. Comparative analysis of suggested approach with existing approaches at different speeds.

Performance Parameters	Techniques	Speed			
		20	**40**	**60**	**80**
Delay (s)	**H-WDFOA-VANET**	90.180295	395.189005	924.638246	1227.591591
	RI-CS	138.843616	686.703072	1210.68834	1565.515813
	EHA-CORP	100.98126	502.120993	1037.0019	1251.455006
Energy consumption (mJ)	**H-WDFOA-VANET**	1320	680	900	1200
	RI-CS	1980	2440	2640	3040
	EHA-CORP	1980	2080	2040	2400
Drop	**H-WDFOA-VANET**	13.334678	7.151254	9.450511	12.405151
	RI-CS	19.80027	24.58287	26.7894	31.195048
	EHA-CORP	19.858595	20.86455	20.441628	24.709388
Throughput (Bps)	**H-WDFOA-VANET**	199	24	14	10
	RI-CS	16	5	2	1
	EHA-CORP	96	21	10	6
Fairness Index (FI)	**H-WDFOA-VANET**	3	2	1	1
	RI-CS	4.0746	2.819775	2.0302	1.672425
	EHA-CORP	6	3	2	2

Additionally, the throughput achieved by the new approach is higher than that of the existing approaches. One key difference between the proposed and existing approaches is that the proposed technique includes a fairness score. With the suggested method, the values for energy consumption, delay, and drop at speed 40 were 680 mJ, 395.189005 s, and 7.151254, respectively. In addition, greater throughput could be achieved using the new

method than existing methods. The proposed method reduces energy usage to 900 mJ, delay to 924.638246 s, and drop value to 9.450511 when traveling at 60 mph.

7. Conclusions

This study's findings mainly centered on detecting DDoS attacks and allocating bandwidth in the VANET, based on a machine learning-hybrid optimization approach. DDoS attack detection was performed using the SVM-HHO technique, i.e., a modified SVM. Then, the impact of DDoS attacks in the VANET was mitigated using a hybrid whale dragonfly optimization technique (H-WDFOA-VANET). Using a unique optimization method called the hybrid whale dragonfly optimization approach (H-WDOA-VANET), which hybridizes the notion of rider integrated cuckoo search, this study found the best route selection with minimal routing costs (RI-CS). To utilize the predicted information after validation, fast and reliable communication was achieved for stability in vehicular networks. Energy consumption, throughput, latency, and packet delivery ratioswere a few performance indicators used to gauge performance of the suggested approach. To increase the effectiveness of the routing process in VANETs, a new protocol, H-WDFOA-VANET, was introduced.

The DCM and SBACO pairing at H-WDFOA-VANET sped up packet processing, decreased ant search time, and eliminated blind broadcasting packets while preventing stagnation issues. The packet delivery ratio and throughput benefit from H-WDFOA-VANET allowed for the use of the shortest channel without stagnation, due to its rapid processing of packets and fast convergence speed. Finally, the suggested hybrid whale dragonfly optimization approach (H-WDFOA-VANET) was evaluated in comparison with industry standard models, such as rider integrated cuckoo search (RI-CS) and enhanced hybrid ant colony optimization routing protocol (EHACORP). The H-WDFOA-VANET system had values of 2.00000 mJ for energy consumption, 15.61668 s for latency, and 0.15759 for drop at node 60. A higher throughput was achieved using the new approach as well. With the suggested method, the energy consumption target, delay value, and drop value at node 60 may be attained. At node 80, the suggested approach minimized the drop value to 0.15504, delay time to 15.64318 s, and energy consumption to 2.00000 mJ. These results demonstrate the efficacy of the proposed H-WDFOA-VANETapproach. Therefore, the efficiency of the proposed system was greater than that of existing systems.

Author Contributions: Conceptualization: G.P.K.M., S.T. and A.J.; writing—original draft preparation: P.K.M., A.A. and C.O.S.; methodology: M.S., R.S., C.O.S. and T.C.M.; writing—review and editing: P.K.M., R.S. and G.P.K.M.; investigation: C.O.S., A.J. and S.T.; supervision: A.A., M.S. and S.T.; visualization: P.K.M. and T.C.M.; software: T.C.M., R.S. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: National Research Development Projects to finance excellence (PFE)-14/2022-2024 granted by the Romanian Ministry of Research and Innovation.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Abbreviations	Full Form
ADMM	Alternating Direction Multiplier Method
AMMSE	Adaptive Mathematical Morphology Spectrum Entropy
ANN	Artificial Neural Network
CRS-MP	Centralized Routing Memechanism with Mobility Prediction
CS	Cuckoo Search
CNN	Convolutional neural network
DCM	Distance Calculation Method
DDoS	Distributed Denial of Service
DF	Dragon Fly
DNS	Domain Name System
EHACORP	Enhanced Hybrid Ant Colony Optimization Routing Proctocal

