Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6876688, 16 pages
https://doi.org/10.1155/2021/6876688

Research Article

Fog Big Data Analysis for IoT Sensor Application Using Fusion Deep Learning

Anand Singh Rajawat,[1] Pradeep Bedi,[2] S. B. Goyal ,[3] Adel R. Alharbi,[4] Amer Aljaedi,[4] Sajjad Shaukat Jamal ,[5] and Piyush Kumar Shukla [6]

[1]*Department of Computer Science Engineering, Shri Vaishnav Vidyapeeth Vishwavidyalaya Indore, Indore, India*
[2]*Lingayas Vidyapeeth, Faridabad, India*
[3]*City University, Petaling Jaya, Malaysia*
[4]*College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia*
[5]*Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia*
[6]*Computer Science & Engineering Department, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, (Technological University of Madhya Pradesh), Bhopal 462023, India*

Correspondence should be addressed to Piyush Kumar Shukla; pphdwss@gmail.com

Received 9 July 2021; Accepted 14 September 2021; Published 20 October 2021

Academic Editor: Vijay Kumar

The IoT sensor applications have grown in extreme numbers, generating a large amount of data, and it requires very effective data analysis procedures. However, the different IoT infrastructures and IoT sensor device layers possess protocol limitations in transmitting and receiving messages which generate obstacles in developing the smart IoT sensor applications. This difficulty prohibited existing IoT sensor implementations from adapting to other IoT sensor applications. In this article, we study and analyze how IoT sensor produces data for big data analytics, and it also highlights the existing challenges of intelligent solutions. IoT sensor applications required big data classification and analysis in a Fog computing (FC) environment using computation intelligence (CI). Our proposed Fog big data analysis model (FBDAM) and BPNN analysis model for IoT sensor application using fusion deep learning (FDL) pose new obstacles for potential machine-to-machine communication practices. We have applied our proposed FBDAM on the most significant Fog applications developed on smart city datasets (parking, transportation, security, and sensor IoT dataset) and got improving results. We compared different deep and machine learning algorithms (SVM, SVMG-RBF, BPNN, S3VM, and proposed FDL) on different smart city dataset IoT application environments.

1. Introduction

IoT sensor systems are limited due to their processing capacity and network bandwidth. On the other side, smart apps need vast data and computational power for deep learning (DL) based research. Therefore, present smart applications respond to certain constraints by offering deep learning (DL) research at the gateway or cloud. Applications need input from the person or other smart machines involving duplex communication. Additionally, input is processed by computational intelligence (CI) algorithm that needs more computation time than restricted devices can accomplish. Therefore, IoT systems [1, 2] have a restricted capacity to understand, develop, and share information autonomously. IoT implementations use ontologies, which are databases that characterise items by their properties and relationships, to overcome their shortcomings. This enables intelligent application systems by allowing IoT solutions to find inference rules or information built into other machines or the cloud. Adopting one Fog computing to another sector has proven difficult, given the widespread use of Fog computing [3] to construct smart applications. When using deep learning from another domain, an IoT application's ability to semi-identify the context of messages deteriorates.

Thus, hundreds of thousands of data records were obtained from various data sources in various aspects, such as Smart Cities [4] datasets, including development on smart city datasets (parking, transportation, security, and sensor IoT dataset), and several data preparation and preprocessing processes were carried out before the key testing phase of the proposed algorithm. We described that F2C data management (from distributed to centralized) has a great possibility to handle all data life stages (from creation to conception) concerning the DLC concepts. We contributed to different smart city scenarios to demonstrate our proposed big data architecture [5] for the smart cities. IoT sensor applications required big data classification [6] and analysis in Fog computing environment using computation intelligence. We proposed a new deep learning-based Fog big data analysis model (FBDAM) for classification of IoT application generated big dataset. To perform the simulation, we used a Hadoop framework based spark tool, and we compared different deep and machine learning algorithms (SVM, SVMG-RBF, BPNN, S3VM, and proposed FDL) on different smart city dataset IoT application environments.

The organization of the article is as follows. Section 2 examines the related research about Fog big computing data analytics model and compares key differences between deep learning techniques based on Fog big data analysis for IoT sensor application. Section 3 describes basic Fog computing technique. Section 4 explains the Fog computing environment for IoT sensor big data processing. Section 5 describes the proposed methodology. Section 6 presents results' analysis. Section 7 describes conclusion and future work.

