International Journal of Computer Engineering in Research Trends
ISSN: 2349-7084/ https://doi.org/10.22362/ijcert.v10i9.853
Volume 10, Issue 9, September 2023

Research Paper

Outlier Detection using Artificial Rabbit Optimizer with Hopfield Neural Network

[1*]M. Kalpana Devi, [2]R. Padmaja

[1]*Professor,MCA Department,SITAMS, Chittoor, Andhra Pradesh, India*
[2]*Assistant Professor,MCA Department,SITAMS, Chittoor, Andhra Pradesh, India*

[1]*kalpana40.melpadi@gmail.com*, [2]*rpadmajamca@gmail.com*

**Corresponding Author: kalpana40.melpadi@gmail.com*

Received: 22/08/2023, Revised: 08 /09/2023, Accepted: 17/09/2023 Published: 28/09/2023

Abstract: *- Nowadays, anomaly detection becomes a difficult task in realtime computer networks as there is a continuous increase of high-volume, high-speed data streams, and high dimensional, where ground truth data was not available. In simple, Outlier detection can be referred to as process of detecting data points in the dataset that are different from other data. This is significant in many domains, which include statistics, data mining, and machine learning, as outliers can have major effect on the outcomes of data analysis. At present, several OD solutions are presented that could compute anomaly scores while managing data stream. Therefore, this study presents an artificial rabbit optimizer with Hopfield neural network based outlier detection with data classification (AROHNN-ODDC) technique. The presented AROHNN-ODDC technique focuses on the removal of outliers and classifies high dimensional data. In the presented AROHNN-ODDC technique, initial stage of data pre-processing is performed. Next, the OD process is performed by the Local Outlier Factor (LOF) model. Followed by, the ARO approach is applied for the effectual selection of subset of features. Finally, the HNN classifier is used for data classification purposes. For assessing the enhanced data classification results of the proposed method, a wide range of simulations was carried out against benchmark datasets. The experimental outcomes stated the promising performance of the AROHNN-ODDC algorithm over other existing techniques.*

Keywords*- Outlier detection; Data classification; Artificial rabbit optimizer; Hop field neural network; Feature selection*

---

1. Introduction

The two significant tasks in machine learning (ML) and data analysis are data classification and outlier detection (OD). OD is to remove or treat and identify data points that deviate from the rest of the data [1], whereas data classification is to classify data points into different categories or classes. Rarely, OD can be a pre-processing step before data classification [2]. This is due to outliers affecting the reliability and accuracy of classification method. Outliers can probably make bias in the training dataset, leading to models that are less effective at forecasting the correct class labels for new data [3]. One typical technique OD was to use statistical approaches like interquartile range (IQR) analysis or Z-score analysis for finding data points that are diverse from the rest of the dataset [4]. Such techniques involve computing a threshold related to the distribution of data, and finding any data points that fall outside of this threshold [5].

Once outliers were detected and treated or eliminated, data classification is executed with the use of different machine learning methods [6], like neural networks (NN) or support vector machines (SVM), decision trees (DT), and k-nearest neighbors (KNN). In the dataset, such techniques utilize paradigms for classifying new data points into their suitable class labels [7]. Rarely, data classification outlier and detection can be executed concurrently, utilizing techniques that are robust to outliers, like robust PCA or robust regression [8]. Such techniques can concurrently categorize data points and find outliers, without requiring separate pre-processing steps [9]. The high-speed data, emerging high-volume, and high-dimensional network structures demanded an increasing need to stream analytics in a record-by-record manner for OD, also called real-time or online detection or detection on Streaming Data (SD). For enhancing the outcome of such methods [10], Feature Selection (FS) can be utilized for irrelevant, eradicate noisy, or redundant features, resulted in superior prediction values along with that a minimal computational cost.

This study presents an artificial rabbit optimizer with Hopfield neural network based outlier detection with data classification (AROHNN-ODDC) technique. The presented AROHNN-ODDC technique focuses on the removal of outliers and classifies high dimensional data. In the presented AROHNN-ODDC technique, initial stage of data pre-processing is performed. Next, the OD process is performed by the Local Outlier Factor (LOF) model. Followed by, the ARO algorithm was applied for the effectual selection of subset of features. Finally, the HNN classifier is used for data classification purposes. For assessing the enhanced data classification outcomes of the proposed method, a wide range of simulations were carried out against benchmark datasets.

