Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 32s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

A Modified Possibilistic Fuzzy C Means (MPFCM) Clustering based Intrusion Detection Framework for VANETs Using Improved Whale Optimization with Enhanced Deep Neural Networks

Mrs. S.DE. Kalaivani[1], Dr. M.Santhalakshmi[2]

[1]Research Scholar, Kamban College of arts and Science, Sulthanpet.

[2]Assistant Professor of Computer Science, Kamban College of arts and Science, Sulthanpet

Cite this paper as: Mrs. S.DE. Kalaivani, Dr. M.Santhalakshmi, (2025) A Modified Possibilistic Fuzzy C Means (MPFCM) Clustering based Intrusion Detection Framework for VANETs Using Improved Whale Optimization with Enhanced Deep Neural Networks. *Journal of Neonatal Surgery*, 14 (32s), 6631-6649.

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) operate in highly dynamic and high-velocity environments, making them especially vulnerable to a wide range of intrusions and malicious attacks. To address these challenges, this paper proposes a hybrid IDS framework that combines multiple state-of-the-art techniques tailored for VANET traffic. Initially a Z-Score Normalization ensures all features share a consistent scale, improving model convergence and preventing domination by outliers in vehicular data. Next, an Improved Whale Optimization Algorithm (IWOA) is employed to select the most discriminative feature subset by simulating whale-hunting strategies with adaptive parameters for robust global search in rapidly changing VANET conditions. Then an Improved Deep Neural Network (IDNN) functions as a signature-based classifier, rapidly identifying known intrusions through enhanced architecture, optimization, and regularization. Data classified as normal or inconclusive is then passed to a Modified Possibilistic Fuzzy C Means (MPFCM) clustering based anomaly detection module, leveraging fuzzy logic to isolate novel or zero-day threats that deviate from typical vehicle communication patterns. The framework's performance is evaluated using accuracy, precision, recall, and f-measure, ensuring a balanced view of both detection thoroughness and correctness. Experimental results demonstrate that this integrated solution achieves high detection rates, low false alarms, and robust adaptability to evolving attack behaviors in VANETs, making it a promising approach to secure next-generation intelligent transportation systems.

Keywords: *Vehicular Ad-hoc Network, Road Side Units, Z-Score Normalization, Improved Whale Optimization Algorithm (IWOA), Fuzzy C-Means, Anomaly detection, Security.*

1. INTRODUCTION

Communication technology plays an essential role in recent vehicle network management. In conventional methods, the wired communication protocols were developed to control the various internal parts of the vehicle under single devised system architecture. This wired method increases the system and maintenance cost of the vehicle. In order to improve cost efficiency, modern vehicular networks use wireless protocols to transfer or receive the data wirelessly within or outside the vehicles. VANET is the recent efficient wireless technology which is presently used in many intelligent transportation networks [1–2], carpooling [3], and even using fifth-generation small-cell networks [4]. Each vehicle belonging to VANET system consists of multiple wireless sensors, converting equipment, and mapping units. Also, the VANET system consists of two interfacing modules, ad On-Board Unit (OBU) and Road Side Units (RSU). The OBU module is integrated within the vehicle and connects to all the wireless sensors within the vehicle. The RSU module is fitted in roadside buildings with individual transmitter and receiver units to communicate each vehicle's OBU module. When the vehicle enters the VANET system, it senses the vehicle information through the multiple sensors fitted in it and sends all this data to the RSU module. The accidents will be prevented if there is confident coordination between the vehicle OBU module and RSU module. The performance between the OBU module and RSU module will be affected by the presence of intruders. Hence, there is a need to provide a security mechanism between OBU and RSU modules.

By implementing these VANET techniques, accidents are significantly reduced by exchanging the vehicle information with its nearby or surrounding vehicles. This VANET can be categorized into Vehicle-to-Vehicle (VV) and Vehicle-to-Infrastructure module (VI). The VV module of the VANET system transfers the information between vehicle and vehicle. In the case of VI module, if the VANET system, the information is transferred from one vehicle to the centralized system or controller. The real-time environment scenario of the VANET is dynamic, and its topology system is changing with respect to the distance and location of vehicles. The factors such as environmental noises affect the quality of the information passage

between the vehicles [5–7]. This type of vehicle environment is called a rugged VANET environment, which is easily affected by external attacks such as eavesdropping and hacking the data. Figure 1 illustrates the VANET environment where all the vehicles are connected wirelessly to the centralized controller. The vehicles in VANET and centralized controller are attacked by the attacker. Figure 1 shows the VANET systems, which connect multiple vehicles to the centralized controller, which is called as RSU module. The attackers mostly affected the interference between each vehicle and the interference between the vehicle and the centralized controller.

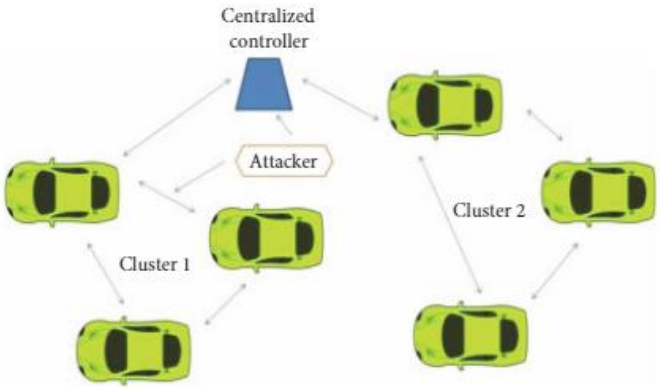

Figure 1: VANET system

The attackers generate different types of attacks to collide with the functional activities of the network environment in VANET system. These will also affect the lives of people who are driving vehicles in these environmental conditions. Therefore, detecting the attacks in VANET system is very important to provide more secure and reliable communication between all vehicles in the system. The attacks in VANET are classified into either external attacks or internal attacks. Internal attacks can be detected or identified using cryptography methods, which use a digital signature to perform encryption and decryption. These methods are not able to detect the external attacks [8–9]. Therefore, the IDS is required in VANET to provide security from external attacks. The external attacks are categorized into Denial of Service (DoS) attacks, Botnet attacks, PortScan attacks, and Brute Force attacks. DoS attack can be generated by any attacker who statically launches the attack in a particular location, or any moving vehicle can launch the attack. The attacker intends to disrupt the network services that are used by the vehicles. Botnet attack is generated due to the devices which are affected by malware. The ports in the device can be affected or attacked by the PortScan attack. The login credentials and passwords of the network devices can be affected by the Brute Force attack.

Despite the promising future of VANETs, they are known to be sensitive to various misbehaves, ranging from malicious attacks to random failures [10]. Considering the safety of vehicles is directly related to human lives, security is one of the main challenges in VANETs. Various detection methods have been proposed in the past decade to detect and mitigate Intrusions in VANETs [11]. Most of these presented methods overlook the security of senior units or just simply rely on a set of predefined and fixed threshold(s) to secure the senior units [12]. However, senior units, Road Side Units (RSUs) and Cluster Heads (CHs), are not guaranteed to be safe in a VANET. Although RSUs are built to be robust, yet intruders can still impair the system through physical attacking RSUs or impersonating as an RSU. Not to mention that CHs are easier than RSUs to be impersonated or overtook. The overlook of those senior units' security can lead to serious consequences. Furthermore, considering the highly dynamic nature of VANETs, it is not achievable to find a set of fixed thresholds to detect malicious nodes [13]. In contrast, Machine Learning based (ML-based) intrusion detection method can automatically determine whether a node is malicious or not considering all available data from the VANET [14]. In existing work introduced the limitations of traditional intrusion detection approaches (e.g., high false positives, lack of adaptability to zero-day attacks). Hence a hybrid solution is proposed to catch both known and unknown threats. Also the Min-max normalization is often used because of its simplicity—it rescales feature values into a fixed range (e.g., [0, 1]). However, min-max normalization can be sensitive to outliers and may lead to suboptimal performance when your data has skewed distributions or a broad range of values. The primary aim of the research is to develop a robust, efficient, and adaptive intrusion detection system (IDS) tailored for Vehicular Ad hoc Networks (VANETs). This framework seeks to effectively identify and classify network intrusions by integrating three key components:

- Modified Possibilistic Fuzzy C Means (MPFCM) clustering: To group and preprocess network traffic data by uncovering inherent patterns and segmenting normal and abnormal behaviors.
- Improved Whale Optimization Algorithm: To optimize feature selection and fine-tune the model's parameters, thereby enhancing computational efficiency and reducing the dimensionality of the input data.

- Enhanced Deep Neural Networks: To accurately learn complex intrusion patterns and perform precise classification of potential security threats.

2. RELATED WORKS

In [15] proposed an anomaly detection framework for VANETs based on deep neural networks (DNNs) using a sequence reconstruction and thresholding algorithm. In this framework, the DNN architectures are deployed on the roadside units (RSUs) which receive the broadcast vehicular data and run anomaly detection tasks to classify a particular message sequence as anomalous or genuine. Multiple DNN architectures are implemented in this experiment and their performance is compared using key evaluation metrics. Performance comparison of the proposed framework is also drawn against the prior work in this area. Our best performing deep learning-based scheme detects anomalous sequences with an accuracy of 98%, a great improvement over the set benchmark.