2. Related Work

Several types of research have been shown in the arena of large data manipulation and determining Fog computing with big data analysis in smart cities. In this research work, the recent research studies will be reviewed and discussed, showing the advantages and good features and debating the drawbacks and weak points to be taken as a reference and utilized to the proposed model. Context-based offloading [7] is used to satisfy the performance needs of IoT-enabled services. The study [8] gave an overview of an e-health monitoring system in the context of Fog computing development and testing. The analytic network process (ANP) [9] was used to identify and rank FC-based IoT for health monitoring systems in the suggested research. Fog computing [10] is an enhancement of the edge of the network's cloud computing resources to reduce latency and network congestion. These VMs may sustain malware attacks or a device malfunction from the physical server storage, which leads to services and resources becoming inaccessible. Therefore, a computational smart live copying precopy method for VM replication is given which calculates the downtime after each iteration to decide if a device malfunction or an attack on a Fog machine node can proceed to the stop-and-copy point. It would reduce downtime and conversion time and ensure that the end-users of Fog computing have access and infrastructure and support. The research in [11] offers a DL framework for quickly and accurately diagnosing pneumonia illness. For extracting relevant features from chest X-ray pictures, several deep convolutional neural network (DCNN) transfer learning algorithms such as AlexNet, SqueezeNet, VGG16, VGG19, and Inception-V3 are used. The Internet of Things (IoT) is built to connect billions of intelligent devices to the Internet, which will provide smart cities with a bright future. The authors address weakness through data analytics to push processes for the discovery of knowledge towards the limits. Edge devices do, however, have limited computing power. Inherited strengths and weaknesses make this difficult. The authors of [12] identified the possibilities and problems posed by the use of Fog computation in IoV environments for real-time ITS big data processing. After the shortcomings of the previous work linked to the IoV, smart computing, and real-time big data analytics are established, a three-dimensional device design would be introduced. Xu et al. [13] gave an overview of Fog architecture, a hierarchy of different layers, applications, challenges, and research directions. Over the past few years [14], autonomous devices and sensors have increased significantly with IoT implementations, providing a broad range of multimodal and heterogeneous big data (BD) results. Fog computing [15] provides the tools for edge machines and the cloud data centre, a developing concept. It is powered by the need to process huge data from the Internet with low latency at high speed and broad quantities. Fog computing [16] is a specific software extension that puts out a few essential operations at the edge of the consumer and leaves the remainder of the system. Owing to the unique conditions of most IoT implementations, many of these problems posed by cloud computing are addressed further. The ultimate purpose is to build intelligent digital [17] apps that are independent and willing to make good decisions based on big data. Table 1 provides differences between deep learning techniques based on Fog big data analysis for IoT sensor application.

3. Basic Fog Computing Technique

Smart cities are predictable cities, and because of urbanization, they need to handle the data in these smart cities rapidly. To be able to test for multiple purposes, data should be appropriately handled. The smart city data collection and analysis method is very complicated. The key aim of cloud computing is to retain these massive files, but it also has many drawbacks. Many options for cloud storage can be found. The Fog data [23] access is very easy to Fog as opposed to a server. Fogging is a robust cloud processing tool for cloud computing, bringing cloud capabilities down to the ground, i.e., from end-user to the source when data access and storage are required. When talking about the alternatives, low Internet speed and bandwidth make accessing data from the cloud tougher, which leads to difficulties accessing it. For both of these problems, fogging will be the perfect remedy. A detailed description of fogging is given in the following diagram. In large-scale modeling, there are several obstacles to applying Fog computing. The advantages of the suggested plan should be measured. The Fog computing solution may be costly in many places. The Fog can be

Table 1: Key differences between deep learning techniques based on Fog big data analysis for IoT sensor application.

S. no.	Citation	Deep learning technique	Big data analysis	IoT sensor application	Highlights
1	[18]	Fine-tuning AlexNet	Image data processing	Geologic hazards	Determining the state of the landslide
2	[19]	Deep learning regression prediction model	Features of the sites to be assessed in terms of signal strength	Wearable device	Deep learning-based indoor locating algorithm for wearable devices
3	[20]	Mixed-data design	Fog error compensation	Analysis of error theory	Optimization techniques for error compensation
4	[21]	Intelligent mapping algorithm	Principles of Fog and edge computing	IoT data-intensive processes can be automated	IoT data-intensive processes can be automated
5	[22]	Smart algorithm	Edge, Fog, and cloud IoT devices' processing data	Streaming IoT data	Smart parking system based on IoT in the real world

included as a new technological layer. To comply with privacy problems, data protection can also entail changes. Fog computing should also be part of a comprehensive data policy to respond to basic issues such as what data should be gathered and how long the information can be kept. Fog computing is likely to be described as balancing services. As Fog computing could be tailored to various applications, adjusting the effective method based on application specifications would be necessary. Fog engine can be run with a battery to conserve electricity, parking, transportation, security, and sensor IoT dataset [24]. The performance of the Fog computing analysis data is big data, so that classification is especially important when large numbers of them are deployed. Lastly, the management of resources is a difficult task for Fog engine.

4. Fog Computing (FC) Environment for IoT Sensor Big Data Processing

Resource management can be structured and spread in areas where Fog computing (FC) is the first step of big data processing (BDP), and the rest is performed in the cloud. Consequently, a resource manager would deal with resource allocation for BDP with appropriate efficiency and expense in Fog computing [25]. Big data can be very useful in edge computing and in cloud-based applications for analysis. Data can be easily examined in the Fog due to the saturation of its approach and data can easily be observed from the direction and the position of the Fog. There is plenty of evidence that Fog computing helps format and sort data. We can provide a lot of protection for people. Big data can be helpful in edge calculations and in cloud-based computing for the processing of knowledge. Data can easily be processed in the Fog due to the propagation, and the data can be recorded and transmitted easily. There are several proofs that FC can aid in the encoding and processing of data. We provide people with much protection. In this set of data, there are many difficulties.