2. Literature Review

Tra et al. [11] present a supervised multiclass deep AE Gaussian mixture model (S-DAGMM), is an ensemble method of individual unsupervised DAGMM. This system saids S-DAGMM to find ambiguous outliers in testing and training datasets. As well, this study uses DAGMM for pre-training DN. It could thwart DNNs from being stuck on local optima since DAGMM may learn data distribution from unlabelled datasets. Mansour et al. [12] designed an IODML-BDA technique (intelligent OD with ML empowered big data analytics) for MEC. This approach used an adaptive synthetic sampling-based OD method for eliminating the existence of outliers. Further, to choose a potential feature set, oppositional swallow swarm optimization (OSSO) related feature selection algorithm was leveraged. Eventually, LSTM related classifier model was used for finding various class labels.

Kieu et al. [13] devised a structure for OD in time series that, for instance, can be utilized to find hazardous road locations and dangerous driving behavior. In particular, to enhance the feature spaces of raw time series, the authors provide a method that would generate statistical features. Then, to rebuild the enhanced time series, the authors use an AE. The AE performed dimensionality reduction, the representative features of the ed time series to capture utilizing small feature spaces. Dash et al. [14] presented a structure named Inter-Quartile Range (IQR) where a popular statistical method was utilized for identifying outliers in data and dealing with them by Winsor zing approach. Utilizing the preprocessed dataset under this structure, a radial basis function network trained by training a learning-related optimization method. Ijaz et al. [15] present a cervical cancer prediction model (CCPM) that renders initial prediction of CC utilizing risk factors as input. Utilizing OD techniques, the CCPM first eliminates outliers, OD techniques are isolation forest (iForest) and DBSCAN (density-based spatial clustering of applications with noise) and by growing cases in the data in balanced way.

3. Proposed Model

In this study, we have developed a new AROHNN-ODDC algorithm for automated OD and data classification processes. The presented AROHNN-ODDC technique focuses on the removal of outliers and classifies high dimensional data. In the presented AROHNN-ODDC technique, several sub-processes are involved namely data pre-processing, LOF based outlier removal, ARO based feature selection, and HNN based classification. Fig. 1 illustrates the overall flow of AROHNN-ODDC method.

A. *Outlier Detection using LOF Approach*

At this stage, the OD process is performed by the use of the LOF model. The Local Outlier Factor (LOF) refers to a method to find outliers in data hinge on their local density [16]. A density-related technique considered the local density of points relative to its neighbours for detecting points that have lower density than their neighbors. Here are the key steps indulged in the LOF approach:

Determine the k-nearest neighbors of all data points: The initial step was to decide the KNN of each data point. The value of k is ordinarily set to a small number like 5 or 10. Compute reachability distance of all data points: The reachability distance of data point can the distance to its k-NN. It is measure of local density around data points.

Calculate the local reachability density of each data point: The local reachability density (LRD) of data point inverse of average reachability distance of its KNNs. The LRD offers a measure of how isolated point was from its neighbors. Points with low LRD values are to be expected to be outliers.

