International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I3P229

Volume 71 Issue 3, 282-294, March 2023
© 2023 Seventh Sense Research Group®

Original Article

An Efficient Approach for Discovering Objects in the Internet of Things using Clue-Based Search Engine

R. Raghu Nandan[1], N. Nalini[2]

[1]*Department of Computer Science and Engineering, Navkis College of Engineering, Hassan, Karnataka, India*
[2]*Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka, India*

[1]*Corresponding Author : aghu.siet@rediffmail.com*

Received: 12 December 2022 Revised: 02 March 2023 Accepted: 22 March 2023 Published: 25 March 2023

Abstract - *Internet of Things (IoT) technology is a form of a network employing different data-sensing devices that capture actual information and integrate any physical object with a pre-established protocol. Recently, Deep Learning and its variations have been used to detect objects in IoT. However, recognition of objects connected in the network still has major problems because of its troubled networks. Thus, accuracy is not attained in various methods due to their complex methodology. An effective method for discovering IoT objects using a clue-based search engine is proposed to address these concerns, mainly designed for identifying the objects connected in a network. In the initial phase, pcap files are gathered. The preprocessing phase includes flow generation missing field rejection, and normalization techniques. The RFO-based Deep Neural Network (DNN) methodologies are used to identify the objects connected to the network. The objects Discovery Search Engine in the IoT network are tested to verify its performance. Also, the proposed model is compared with several existing models, namely DBN, ANN, SVM and KNN, to estimate their effectiveness. The classification accuracies of DNN, DBN, ANN, SVM and KNN are 95.3%, 92.6%, 87.4%, 84.8% and 80.3%, respectively. The proposed work is based on the clues specified by the requester, which is the first of its kind so far. It is found that the accuracy of the DNN model gives better results when compared with the other models.*

Keywords - *Object recognition, Internet of Things, DNN, Clue Based Search Engine, pcap file*.

1. Introduction

The Internet of Things (IOT) is a system of interconnected objects and technology that enables communications between them and the internet. IoT technology allows electronic devices to communicate with one another through the internet, thereby improving the lifestyle of humanity. Smart devices and the internet, leading to IoT, help to provide appropriate information to the problems faced by business, government and public/private sectors worldwide. IoT is rapidly gaining importance and therefore is widespread in real-world activities. Due to the increasing growth of IoT devices, device management and network security have become major concerns. Since more devices are getting into the IoT network, there is a need to identify the devices precisely to communicate effectively.

Most of the time, network administrators/users are unaware of the number of IoT devices connected to the network, while attackers are increasingly concentrating on the vulnerabilities of IoT devices. The users are not knowledgeable in updating the patches to the IoT devices and reducing the vulnerabilities. Recently, researchers working on identifying IoT devices have used supervised machine learning techniques to improve accuracy. For training, these approaches require a large amount of tagged data. Asset management and IoT security rely on recognising various IoT devices and keeping track of their state to ensure they act appropriately. Therefore, gathering a large amount of labeled data requires much time and effort and cannot be extended to an environment with many IoT devices. Identification of IoT devices is the essential step in accomplishing these targets.

In the past, IoT detection initiatives were focused on supervised learning and a large amount of labeled data, which is time-consuming and not scalable. These are the major concerns that arise about IoT object discovery during transmission. IoT devices do not explore the enquiry procedure to find the feature of device biometrics. Machine learning models necessitate a large amount of training examples, specifically when factoring in the bias-variance trade-off, which is another significant difficulty. There are three major issues while using semi-supervised learning to identify IoT devices. Firstly, they may depend on a small number of labeled data and a large number of unlabeled data. Therefore, the selected features will identify various devices as far as reasonable. Secondly, IoT devices frequently operate at different phases and provide various features, which might affect accuracy.

Additionally, some network operations, such as scanning, alter IoT devices' functionality by affecting their traffic. Finally, there is a number of non-IoT devices included in the collection. They could have incredibly dissimilar characteristics that alter the classification outcome. The majority of the research carried out so far does not identify the devices by the query given by the user for discovering the IoT device. However, the proposed research aims to discover the devices by considering the user query to identify the IoT devices appropriately.

Thus, various existing approaches have not attained accuracy effectively, which is the main concern that arises in the device discovery process. Also, this can lead to misconfiguration among the connected devices. There is a need for an efficient approach to discovering the object in the Internet of things environment to have fruitful communication even if the person does not know the technical way of presenting a query in the process of identifying the devices. The discovery of IoT devices is a pre-requisite to characterize, monitor and protect these devices.

This Paper focuses on identifying the devices based on the clues in the description provided by the user to discover the IoT Devices. The remaining part of the work carried out is organized into 4 sections. Section 2 describes state of the art concerned with the current object identification methods used in IOT. The proposed methodology of object identification is described in Section 3. The outcomes and performance indicators of the suggested framework are described in Section 4. The conclusion of the work carried out is given in section 5.

2. Literature Review

To know the state-of-the-art in the related work, a survey is carried out, and the following is the gist of the papers:

Pashaei *et al.* [1] presented Distributed Learning Automata-based algorithm (DLA) for finding objects with high popularity or influence in the IoT environment. The fundamental idea is that an item might use friends or friends of friends to choose the best service provider. This approach has produced a brand-new centrality metric that quantifies a node's significance level inside the SIoT while considering its graph properties. A DLA technique is applied for choosing the most important nodes in the network by embedding a Learning Automata in each object. If there is no matching of words presented in the query description, then this model will not work. Therefore the queries of the people in their words will not yield a fruitful result of the device discovery.

Mardini *et al.* [2] introduced a hyperlink selection method that uses the Genetic Algorithm (GA) to locate the nearly ideal response. There are too many objects, services and interactions on the IoT, which are heavily populated. In order to deliver a particular service, it is crucial to have the capacity to find the appropriate object. This is made feasible by the Social Internet of Things (SIoT), a combination of IoT and Social Networking. The fundamental element of the SIoT is that every IoT object can use its relationships with friends and friends-of-friends to find a particular service. Because each node must handle a large number of friends, and this process is typically slow.

Zannou *et al.* [3] designed an IoT service selection and discovery method. An edge server employing a neural network carries out the discovery process. Ant Colony Optimization (ACO) is employed in the selection phase by nodes to choose a suitable node from the set of relevant nodes.

Koroniotis *et al.* [4] designed a methodology for identifying and tracing attack behaviors in IoT networks using a new network forensics framework. It is also known as the Particle Deep Framework (PDF). The suggested scheme adds three new functionalities: (1) extracting network data flows and clarifying their integrity to deal with encrypted networks; (2) Using a Particle Swarm Optimization (PSO) algorithm to adapt deep learning parameters automatically; and (3) generating a Deep Neural Network (DNN) depending on the PSO algorithm to find and detect anomalous behavior from IoT network of smart buildings.

Ashraf *et al.* [5] had presented a deep learning-based Intrusion Detection System (IDS) for Intelligent Transport Systems (ITS), particularly to identify abnormal network activities of In-Vehicle Networks (IVN), vehicles to vehicles (V2V) communications and vehicles to equipment networks. A Long-Short Term Memory (LSTM) auto-encoder technique based on Deep Learning architecture is used to recognize intrusive events from the central network gateways of Automatic Vehicles (AV).

Noh *et al.* [6] developed Deep Learning technology which is presently being a hot research area for numerous research teams and Information Technology (IT) organizations due to the extensive use of object identification methods employing unsupervised learning and the technology’s high potential and efficiency. The fabric industry utilizes a significant amount of personnel in all stages, including gathering raw materials, colouring, cleaning and stitching. The discarded items can minimize or eliminate environmental pollution. As an outcome of this test, it was verified that effective categorization work could be carried out without depending on the knowledge and skills of actual personnel in the shuttered item identification workplace. It will result in a fresh approach to the item classification task previously done by workers in the current workforce.