The challenges may include protection of the IoT environment [26], innovations that seek to improve social and economic conditions in society (parking, transportation, security, and sensor IoT dataset), and decision-making respected by all. We should develop a smart city, and overcoming these problems will serve to boost citizens' social lives. This model aims to provide enhanced data calcification and analysis of today's intelligent cities through different methods for gathering, incorporating, and analysing data to better citizens' lives. We use the pattern mining algorithms such as SVM (support vector machine), SVMG-RBF (radial basis function kernel), BPNN (backpropagation neural network), and S3VM (semisupervised support vector machines). We have examined several challenges at data, model, and device level (Fog computing environment (FCE)) [27] to explain the definition of large volume of big data. From the above literature survey, researchers face the following challenges in big data:

Analytics architecture: the architecture of big data analytics does not give a proper solution to understand. This is a large area of study.

Statistical significance: when evaluating big data, an important issue is whether the statistical significance of the calculated coefficients should be recorded at the 1 percent level instead of the more traditional 5 percent level. It has been known that rejecting the null hypothesis of no statistical significance becomes "too simple" when using large data.

Heterogeneity of data: As we know, big data sources such as Smart Cities (parking, transportation, security, and sensor IoT dataset) [28] and social media are composed of different types of datasets. Filtering out useful data from a large set of data is a challenge in this area.

5. Proposed Methodology

A city has three important dimensions given as follows: the technologies applied in the city, citizens and other people living in the city, and communities running in the city. Depending on the functionality level of these three dimensions, a city can be defined in many ways, e.g., digital city, ubiquitous city, creative city, and smart community city [29]. The smart city has a high level of commitment and functionality of the above three dimensions. The proliferation of human beings is a serious issue for the ruling government in all countries worldwide. Governments have to face the challenges in providing the resources to inhabitants at economical prices without shortage and maintaining the supply regarding demands (parking, transportation, pollution, and sensor IoT dataset). It also has to be perceptive to the environment and avoid wastage juxtaposed with optimum utilization of the available resources. Maintaining security standards and managing the increasing traffic rush on the roads through better manageable techniques are considered. In urban development plans, it is proposed to bring information and communication technology (ICT) to improve the quality of life. ICT enhances urban services' quality and performance, reduces resource consumption [30] and its associated cost, and establishes a healthy and fruitful contact between government and citizens—Fog big data analysis for IoT sensor application using fusion deep learning. The integration of ICT in the urban development plan has introduced smart cities [31]. All cities (parking, transportation, pollution, security, and sensor IoT dataset), etc. will be managed by applying the modern technological solutions, Hader. City at the major level would be keeping up with the track and maintaining affordability at the individual level. The big data analysis process is shown in Figure 1. In Table 2, we explore the formulation of an existing framework solution for the city after implementing Fog computing.

5.1. IoT Data Analysis and Collection Life Cycle. Despite the growth in edge and Fog resources, a standard abstract or runtime programming environment has not been developed to describe and run distributed IoT applications on these resources. A hierarchical pattern for the composition of applications, generating edge data and progressively aggregating and processing on the Fog and cloud layers, has been established for preliminary work. The spatial partitions are used to delegate boundary instruments to Fogs. The architecture provides an application data flow model for clustered runtime engines on fog and cloud services. A declarative design specification and big data framework are required to simplify design construction in these dynamic environments. System implementation and resource management are also linked to this. VMs are used to configure the appropriate environment in the cloud, but can prove too resource-intensive for Fog. Some also looked at utilizing just a subset of the VM's footprint on the Fog and migrated this representation through infrastructure to map the accessibility of users using its services. Figure 2 describes IoT data analysis and collection life cycle. Research of this type needs to be updated as the architectural models, and Fog computing implementation become clearer with tools providing new challenges for mobility, affordability, and energy uses.

5.1.1. Fog Layer 1. The end-user and pilot IoT equipment are adjacent ground. In the Fog region (includes various building types and their neighborhoods) and in Fog Geräte (it is the most robust), this layer is a multitude of different IoT sources, including sensors and smartphones. As seen below, this layer can perform many functions in big data analysis.

5.1.2. Storage Type. Fog layer 1 stores the data in real time. In data management architecture, this layer is part of the framework for distributed data processing.

5.1.3. IoT-Hub in Fog Layer 1. IoT-Hub is the largest node for data analysis and storage. In addition, this layer is in the city of the operator but not near IoT devices [32] such as Fog layer 1. This layer can perform many big data processing activities as follows.

5.1.4. Data Type. Fog layer 2 is the location where the most current data are stored. This layer again takes the hierarchical data management architecture in mind.

5.1.5. DLC Model. IoT-Hub [33] is responsible for high-level activities under blocks for data storage and analysis (medium color). The cloud is also responsible for specialized computing and storage functions. In comparison, the data collection block has less liability (lighter color level) than the lower layer since the data sources are smaller than the lower layer. In a dominant position is the cloud layer. The cloud accumulates with the most efficient computing and storage tools. Data type: Fog layer 1 includes all relevant data storage facilities. As shown below, this layer can coordinate multiple tasks for handling big data. Figure 3 describes and explores the convergence of IoT architecture and big data analytics.