In [16] presents a novel scheme for minimizing the invalidity ratio of VANET packets transmissions. In order to detect unusual traffic, the proposed scheme combines evidences from current as well as past behaviour to evaluate the trustworthiness of both data and nodes. A new intrusion detection scheme is accomplished through a four phases, namely, rule-based security filter, Dempster–Shafer adder, node's history database, and Bayesian learner. The suspicion level of each incoming data is determined based on the extent of its deviation from data reported from trustworthy nodes. Dempster–Shafer's theory is used to combine multiple evidences and Bayesian learner is adopted to classify each event in VANET into well-behaved or misbehaving event. The proposed solution is validated through extensive simulations. The results confirm that the fusion of different evidences has a significant positive impact on the performance of the security scheme compared to other counterparts.

IN [17] presents a hybrid approach for intrusion detection in vehicular networks that effectively balances classification performance and model efficiency. Our method, which combines automated feature engineering through correlation-based feature selection (CFS) and principal component analysis (PCA) with optimized deep learning techniques, has yielded promising results. The proposed hybrid model significantly improved classification performance, increasing the F1-score from 96.48% to 98.43%. Moreover, our feature engineering techniques reduced the model size from 28.09 KB to 20.34 KB, optimizing both performance and resource usage. Post-training quantization further compressed the model to 9 KB, demonstrating substantial optimization potential. Experiments using the CICIDS 2017 dataset validated the model's effectiveness in classifying malicious traffic in vehicular network scenarios. These findings prove that it is possible to develop high-performing intrusion detection systems with low computational complexity, making them suitable for deployment in resource-constrained vehicular ad-hoc network (VANET) environments.

Danalakshmi et al. [18] proposed an IDS technique in which the Deep Belief Network is used by enhancing with a rule-based technique to improve the accuracy of the detection rate and reduce the false alarm rate as well. The authors concentrated only on False Data Injection and DoS attacks. To identify the port scan attacks, efficient detection rules are generated in the IDS, which detects the real-time native port scan attacks using Snort. But the snort has some limitations that, due to noise, can limit the effectiveness of IDS.

Guangzhen Zhao et al. [19] utilized two classification models, DBF and Probabilistic Neural Network (PNN), for detecting intrusion in the VANET system. The authors analyzed the performance of the system by implementing the IDS system with these two classification models. Hao [20] used an encryption-based key management system for detecting the various attacks in IDS system of the VANET environment. The developed key management system provided different encrypted keys for handling a large amount of data between the roadside unit and the vehicles. Daeinabi et al. [21] proposed a vehicular weighing clustering algorithm (VWCA) for improving the security level of the nodes in VANET. The authors constructed a weight-based clustering framework for detecting the nodes being attacked by the host node.

Mengting Yao et al. [22] proposed mutual authentication method for improving the security enhancement in VANET using a forward secrecy approach. The shared key in this method was verified through the batch normalization process. The authors detected impersonation and forgery attacks using this mutual authentication technique. Shen et al. [23] developed a data aggregation approach for VANET to improve security performance. The authors used a batch verification approach between each transmission process of sender and receiver in order to provide a trust behavior network.

Gope et al. [24] improved security authentication of VANET by developing a privacy-preserving approach between vehicle and grid. The authors applied and tested the developed security authentication scheme in different rouged environments. Gayathri et al. [25] used certificateless approach to prove the authentication scheme in the VANET environment. This method used certificateless keys between the roadside units and central units in VANET. The effectiveness of this keyless approach was analyzed using hit rate and miss rate analysis parameters.

Nayyar et al. [26] developed a hybrid data model for the detection of intrusion in the VANET environment. This method was based on the hybrid model and integrated with the non-linear prediction flow to determine the intrusion activities in VANET. Naqvi et al. [27] detected the malicious activities or any misbehavior activities of the vehicle in VANET using IDS flow. The authors mainly focused on providing more reliability and security for the vehicle nodes in VANET. The

experimental results of this proposed method were compared with other similar algorithms in the same VANET environment. Jabar Mahmood et al. [28] analyzed and synthesized various problems faced in the security flow of the VANET system. The determined countermeasures in VANET identified the major security threats and resolved the issues in VANET, which also optimized the efficiency of the entire network. Irshad et al. [29] proposed a security key authentication system for the VANET system to provide reliable and secure access to vehicle users. The authors discussed various authentication protocols to improve the security of the VANET system.

3. PROPOSED METHODOLOGY

This work presents a hybrid Intrusion Detection System (IDS) framework specifically designed for VANET traffic is shown in figure 2. The framework begins with Z-Score Normalization to standardize feature scales, followed by an Improved Whale Optimization Algorithm (IWOA) for discriminative feature selection. An Improved Deep Neural Network (IDNN) is then employed as a signature-based classifier to identify known intrusions rapidly. Data that is classified as normal or inconclusive is further analyzed by a Modified Possibilistic Fuzzy C Means (MPFCM) clustering module to detect anomalies or zero-day attacks.

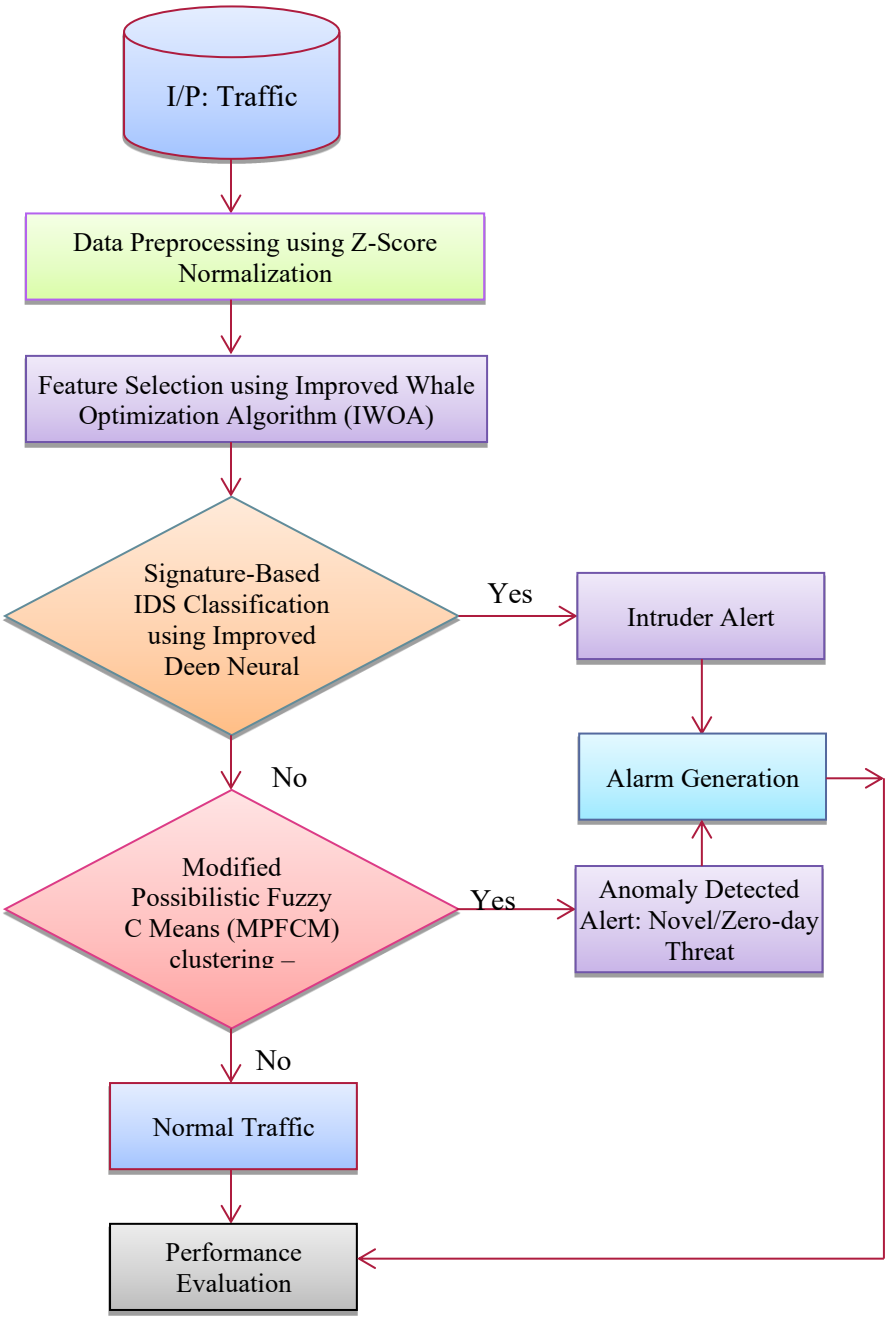

Figure 2: Proposed Flow Diagram

3.1 Data Preprocessing using Z-Score Normalization

Initially load the CICIDS2017 dataset. The dataset is undergone preprocessing sing Z score normalization which identify and handle any missing values. Then remove features that aren't needed for your analysis or model. Finally determine which columns are numerical, as these will be normalized. This technique scales the values of a feature to have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean of the feature from each value in the dataset, and then dividing by the standard deviation. It is mathematically defined as follows:

$$Z = \frac{(x-\mu)}{\sigma} \quad (1)$$

Where, x refers the original value in the dataset, μ indicates the mean of data, and σ denotes the standard deviation.