Coquin *et al.* [7] had introduced a significant robotic object detection method using sensors. Moreover, various variables, like illumination, conflict, obstruction, device alignments, etc., can restrict the accuracy of a robot's identification, especially a robot connected with a vision and operating in an unknown situation. According to these limitations, a simple robotic image may only be capable of providing limited and unreliable data. It limits the ability to handle complicated detection scenarios, especially in ambiguous and uncertain surroundings. Several sensors may be compatible, and their mixture boosts the recognition rate of one robotic lens. Through network protocols, the IoT platform enables the communication between robotic as well as several linked devices.

Sachin Kumar *et al.* [8] have proposed a workshop on the importance of the Internet of Things in our daily life. The occupancy of smart devices in daily life is increasing in the areas of security and surveillance, Agriculture automation, smart cities, home-energy consumption, healthcare and so on. The authors stated that the future of everything would be connected to the internet, which is a big challenge in identifying the right object for communication. It also shows concerns about analyzing the huge amount of data that the devices produce.

Kashif Naseer Qureshi et al. [9] have presented a paper on an object detection model that offers many object detection and recognition. The technique used is a CNN-5GIoT model. It has advanced CNN-based object detection and recognition systems with 5 G-based data communication services for IoT networks. The object detection model proposed gives better results and scaling percentage localization and recognition. The large and complex IoT data is processed with 5G networks and edge and cloud computing models, which provide fast data delivery with large bandwidth capabilities.

Hyoduck Seo et al. [10] proposed an object recognition technique based on the gesture recognition of nearby IoT devices. It is capable of performing multi-sensor MEV-based gesture recognition techniques. The proposed system uses multiple sensors installed in an MEV to log driving data as the vehicle operates and to recognize objects surrounding the MEV to remove blind spots. The proposed system can be utilized in different fields in conjunction with a wide range of technologies. These advantages facilitate innovations in the form of new algorithms and technologies through union with technologies in various fields, such as 5G, self-directed driving, and AI.

Fang-Qi Li et al. [12] presented an article on the pervasive deployment of IoT that has radically facilitated manufacturing. Despite the efforts on feature engineering and classification backend design, they are nonflexible against traffic distribution. A fuzzy system for the online defense of IoT is designed to deal with such drawbacks. The proposal is tested on settings and is compared with various competitors. Experimental results have shown the advantages of the proposed system for the accuracy of results.

Jie Tang et al. [14] have proposed work on efficiently integrating more deep learning workloads with IoT Devices. The authors have developed an OS consisting of a sensor interface for consuming standard sensor inputs and a compiler based on NNVM to compile and optimize existing deep-learning models into executable code. A message-passing structure based on nano-msg to connect all the nodes is developed.

Raghu Nandan R et al. [15], authors have proposed a semantic search engine for detecting objects in the Internet of Things using the user query. The proposed search engine can identify the devices based on the semantic meaning present in the user query. Often the user is not aware of the technical terms to write in the query, so that the users will define them in their own terms. The proposed search engine will accurately identify the devices by extracting the semantic meaning hidden in the search query.

According to the above-reviewed articles, several concerns arise during the detection process. In authors [1], the results of the navigation process are highly scalable and use low navigation time in the distributed nature. In the authors [2], five ways of analysis are described, which are used in the field to address the problem of link selection of friends. In authors [3], designers suggested a service discovery and selection method in the IoT. The author [4] describes the flaws commonly occurring in cyber hazards that impair the operation of their communication networks, considering their small size and low energy usage. In authors [5], autonomous vehicles (AVs), in particular, are susceptible to security and safety vulnerabilities that put human lives in danger are discussed. In authors [6], object identification technologies are used that employ IoT and AI to develop a classification system for recycled items is developed. In the author [7], object recognition systems and evidence theory combine the data released from the IoT sources to make the identification judgment.

Thus, there is a need for a methodology to overcome the issues that exist in the current state of the art. The discovery of objects in the IoT network is significant because of many issues, like ensuring security during information exchange. Therefore, proposing a search engine that searches for the accurate object more efficiently and in less time is essential. Hence, the work entitled “An Efficient Approach for Discovering Objects in the Internet of Things using Clue Based Search Engine” is proposed.

3. Proposed Methodology

The architecture of the proposed methodology is shown in Figure 1. The methodology consists of five stages: Query process, clue-based clustering, Preprocessing, Feature extraction, Feature reduction and Feature matching. The dataset contains pcap files as input data for identifying objects in the connected network, which is gathered from the online website [21]. A total of 45950 samples are taken for experimentation in the proposed work.

The device detection system contains six phases: query process, clue-based clustering, preprocessing, feature extraction, feature reduction and feature matching. The pcap files are considered for discovering the object in the IoT network. The pcap file is initially gathered during the query process. Then, clue based clustering algorithm is applied for labeling the samples. Further, the raw data is subjected to preprocessing to improve the discovery accuracy. Feature reduction is used to reduce the size of Latent Semantic Indexing (LSI). In order to identify the objects within the network, an RFO-based Deep Neural Network (DNN) is used.

3.1. Query Process

During the query process, a server sends a packet to a local computer using an IP address and receives an acknowledgement. The host then responds with a set of labeled packets (request, response) and sends information in the pcap file.

The pcap files contain information about IoT devices, IP address, hostname, source ID, destination ID, time interval features, traffic volume features, protocol features and TLS-related features. These are the network's important packet features that help identify the objects.

3.2. Clue-Based Clustering

The features from the pcap file are used as inputs for the clue-based clustering strategy. The labelling-acquiring technique known as Clustering Uncertainty-weighted Embedding (CLUE) finds a variety of target samples. In order to illustrate ambiguous parts of the feature set, the deep embedding of target samples is first reweighted based on model entropy. These uncertainty-weighted embeddings are clustered. The samples closest to each cluster centroid are obtained for labelling to select a variety of samples. The adjusted model is then utilised to provide well-classified target data after the target labels are gathered [17].

Bezdek initially developed the FCM clustering algorithm, which is a method of clustering that moves each data pixel into at least two distinct groups. Data is more likely to be related to a particular cluster if it is closer to the cluster centroid. To reduce an objective function with respect to the fuzzy membership set U of the cluster centroids is essential.

$$J_m(U,V) = \sum_{j=1}^{N}\sum_{i=1}^{C}(u_{ij})^m\left(\|x_j - v_j\|\right)^2; 1 \le m \le \infty \quad (1)$$