Abbreviations	Full Form
ERM	Empirical Risk Minimization
GPS	Global Positioning System
HHO	Harris Hwaks Optimization
H-WDFOA	Hybrid Whale Dragonfly Optimization Algorithm
ICMP	Internet Control Messge Protocal l
IDs	Intrusion Detections
MANET	Mobile Adhoc Network
ML	Machine Learning
MMS	Mathematical Morphology Spectrum
MSVM	Multiclass Support Vector Machine
MVSA	MultivariantStream Analysis
NSL-KDD	Network Security Laboratory Knowledge Discovery Dataset
NTP	Network Time Protocal
PML-CIDS	Privacy-preserving Machine-learning BasedCollaborative Intrusion Detection
PPS	Packets Per Second
QoS	Quality of Service e
RI-CS	Rider Intergration Cuckoo Search
ROA	Rider Optimization Algorithm
RPS	Requests per Seconds
RSU	RoadSide unit
SBACO	Source-based Ant Colony Optimization
SDN	SoftwareDefined Networking
SVC	Support Vector Classifier r
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TOPSIS	Technique for Order Performance by Similarity to Ideal Solution
UDP	User Datagram Protocol
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VANET	Vehicle Adhoc Network
VNS-NSGA	Virtual Network System Nondominated Sorting Genetic Algorithm
WAVE	Waveform Audio File Format
WOA	Whale Optimization Algorithm

References

1. Gao, Y.; Wu, H.; Song, B.; Jin, Y.; Luo, X.; Zeng, X. A Distributed Network Intrusion Detection Systemfor Distributed Denialof Service Attacksin Vehicular Ad Hoc Network. *IEEE Access* **2019**, *7*, 154560–154571. [CrossRef]
2. Cheng, N.; Lyu, F.; Chen, J.; Xu, W.; Zhou, H.; Zhang, S.; Shen, X. Big Data Driven Vehicular Networks. *IEEE Netw.* **2018**, *32*, 160–167. [CrossRef]
3. Khan, Z.; Fan, P.; Fang, S.; Abbas, F. An Unsupervised Cluster-Based VANET-Oriented Evolving Graph (CVo EG) Modeland Associated Reliable Routing Scheme. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3844–3859. [CrossRef]
4. Correia, S.; Boukerche, A.; Meneguette, R.I. An Architecture for Hierarchical Software-Defined Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 80–86. [CrossRef]
5. Tan, L.; Pan, Y.; Wu, J.; Zhou, J.; Jiang, H.; Deng, Y. A New Framework for DDoS Attack Detectionand Defensein SDN Environment. *IEEE Access* **2020**, *8*, 161908–161919. [CrossRef]
6. Zhijun, W.; Qing, X.; Jingjie, W.; Meng, Y.; Liang, L. Low-Rate DDoS Attack Detection Basedon Factorization Machinein Software Defined Network. *IEEE Access* **2020**, *8*, 17404–17418. [CrossRef]
7. Shu, Z.; Wan, J.; Li, D.; Lin, J.; Vasilakos, A.V.; Imran, M. Securityin Software-Defined Networking: Threatsand Countermeasures. *Mob. Netw. Appl.* **2016**, *21*, 764–776. [CrossRef]
8. Polat, H.; Turkoglu, M.; Polat, O. Deep network approach with stacked sparse autoencoders in detection of DDoS attacks on SDN-based VANET. *IET Commun.* **2020**, *14*, 4089–4100. [CrossRef]
9. Adhikary, K.; Bhushan, S.; Kumar, S.; Dutta, K. Evaluatingthe Impactof DDo SAttacksin Vehicular Ad-Hoc Networks. *Int. J. Secur. Priv. Pervasive Comput.* **2020**, *12*, 1–18. [CrossRef]
10. Sahoo, K.S.; Tripathy, B.K.; Naik, K.; Ramasubbareddy, S.; Balusamy, B.; Khari, M.; Burgos, D. An Evolutionary SVM Modelfor DDOS Attack Detectionin Software Defined Networks. *IEEE Access* **2020**, *8*, 132502–132513. [CrossRef]