This preprocessing step is crucial when dealing with datasets like CICIDS2017, where features can vary widely in scale. Hence by normalizing it may help to classification algorithm to provided converge faster and perform better.

3.2 Feature Selections Improved Whale Optimization Algorithm (IWOA)

The next critical step in the methodology is Feature Selection using Improved Whale Optimization Algorithm (IWOA). The Whale Optimization Algorithm (WOA) is a metaheuristic inspired by the bubble-net hunting strategy of humpback whales. The improved version (IWOA) incorporates adaptive parameters to efficiently explore the search space in dynamic VANET conditions.

Input pre-processed CICIDS2017 database might have more features and consume more time to get classification results. This work uses feature selections based on Improved Whale Optimization (IWOA). Recently, the revolutionary stochastic population-based optimization approach called WOA which draws inspiration from nature, finds optimal solutions utilizing groups of search agents for optimizations. By using bubble-net hunting technique, WOA mimics actions of humpback whales as they pursue their preys. The three general processes of WOAs are to surround preys, assault using bubble nets, and hunt for best preys.

Whales enclose their prey including fishes while updating their positions to find optimal solutions. Mathematically Eqs. (12) and (13) depict WOA.

$$X(t+1) = X^*(t) - A.|C.X^*(t) - X(t)| if p < 0.5 \quad (2)$$

$$X(t+1)|C.X^*(t) - X(t)|.e^{bl}\cos(2\pi t) + X^*(t) if p \geq 0.5 \quad (3)$$

Where X implies vectors of whales' positions; t represents times or iteration indices; X* implies best solutions found so far; A=2a. (r-a);C=2.r; a are coefficient vectors that linearly decrease from 2 to 0 in iterations; r stands for random vectors between 0 and 1; b implies constant values that define shapes of logarithmic spirals based on paths and is 1 for this work; l stands for random numbers between - 1 and 1; p implies random numbers between 0 and 1 and used to switch between (10) and (11) while updating whales' positions; Eqs. (10) and (11) have 50% probabilities implying during optimizations whales select paths randomly with equal chances. During bubble-net phases, A's random values are in the range [- 1, 1], however in searching phases, these values may be larger than or less than 1. Search processes are illustrated as Eq. (4).

$$X(t+1) = X_{rand} - A.|C.X_{rand} - X(t)| \quad (4)$$

Random searches with values of |A| > 1 emphasize searches and require WOA algorithm to do global searches. WOA searches begin with generations of random solutions. The responses are then updated in iterations and searches continue until present max. iterations are reached.

IWOA

The good trade-off between exploration and exploitation, two critical components of an optimization algorithm, allows for a precise solution to be obtained by escaping the local optima. In WOA, a search agent's step size decreases linearly as iteration counts increases. This step size is determined by a parameter known as A. Nonetheless, it has been demonstrated that insufficient divergence restricts WOA's capacity to capture a local optimum in later rounds.

This paper employs an updated whale optimization approach to get around such problems. This changes the value A by introducing the levy flying function. Levy flight is a mathematical concept used to describe a type of random walk in which step lengths have a probability distribution that is heavy-tailed (e.g., follows a Levy distribution). In optimization algorithms, such as the Improved Chicken Swarm Algorithm (ICSO), Levy flight is applied as a search strategy to improve efficiency and avoid premature convergence to local optima. It improves WOA's capacity for simultaneous exploration and exploitation.

The Levy probability distribution function, a power-law function, is utilized in Levy flight to determine jump sizes where Levy distributions can be mathematically formulated as:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{3/2}} & if\ 0 < \mu < \infty \\ 0 & if\ s \leq 0 \end{cases} \quad (5)$$

Where μ, γ, and s are positions and scales which control scale distributions and samples in distributions respectively.

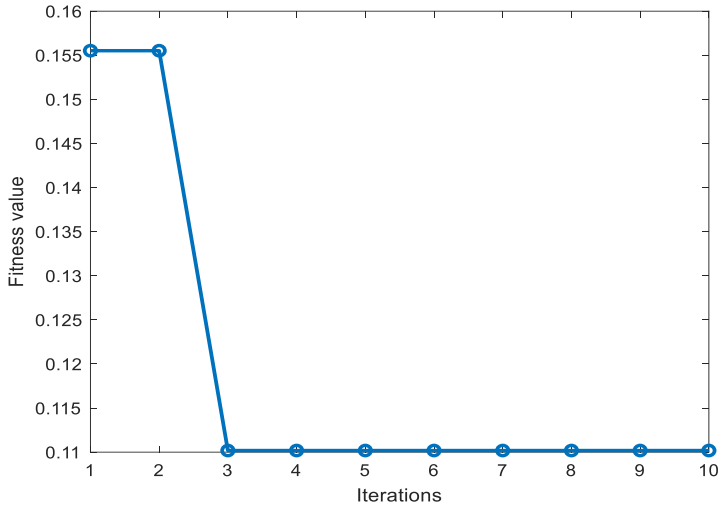

Figure 3: Convergence curve of IWO

The figure 3 depicts a line graph showing the relationship between iterations and fitness values in an optimization algorithm. The graph reveals a significant drop in the fitness value from approximately 0.155 at iteration 1 to about 0.11 at iteration 2. After the second iteration, the fitness value stabilizes and remains constant at approximately 0.11 through iterations 3 to 10, indicating that the optimization process has likely converged. Using Mean Squared Error (MSE) as the fitness function in the Improved Whale Optimization Algorithm not only aids in effectively assessing the quality of solutions but also demonstrates the algorithm's capability to minimize errors in predictive modelling.

In the IWOA, Levy flights play critical roles in enhancing explorations and exploitations of the algorithm, particularly to avoid issues of getting trapped in local optimums during later iterations.

Role of Levy Flight in IWOA:

1. Exploration and Exploitation Balance:
    - Explorations refers to algorithm's abilities to search broadly across solution spaces, while exploitations refer to focusing on refining current best solutions.
    - In standard Whale Optimization Algorithm (WOA), the parameter A controls the step size of the whales' movement. However, as iterations increase, A decreases linearly, reducing the step size and thus restricting the ability to explore new regions of the search space.
    - To enhance both exploration and exploitation simultaneously, Levy flight is used to introduce random long-distance jumps, which allows for more diverse movements even in later stages of the search. This nonlinear random walk based on the Levy distribution helps escape local optima by allowing whales to make larger, random steps.

Algorithm for IWO

START

1. import data
2. Set initial locations of whales **X**
3. Compute whales fitness
4. Set initial values of **a** and **r**, calculate **A** and **C**
5. Set initial **X*** as best hunters' whale locations

6. initialize $t = 1$
7. **while** $t \leq$ max iterations **do**
8. **for** each hunting whale **do**
9. **if** $p < 0.5$
10. **if** $|\mathbf{A}| < 1$
11. update existing locations of hunting whales with (3)
12. **else if** $|\mathbf{A}| \geq 1$
13. another search agent randomly
14. update existing locations of hunting whales with (4)
15. **end if**
16. **else if** $p \geq 0.5$
17. update existing locations of hunting whales with (5)
18. **end if**
19. **end for**
20. update **X*** on better solutions
21. $t = t + 1$
22. **end while**
23. output **X*** Best Features
24. **END**

The IWOA is a key component of the proposed methodology for feature selection optimization. By reducing the dataset's dimensionality, IWOA identifies the most relevant features, enhancing the classification model's overall performance while simultaneously reducing time complexity and energy consumption.

3.3 Signature-Based Classification using Improved Deep Neural Network (IDNN)

Once get the feature subset from the feature selection phase need to classify those features to detect Signature based attacks in Vanet [21]. The Improved Deep Neural Network (IDNN) enhances the traditional deep learning architecture by incorporating optimization and regularization techniques, enabling rapid and accurate detection of known intrusions. In this work, we implement a signature-based classification approach using an Improved Deep Neural Network (IDNN) to effectively distinguish between benign and malicious network traffic. The signature-based methodology leverages distinct characteristics extracted from network flows to capture the unique "signatures" of various attack types. The IDNN is designed to robustly learn these signatures by incorporating improvements such as multi-layer architecture, dropout regularization, and batch normalization.