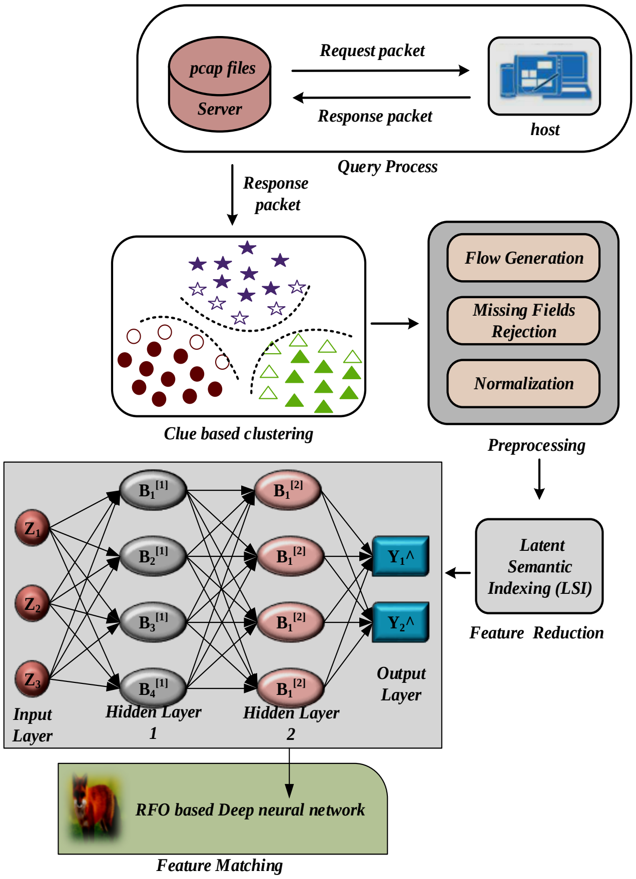

Fig. 1 The architecture of Clue-Based Search Engine

In Equation (1) $X = x_1, x_2, \dots, x_n$ is a $P \times N$ data matrix, and m is any real number higher than 1. The numerals P, N and C represent the size of each x_j feature vectors, the total number of feature vectors (pixel numbers in the image) and the total number of clusters appropriately.

The membership function of the vector x_j to the ith cluster, $U_{ij} \subseteq U(P \times N \times C)$ is specified as having $U_{ij} \in [0 \;\; 1]$ and $\sum U_{ij} = 1, j = 1,2, \dots, N$. Equation (2) gives the membership function for vector Uij.

$$U_{ij} = \sum_{k=1}^{c}\left(\frac{(\|x_j - v_i\|)}{(\|x_j - v_k\|)}\right)^{\left(\frac{-2}{m-1}\right)} \quad (2)$$

Where, V is a $P \times C$ matrix with the values $v_1, v_2, v_3, \dots, v_i, \dots, v_c$. The ith feature centre is calculated using equation (3).

$$V_i = \frac{\sum_{j=1}^{N}(U_{ij})^m \times J}{\sum_{j=1}^{N} U_{ij}{}^m} \quad (3)$$

The level of fuzziness $d^2(x_j, v_i)$, Where m is any real number larger than 1 is controlled. Equation (4) describes the measure of similarity between x_j and v_j.

$$d^2(x_j, v_i) = \left\|x_j - V_j\right\|^2 \tag{4}$$

Here, || || is used to indicate either a simple Euclidean distance or its generalisation, the Mahalanobis distance. The pixel intensity $p = k$ is represented by the feature vector X in the MR picture. Until || $(U_{ij})^{(k)} - (U_{ij})^{(k+1)}$|| $\le \epsilon, \epsilon$ 0 to 1, k is the amount of iterations, the FCM strategy repeatedly optimizes $J_m(U, V)$ with the continuous update of U and V. The algorithm for traditional fuzzy C-means clustering is shown in Table 1.

The cluster size must lie between 2 and n, where n is the amount of data points.

3.3. Preprocessing

In the preprocessing phase, the pcap files, which include the information of packets corresponding to objects in the network, are considered. The pcap files are preprocessed using different techniques, namely, flow generation, missing field rejection and normalization, to improve classification accuracy.

3.3.1. Flow Generation

It will change the initial network flow into identical five-tuple information (source IP address, source port, destination IP address, destination port and transport layer protocol).

3.3.2. Missing Fields Rejection

Missing fields in the pcap files are rejected using the missing field rejection technique.

3.3.3. Normalization

In normalization, the min-max normalization approach is applied. With min and max being the minimum and maximum values, it is a method that linearly converts the variables using the equation (5).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{5}$$

3.3.4. Feature Extraction

There are 10 different kinds of features in the dataset, namely, Network, IP, Host, Host_full, Traffic_cv, Packet_snd, Packet_rcv, Country and Party, which are given in Table 2. The features are given to into clue-based FCM algorithm for the purpose of labelling, i.e. clustering the samples in the dataset. After clustering the samples, the features are given to preprocessing phase to perform preprocessing, thereby enriching the data quality.

3.4. Feature Reduction

In the feature reduction phase, the features that do not contribute to the classification accuracy are eliminated using Feed Backward Selection (FBS) technique. In this phase, the features, namely net, hostful, country and party, are eliminated—the sample feature values of objects after the feature reduction phase are given in Table-3.

A unique feature reduction system based on latent semantic analysis is considered. In this phase, the features that do not contribute to the performance are identified and eliminated, reducing the computation cost. Data is transformed from a high-dimensional space into a new low-dimensional space, and Latent Semantic Indexing (LSI) is used. LSI is a common data capture method created using linear algebra, frequently used through a range of fields to retrieve important linguistic information.

For document analysis, LSI initially constructs m x n matrix A and, given in equation (6), displays the frequency with which every term appears throughout the documents, where m represents the number of identical words and n represents the total number of files. Singular Value Reduction is used to form two orthonormal matrices U and V. Where, U is a $m \times t$ term-concept matrix, V is a $n \times t$ file matrix, and t is minimum(m, n).

$$A = U.S.V^1 \tag{6}$$

Table 1. Fuzzy C-means clustering algorithm

Assuming that the collection of data points $X = x_1, x_2, x_3, \dots, x_n$ be and the set of vertices $V = v_1, v_2, v_3, \dots, v_c$
1. The size of the clusters must be set at $c, 2 \le c \le n,$ where n is the amount of data points. Modify where $1 < m < \infty$. Select any internal product-induced standard metric ‖.‖. 2. The fuzzy c partition $U^{(0)}$ should be established. 3. If $b, b = 0,1,2, \dots,$ 4. Use the Equation to determine the fuzzy member function U_{ij} (2). 5. Utilize the Equation to calculate the fuzzy centres $'V_i'$ (3). 6. Repeat steps 2 and 3 until the least $'J'$ value is achieved or $\|{U_{ij}}^{(k+1)} - {U_{ij}}^{(k)}\| < \varepsilon$

Table 2. Sample Features values of pcap files

Network	IP	Host	Host_full	Traffic_snd	Traffic_rcv	Packet_snd	Packet_rcv	Country	Party
neu	192.168.20.254	192.168.20.254	192.168.20.254	147365	110985	972	865	192.168.20.254	0
neu	33:33:ff:d4:26:ae	33:33:ff:d4:26:ae	33:33:ff:d4:26:ae	156	0	2	0	XX	0
neu	35.190.54.210	nest.com	logsink.devices.nest.com	2028573	1193342	3776	2597	SE	0
neu	02:e5:43:8e:83:f1	02:e5:43:8e:83:f1	02:e5:43:8e:83:f1	6102	0	54	0	XX	0
neu	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	ff:ff:ff:ff:ff:ff	2018	0	40	0	XX	0
neu	8a:60:8a:92:83:f6	8a:60:8a:92:83:f6	8a:60:8a:92:83:f6	42600	42600	710	710	XX	0
neu	3.87.98.35	amazonaws.com	amazonaws.com	29952	15678	234	117	US	0
neu	52.16.28.178	amazonaws.com	amazonaws.com	104016	68612	1204	610	IE	0
neu	162.88.193.70	dyndns.com	dyndns.com	798	1202	10	10	US	0

Table 3. Feature values After the Feature reduction phase

Ip	Host	Traffic_Send	Traffic_Rcv	Packet_Send	Packet_Received	Network
206.55.191.142	usinternet.com	2659	7947	16	14	icl
208.75.88.4	ntp.org	4927	6032	21	20	icl
114.118.7.161	ntp.org.cn	70168	70168	716	716	icl
202.118.1.130	ntp.org.cn	2713	8001	17	15	icl
98.152.165.38	apnic.net	319879	349272	1161	864	icl
45.43.30.59	amazonaws.com	458722	34621	437	321	icl
216.6.2.70	misaka.io	7779	21023	45	39	icl

A $t \times t$ diagonal matrix S, reflecting singular values collected from term matrix A, is also formed by Singular Value Decomposition (SVD) throughout, as well as two orthonormal matrices U and V. This matrix S is sorted in descending order to approximate the initial term-document matrix (A) effectively. The top k rows from S are obtained with $k < t$.

$$A \approx U_k . S_k . V_k^1 \tag{7}$$

U_k and $V_k$are k rank matrices that best approximate A. In k-dimensional space that has been compressed, each row of V_k represents a feature for a document. This permits LSI to represent the documents in a simplified k-dimensional space without losing any significant amount of the valuable information originally carried [18].