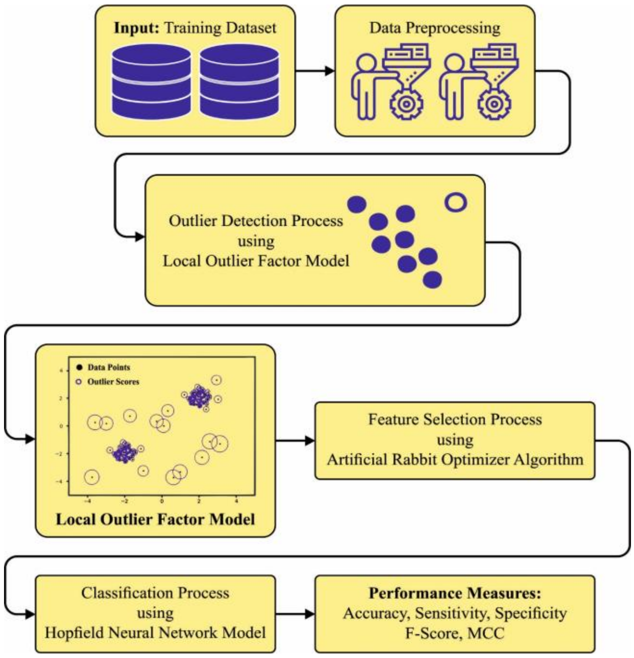

Fig. 1. Overall flow of AROHNN-ODDC approach

Calculate the Local Outlier Factor (LOF) for each data point: The LOF of data points was the average ratio of LRD of its k-NN to its own LRD. It can be measure of how much local density of point varies with neighbors. Points with higher LOF values were probably to be outliers.

Set a threshold for identifying outliers: Lastly, threshold values are set to detect data points with LOF values above threshold as outliers. The threshold value can be generally set related to domain knowledge and empirical observations. The LOF system has various merits over other OD methods. It is powerful to noise and can manage datasets with higher dimensionality. It could be is also capable of detecting outliers in data have complicated and irregular distributions.

B. Feature Selection using ARO Algorithm

In this work, the ARO algorithm was applied for the effectual selection of subset of features. The rabbits use random concealment and detour foraging strategies during the exploitation and exploration stages of life cycle in order to survive [17]. The mathematical modeling of this strategies in ARO is shown below.

Rabbit don't focus on what is near while finding food, and rather look in the distance. Once they munch grass from area except for their home range, is called detour foraging. Assume a swarm of bunnies, each having its own territory filled with burrows and grass, and randomly visiting others to eat. Rabbitslook for the area with richness of food. Every search individual tends to upgrade the location towards the random swarm member and contributed disturbance once the ARO is searching for food and it can be expressed as follows:

$$\overline{A}_{pi}^{k+1} = \overline{A}_{pi}^{k} + k_r \times (A_{pi}^k - A_{pj}^k) + round$$

$$\left(0.5 \times (0.05 + r_1)\right) \times k_{r1}, i, j = 1{:}n_p, j \neq i \quad (1)$$

$$k_r = a_L \times \rho_{(m)} \quad (2)$$

$$a_L = (e - e(\kappa - \frac{1}{k_{\max}})^2) \times sin(2\pi r_2) \quad (3)$$

$$\rho_{(m)} = \begin{cases} 1 & \text{if m} == x(y) \\ 0 & \text{else} \end{cases}, m = 1{:}n_d \& y = 1,2,\ldots,(r_3 n_d) \quad (4)$$

$$\chi = randperm(n_d) \& k_{r1} \sim N(0,1) \quad (5)$$

where a_L shows the running length of rabbit for dynamic behaviors of rabbit that results for exploration stage with exploitation phase or long step with short step size, $\overline{A}_{pi}^{k+1}$ and A_{pi}^k denotes the $i\text{-}th$ rabbit at $(k+1)$ and k iteration, correspondingly; $k_{\max}$ denotes the amount of maximum iterations, n_p and n_d indicates the number of rabbits (population) and number of burrows (dimension), correspondingly;$\rho_{(m)}$ denotes the mapping vector that assist the algorithm arbitrarily modify search individual foraging behaviors, r_1, r_2 and r_3 indicate the random integers with standard distribution and k_{r1} denotes the uniformly distributed random integer; k_{r1} shows the running features of a rabbit.

Eq. (1) demonstrates that searcher finds food in a random manner based on the locations. This allows a rabbit go to houses of other rabbits and this tendency of rabbits to visit other nests rather than their assist in exploring and ensuring that ARO technique could search globally.Rabbit $constructn_p$ burrows around their nests as means of child protection. In this work, the rabbit often excavates tunnels around itself in each direction. Next, to decrease the probability of being eaten, it arbitrarily elects a hole for hiding in. Eq. (6) gives the random concealment process,

$$\overline{A}_{pi}^{k+1} = A_{pi}^k + k_r \times \left(r_4 B_{ij}^k - A_{pj}^k\right) i = 1{:}n_p \quad (6)$$

Now, B_{pi}^k denotes the burrow selected at random manner to shield from $itsn_d$ burrows, and it is expressed as follows,

$$B_{pi}^k = A_{pi}^k + N_r \gamma A_{pi}^k, i = 1{:}n_p, m = 1{:}n_d \quad (7)$$

$$N_r = r_4 \times \left(\frac{\kappa_{\max - k + 1}}{k_{\max}}\right) \quad (8)$$

$$\rho_{(m)} = \begin{cases} 1 & \text{if m} == [r_5 \times n_d] \\ 0 & \text{else} \end{cases}, m = 1{:}n_d \quad (9)$$

Where r_4 and r_5 denotes the uniformly distributed random number within [0, 1], correspondingly.