DNN is a deep learning which consists of three layers namely input, hidden and output layer. A DNN consists of multiple hidden layers between input and output layers. In a DNN (Joshi and Kumar 2020), the input layer assigns weights to the input parameters and transfers those to the next layer. Each subsequent layer also assigns weights to their input and generates their output. At the output layer, the final output value is obtained and error function E_j is calculated to determine how correctly learned those data for identifying the crops. This training cycle is repeated until the relationship between countermeasures and the extracted data is learned. Using training data, the class label of crops are generated from that the crop recommendation can be processed. The probabilities are denoted as $\mathfrak{P}(x) = x$ are given to the input layer of neurons. DNN consists of multiple hidden layers which can handle huge volume of data as shown in figure 3.4. Each hidden layer of DNN is defined as sigmoid transfer function which is given as follows:

$$\mathfrak{P}(x) = \frac{1}{1+e^{-x}} \qquad (6)$$

For (each node j in output layer) the error function Er is calculated as

$$Er_j = \frac{1}{2}\left(to_j^n - op_j^n\right)^2 \qquad (7)$$

Where to_j^n is the desired target output for the n-th observation and the op_j^n is the actual output for the n-th observation. Deep Learning models have so much flexibility and capacity that overfitting can be a serious problem, if the training dataset is not big enough. The standard way to avoid overfitting is called L2 regularization. It consists of appropriately modifying the error function:

$$op_{regularized} = Er_j + \sum_{i=1}^{n} \frac{\mathfrak{r}}{2f} \|w_i\|^2 \quad (8)$$

Where $\mathfrak{r}$ is the regularization parameter and f is the total number of features. It is the hyperparameter whose value is optimized for better results using COA-GS. L2 regularization is also known as weight decay as it forces the weights to decay towards zero (but not exactly zero). Each input feature has its own weight values as $wv_1, wv_2, \ldots, wv_n$ which is calculated by using fuzzy membership function

$$wv = \mathfrak{Z}_{A_i}(x) \; for \; i = 1,2 \quad (9)$$

Where, input nodes are represented by, $\mathfrak{Z}_{A_i}$ is the linguistic labels, (x) is signified as the membership functions, highest and lowest values equivalent to 1 and 0, correspondingly and the weighted sum of the inputs is done by the adder function $\mathcal{A}$ as follows,

$$\mathcal{A} = \sum_{i=1}^{n} wv_i x_i \quad (10)$$

The output layer of IDNN is described by the following equation.

$$y = \mathfrak{P}(\sum_{i=1}^{n} w_i x_i + b_i) \quad (11)$$

In the equation (11), y is the output neuron value; $\mathfrak{P}(x)$ is the transfer function, wv_i refers the weight values, x_i denotes input data values and b_i refers to the bias value. Based on the output neuron values, the relationship between countermeasures and the considered parameters is learned, identifying the types of attacks. Initially, the DNN is loaded with pre-processed attribute vectors, which are then attached to the output matrix as the first layer. Then, in every loop of computation, the result will use the matrix multiplication to multiply weight matrix wv and then add the bias b. In the multiple hidden layers, sigmoid function is applied and the result is given to the next layer. Based on the error values, the weight and bias values of IDNN network is updated. Finally, the output layer returns probability.

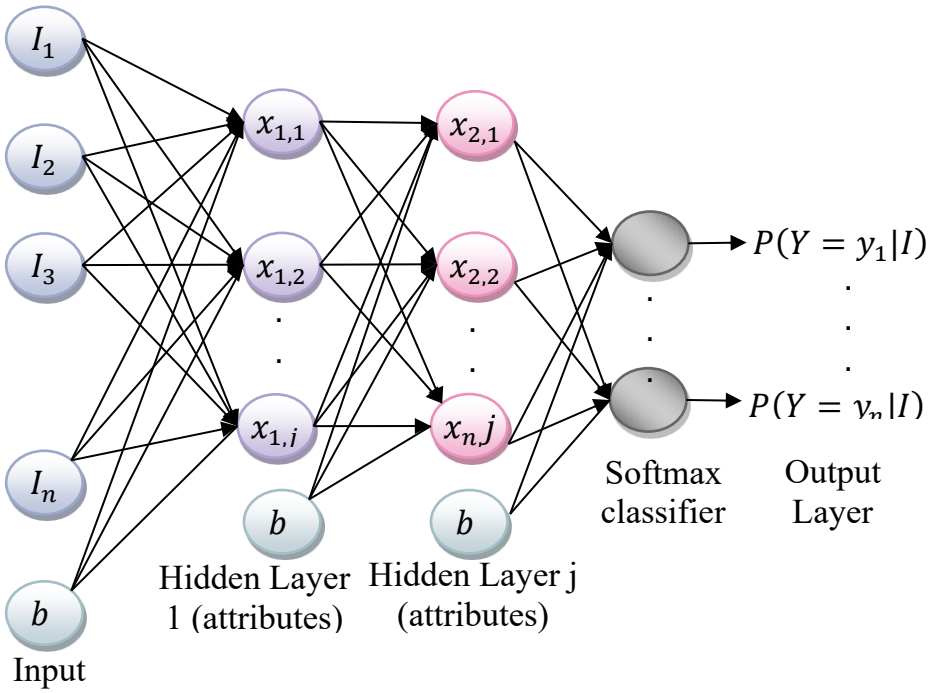

Figure 3.4. The network structure of DNN model

Based on that probability value the IDNN approximate the attack classification. In this work, the objective function based on accuracy of this COA is optimally select the hyperparameters including the number of hidden layers, the number of neurons in each layer, level of learning, momentum, initialization of neuron weights in IDNN for training to improve the precision of IDS model. Once the IDNN architecture has been chosen to be trained, a variable number of hyperparameters must be tuned in order to achieve the best possible model with that architecture. These hyperparameters define how the network functions and are key to their validity and accuracy. Their values depend ultimately on the problem that is being addressed, type of available data and expected output. Moreover, these parameters are interdependent, meaning that modifying a single hyperparameter might necessitate changes to others. Some of the most common or important hyperparameters to adjust within IDNN are (Domhan et al 2015):

Number of hidden layers: Increasing the number of hidden layers is usually believed to increase model accuracy.

Parameter initialization: Weights must be initialized during the first pass. These values can be set to zero or obtained with a random function. This can lead to vanishing or exploding gradients, which reinforces the need to find a way of easily initializing them without compromising its training.

Learning rate: It represents the amount weights are increased during training. It usually has a positive value from 0 to 1. This value is one of the most relevant ones, as it can lead to heavy and long training process after which the process could get stuck (low values) or too fast and cause the training process to become unstable and achieve sub-optimal sets of weights (large values).

An epoch is a forward pass and a backward pass of all the data samples in the training dataset. This value is set when the number of variables is too large to be run in one single epoch, or when the complexity of the model makes it necessary. The number of iterations is the number of backward and forward passes using the information contained in each batch.

Dropout regularization: Defines the number of neurons not trained in each epoch in order to avoid overfitting of the model during training.

Optimizer algorithm and momentum: The Adam optimizer is in charge of running gradient descend in order to actualize the weights of each variable. The way to do so varies from one optimizer to another which can impact the model performance.

COA based Hyperparameter optimization of IDNN: The coyote optimization algorithm (COA) is a population-based algorithm (Pierezan and Coelho 2018). Inspired by the social and evolutionary behavior of canines, COA is divided into two components: group intelligence and evolutionary heuristics The COA has different algorithmic structure settings, and it does not pay attention to the social hierarchy and domination rules of these animals, even if alpha is used as a group leader. In addition, the COA focuses on the social structure and exchange of experiences of wolves, rather than hunting prey as in GWO. The social condition $\mathfrak{S}$ (set of decision variables dec such as hidden layers, Parameter initialization, learning rate, epoch, number of iterations, Dropout regularization, Optimizer algorithm and momentum) of cth wolf at tth instant popth is written as

$$\mathfrak{S}_c^{pop,t} = (x_1, x_2, \dots, x_{dec}) \tag{12}$$

It means the adaptation of the wolf to the environment (the accuracy of the objective function $fitness_c^{pop,t} = f\left(\mathfrak{S}_c^{pop,t}\right)$). The first step of the COA is to initialize the global coyote population. It is achieved by assigning random values in the search space of the cth coyote of the popth package in the jth dimension, as follows:

$$\mathfrak{S}_{c,j}^{pop,t} = \ell\mathfrak{b}_j + rand_j \cdot \left(u\mathfrak{b}_j - \ell\mathfrak{b}_j\right) \tag{13}$$

where $\ell\mathfrak{b}_j$ and $u\mathfrak{b}_j$ respectively denote the lower and upper bounds of the jth decision variable, $\mathfrak{D}$ is the dimension of the search space, and $rand_j$is the real random number generated in the range [0,1] using uniform probability. Considering the two main biological events, birth and death, COA calculates the age (in years) of the coyote and expresses it as the age $age_c^{pop,t} \in N$. In order to express the cultural interaction within the ethnic group, COA assumes that the coyotes are affected by alpha (α_1) and ethnic influence (α_2). The first refers to a cultural difference $\mathfrak{cd}$, from a random coyote pack ($\mathfrak{cd}_1$) to an alpha coyote, while the second is a cultural difference, from a random coyote ($\mathfrak{cd}_1$) to a culturally inclined pack. The random coyote is selected by the uniform distribution of probability, α_1 and α_2 are written as

$$\alpha_1 = alpha^{pop,t} - \mathfrak{S}_{\mathfrak{cd}_1}^{pop,t}$$

$$\alpha_{2=}\mathfrak{cd}^{pop,t} - \mathfrak{S}_{\mathfrak{cd}_2}^{pop,t} \tag{14}$$

Therefore, the new social $\mathfrak{NS}$ conditions of the coyotes are updated using the following equations using alpha and group influence: selection by uniform probability distribution, α_1 and α_2 are written as

$$\mathfrak{NS}_c^{pop,t} = \mathfrak{S}_c^{pop,t} + rand_1 \cdot \alpha_1 + rand_2 \cdot \alpha_2 \tag{15}$$

where $rand_1$ and $rand_2$ were defined as random numbers in the range [0,1] generated with uniform probability. Then, assess the new social situation: $newfit_c^{pop,t} = f\left(\mathfrak{NS}_c^{pop,t} - \mathfrak{S}_c^{pop,t}\right)$

The cognitive ability of the wolf determines whether the new social conditions are more suitable to maintain it than the old social conditions, which means

$$\mathfrak{S}_c^{pop,t+1} = \begin{cases} \mathfrak{NS}_c^{pop,t} & newfit_c^{pop,t} < fit_c^{pop,t} \\ \mathfrak{S}_c^{pop,t} & otherwise \end{cases} \tag{16}$$

Finally, the social conditions of the coyote that best adapt to the environment are chosen as the overall solution to the problem. The algorithm runs for a fixed number of iterations (or until convergence) and returns the best hyperparameter set found.