Form of data: the cloud is responsible for storing historical records.

Architecture for data processing: this layer is the foundation for structured data storage.

DLC model: the cloud technology has almost unlimited resources for all inquiries of data. Then, all related tasks will be done in the cloud environment (darker color level).

5.2. FBDA Model. It is a model based on FBDA for IoT, deep learning, and information exploration. The inadequacy of adaptive learning machines becomes the bottleneck in growing intelligent IoT computer systems, despite the pervasiveness of IoT applications [34]. Nothing has been done to look into how intelligent IoT device systems will exchange information on their own as data come in from diverse settings or case studies. As a result, this section proposes a paradigm for how sensor devices should autonomously communicate knowledge, develop new knowledge (fusion-based deep learning), and alter knowledge to be extended across domains or case studies. The paradigm is based on how people research, identify, and analyse data. First, this section goes over the various case studies that have been linked to smart IoT big data analytics.

An examination of the numerous case studies that used big data analytics to solve IoT problems using fusion deep learning techniques, as well as a study of smart big data analytics for IoT applications, is conducted. There are numerous domains based on which this study work is divided: parking, transportation, pollution, security, and sensor IoT dataset. Issues in the smart cities domain are having the best traffic path, anticipating waste bin filling trends for

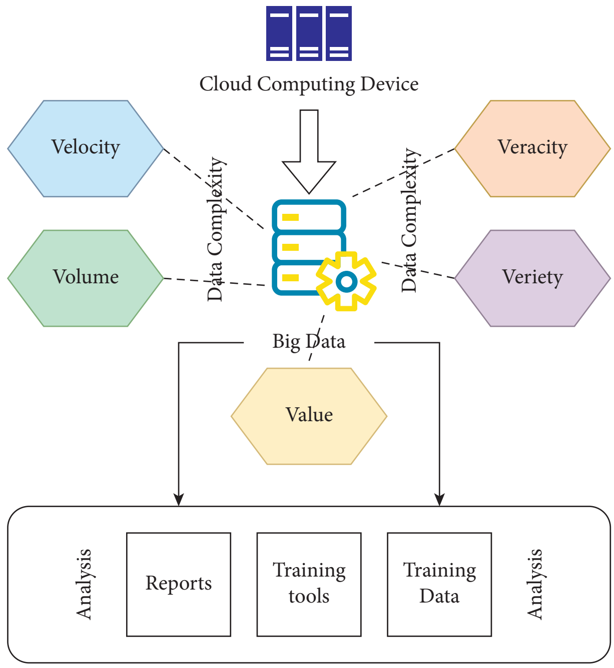

FIGURE 1: Big data analysis process.

TABLE 2: Formulation of an existing framework solution for the city after implementing Fog computing.

Existing framework	Working
FS-1	It would include controlling traffic lights as per the traffic density and help create a diversion in a heavy traffic rush situation
FS-2	It would consider and cover up all public places prone to theft and misshappening, thereby maintaining security standards and providing a high-end care-free experience
FS-3	It would sort out the cavalier issue of power theft by implementing a dual metering system

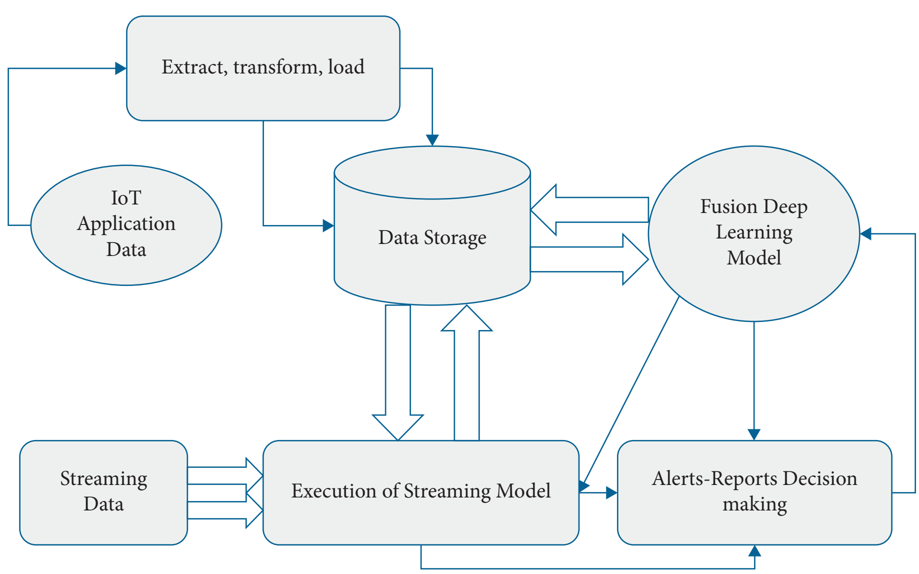

FIGURE 2: IoT data analysis and collection life cycle.