3.5. Feature Matching

Deep Neural Network (DNNs) is applied in the feature-matching process, which is a type of Neural Network that places distributions over their weights. The Red Fox optimization (RFO) technique is used to choose the ideal weight and minimize error values taken into account as a fitness function. This method categorises a feature vector employing DNN into one of the k classes. Once the devices are matched, the remote host device labels' types, sellers and products are identified. The interaction between the server and the remote host is a communication process that does not rely on the participation of the remote hosts.

3.5.1. DNN

Improvements in conventional networks using prior analysis are referred to as Deep Neural Networks (DNN) in order to prevent over-fitting. In a larger sense, the Deep idea uses statistical methods such that all variables, especially design variables, are associated with a probability distribution (weights and biases in neural networks). Frequently, the vibrational inference is used to estimate the algorithm for Deep Learning posterior probability. The structure of RFO-based DNN is Considering the features reduced data from LSI as A, taken as the input for the model with N number of input pairs denoted as $D = [(X^n, Y^n)|n = ..1,2, N]$.

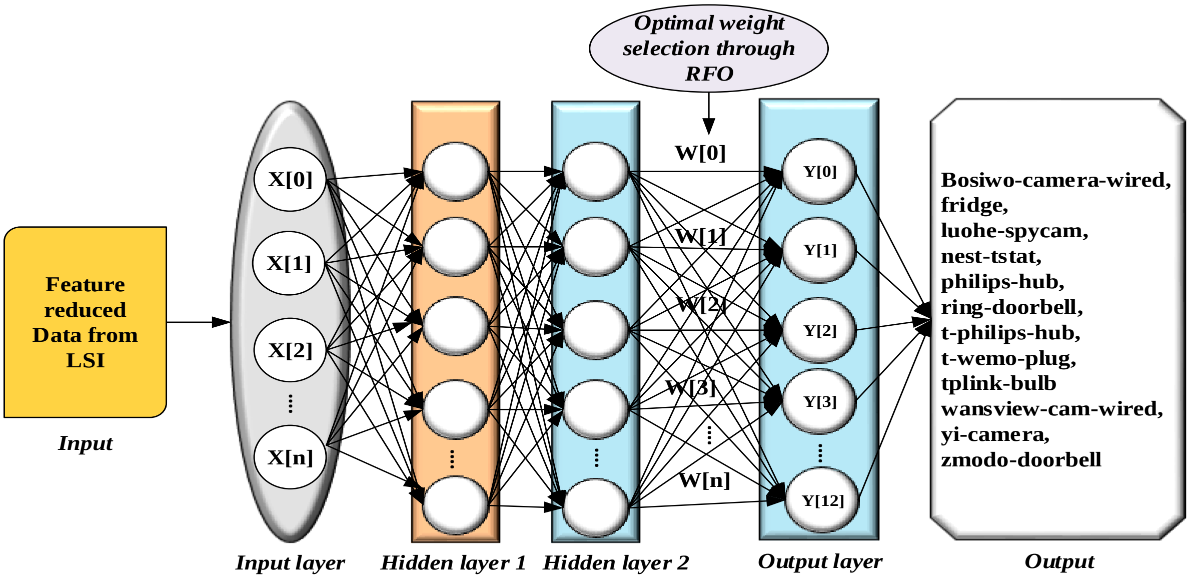
Optimal weight selection through RFO
Feature reduced Data from LSI
Input
X[0]
X[1]
X[2]
X[n]
W[0]
W[1]
W[2]
W[3]
W[n]
Y[0]
Y[1]
Y[2]
Y[3]
Y[12]
Bosiwo-camera-wired, fridge, luohe-spycam, nest-tstat, philips-hub, ring-doorbell, t-philips-hub, t-wemo-plug, tplink-bulb wansview-cam-wired, yi-camera, zmodo-doorbell
Input layer
Hidden layer 1
Hidden layer 2
Output layer
Output

Fig. 2 Structure of RFO-based Deep Neural Network

The DNN network typically comprises three layers, namely the input layer, the hidden layer and the output layer. The neurons of each layer are linked to the neurons present on the subsequent layer. The weight parameter is used to establish the connection between the neurons. Additionally, the input layer contains the activation function or identity function. Similarly, the hidden and output layers have activation functions called softmax or rectifiers. In multilayer architecture, some neurons contain non-linear activation functions, and others have linear activation functions. All neurons do not contain linear functions in all cases. These activation functions are responsible for developing the relationships between the output and the input. The neurons defined based on several activation functions are illustrated as given in equation (8).

Equation 8 defines the ReLU unit according to the activation function.

$$\sigma(x) = max(x, 0) \tag{8}$$

Sigmoid: According to the activation function, the sigmoid unit is defined by equation 9.

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{9}$$

Equation 10 defines the *Tanh* unit based on the activation function.

$$\sigma(z) = \tanh \llbracket(z)\rrbracket \tag{10}$$

The activation functions identified in the above formulations are the standard and basic activation functions present in every neuron. In this case, the network structure input layer is designated as Q, the hidden layer is designated as M, and the output layer is designated as N. Based on the parameters, the weight matrix for the hidden layer is identified and is given by equations (11) and (12).

$$w_A \in R^{Q \times M} \tag{11}$$

$$w_B \in R^{M \times N} \tag{12}$$

The above equations indicate the weight matrices and the matrix row represents the weight vector associated with each neuron. For example, consider the (a_i, b_i) as set P. The corresponding input is denoted as b_i and the desired output is denoted as a_i. Thus the training samples of the input and the output matrices are expressed as given in (13) and (14).

$$B = [b_1, b_2, b_3 \ldots\ldots b_p] \tag{13}$$

$$A = [a_1, a_2, a_3 \ldots\ldots a_p] \tag{14}$$

The hidden layer is represented on the output matrix is given in equation (15).

$$z = f(AW_A + B_1) \tag{15}$$

The bias matrix B_1, the bias vector b_1 and the activation function f are represented in equation (15). The intended output matrix of the neural network for $Y^\wedge \in R^{P \times N}$ is expressed by equation (16).

$$Y^\wedge = g(YW_B + B_2) \tag{16}$$

In equation (17), the activation function for the output layer is represented as g. B_2 stands for the bias matrix that extends along the matrix column. The additional formulation for the projected output matrix is given in equation (17).

$$Y = Y^{\wedge} + E = g(YW_B + B_2) + Error \tag{17}$$

The error matrix of the DNN is denoted by the expression $E = [e_1, e_2, e_3 \ldots\ldots e_p]$ in equation (17). Error matrix refers to the discrepancy between expected and actual results. Equation (18) provides the neural network's error estimation:

$$\varepsilon = \frac{1}{M}\sum_{m=1}^{M}\left(Actual(Y^{\wedge}) - Predicted(Y)\right)^2 \tag{18}$$

In equation (18), $Predicted(Y)$ stands for estimated output and $Actual(Y^{\wedge})$ for actual production. An error must be reduced in order to achieve effective network performance. In order to reduce the error, weight value optimization is carried out. The weight value is optimised, and the error is decreased as iterations progress. The weight parameter in DNN is optimised using the Red Fox optimization algorithm.