Algorithm: Signature Based Classification _IDNN_COA

Input:

- Feature subset F
- Dataset D (training, validation, test)
- Hyperparameter search space S with bounds {lb_j, ub_j} for j = 1,...,dec
- COA parameters: Population size N, Max iterations T, Abandonment probability p_a

Output:

- Optimized hyperparameters x_best
- Trained IDNN model for signature-based attack classification

Phase 1: COA-Based Hyperparameter Optimization

1. For c = 1 to N do:
   For j = 1 to dec do:
   $S_c^{(0)}[j] = lb_j + rand_j*(ub_j - lb_j)$ (Equation 3.27)
2. For each candidate $S_c^{(0)}$, compute fitness $f(S_c^{(0)})$ = ValidationError(IDNN trained with $S_c^{(0)}$)
3. Set x_best = candidate with minimum $f(S_c^{(0)})$
4. For t = 0 to T-1 do:
   For each candidate $S_c^{(t)}$:
   a. Compute $S_{social}^{(t)}$ = Mean({$S_c^{(t)}$ for all c})
   b. Select two random candidates $S_{cd1}^{(t)}$ and $S_{cd2}^{(t)}$
   c. Compute $\alpha_1 = \alpha^{(t)} - S_{cd1}^{(t)}$ and $\alpha_2 = S_{cd2}^{(t)} - \alpha^{(t)}$ (Equation 3.28)
   d. Update candidate:
   $NS_c^{(t)} = S_c^{(t)} + rand_1*\alpha_1 + rand_2*\alpha_2$ (Equation 3.29)
   e. Enforce bounds on $NS_c^{(t)}$
   f. If $f(NS_c^{(t)}) < f(S_c^{(t)})$ then:
   $S_c^{(t+1)} = NS_c^{(t)}$
   Else:
   $S_c^{(t+1)} = S_c^{(t)}$ (Equation 3.30)
   End For
   Update x_best = candidate with minimum $f(S_c^{(t+1)})$
   With probability p_a, reinitialize the worst-performing candidates.
5. End For
6. Return optimized hyperparameters x_best.

Phase 2: Signature-Based Classification Using IDNN

7. Configure the IDNN using x_best:
   - Input Layer: Load feature vectors F.
   - For each hidden layer l:
   Compute $z^{(l)} = W^{(l)}*a^{(l-1)} + b^{(l)}$

Compute a^(l) = 1/(1+exp(-z^(l))) (Equation 3.20)

Apply dropout and batch normalization.

- For each input feature i, compute fuzzy weight: wv_i = Z_Ai(x) (Equation 3.23)

Compute adder A = Σ_i (wv_i * x_i) (Equation 3.24)

- Output Layer: y = P(Σ_i (w_i*x_i + b_i)) (Equation 3.25)

8. Train IDNN:

For each epoch:

- Perform forward propagation.
- Compute error: Er_j = 1/2*(to_j - op_j)^2 (Equation 3.21)
- Apply L2 regularization: Er_regularized = Er_j + Σ_i (r/(2f))*||w_i||^2 (Equation 3.22)
- Update weights using backpropagation (e.g., Adam optimizer).

9. For each test instance:

- Compute output y via forward propagation.
- Classify as attack if y ≥ threshold; else classify as benign.

10. End.

Output: Trained IDNN model and classification results.

3.4 Modified Possibilistic Fuzzy C Means (MPFCM) clustering for Anomaly Detection using Anomaly Detection

In this phase anomaly based intrusion detection is performed on the features which are classified as a normal traffic in the signature based intrusion detection phase. In this work anomaly based intrusion detection is performed using MPFCM extends the standard Possibilistic Fuzzy C-Means (PFCM) algorithm by introducing modifications aimed at improving robustness against noise and outliers. In anomaly detection, the goal is to identify data points that do not belong well to any cluster. MPFCM achieves this by combining fuzzy membership values with possibilistic typicality values and, optionally, adding regularization terms.

In standard PFCM, each data point x_j is assigned:

- **Fuzzy Membership** u_{ij} for each cluster i. These values satisfy:

$\sum_{i=1}^{c} u_{ij} = 1$ for each j

- **Typicality** t_{ij} which measures the absolute degree of belonging of x_j to cluster i. Anomalies typically exhibit low typicality across all clusters.

The standard PFCM objective function is:

$$J_{PFCM} = \sum_{i=1}^{c}\sum_{j=1}^{n}\left[u_{ij}^{m}\|x_i - v_i\|^2 + \eta_i\left(1 - t_{ij}\right)^{m}\right] \quad (17)$$

where:

- c is the number of clusters,
- n is the number of data points,
- v_i is the center of cluster i,
- $m > 1$ is the fuzzifier parameter,
- η_i is a scaling parameter for cluster i.

MPFCM introduces the following modifications:

- Robust Distance Measure: The standard Euclidean distance $\|x_i - v_i\|$ can be replaced or weighted to lessen the effect of outliers.
- Adaptive Scaling: The parameter η_i is updated adaptively to reflect the spread (variance) within each cluster.

- Regularization Term: An additional term may be included to penalize clusters that become too close together or to prevent trivial clustering solutions.

The modified objective function takes the form:

$$J_{MPFCM} = \sum_{i=1}^{c}\sum_{j=1}^{n}\left[u_{ij}^{m}d^2(x_i, v_i) + \eta_i\left(1 - t_{ij}\right)^m\right] + \lambda R(\{v_i\}) \quad (18)$$

where:

- $d(x_i, v_i)$ is a (possibly robust) distance measure,
- $R(\{v_i\}$ is a regularization term (for example, one that enforces separation between cluster centers),
- λ is the regularization parameter.

Membership Update:

- The membership u_{ij} of data point x_j in cluster i is updated as:

$$u_{ij} = \left(\sum_{k=1}^{c}\left(\frac{d^2(x_j,v_i)}{d^2(x_j,v_k)}\right)^{\frac{1}{m-1}}\right)^{-1} \quad (19)$$

In MPFCM, the distance $d(x_j, v_i)$ might incorporate a robust weighting function.

Typicality Update

The typicality t_{ij} indicates the absolute degree of belonging and is updated by:

$$t_{ij} = \frac{1}{1+\left(\frac{d^2(x_j,v_i)}{\eta_i}\right)^{\frac{1}{m-1}}} \quad (20)$$

The scaling parameter η_i is updated adaptively as:

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} d^2(x_j,v_i)}{\sum_{j=1}^{n} u_{ij}^{m}} \quad (21)$$

This adjustment helps each cluster adapt to the data's spread.

Cluster Center Update

Cluster centers v_i are computed using the weighted average:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \quad (22)$$

A common regularization term used to maintain cluster separation is:

$$R(\{v_i\}) = \sum_{i=1}^{c}\sum_{\substack{k=1 \\ k\neq i}}^{c}\frac{1}{\|v_i - v_k\|^2} \quad (23)$$

The term is multiplied by λ to control its influence on the overall objective.

After clustering, anomalies are detected by examining the typicality values. For each data point x_j, compute:

$$t_j^{max} = \underset{i\,=\,1,..,c}{max}\{t_{ij}\} \quad (24)$$

If t_j^{max} is below a predetermined threshold τ, then x_j is flagged as an anomaly. This is because an anomalous point will not fit well (i.e., will have low typicality) into any cluster.

Algorithm HybridIntrusionDetection_VANETs

Input:

D // Raw VANET dataset (e.g., CICIDS2017)

Params: // Parameters for normalization, IWOA, COA, IDNN, MPFCM, etc.