In this phase, ARO changes the solution and global position using the hiding or detour foraging behaviors as follows,

$$A_{pi}^{k+1} = \begin{cases} A_{pi}^k & f(A_{pi}^k) \leq f(\overline{A}_{pi}^{k+1}) \\ \overline{A}_{pi}^{k+1} & f(A_{pi}^k) > f(\overline{A}_{pi}^{k+1}) \end{cases} \quad (10)$$

The above formula indicated that if fitness of $i\text{-}th$ rabbit's location is superior to fitness of the present location, the rabbit abandon present location and remain at candidate location made either by Eq. (1) or (6).

In ARO, rabbits hide at the end and forage at the beginning of iterations. Detour makes foraging more difficult. This search technique exploits rabbit energy that shrinks over period. Exploration to exploitation stage needs the energy factor model. The energy factor $E_{f(k)}$ in ARO can be expressed as follows,

$$E_{f(k)} = 4 \times ln\left(\frac{1}{r_6}\right) \times \left(1 - \frac{1}{k_{\max}}\right) \quad (11)$$

When rabbit has a higher energy factor, it forages in a novel region. Lower energy makes a rabbit lesser active and needs to hide. If $E_{f(k)} > 1$, then they detour browse in other rabbit territories and if $E_{f(k)} \leq 1$, then they arbitrarily use their burrows.

The fitness function has considered the selected features and the classifier accuracy. It has minimized the set size of the selective features and maximized the classifier accuracy. Hence, the following fitness function was utilized for assessing individual solutions, as seen in Eq. (12).

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \quad (12)$$

Here $\#SF$ is the count of selective features and $\#All\_F$ is the total attributes in the original dataset. ErrorRate signifies the classification error rate by making use of the selective features. ErrorRate was computed as the percentage of incorrect classifiers to classifications made, stated as a value between zero and 1. α is employed for controlling the significance of subset length and classification quality. In this experiment, α is set to 0.9.

C. Data Classification using HNN Model

At the final stage, the HNN classifier is used for data classification purposes. The most popular recurrent neural network (RNN) is the HNN [18]. This NN has been employed in a variety of applications such as optimization, classification, control, image processing, and resolving blind detection problems and is viewed as an associative memory. Fig. 2 defines the framework of HNN.

A HNN includes set of neurons N that upgrade the activation value independently and asynchronously of other neurons. A neuron i was represented by its state $S_i = \pm 1$. The principle of HNN was to store binary pattern of form $\{+1, -1\}^N$, and to utilize a rule, named Hebb's rule, to study them. The "energy" of HNN can be represented as follows:

$$E = -\frac{1}{2}\sum_{i,j}^{N} S_i S_j w_{ij} \tag{13}$$

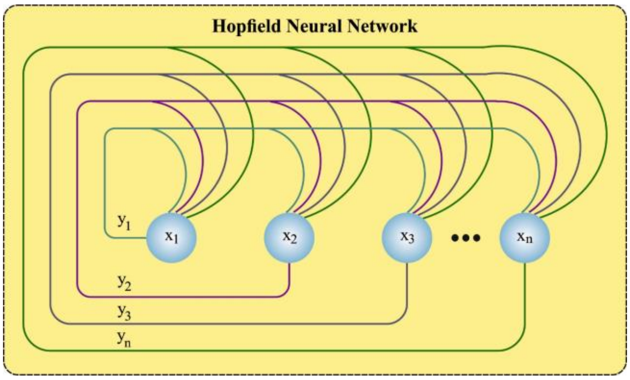

Fig. 2. Structure of HNN

In Eq. (13), w denotes the weight connected to the neurons i and j. S_i indicates state of neuron i. This quantity can be regarded as a Lyapunov function; it remains stable or reduces once network state is upgraded. Thus, the optimum value of weight is those that minimalize energy function. Moreover, theoretical storage capability of HNN, under assumption of stability of each the patterns, can be represented as follows:

$$P_{\max} = \frac{N}{4 \ln N} \tag{14}$$

In Eq. (14), $P_{\max}$ represents the maximal amount of uncorrelated patterns that are saved in N- neuron network. Every pattern saved corresponding to local minima of energy determined in (13).