3.5.2. Red Fox Optimizer (RFO)

The Red Fox Optimizer (RFO) is developed based on the hunting behaviour of red foxes, and it is one of the most newly established meta-heuristic optimization techniques. The red fox is mainly selected to attain high efficiency and quick convergence to the optimum in many optimization problems. Compared to the other meta-heuristic algorithms, the red fox algorithm comprises exploration as well as exploitation.

The Squared Euclidean distance is given by equation (19).

$$D(((X)^i)^t, (X_{best})^t) = \sqrt{((X)^i)^t - (X_{best})^t} \tag{19}$$

The optimum solution is given in equation (20).

$$((X)^i)^t = ((X)^i)^t + \propto \times\ sgn((X_{best})^t - ((X)^i)^t) \tag{20}$$

In equation (20) $\propto$ specifies a random value. The applicants' new location should suggest a suitable solution. If not, the prior location would still exist. It needs to be defined by the application of the Red Fox Optimization approach. This is modelled by assuming an integer number for r in the range [0, 1] and is given in equation (21):

$$\begin{cases} move\ closer\ \ if\ r > \frac{3}{4}, \\ stay\ and\ hide\ \ if\ r \leq \frac{3}{4}. \end{cases} \tag{21}$$

Further, the motion of each member is computed utilizing an improved cochleoid formula. The radius-specific variable serves as a condition for the subsequent word based on two variables. The variable ϕ_0 represents a number between $[0, 2\pi]$ and the fox viewing angle is determined by a random number in the interval [0, 0.2]. The mathematical term is modelled as given in equation (22).

$$r = \begin{cases} a \times \frac{\sin(\phi_0)}{\phi_0}, if\phi_0 \neq 0, \\ \gamma, \qquad\qquad if\phi_0 = 0, \end{cases} \tag{22}$$

Equation (22) γ stands for any number between 0 and 1. Equation (23) gives the mathematical expression for the fox number surrounding the prey.

$$\begin{cases} x_0^{New} = a \times r \times \cos(\phi_1) + x_0^{actual} \\ x_1^{New} = a \times r \times \sin(\phi_1) + a \times r \times \cos(\phi_2) + x_1^{actual} \\ x_1^{New} = a \times r \times \sin(\phi_1) + a \times r \times \sin(\phi_2) + a \times r \times \cos(\phi_3) + x_2^{actual} \\ \vdots \\ x_{n-1}^{New} = a \times r \times \sum_{k=1}^{n-2}\sin(\phi_1) + a \times r \times \cos(\phi_3) + x_{n-2}^{actual} \\ x_{n-1}^{New} = a \times r \times \sin(\phi_1) + a \times r \times \sin(\phi_2) + \cdots + a \times r \times \sin(\phi_{n-1}) \\ \quad + a \times r \times \sin(\phi_{n-1}) + x_{n-a}^{actual} \end{cases} \tag{23}$$

A fixed population size is maintained by eliminating 5% of the worst members. In iteration t, two optimum members $(X(1))^t and\ (X(2))^t$ are obtained as an alpha couple. The territorial centre is then determined by applying equation (24):

$$H_c^t = \frac{1}{2}(X(1))^t - \left(X(2)\right)^t \tag{24}$$

A measure of the region diameter through Euclidean distance is given in Equation (25).

$$H_d^t = \sqrt{(X(1))^t - (X(2))^t} \tag{25}$$

A random value is denoted as σ, chosen in this process which lies between 0 and 1:

$$\begin{cases} New\ nomadic\ candidate, if\sigma > 0.45, \\ Reproduction\ of\ the\ alpha\ couple, if\ \sigma \leq 0.45 \end{cases} \tag{26}$$

The search space yields random locations. The alpha couple then introduces the new members to one another and is given in Equation (27).

$$(X^{rep})^t = \frac{\sigma}{2}(X(1))^t - (X(2))^t \tag{27}$$

The RFO study's incorporated parameters are a =0.2 and $\phi_0 = 1$.

3.5.3. Enhanced DNN

The features are identified using a simplified DNN classifier. Only a few number of neurons are undermatching because a large number of neurons are overfitting. The size and number of hidden layer neurons are carefully determined for the aforementioned purpose. The actual computation is carried out for each layer. The activation function supports DNN's problem-solving capacity and helps in defect learning. Activating functions are chosen for the linear transfer unit (ReLU) and Gaussian. It is difficult to calculate the weight parameter in a neural network since it is uncertain. Enhancing prediction accuracy requires finding the ideal weight. RFO is performed to identify the right weight factor. The error is reduced by increasing the weight parameter.

Training Phase

In the training phase, the bias and weights are reorganized to fit a specific sample input and its associated output. Because it helps develop an unidentified key link between inputs and output, the network can estimate the output class depending on input features after training. The steps involved in weight optimization using RFO are as follows:

Step 1: Initialization

In this step, the random initialization of weights is done:

$$w = \{w_1, w_2, w_2 \ldots\ldots w_n\} \tag{28}$$

Step 2: Evaluation of Fitness Function

A fitness function is used with the goal of reducing generalization error and achieving the optimum classification result.

$$f_{H(u)} = minimize\ (Error) \tag{29}$$

Step 3: Weight Updating

In order to arrive at the ideal solution, weight is updated as iterations proceed, and the equation for weight update is given in Equation (30)

$$Error = \sum \frac{(y_i - \hat{y}_i)^2}{n} \tag{30}$$

Step 4: Termination

The process is terminated when the correct solution is identified. The feature-reduced data is then provided to the DNN model as input. In the training and testing phases, 80% and 20% of data are used, respectively.

Testing Phase

The proposed methodology classifies 12 different kinds of objects connected in the network with different classification accuracies, as shown in Figure 4. If any unknown objects enter the network, the proposed model recognizes the unknown objects in the IoT network. This can avoid the occurrence of attacks from unknown objects. The performance of the proposed model is discussed in section 4. The overall process is provided in algorithm 1.

Algorithm 1: Pseudocode for object detection

```
Input: Dataset= A
#clue based clustering
cc= clue-based FCM (A)
# Preprocessing
F= Flow Generation (cc)
M= Missing Fields Rejection (F)
N= Normalization (M)//By using equation (8)
#Dimensionality Reduction
L= LSI (N)//using equation (10)
#Feature matching
D= Training_IDNN (L)
D1= Testing_IDNN (D)
If (D1=0)
class 0= bosiwo-camera-wired
If (D2=1)
class 1= fridge
If (D3=2)
class 2= luohe-spycam
If (D4=3)
class 3= nest-tstat
If (D5=4)
class 4= philips-bulb
If (D6=5)
class 5= ring-doorbell
If (D7=6)
class 6= t-philips-hub
If (D8=7)
class 7= t-wemo-plug
If (D9=8)
class 8= tplink-bulb
If (D10=9)
Class 9= wansview-cam-wired
If (D11=10)
Class 10= yi-camera
Else
Class 12= zmodo-doorbell
Output: Predicted objects
```

4. Results and Discussion

The experiments are conducted to know the accuracy of device identification. The performance metric values, namely accuracy, precision, recall, error, specificity, F1-Score and negative predictive values, are used to compute the classification accuracy. To compare and contrast, the proposed methodology is also compared with the other existing models, such as DBN, ANN, SVM and KNN. The metric values of the proposed and other models are given in Table 3.