Output:

Final_Results // Intrusion detection results (signature-based and anomaly alerts)

BEGIN

```
// Phase 1: Data Preprocessing using Z-Score Normalization
D_normalized = ZScoreNormalization(D)
[D_train, D_val, D_test] = SplitDataset(D_normalized)
  // Phase 2: Feature Selection using Improved Whale Optimization Algorithm (IWOA)
FeatureSubset = IWOA_FeatureSelection(D_train, D_val)
// IWOA_FeatureSelection:
//   Initialize population of candidate feature subsets
//   For t = 1 to max_iter_IWOA:
//      For each candidate subset S in population:
//         Evaluate fitness f(S) (e.g., via classification error on D_val)
//      Update candidate positions using IWOA update rules (including Levy flights)
//      Update global best subset
//   End For
//   Return global best subset as FeatureSubset
  SelectedFeatures = FeatureSubset
  // Phase 3: Signature-Based Classification using IDNN with COA-Based Hyperparameter Optimization
BestHyperparams = COA_HyperparameterOptimization(D_train[SelectedFeatures], D_val[SelectedFeatures])
// COA_HyperparameterOptimization:
//   Initialize population {S_c^(0)} of candidate hyperparameter vectors (for IDNN)
//   For t = 1 to max_iter_COA:
//      For each candidate S_c:
//         Configure IDNN with S_c and train on D_train[SelectedFeatures]
//         Evaluate fitness f(S_c) (e.g., validation error on D_val[SelectedFeatures])
//         Update candidate:
//            S_c_new = S_c + rand_1*(S_social - S_c) + rand_2*(S_best - S_c)
//            Enforce bounds; if f(S_c_new) < f(S_c), then S_c = S_c_new
//      End For
//      Update global best S_best among all candidates
//      With probability p_a, reinitialize worst candidates
//   End For
//   Return S_best as BestHyperparams
  // Train final IDNN model using BestHyperparams
IDNN_model = TrainIDNN(D_train[SelectedFeatures], BestHyperparams)
SignatureResults = IDNN_model.Predict(D_test[SelectedFeatures])
// For each test instance, output probability and label it as "attack" if above threshold;
// otherwise, label it "normal/inconclusive".
  // Phase 4: Anomaly Detection using Modified Possibilistic Fuzzy C-Means (MPFCM)
NormalData = ExtractNormalInstances(D_test[SelectedFeatures], SignatureResults)
// NormalData contains samples labeled as normal/inconclusive by the IDNN.
  MPFCM_Result = MPFCM_Clustering(NormalData)
// MPFCM_Clustering:
```

```
// Initialize cluster centers {v_i} randomly from NormalData for i = 1 to c
// For each data point x_j in NormalData and cluster i:
//     Initialize membership u_ij (e.g., uniformly) and set typicality t_ij = 1
// Set iter = 0 and J_old = ∞
// While (iter < max_iter_MPFCM) and (|J_new - J_old| > ε) do:
//     For each cluster i:
//         Update cluster center:
//             v_i = (Σ_j u_ij^m * x_j) / (Σ_j u_ij^m)
//         Update scaling parameter:
//             η_i = (Σ_j u_ij^m * d^2(x_j, v_i)) / (Σ_j u_ij^m)
//     For each x_j and cluster i:
//         Update membership:
//             u_ij = ( Σ_{k=1}^{c} (d^2(x_j, v_i)/d^2(x_j, v_k))^(1/(m-1)) )^(-1)
//         Update typicality:
//             t_ij = 1 / ( 1 + (d^2(x_j, v_i)/η_i)^(1/(m-1)) )
//     Compute objective function:
//         J_new = Σ_i Σ_j [ u_ij^m * d^2(x_j, v_i) + η_i*(1 - t_ij)^m ] + λ*R({v_i})
//     If |J_new - J_old| < ε, break; else, set J_old = J_new and iter = iter + 1
// End While
// For each data point x_j:
//     t_j_max = max_{i} {t_ij}
//     If t_j_max < anomaly_threshold τ, flag x_j as anomaly
// Return MPFCM_Result containing cluster centers, membership & typicality matrices, and anomaly flags
  AnomalyFlags = MPFCM_Result.AnomalyFlags
  // Combine classification and anomaly detection results
 Final_Results = CombineResults(SignatureResults, AnomalyFlags)
  // Step 5: Evaluate Performance (accuracy, precision, recall, f-measure)
 Metrics = EvaluateMetrics(Final_Results, D_test.Labels)
  Output Final_Results, Metrics
END
```

4. EXPERIMENTAL RESULTS

This section discusses the experimental results of the proposed model in detail. Proposed model is implemented in mat lab. Proposed IFCM is compared with the existing KNN and W-KMC models interms of preicison, accuracy, recall and f-measure. In this work, use the most recent dataset CICIDS2017 that contains the most up to date network attack scenarios, including the common type of attacks in VANET (like DoS Slowloris attacks and DDoS attacks). CICIDS2017 dataset covers the most cutting-edge frequent attack scenarios based on simulation of seven attack families, namely: brute force attack, heart-bleed attack, botnet, DoS attack, DDoS attack, web attack, and infiltration attack. A total number of 80 features were extracted based on the information present in the pcap file. The total number of records used in this experiment is 273 097. The dataset is divided into two parts using train-test_split, 80% for training and 20% for testing the model.

Table 1.Shows the simulation parameters

Parameter	Value
Simulation time	300 sec
Simulation Grid	1000 x 1000
Scenario	Two-way highway
Number of vehicles	100
Vehicle speed	15–45m/s
Data transfer rate	6, 12, 18, 27Mbps
Mac Layer	IEEE 802.11p
Packet size	100 bytes

Below are the standard performance metric formulas (Accuracy, Precision, Recall, and F-measure) along with their typical interpretations in the context of intrusion detection systems (IDS). In this context:

- True Positive (TP): The system correctly classifies an intrusion as an intrusion.
- True Negative (TN): The system correctly classifies normal traffic (non-intrusion) as normal.
- False Positive (FP): The system incorrectly classifies normal traffic (non-intrusion) as intrusion.
- False Negative (FN): The system fails to identify an intrusion (i.e., it is an intrusion but classified as normal).

Precision: Precision is defined as the instances predicted as attacks, how many are actually attacks. Precision is crucial in assessing false alarms. A high precision means fewer false alarms, so if the system classifies a traffic as attack, it is likely correct.

$$Precision = \frac{TP}{TP+FP} \quad (25)$$

F-measure: F-measure is the harmonic mean of precision and recall, balancing both false positives and false negatives.

$$\mathrm{F-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (26)$$

Recall: It is defined as Of the actual attacks, how many are caught by the system?". Recall is crucial to ensure no attack goes undetected. A high recall means fewer missed attacks (false negatives).

$$Recall = \frac{TP}{TP+FN} \quad (27)$$

Accuracy: Accuracy measures the proportion of total instances (both normal and attack) that are correctly classified by the IDS.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (28)$$

It gives a quick overview of the system's overall correctness.

Table 1 compares the performance of KNN, W-KMC, IFCM and IDNN_COA with MPFCM across multiple performance metrics commonly used in classification tasks: Accuracy, Precision, Recall, F-Measure, and Error.

Table 1: The numerical outcome of suggested and current approaches depends on various performance metrics

Metrics	KNN	W-KMC	IFCM	IDNN_COA with MPFCM
Accuracy	82	89	96	97.5
Precision	74	82	90	92
Recall	73.5	84.5	88	91
F-Measure	76	81	89	92

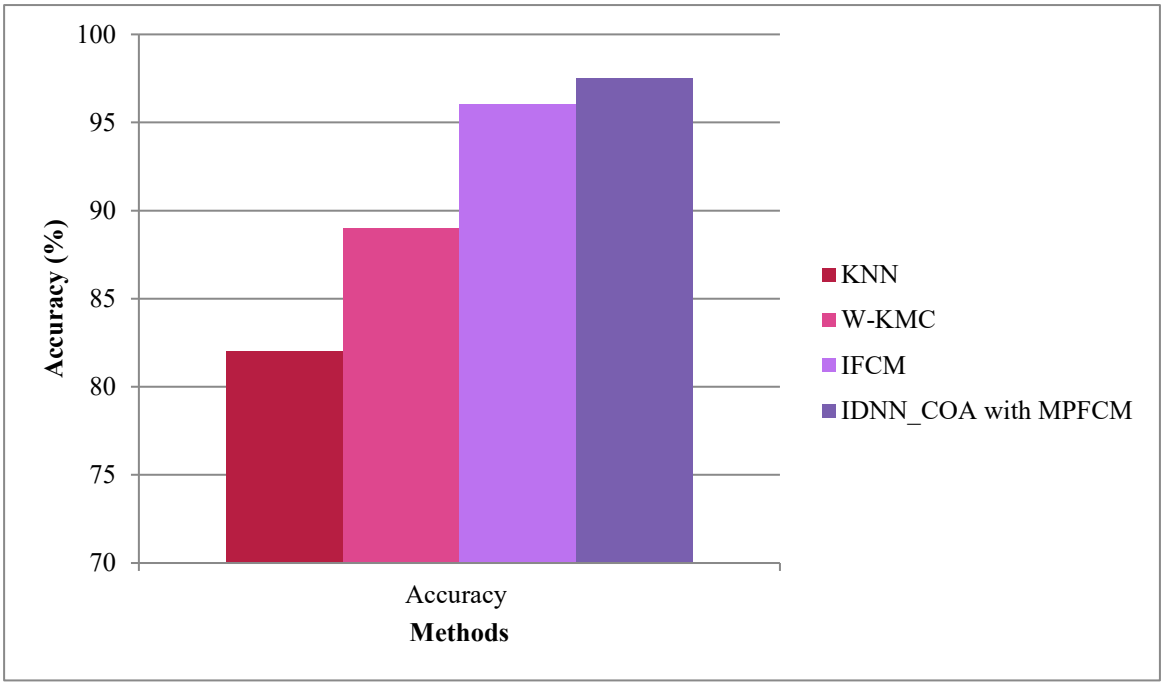

Figure 3: Accuracy Comparison Results

Figure 3 shows the accuracy performance metric comparison between existing KNN, W-KMC and IFCM methods and proposed IDNN_COA with MPFCM for intrusion detection. In the above figure X-axis represents the methods and the y-axis represents the accuracy results. This work using improved support vector machine and it increases the accuracy results. From the results it is concluded that the proposed IDNN_COA with MPFCM model produces the higher accuracy results of 97.5% while the existing KNN, W-KMC and IFCM models produces only 82%, 89% and 96% accordingly.