4. Results Analysis

In this study, the experimental validation of the AROHNN-ODDC method is studied on the KDD-Cup 1999 dataset [19]. The dataset holds 41 features and the ARO algorithm has chosen a set of 23 features. Table 1 represents the OD results of the AROHNN-ODDC model. The results indicate that the AROHNN-ODDC model identifies the outliers effectively. On normal class with 750 samples, 667 samples are recognized after outlier removal process. Likewise, on DoS class with 750 samples, 682 samples are recognized after outlier removal process. Meanwhile, on Probe class with 750 samples, 694 samples are detected after outlier removal process. Finally, on R2L class with 750 samples, 639 samples are identified after outlier removal process.

Fig. 3 shows the classifier results of the AROHNN-ODDC technique under test dataset. Fig. 3a depicts the confusion matrix offered by the AROHNN-ODDC algorithm under 70% of TRS. The figure denoted that the AROHNN-ODDC approach has identified 485 instances under normal, 447 instances under Dos, 459 instances under Probe, and 424 instances under R2L. Besides, Fig. 3b depicts the confusion matrix offered by the AROHNN-ODDC method under 30% of TSS.

TABLE I

OD OUTCOME OF AROHNN-ODDC APPROACH WITH DISTINCT CLASSES

Class	Samples Before Outlier	Samples After Outlier
Normal	750	667
DoS	750	682
Probe	750	694
R2L	750	639
Total Samples	3000	2682

The figure denoted that the AROHNN-ODDC model has identified 168 instances under normal, 206 instances under Dos, 224 instances under Probe, and 183 instances under R2L. Similarly, Fig. 3c displays the PR analysis of the AROHNN-ODDC model. The figures reported that the AROHNN-ODDC model has obtained maximum PR performance under all classes. Finally, Fig. 3d illustrates the ROC investigation of the AROHNN-ODDC model. The figure shown that the AROHNN-ODDC method has proficient results with maximal ROC values under distinct class labels.