collection, offering a location-based product, and predicting smart metre energy usage, just to mention a few. Back-propagation neural networks were used to evaluate a building's past energy use as well as environmental variables such as temperature and humidity, resulting in a projected energy use difference of 1.7 kWh in a 30-day window

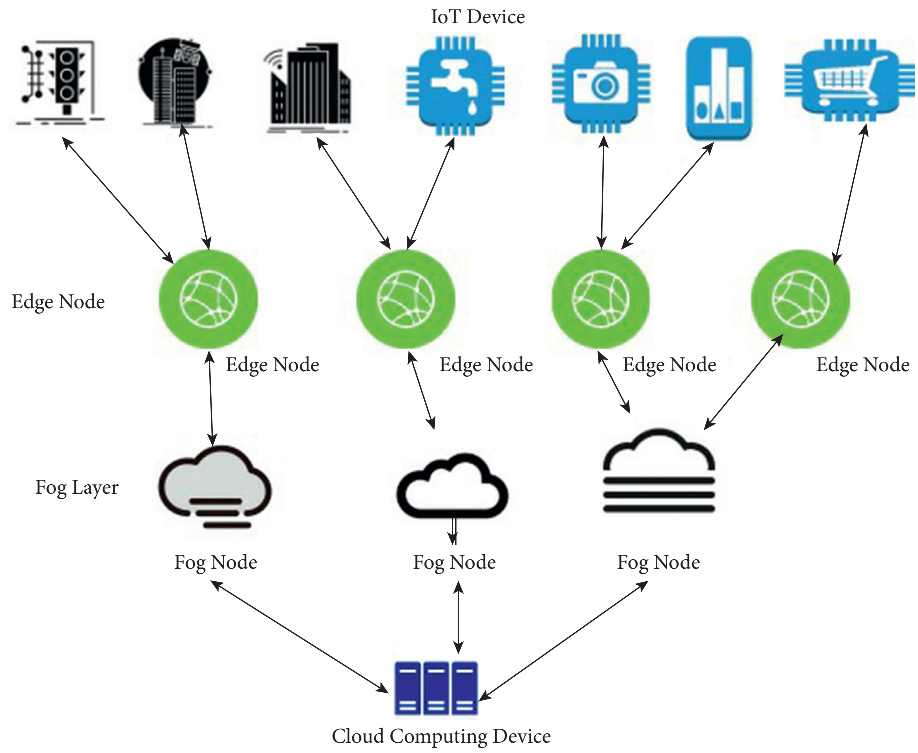

Figure 3: IoT architecture and big data analytics converge.

between real and predicted consumption. Active, regular, and cow-dormant behaviours were identified using deep learning algorithms. Parking, transportation, pollution, security, and sensor IoT dataset should be considered a bridge between the smart city and manufacturing domains, as their concepts can be applied to any domain. The wellbeing sector could be the most common implementation when IoT sensors are connected to individuals, e.g., estimating a workplace degree of thermal comfort and detecting when an individual falls at home.

The Fog engine, which primarily hires low-end instruments, is combined with the IoT because

(a) It is straightforward and flexible.

(b) Fog engine applied to IoT systems should not negatively affect the current machine. The Fog motor consists of the following 3 components:

  (i) A preprocessing data collection and storage unit (parking, transportation, pollution, security, and sensor IoT dataset), for both analysis and storage of data

  (ii) A messaging and networking platform composed of peer-to-peer network connections and cloud and IoT connectivity

  (iii) An orchestrating device in which the Fog motors and the cloud are synchronized

In this research, we have explored ways to increase the parking, transportation, pollution, and security in cities and use emerging technology to build sustainable communities. Compared to the cloud, data can be processed more effectively in Fog-based computing. It also contrasts cloud models to enhance data storage. It also differentiates between Fog and cloud computing, which can be rendered based on certain functionality that stresses the safety concerns accompanying cloud Fog. Finally, we believe that we can make smart cities by using Fog computing technologies.

5.3. BPNN Analysis Model. Deep learning [35] is an intelligent data processing and classification technique. This technique aims to evaluate the concept of human brain building and its purpose. This is a self-organizing, robust, noise struggle type of featured technique. The complex modeling process is not essential in the training stage and in predicting enhanced consequences for the unidentified model. However, among them, backpropagation (BP) neural network is extensively used in prediction and classification. BPNN is one of the major module analysis techniques that can classify real-time data with flume (parking, transportation, security, and sensor IoT dataset) in the smart city domain. The BP network approach is intelligent to analyse the IoT sensor application's real-time data and acquired excellent consequences.

The proposed algorithm is given in Algorithm 1.

Figure 4 describes the BPNN analysis model. The major stage to create BP neural network classification has some requirements, but not limited to training network. Training network has the ability of self-learning and grouping through training samples. The training stages of the BP neural network are as follows:

IoT sensor big data can be categorized by BP neural network classification.

Step 1: starting the network weights, learning rate, and threshold error. Make iterations 0.
Step 2: open the image training set file.
Step 3: total error = 0; iterations+1
Step 4: input one vector from the file to the layer units.
Step 5: initialize the vector's goal output.
Step 6: determine hidden layer unit outputs.
Step 7: determine output layer units.
Step 8: input = desired output–actual output
Step 9: total error + error
Step 10: measure the output layer's delta sigma and then change the hidden layer's weights.
Step 11: tend to your hidden layer units' delta sigma and alter the weights among them.
Step 12: if the file has more vectors, move to step 4.
Step 13: stop if threshold error > total error.