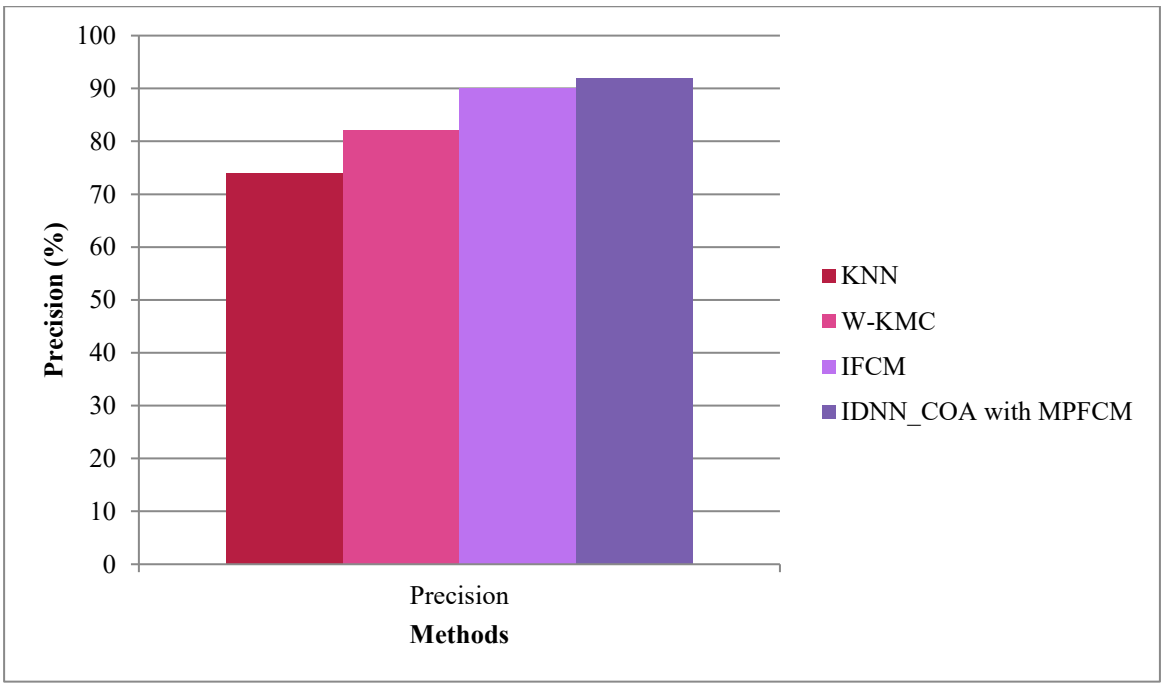

Figure 4: Precision Comparison Results

Figure 4 shows the precision performance metric comparison between existing KNN, W-KMC and IFCM methods and

proposed IDNN_COA with MPFCM for intrusion detection. In the above figure X-axis represents the methods and the y-axis represents the accuracy results. From the results it is concluded that the proposed IDNN_COA with MPFCM model produces the higher precision results of 92% while the existing KNN, W-KMC and IFCM models produces only 82%, 89% and 90% accordingly.

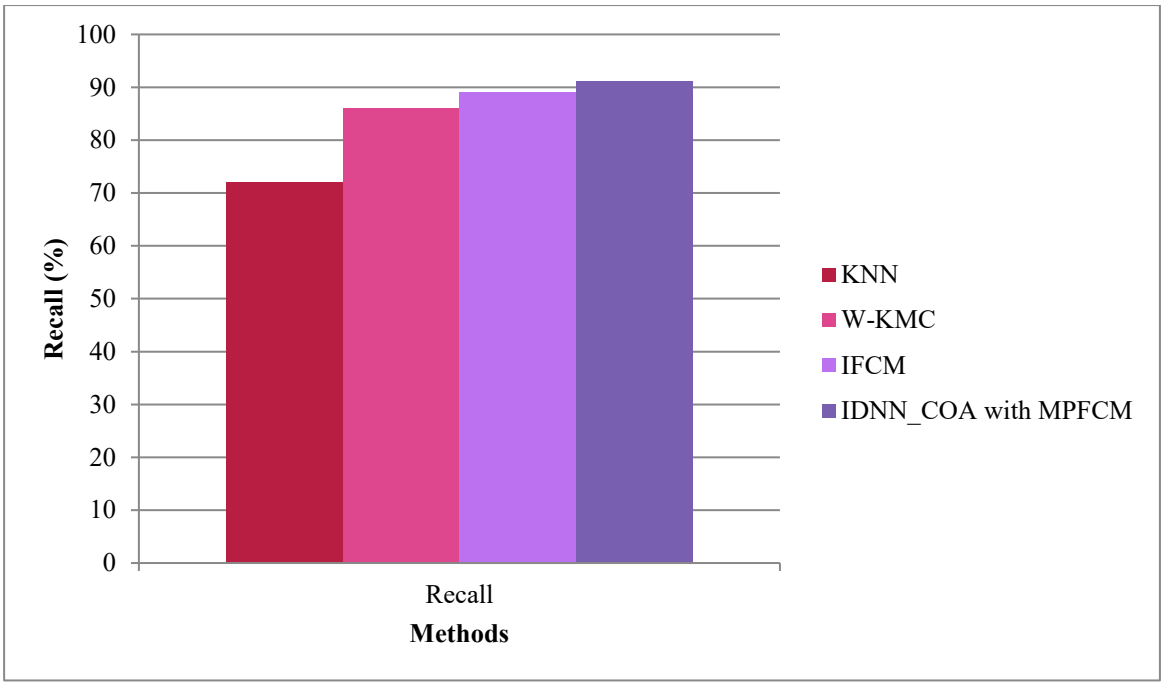

Figure 5: Recall Comparison Results

Figure 5 shows the recall performance metric comparison between existing KNN, W-KMC and IFCM methods and proposed IDNN_COA with MPFCM for intrusion detection. In the above figure X-axis represents the methods and the y-axis represents the recall results. From the results it is concluded that the proposed IDNN_COA with MPFCM model produces the higher recall results of 91% while the existing KNN, W-KMC and IFCM models produces only 73.5%, 84.5% and 88% accordingly.

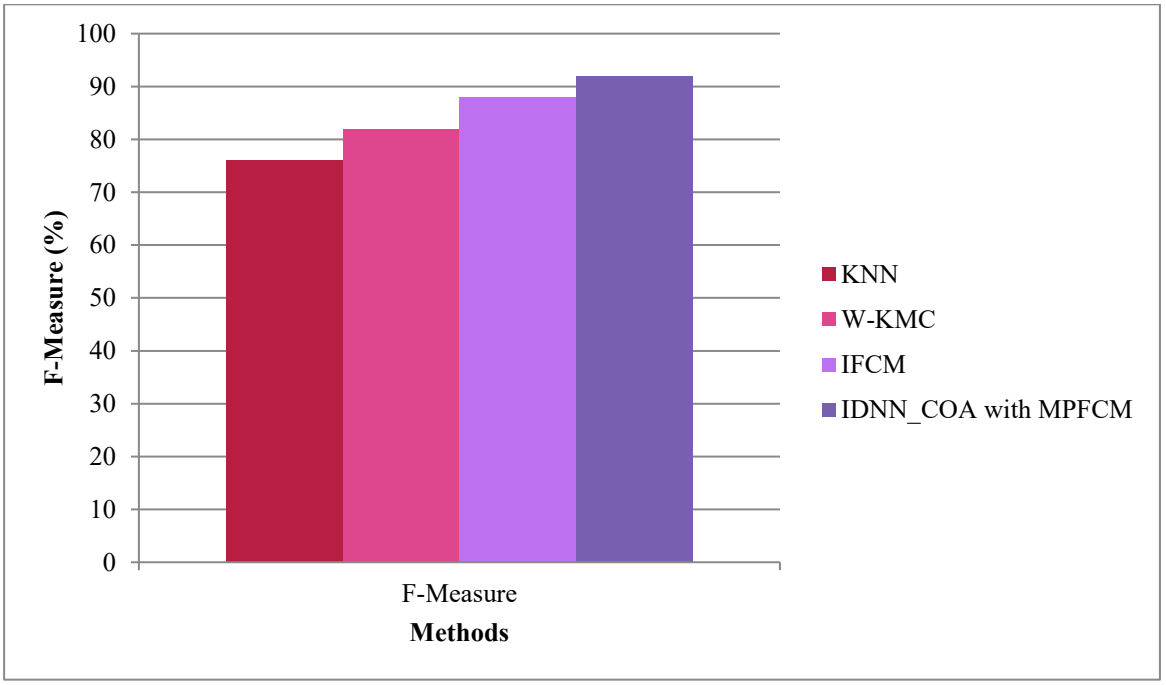

Figure 6: F-Measure Comparison Results

Figure 5 shows the F-Measure performance metric comparison between existing KNN, W-KMC and IFCM methods and proposed IDNN_COA with MPFCM for intrusion detection. In the above figure X-axis represents the methods and the y-axis represents the F-Measure results. From the results s concluded that the proposed IDNN_COA with MPFCM model produces the higher F-Measure results of 92% while the existing KNN, W-KMC and IFCM models produces only 76%, 81% and 89% accordingly.