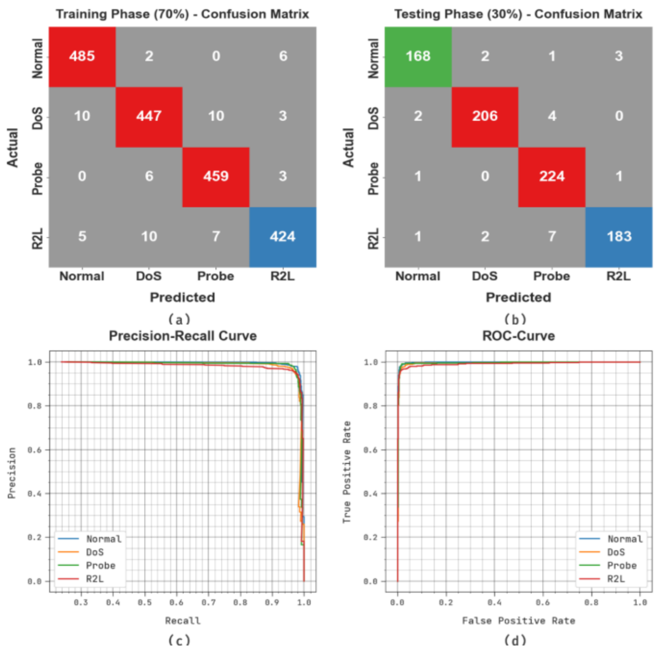

Fig. 3. Classifiers of (a-b) TRS/TSS of 70:30, (c) PR curve, and (d) ROC curve

In Table 2 and Fig. 4, the overall outcomes of the AROHNN-ODDC model are provided under 70% of TRS. The experimental values indicate that the AROHNN-ODDC method reaches effectual outcomes under all classes. On normal class, it is noticed that the AROHNN-ODDC model achieves $accu_y$ of 98.77%, $sens_y$ of 98.38%, $spec_y$ of 98.92%, F_{score} of 97.68%, and MCC of 96.86%. Meanwhile, on DoS class, it is noticed that the AROHNN-ODDC model achieves $accu_y$ of 97.82%, $sens_y$ of 95.11%, $spec_y$ of 98.72%, F_{score} of 95.61%, and MCC of 94.16%. Eventually, on R2L class, it is noticed that the AROHNN-ODDC model achieves $accu_y$ of 98.19%, $sens_y$ of 95.07%, $spec_y$ of 99.16%, F_{score} of 96.15%, and MCC of 94.97%.

TABLE II

OVERALL RESULTS OF AROHNN-ODDC APPROACH ON 70% OF TRS

Training Phase (70%)					
Class	Accuracy	Sensitivity	Specificity	F-Score	MCC
Normal	98.77	98.38	98.92	97.68	96.86
DoS	97.82	95.11	98.72	95.61	94.16
Probe	98.61	98.08	98.79	97.25	96.33
R2L	98.19	95.07	99.16	96.15	94.97
Avg.	98.35	96.66	98.90	96.67	95.58

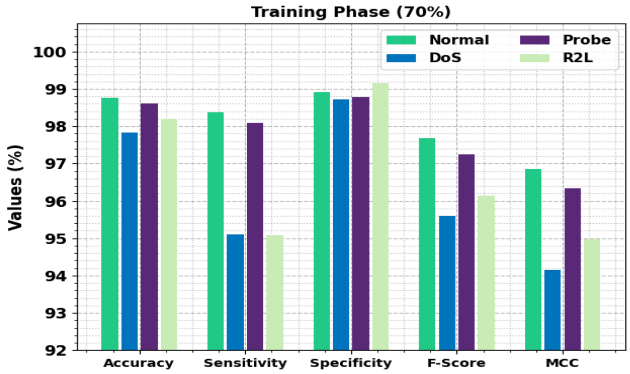

Fig. 4. Overall results of AROHNN-ODDC approach on 70% of TRS

In Table 3 and Fig. 5, the overall results of the AROHNN-ODDC model are provided under 70% of TRS. The experimental values indicate that the AROHNN-ODDC method reaches effectual outcomes under all classes. On normal class, it is noticed that the AROHNN-ODDC method gains $accu_y$ of 98.76%, $sens_y$ of 96.55%, $spec_y$ of 99.37%, F_{score} of 97.11%, and MCC of 96.32%. Meanwhile, on DoS class, it is noticed that the AROHNN-ODDC technique achieves $accu_y$ of 98.76%, $sens_y$ of 97.17%, $spec_y$ of 99.33%, F_{score} of 97.63%, and MCC of 96.79%. Eventually, on R2L class, it is noticed that the AROHNN-ODDC attains achieves $accu_y$ of 98.26%, $sens_y$ of 94.82%, $spec_y$ of 99.35%, F_{score} of 96.32%, and MCC of 95.20%.

TABLE III . OVERALL RESULTS OF AROHNN-ODDC APPROACH ON 30% OF TSS

Testing Phase (30%)					
Class	Accuracy	Sensitivity	Specificity	F-Score	MCC
Normal	98.76	96.55	99.37	97.11	96.32
DoS	98.76	97.17	99.33	97.63	96.79
Probe	98.26	99.12	97.93	96.97	95.80
R2L	98.26	94.82	99.35	96.32	95.20
Average	98.51	96.91	98.99	97.01	96.03

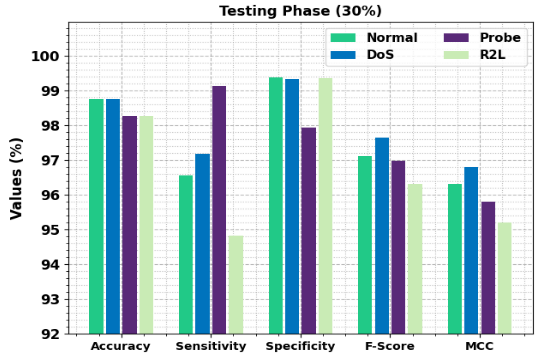

Fig. 5. Overall results of AROHNN-ODDC approach on 30% of TSS

The TACY and VACY of the AROHNN-ODDC model have investigated on outlier detection performance in Fig. 6. The outcomes implied that the AROHNN-ODDC model has shown improved performance with increased values of TACY and VACY. Especially, the AROHNN-ODDC method has reached maximum TACY outcomes.