ALGORITHM 1: Steps in the BP simulation program.

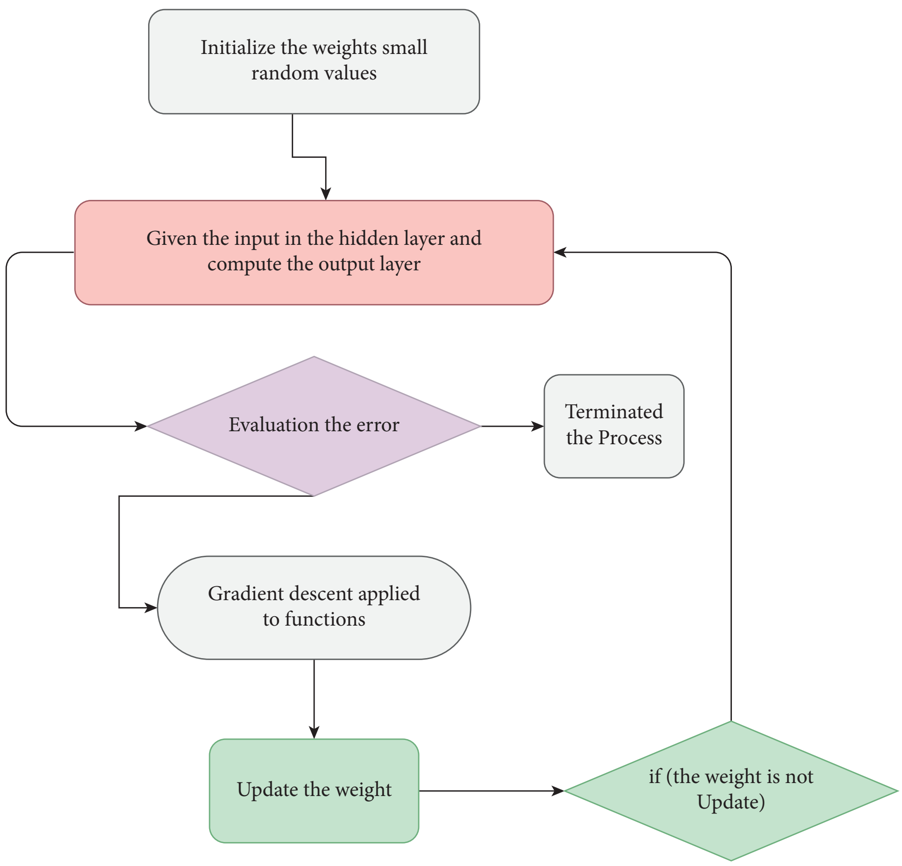

FIGURE 4: BPNN analysis model.

This classifier, the parking, transportation, security, and sensor IoT dataset, can be categorized efficiently with training data accounting for 75.2% and the testing data accounting for 30.6%.

This, in turn, increases the accuracy factor that can effectively progress the BP network performance. When the accuracy factor is reduced, the error and noisy data of fusion become faster.

By accepting suitable numerous learning rate, the BP classifier can also decrease the target error and preserve accuracy.

The BPNN has a sequence of nonlinear and self-learning structures, which are extensively used to resolve dissimilar, cost-effective, or even collective difficulties. In particular, the BPNN preserves three-layer construction and is significantly preferred by the common or knowledgeable. It has been established to explain several nonlinear difficulties. However, there are many characteristic problems in the traditional BP network technique, such as simply reducing interest in a local minimum, no speculative supervisions, low possibility to consider as a special, the hidden layer node number, and no technique to receive the inherent illustration design consequences.

It is acceptable that numerous researchers and specialists have anticipated several approaches to resolving these problems, such as increasing information extraction [28], simulating an adaptive learning rate, and adopting learning and extraction and other approaches to increase the convergence accuracy and incredulous of the local minima difficulties. However, there are no resolutions available for classifying the hidden layer nodes quantity and receiving training consequences due to the old samples to analyse that approach and proposed fusion deep learning-based BNNN in the field of artificial neural networks.

5.4. Big Data Analysis Algorithm. For analysis of big data, our proposed technique is used. Using our proposed models to perform the training, the split approach divided the data into subsets. The datasets are then distributed among various system nodes in both ways, i.e., globally and locally. Using both global and local levels, the computation can be performed to perform the task in two ways. The first one is to execute the training function that divides the things into subsets and also forms the global subsets.

The other one is a reducer that can merge the results produced by the mapper. The obtained new support vectors are gathered with the global reduced vectors in phases. This can be explained in a precise and clear manner through this algorithm.

We represent the symbols that are used for various terms in our algorithms as follows: x represents the number of iteration, l represents the size of web-produced functions, h_x symbolizes the theory at iteration x, D_x symbolizes subsets in datasets, SV_G represents the support vector as a global, and SV_l represents the support vector through SV and l for the computer node.