5. CONCLUSION

Vehicular Ad Hoc Networks (VANETs) operate in extremely dynamic and high-velocity environments, which expose them to a broad spectrum of intrusions and malicious attacks. This paper has presented a hybrid Intrusion Detection System (IDS) framework that leverages a combination of advanced techniques tailored specifically for VANET traffic. By first applying Z-Score Normalization, the framework ensures that all features share a consistent scale, significantly improving model convergence and mitigating the influence of outliers. Feature selection is efficiently achieved through an Improved Whale Optimization Algorithm (IWOA), which mimics whale-hunting strategies to robustly explore the search space and select the most discriminative feature subset under the rapidly changing conditions of VANETs. Subsequently, an Improved Deep Neural Network (IDNN) is employed as a signature-based classifier. The enhanced architecture of the IDNN, incorporating advanced optimization and regularization techniques, facilitates the rapid and accurate identification of known intrusions. To address the challenges posed by novel or zero-day attacks, data that is classified as normal or inconclusive by the IDNN is further analyzed using a Modified Possibilistic Fuzzy C-Means (MPFCM) clustering module. This anomaly detection phase leverages fuzzy logic to isolate threats that deviate from established vehicle communication patterns. Comprehensive evaluations using accuracy, precision, recall, and f-measure demonstrate that the proposed framework achieves high detection rates and low false alarm rates while maintaining robust adaptability to evolving attack behaviors. Overall, the integration of these state-of-the-art techniques makes the proposed hybrid IDS a promising approach for securing next-generation intelligent transportation systems. Future research will explore further refinements in each module and validate the system's performance in real-world VANET environments.

REFERENCES

[1] F. Chiti, R. Fantacci, Y. Gu, and Z. Han, "Content sharing in Internet of Vehicles: two matching-based user-association approaches," Vehicular Communications, vol. 8, pp. 35–44, 2017.

[2] F. A. Ghaleb, M. A. Maarof, A. Zainal, B. A. Saleh Al-Rimy, Alsaeedi, and W. Boulila, "Ensemble-based hybrid context-aware misbehavior detection model for vehicular ad hoc network," Remote Sensing, vol. 11, no. 23, p. 2852, 2019.

[3] F. Zafar, H. A. Khattak, M. Aloqaily, and R. Hussain, "Car- pooling in connected and autonomous vehicles: current solutions and future directions," ACM Computing Surveys, vol. 54, no. 10s, pp. 1–36, 2022.

[4] R. Gopi and A. Rajesh, "Securing video cloud storage by ERBAC mechanisms in 5g enabled vehicular networks," Cluster Computing, vol. 20, no. 4, pp. 3489–3497, 2017.

[5] J. Liang, J. Chen, Y. Zhu, and R. Yu, "A novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) based on differences of traffic flow and position," Applied Soft Computing, vol. 75, pp. 712–727, 2019.

[6] F. A. Ghaleb, M. Aizaini Maarof, A. Zainal, B. A. S. Al-Rimy, F. Saeed, and T. Al-Hadhrami, "Hybrid and multifaceted context-aware misbehavior detection model for vehicular ad hoc network," IEEE Access, vol. 7, pp. 159119–159140, 2019.

[7] D. Huang, S. Misra, M. Verma, and G. Xue, "PACP: an efficient pseudonymous authentication-based conditional privacy protocol for VANETs," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 3, pp. 736–746, 2011.

[8] M. Zhou, L. Han, H. Lu, and C. Fu, "Distributed collaborative intrusion detection system for vehicular Ad Hoc networks based on invariant," Computer Networks, vol. 172, p. 107174, 2020.

[9] T. Zhang and Q. Zhu, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," IEEE Transactions on Signal and Information Processing over Networks, vol. 4, no. 1, pp. 148–161, 2018.

[10] Belenko, V., Krundyshev, V. and Kalinin, M., 2018, September. Synthetic datasets generation for intrusion detection in VANET. In Proceedings of the 11th international conference on security of information and networks (pp. 1-6).

[11] Shu, J., Zhou, L., Zhang, W., Du, X. and Guizani, M., 2020. Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach. IEEE Transactions on Intelligent Transportation Systems, 22(7), pp.4519-4530.

[12] Liang, J., Chen, J., Zhu, Y. and Yu, R., 2019. A novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) based on differences of traffic flow and position. Applied Soft Computing, 75, pp.712-727.

[13] Alsarhan, A., Alauthman, M., Alshdaifat, E.A., Al-Ghuwairi, A.R. and Al-Dubai, A., 2021. Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. Journal of Ambient Intelligence and Humanized Computing, pp.1-10.

[14] Ben Rabah, N. and Idoudi, H., 2022. A machine learning framework for intrusion detection in VANET communications. In Emerging trends in cybersecurity applications (pp. 209-227). Cham: Springer International Publishing.

[15] T. Alladi, B. Gera, A. Agrawal, V. Chamola and F. R. Yu, "DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs," in IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 12013-12023, Nov. 2021, doi: 10.1109/TVT.2021.3113807.

[16] Alsarhan, A., Al-Ghuwairi, AR., Almalkawi, I.T. et al. Machine Learning-Driven Optimization for Intrusion Detection in Smart Vehicular Networks. Wireless Pers Commun 117, 3129–3152 (2021). https://doi.org/10.1007/s11277-020-07797-y

[17] Hassan F, Syed ZS, Memon AA, Alqahtany SS, Ahmed N, Reshan MSA, Asiri Y, Shaikh A. A hybrid approach for intrusion detection in vehicular networks using feature selection and dimensionality reduction with optimized deep learning. PLoS One. 2025 Feb 6;20(2):e0312752.

[18] Durairaj D., Venkatasamy T. K., Mehbodniya A., Umar S., and Alam T., Intrusion detection and mitigation of attacks in microgrid using enhanced deep belief network, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. (2022) 1–23,

[19] Zhao G., Zhang C., and Zheng L., Intrusion detection using deep belief network and probabilistic neural network, 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), July 2017, Guangzhou,

[20] Hao Y., Cheng Y., Zhou C., and Song W., A distributed key management framework with cooperative message authentication in VANETs, IEEE Journal on Selected Areas in Communications. (2011) 29, no. 3, 616–629,

[21] Daeinabi A., Pour Rahbar A. G., and Khademzadeh A., VWCA: an efficient clustering algorithm in vehicular ad hoc networks, Journal of Network and Computer Applications. (2011) 34, no. 1, 207–222

[22] Yao M., Wang X., Gan Q., Lin Y., and Huang C., An improved and privacy-preserving mutual authentication scheme with forward secrecy in VANETs, Security and Communication Networks. (2021) 2021, 12.

[23] Shen J., Liu D., Chen X., Li J., Kumar N., and Vijayakumar P., Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs, IEEE Transactions on Vehicular Technology. (2020) 69, no. 1, 807–817

[24] Gope P. and Sikdar B., An efficient privacy-preserving authentication scheme for energy internet-based vehicle-to-grid communication, IEEE Transactions on Smart Grid. (2019) 10, no. 6, 6607–6618,

[25] Gayathri N. B., Thumbur G., Reddy P. V., and Ur Rahman M. Z., Efficient pairing-free Certificateless authentication scheme with batch verification for vehicular ad-hoc networks, IEEE Access. (2018) 6, 31808–31819

[26] Nayyar A., Flying adhoc network (FANETs): simulation based performance comparison of routing protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP, 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), August 2018

[27] Naqvi I., Chaudhary A., and Rana A., Intrusion detection in VANETs, 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), September 2021,

[28] Mahmood J., Duan Z., Yang Y., Wang Q., Nebhen J., and Bhutta M. N. M., Security in vehicular ad hoc networks: challenges and countermeasures, Security and Communication Networks. (2021)

[29] Irshad A., Usman M., Ashraf Chaudhry S., Naqvi H., and Shafiq M., A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework, IEEE Transactions on Industry Applications. (2020) 56, https://doi.org/10.1109/TIA.2020.2966160.