11. Bouyeddou, B.; Kadri, B.; Harrou, F.; Sun, Y. DDOS-attacks detection using an efficient measurement-based statistical mechanism. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 870–878. [CrossRef]
12. Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS Attack Detection Method Based on SVM in Software Defined Network. *Secur. Commun. Netw.* **2018**, *2018*, 9804061. [CrossRef]
13. Hong, K.; Kim, Y.; Choi, H.; Park, J. SDN-Assisted Slow HTTP DDoS Attack Defense Method. *IEEE Commun. Lett.* **2018**, *22*, 688–691. [CrossRef]
14. Kadam, N.; Sekhar, K.R. Machine Learning Approach of Hybrid KSVN Algorithmto Detect DDoS Attackin VANET. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2021**, *12*, 718–722. [CrossRef]
15. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on To N-IoT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
16. Wu, D.; Wu, C. Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows. *Agriculture* **2022**, *12*, 793. [CrossRef]
17. Poongodi, M.; Hamdi, M.; Sharma, A.; Ma, M.; Singh, P.K. DDoS Detection Mechanism Using Trust-Based Evaluation Systemin VANET. *IEEE Access* **2019**, *7*, 183532–183544. [CrossRef]
18. Adhikary, K.; Bhushan, S.; Kumar, S.; Dutta, K. Hybrid Algorithm to Detect DDoS Attacksin VANETs. *Wirel. Pers. Commun.* **2020**, *114*, 3613–3634. [CrossRef]
19. Kolandaisamy, R.; Noor, R.M.; Ahmedy, I.; Ahmad, I.; Z'aba, M.R.; Imran, M.; Alnuem, M. A Multivariant Stream Analysis Approachto Detectand Mitigate DDoS Attacksin Vehicular AdHoc Networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 2874509. [CrossRef]
20. Marwah, G.P.K.; Jain, A. A hybrid optimization with ensemble learning to ensure VANET network stability based on performance analysis. *Sci. Rep.* **2022**, *12*, 10287. [CrossRef]
21. Bangui, H.; Ge, M.; Buhnova, B. A hybrid machine learning model for intrusion detection in VANET. *Computing* **2022**, *104*, 503–531. [CrossRef]
22. Tang, Y.; Cheng, N.; Wu, W.; Wang, M.; Dai, Y.; Shen, X. Delay-Minimization Routing for Heterogeneous VANETs With Machine Learning Based Mobility Prediction. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3967–3979. [CrossRef]
23. Zhang, T.; Zhu, Q. Distributed Privacy-Preserving Collaborative Intrusion Detection Systems for VANETs. *IEEE Trans. Signal Inf. Processing Over Netw.* **2018**, *4*, 148–161. [CrossRef]
24. Gnanasekar, T.S.; Samiappan, D. VANET routing Protocols: Implementation and Analysis Using NS3 and SUMO. *Int. J. Commun. Syst.* **2021**, *34*, e4954. [CrossRef]
25. Ramamoorthy, R.; Thangavelu, M. An enhanced hybrid ant colony optimization routing protocol for vehicular ad-hoc networks. *J. Ambient. Intell. Hum. Comput.* **2022**, *13*, 3837–3868. [CrossRef]
26. Abusitta, A.; Bellaiche, M.; Dagenais, M. An SVM-based framework for detecting DoS attacks in virtualized clouds under changing environment. *J. Cloud Comp.* **2018**, *7*, 9. [CrossRef]
27. Yao, R.; Guo, C.; Deng, W.; Zhao, H. A novel mathematical morphology spectrument ropy based on scale-adaptive techniques. *ISA Trans.* **2022**, *126*, 691–702. [CrossRef]
28. Hossain, M.A.; Noor, R.M.; Yau, K.-L.; Razalli, S.; Zaba, M.; Ahmedy, I.; Jabbarpour, R. Multi-Objective Harris Hawks Optimization Algorithm based 2-Hop Routing Algorithm for CR-VANET. *IEEE Access* **2021**, *9*, 58230–58242. [CrossRef]
29. Prabakeran, S.; Sethukarasi, T. Optimal solution for malicious node detection and prevention using hybrid chaotic particle dragonfly swarm algorithm in VANETs. *Wirel. Netw.* **2020**, *26*, 5897–5917. [CrossRef]
30. Nisha, A.; Gaurav, S. Mukhopadhyay, Debajyoti (2020), "DDOS attack SDN Dataset", Mendeley Data, V1. Available online: https://data.mendeley.com/datasets/jxpfjc64kr/1 (accessed on 10 September 2022).
31. Khalaf, O.I.; Ogudo, K.A.; Singh, M. A Fuzzy-Based Optimization Technique for the Energy and Spectrum Efficiencies Trade-Off in Cognitive Radio-Enabled 5G Network. *Symmetry* **2021**, *13*, 47. [CrossRef]
32. Walia, G.S.; Singh, P.; Singh, M.; Abouhawwash, M.; Park, H.J.; Kang, B.; Mahajan, S.; Pandit, A.K. Three Dimensional Optimum Node Localization in Dynamic Wireless Sensor Networks. *Comput. Mater. Contin.* **2022**, *70*, 305–321.
33. Singh, M.; Kumar, M.; Malhotra, J.; Tiwari, S.; Trivedi, M.; Kohle, M.L. Energy efficient cognitive body area network (CBAN) using lookup table and energy harvesting. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1253–1265. [CrossRef]
34. Majumder, S.; Mathur, A.; Javaid, A.Y. A study on recent applications of blockchain technology in vehicular adhoc network (VANET). In *National Cyber Summit*; Springer: Cham, Switzerland, 2019; pp. 293–308.
35. Wahab, O.A.; Mourad, A.; Otrok, H.; Bentahar, J. CEAP: SVM-based intelligent detection model for clustered vehicular ad hoc networks. *Expert Syst. Appl.* **2016**, *50*, 40–54. [CrossRef]
36. Zhang, C.; Chen, K.; Zeng, X.; Xue, X. Misbehavior detection based on support vector machine and Dempster-Shafer theory of evidence in VANETs. *IEEE Access* **2018**, *6*, 59860–59870. [CrossRef]
37. Thilak, K.D.; Amuthan, A.J.F.G.C.S. Cellular automata-based improved ant colony-based optimization algorithm for mitigating ddos attacks in vanets. *Future Gener. Comput. Syst.* **2018**, *82*, 304–314. [CrossRef]