Figure 6 compares existing and proposed approaches based on the accuracy (%). The accuracy is determined to be higher for the proposed data analysis. The classification accuracies of DBN, ANN, SVM, KNN, and DNN are 92%, 87%, 84%, 80% and 98%, respectively. It is found that the proposed model's accuracy is higher when compared with the other classification models. Figure 7 shows a comparative analysis based on Precision (%) between existing and proposed techniques. The precision results for DBN, ANN, SVM, KNN, and DNN are 75%, 70%, 67%, 65% and 81%, respectively. It is found that the Precision for the DNN model is higher.

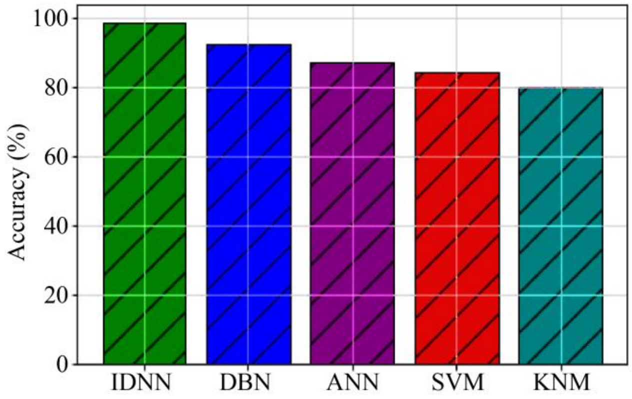

Fig. 3 Comparison of Accuracy

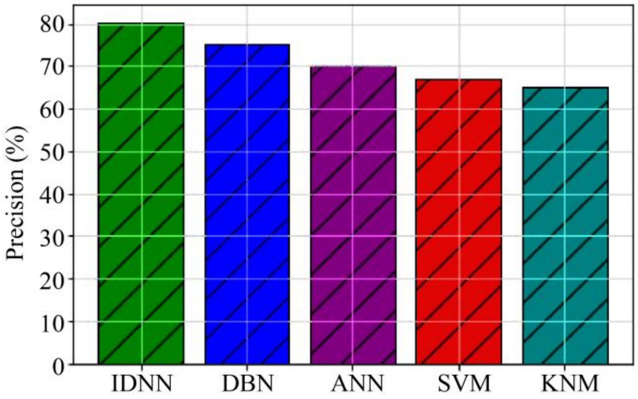

Fig. 4 Comparison of Precision

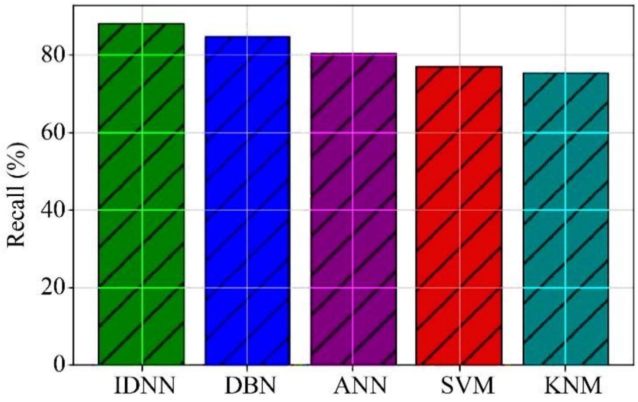

Fig. 5 Recall

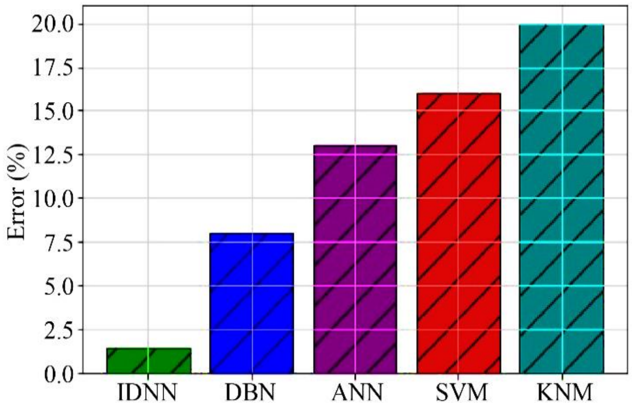

Fig. 6 Comparison of Error

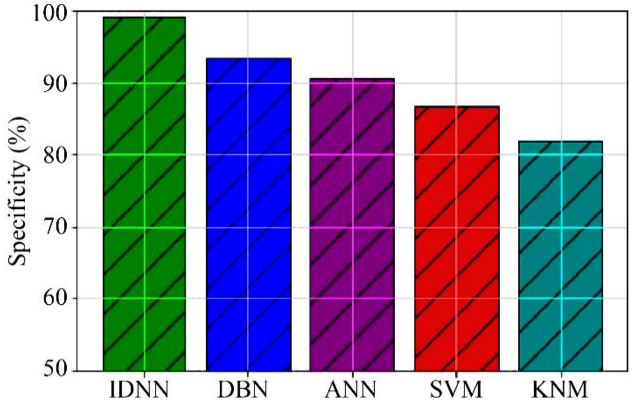

Fig. 7 Comparison of Specificity

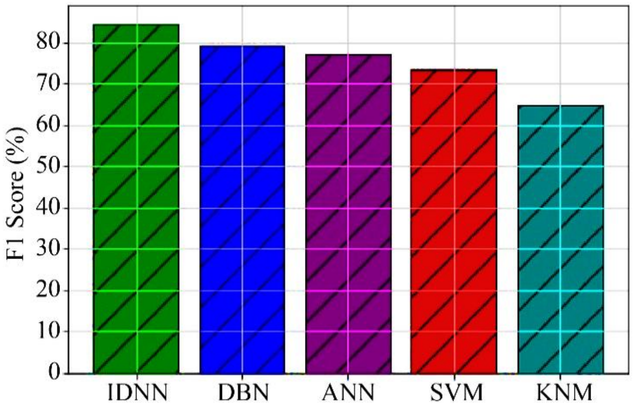

Fig. 8 Comparison of F1_Score

Similarly, Recall, Error, Specificity, F1_Score, Negative predicted value and False Positive rate are shown in Figure 5 thru Figure 10. The Recall, Specificity, F1_Score, and Negative predicted values are found to be higher than the other models. The Error and False Positive rates are found to be the least compared to the other models. Hence, the proposed model's performance is found to be better than the other models considered.