Fig. 6. TACY and VACY outcome of AROHNN-ODDC approach

Fig. 7. TLOS and VLOS outcome of AROHNN-ODDC approach

The TLOS and VLOS of the AROHNN-ODDC model are tested on outlier detection performance in Fig. 7. The results show that the AROHNN-ODDC technique has better performance with least values of TLOS and VLOS. Visibly, the AROHNN-ODDC model has reduced VLOS outcomes.

In Table 4 and Fig. 8, a brief $accu_y$ examination of the AROHNN-ODDC method with existing methods is provided [20]. The experimental values stated that the AROHNN-ODDC technique gains higher $accu_y$ of 98.51%. Contrastingly, the existing techniques such as ODFST-SDC, CIDD-ADODNN, OC-SVM, NB, Gaussian, and DNN-SVM models have resulted in lower $accu_y$ of 97.99%, 95.79%, 91.35%, 90.54%, 93.21%, and 92.61% respectively. Thus, the AROHNN-ODDC model gains maximum performance over other existing techniques.

TABLE IV

ACCURACY ANALYSIS OF AROHNN-ODDC APPROACH WITH OTHER SYSTEMS

Methods	Accuracy (%)
AROHNN-ODDC	98.51
ODFST-SDC	97.99
CIDD-ADODNN	95.79
OC-SVM model	91.35
NB model	90.54
Gaussian model	93.21
DNN-SVM model	92.61

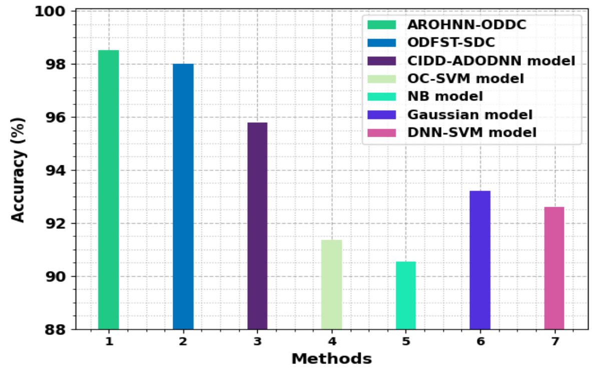

Fig. 8. Accuracy analysis of AROHNN-ODDC approach with other systems

5. Conclusion

In this study, we have developed a new AROHNN-ODDC algorithm for automated OD and data classification processes. The presented AROHNN-ODDC technique focuses on the removal of outliers and classifies high dimensional data. In the presented AROHNN-ODDC technique, several sub processes are involved namely data pre-processing, LOF based outlier removal, ARO based feature selection, and HNN based classification. In the presented AROHNN-ODDC approach, the primary level of data pre-processing can be carried out. Afterwards, the OD process is performed by the use of the LOF model. Followed by, the ARO technique is applied for the effectual selection of subset of features. Finally, the HNN classifier is used for data classification purposes. For assessing the enhanced data classification outcomes of the proposed method, a wide range of simulations were carried out against three benchmark datasets. The experimental outcomes stated the promising performance of the AROHNN-ODDC method over other existing techniques. In future, the performance of the AROHNN-ODDC technique can be enhanced by parameter tuning approaches.

References

[1] Liuliakov, A., Hermes, L. and Hammer, B., 2023. AutoML technologies for the identification of sparse classification and outlier detection models. *Applied Soft Computing*, *133*, p.109942.

[2] Venkateswarlu, Y., Baskar, K., Wongchai, A., Gauri Shankar, V., Paolo Martel Carranza, C., Gonzáles, J.L.A. and Murali Dharan, A.R., 2022. An Efficient Outlier Detection with Deep Learning-Based Financial Crisis Prediction Model in Big Data Environment. *Computational Intelligence and Neuroscience*, *2022*.