Step 1: on the initial level, SV_G is a global support vector.

Step 2: $X = x + 1$.

Step 3: for different nodes, the result obtained is merged with the subsets of training.

Step 4: multilevel classifier performs the training using fusion deep learning for merging sets.

Step 5: obtain the support vectors.

Step 6: on completion of the training phase, get all SVs to be merged and save the output.

Step 7: terminate all conditions if results are not obtained. Then, transfer it to the second step again.

5.5. Fusion Deep Learning (FDL) Model. Deep learning is a subcategory of machine learning techniques. Deep neural networks are conventional networks of algorithms that keep different records in precision for numerous significant problems, including analysis and recommender methods. Its technical term is fusion deep. This is used as several layers occurred in a neural network. A narrow network takes a unique invented hidden layer, and a deep network has a number of hidden layers. Deep neural networks permit multiple hidden layers to learn remarkable information features in a fusion, since simple features (input training set) fuses from one layer to the subsequence layer, a method and other additional complex features, the fusion type of deep learning involves added approaches to deliver better outcomes. Fusion deep learning [36] involves a category of methods that strives to adopt main internal points of data that can take part with similar internal neural network connectivity, working instinct-wise and conceived than three different layers. Figure 5 represents the fusion deep learning model. Under unobserved making, one learning method involves the interconnectivity working with proper organized main layer, and in a primitive time, it is converted in a well-observed method [36]. Under this process, highly positioned aspects can understand the following low-level ones, anyhow the required aspects are distinctly addressed to design in specific categories in the accomplishment consequently.

Fusion deep structure (FDS) [36] is designed probably to increase the extra constituent scholar, unique aspects at peak levels and outside added intelligent aspects in a different form to a very far distant local ups and downs of the input. Allowing for particular current research, deep models can improve nonlinear functions' evaluation than deep representations, which is outstanding according to its capability for accomplishing the maximum performance for numerous responsibilities [37].

A deep convolutional neural network (DCNN) can be created, a further inclusive of the modern deep learning training canister. Among such training, DCNN is a deep discriminative design for the DNN class, which has advanced performance on numerous responsibilities and cooperation in data classification, analysis, and prediction [38]. In DCNN, every component contains a compute layer and a fusion layer. These components are frequently weighted up through one on a maximum of additional to the deep Fusion model (FDM) method of a deep fusion model (FDM).

The compute layer segments various weights, and the fusion layer subsamples the output of this compute layer, which decreases the data percentage as the lower layer [38]. However, DCNNs have been presented through optimistic consequences aimed at the classification responsibilities in many applications. Its residues can be identified in a way it achieves on an extremely imbalanced dataset. Subsequently, in this research, a method for successful DCNNs is directed at imbalanced data cataloguing and additional facts to extend it for improved performance [19]. To be proposed to modify CNNs by applicable fusion it through a backpropagation neural network method can convulsion the high-class features of CNNs.

Dissimilar to the backpropagation neural network in that fusion approach (adaptive selection and random sampling) for discovery of the applicable negatives, our proposed FDL technique fusion can perform oversampling with the decision fusion to boost convolutional neural network

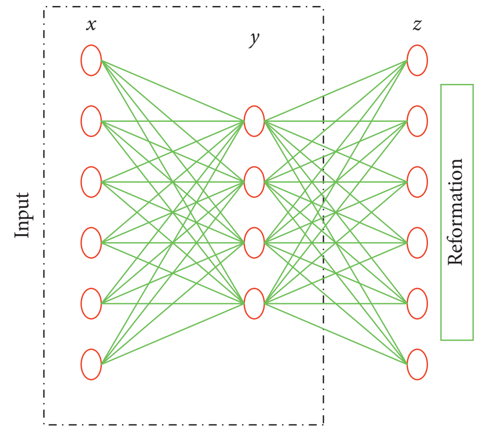

Figure 5: Fusion deep learning for data-specific category (IoT sensor applications data) of one layer for row data-specific categorization. The model achieves a conceived certain aspect "y" since input "x" by advancing it on "z." Steady acceptances are made vital in the network.

performance on row data (IoT sensor applications data) classification through or deprived of imbalanced data disseminations [39].

Proposed FDL models that alter multilayer perceptions are intended to make use of minimal volumes of preprocessing constructed on binary ideas. The major idea is to confine the connections among the input units and hidden units so that every hidden unit attaches to a minor subsection of the input units [40]. This approach of taking nearby associated networks is likewise to the fascinations of the incentive after the biological discovery that neurons have in the visual cortex consumed localized reachable arenas. For additional knowledge, this can be used to decrease the compute complexity in data with the natural dataset having the belongings of existence fixed [41, 42]. This means that the numbers of the unique quantity of the information are similar to several additional parts. As a result, we can yield the features learned by completing some insignificant reinforcements that are arbitrarily sampled from a big data and convolute them to find a dissimilar feature activation value on every data site. Obtaining the features expending difficulty, we can follow the straightforward procedure of their fusion statistics for classification [43]. The fusion statistics are considerably lower in dimension (associated with completion of the mined features) and can similarly advance certain consequences (less overfitting).