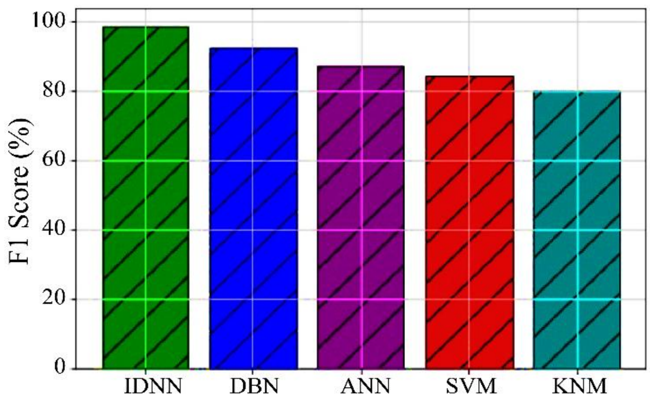

Fig. 9 Negative predictive value

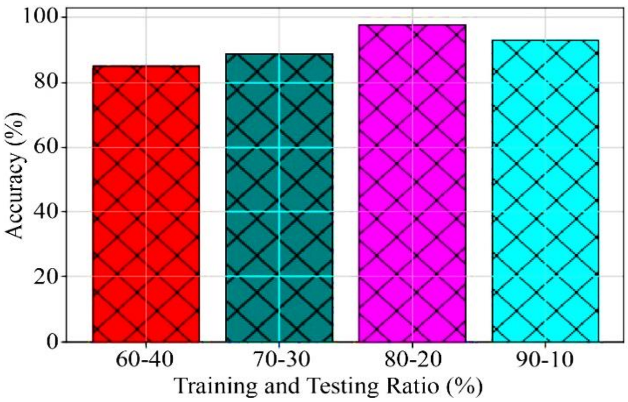

Fig. 11 Comparison with different training and testing ratio

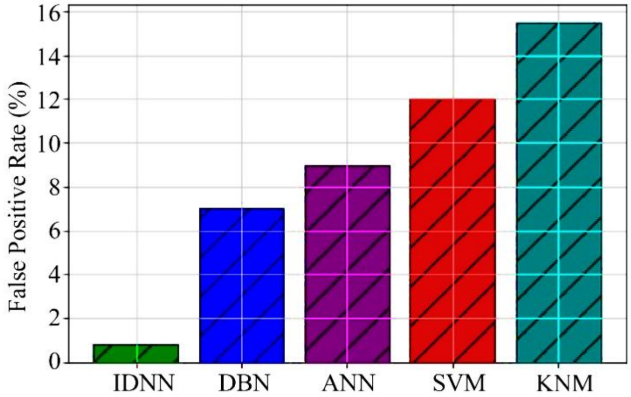

Fig. 10 False positive rate

Table 3. Performance metric values of proposed and existing models

Method and Metrics	IDNN	DBN	ANN	SVM	KNN
Accuracy (%)	98	92	87	84	80
Precision (%)	81	75	70	67	65
Recall (%)	88	84	80	77	75
Error (%)	2	8	13	16	20
Specificity (%)	99	93	90	86	82
F1_Score (%)	85	79	76	73	65
Negative predictive value (%)	97	94	90	85	81

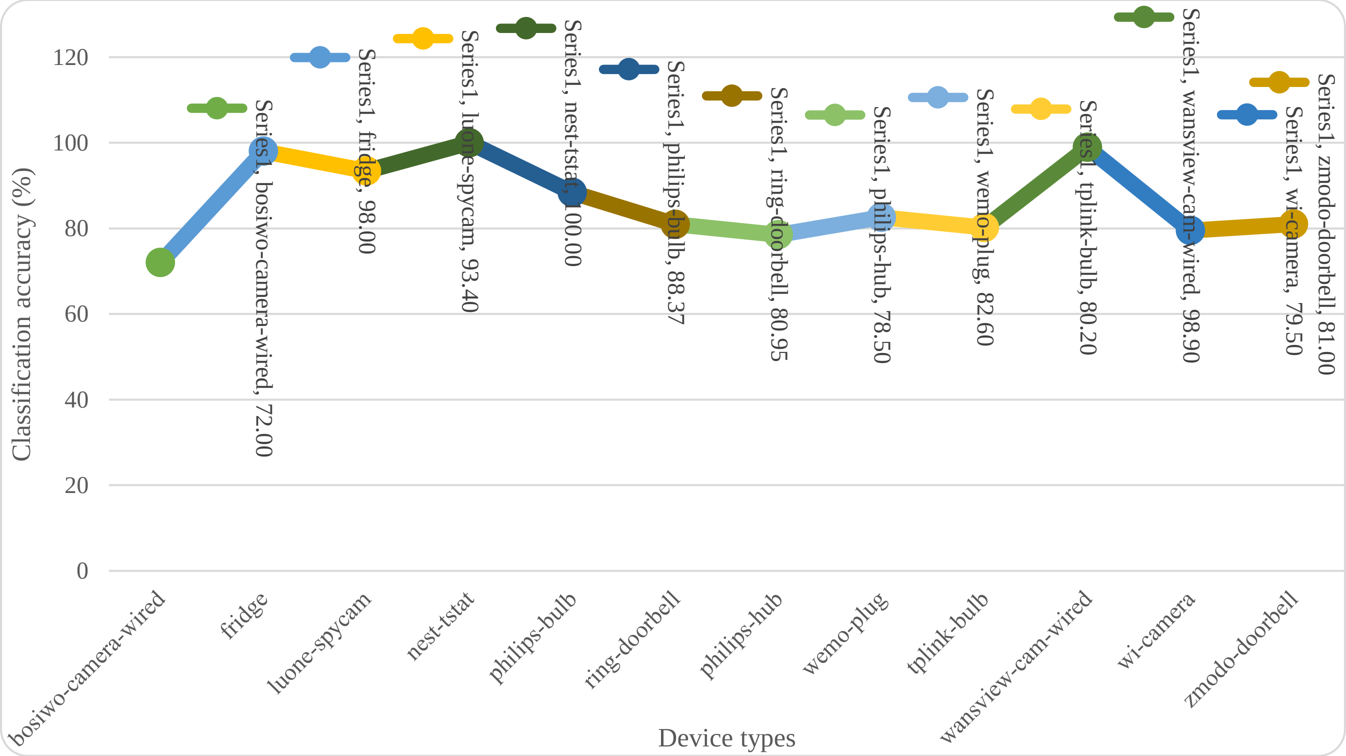

Fig. 12 Classification accuracy using DNN

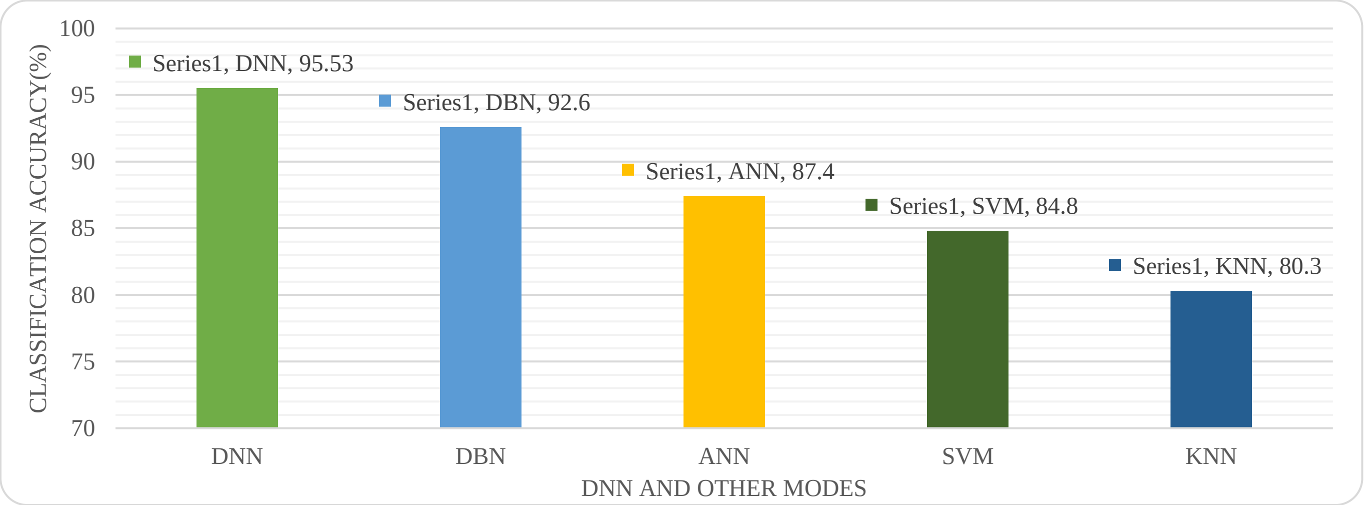

Fig. 13 Accuracies using DNN and other models.