[3] Dwivedi, R.K., Rai, A.K. and Kumar, R., 2020, February. Outlier detection in wireless sensor networks using machine learning techniques: a survey. In *2020 International Conference on Electrical and Electronics Engineering (ICE3)* (pp. 316-321). IEEE.

[4] Schielein, M.C., Christl, J., Sitaru, S., Pilz, C., Kaczmarczyk, R., Biedermann, T., Lasser, T. and Zink, A., 2023. Outlier Detection in Dermatology: Performance of different Convolutional Neural Networks for Binary Classification of Inflammatory Skin Diseases. *Journal of the European Academy of Dermatology and Venereology*.

[5] Ahmed, U., Srivastava, G., Djenouri, Y. and Lin, J.C.W., 2022. Knowledge graph based trajectory outlier detection in sustainable smart cities. *Sustainable Cities and Society*, *78*, p.103580.

[6] Jesus, G., Casimiro, A. and Oliveira, A., 2021. Using machine learning for dependable outlier detection in environmental monitoring systems. *ACM Transactions on Cyber-Physical Systems*, *5*(3), pp.1-30.

[7] Jiang, J., Han, G., Shu, L. and Guizani, M., 2020. Outlier detection approaches based on machine learning in the internet-of-things. *IEEE Wireless Communications*, *27*(3), pp.53-59.

[8] Caroline Cynthia, P. and Thomas George, S., 2021. An outlier detection approach on credit card fraud detection using machine learning: a comparative analysis on supervised and unsupervised learning. In *Intelligence in Big Data Technologies—Beyond the Hype: Proceedings of ICBDCC 2019* (pp. 125-135). Springer Singapore.

[9] Bhatti, M.A., Riaz, R., Rizvi, S.S., Shokat, S., Riaz, F. and Kwon, S.J., 2020. Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *Journal of Communications and Networks*, *22*(3), pp.236-243.

[10] Ghosh, N., Maity, K., Paul, R. and Maity, S., 2019, May. Outlier detection in sensor data using machine learning techniques for IoT framework and wireless sensor networks: A brief study. In *2019 International Conference on Applied Machine Learning (ICAML)* (pp. 187-190). IEEE.

[11] Tra, V., Amayri, M. and Bouguila, N., 2022. Outlier detection via multiclass deep autoencoding Gaussian mixture model for building chiller diagnosis. *Energy and Buildings*, *259*, p.111893.

[12] Mansour, R.F., Abdel-Khalek, S., Hilali-Jaghdam, I., Nebhen, J., Cho, W. and Joshi, G.P., 2021. An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing. *Cluster Computing*, pp.1-13.

[13] Kieu, T., Yang, B. and Jensen, C.S., 2018, June. Outlier detection for multidimensional time series using deep neural networks. In *2018 19th IEEE international conference on mobile data management (MDM)* (pp. 125-134). IEEE.

[14] Dash, C.S.K., Behera, A.K., Dehuri, S. and Ghosh, A., 2023. An outliers detection and elimination framework in classification task of data mining. *Decision Analytics Journal*, p.100164.

[15] Ijaz, M.F., Attique, M. and Son, Y., 2020. Data-driven cervical cancer prediction model with outlier detection and over-sampling methods. *Sensors*, *20*(10), p.2809.

[16] Cheng, Z., Zou, C. and Dong, J., 2019, September. Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems* (pp. 161-168).

[17] Rao, C.R., Balamurugan, R. and Alla, R., Artificial Rabbits Optimization Based Optimal Allocation of Solar Photovoltaic Systems and Passive Power Filters in Radial Distribution Network for Power Quality Improvement.

[18] Keddous, F.E. and Nakib, A., 2022. Optimal CNN–Hopfield network for pattern recognition based on a genetic algorithm. *Algorithms*, *15*(1), p.11.

[19] https://www.kaggle.com/galaxyh/kdd-cup-1999-data

[20] Rajakumar, R. and Sathiya Devi, S., 2023. A Novel Outlier Detection with Feature Selection Enabled Streaming Data Classification. *Intelligent Automation & Soft Computing*, *35*(2).