The classification accuracies of 12 device types were computed with different training and testing data sizes, as shown in Figure 3. It is also found that the classification accuracy varies with different device types. It is found that the accuracy is maximum when 80% of training data and 20% of testing data is considered. The classification accuracies for the same are shown in Figure 11.

Also, the classification accuracies of different models, namely DBN, ANN, SVM and KNN, are computed for various device types considered in the work. Figure 13 shows the classification accuracies computed for DNN, DBN, ANN, SVM and KNN models. It is found that the DNN model gives better results when compared with other models.

5. Conclusion

Clue-Based Search Engine is an effective methodology for identifying devices in IoT. The RFO-based Deep Neural Network is used to identify the objects connected to the network. The classification accuracies are computed for different models, and it is found that the proposed model gives better results when compared with the other models. The proposed work finds applications for additional investigation in detecting malicious behavior due to security flaws in the smart environment using a trajectory-based search engine.

References

[1] Javad Pashaei Barbin, Saleh Yousefi, and Behrooz Masoumi, “Navigation in the Social Internet-of-Things (Siot) for Discovering the Influential Service-Providers Using Distributed Learning Automata,” *The Journal of Supercomputing*, vol. 77, no. 10, pp. 11004-11031, 2021. [CrossRef] [Google Scholar] [Publisher link]

[2] Wail Mardini et al., “Mining Internet of Things for Intelligent Objects Using Genetic Algorithm,” *Computers & Electrical Engineering*, vol. 66, pp. 423-434, 2018. [CrossRef] [Google Scholar] [Publisher link]

[3] Abderrahim Zannou, Abdelhak Boulaalam, and El Habib Nfaoui, “Relevant Node Discovery and Selection Approach for the Internet of Things Based on Neural Networks and Ant Colony Optimization,” *Pervasive and Mobile Computing*, vol. 70, p. 101311, 2021. [CrossRef] [Google Scholar] [Publisher link]

[4] Nickolaos Koroniotis, Nour Moustafa, and Elena Sitnikova, “A New Network Forensic Framework Based on Deep Learning for Internet of Things Networks: A Particle Deep Framework,” *Future Generation Computer Systems*, vol. 110, pp. 91-106, 2020. [CrossRef] [Google Scholar] [Publisher link]

[5] Javed Ashraf et al.,“Novel Deep Learning-Enabled LSTM Auto Encoder Architecture For Discovering Anomalous Events From Intelligent Transportation Systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4507-4518, 2021. [CrossRef] [Google Scholar] [Publisher link]

[6] Sun-Kuk Noh, “Recycled Clothing Classification System Using Intelligent IoT and Deep Learning with AlexNet,” *Computational Intelligence and Neuroscience*, 2021. [CrossRef] [Google Scholar] [Publisher link]

[7] Didier Coquin et al., “Assistance via IoT Networking Cameras and Evidence Theory for 3D Object Instance Recognition: Application for the NAO Humanoid Robot,” *Internet of Things*, vol. 9, p. 100128, 2020. [CrossRef] [Google Scholar] [Publisher link]

[8] Sachin Kumar, Prayag Tiwari, and Mikhail Zymbler, “Internet of Things is a Revolutionary Approach for Future Technology Enhancement: A Review,” *Journal of Big Data*, vol. 6, no. 1, pp. 1-21, 2019. [CrossRef] [Google Scholar] [Publisher link]

[9] Kashif Naseer Quresh et al., “Neuro Computing for Internet of Things: Object Recognition and Detection Strategy,” *Neurocomputing*, vol. 485, pp. 263-273, 2022. [CrossRef] [Google Scholar] [Publisher link]

[10] Hyoduck Seo et al., “Multi-Sensor-Based Blind-Spot Reduction Technology and a Data-Logging Method Using a Gesture Recognition Algorithm Based on Micro E-Mobility in an IoT Environment,” *Sensors*, vol. 22, no. 3, p. 1081, 2022. [CrossRef] [Google Scholar] [Publisher link]

[11] Fang-Qi Li et al., “Online Intrusion Detection for IoT Systems with Full Bayesian Possibilistic Clustering and Ensembled Fuzzy Classifiers,” *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 11, pp. 4605-4617, 2022. [CrossRef] [Google Scholar] [Publisher link]

[12] Manasa R, K Karibasappa, and Rajeshwari J, "Autonomous Path Finder and Object Detection using an Intelligent Edge Detection Approach," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 8, pp. 1-7, 2022. [CrossRef] [Publisher link]

[13] Jie Tang et al., “Enabling Deep Learning on IoT Devices,” *Computer*, vol. 50, no. 10, pp. 92-96, 2017. [CrossRef] [Google Scholar] [Publisher link]

[14] R. Raghu Nandan, N. Nalini, and Prasad Naik Hamsavath, “IoT-CBSE: A Search Engine for Semantic Internet of Things,” *Emerging Research in Computing, Information, Communication and Applications, Springer*, Singapore, pp. 265-271, 2022. [CrossRef] [Google Scholar] [Publisher link]

[15] Esmaeil Mirmahdi, and Omid Ghorbani Shirazi, "Installation of Suitable Sensors for Object Detection and Height Control on Combine Harvester," *SSRG International Journal of Mechanical Engineering*, vol. 8, no. 5, pp. 12-19, 2021. [CrossRef] [Google Scholar] [Publisher link]

[16] Yunlong Gao et al., “A New Robust Fuzzy C-Means Clustering Method Based on Adaptive Elastic Distance,” *Knowledge-Based Systems*, vol. 237, p. 107769, 2022. [CrossRef] [Google Scholar] [Publisher link]

[17] Dawid Połap, and Marcin Woźniak, “Red Fox Optimization Algorithm,” *Expert Systems with Applications*, vol. 166, p. 114107, 2021. [CrossRef] [Google Scholar] [Publisher link]

[18] Sujata Chaudhari et al., "Yolo Real Time Object Detection," *International Journal of Computer Trends and Technology*, vol. 68, no. 6, pp. 70-76, 2020. [CrossRef] [Publisher link]

[19] Hwanjun Song et al., “Learning From Noisy Labels with Deep Neural Networks: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [CrossRef] [Google Scholar] [Publisher link]

[20] [Online]. Available: https://www.kaggle.com/code/annamariamandalari/iotlab/data

[21] Mustafa Cakir, and Akhan Akbulut, “A Bayesian Deep Neural Network Approach to Seven-Point Thermal Sensation Perception,” *IEEE Access*, vol. 10, pp. 5193-5206, 2022. [CrossRef] [Google Scholar] [Publisher link]

[22] T. Ananth Kumar, K. Suresh Kumar, and S Jaisiva, “IoT based Leukemia Detection using Fuzzy C-means clustering Technique,” *Global Journal on Innovation, Opportunities and Challenges in AAI and Machine Learning*, vol. 6, no. 1, 2022. [Google Scholar] [Publisher link]

[23] Wasseem N. Ibrahem Al-Obaydy et al., “Document Classification Using Term Frequency-Inverse Document Frequency and K-Means Clustering,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 3, pp. 1517-1524, 2022. [CrossRef] [Google Scholar] [Publisher link]

[24] Fahrettin Horasan, “Latent Semantic Indexing-Based Hybrid Collaborative Filtering for Recommender Systems,” *Arabian Journal for Science and Engineering*, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher link]