It consists of a checked layer of inputs, a remote hidden layer of sections, and single advancement layer of M units, apart from a running function. By completing one learning process, its first strategy is to aid the Hidden layer and increase the hidden training [44]. The interconnectivity working till now is stable and regular, and after this, it is known as computing. At very first, it is outlined by an output layer that contains the same expansion of the input layer; this is known as modernization. The revised significance is introduced precisely. These two stages can be distinctively pointed out as follows:

$$\begin{aligned} Y &= f\left(M_Y X + b_Y\right), \\ Z &= f\left(M_Z Y + b_Z\right), \end{aligned} \tag{1}$$

where input shows the importance of the hidden layer to generate weights; on the other hand, it shows the vitality of neutrality of hidden apart from output parameters. Hence, it makes true importance of the activation processing. Besides, the causes and their effect relation are examined. There are placements of alternatives targeted at a sigmoid process, hyperbolic refraction, and improved linear process. The following constraint is having weight (M):

$$M_Y = MZ' = M. \tag{2}$$

We can know that this has protected weights, which help to split model measurements. As an unwanted result, three sections of measurements were reluctant to get knowledge [45, 46]. The aim of making someone learn is to remove the incorrectness among input and advanced.

$$\arg M, bYbZ\left[c\left(X, Z\right)\right]. \tag{3}$$

This is very considerable for analysis though it is particular. It is very considerable, required on the analysis though, it is particular also to be checked for the mistakes which can be peculiar in a making choice off.

The three layers in this model are described in Figure 5: the input layer, the hidden layer, and the output layer. As a result, the weight in the advancing rule can be denominated as (somewhere denoted as the learning rate ξ)

$$\begin{aligned} M &= M - \xi\nabla\left(X, Z\right)\nabla M, \\ b_Y &= b_Y - \xi\nabla\left(X, Z\right)bY, \\ b_Z &= b_Z - \xi\nabla\left(X, Z\right)bz. \end{aligned} \tag{4}$$

Simultaneously, after having some input as training values to the input layer, having provided weight on that, and modifying the same on the output layer, it was given the feature value to provide input to the hidden layer and perform its functions. Now, the result produced on the hidden layer having the correct value is recommended for the output layer. Otherwise, it will be again preferred to be the training value unless we are not finding the desired or expected values out of that. With the help of new rules observed from the training set, we can produce the result as per our expectations. The input data, which were actually in the form of data, are used by users and sent to the hidden layer for training and processing to yield results. It was primitively as row stuff that eventually became the right calculation of desired value on the hidden layer for finding it on the output layer. Although this model signifies how much precise input we are providing, the same perfect result we obtain is a training set.

Earlier, we were getting that the nonlinear values would be in the form of linear values. The classification will no uncertainty be enhanced by the weight value and biases computation to perform upcoming of the best-categorized stage or level. On the above training set generation, the information loss is lessened. This methodology can resolve the complexity of consistent big data that we were having through different and varied resources used by the user. That is why we have applied deep fusion learning. The main objective is to analyze the sensor data through deep learning features.

5.5.1. FDL-Fog Computing Environment (FCE). FBEBDBA-based analysis algorithm, which is used for the training for a set of T training samples, different parts for L news event, Q is a systematic established in closing one sample per IoT sensor applications event data classification, r is denoted as a feature for extraction algorithm, β and λ are parameters of the fusion based enhancement big data behavior analysis (FBEBDBA) for IoT sensor applications event data classification algorithm, and pre-eminent algorithm is an systematic set that encompasses the deep fusion model fitting the pre-eminence of IoT sensor applications event dataset (Algorithm 2).

5.6. Data Collection Methodology. The Fog computing environment [37] collects big data for many data sources in the Internet, such as parking, transportation, security, and sensor IoT dataset statistics. Hundreds of thousands of collected data are contained in each dataset. System design or preprocessing: on the acquired big data, preparation processes were run to prepare them for processing by the deep learning algorithms of the new proposed methodology and conventional methods. The FDL system considers the data obtained results with those of the BPNN methods to verify that the FDL algorithm is running properly and to evaluate them in terms of processing speed and storage space to evaluate the benefits that the suggested technique has produced. We implement our proposed algorithm using Python and Scala, Spark ML H2O-based deep learning libraries [38, 47] Here, we performed an experiment based on HADOOP and flume using multiple nodes. Simulations are run in software toolkits such as iFogSim, YAFS, and CloudSimSDN to mimic the proposed architecture. A hierarchical tree-like arrangement of fog devices is feasible in iFogSim, but only between a parent-child pair. However, as compared to cloud, the latency and network use are low, and the delay factor is tied to the system's efficiency. If the band low, less traffic passes through the Fog node, making it more tolerable and efficient, and thus generated a training and testing outcome on a multiple node. Now, we have evaluated the speedup factor by the right utilization of the model. Here, we have selected several subsets and tested their efficiency, and then we have evaluated their testing results. We have evaluated the errors based on these selections. Now, we have improved the efficiency and stability of data models. There are almost several datasets available in the vector subspace. The operation and process we perform will be in tenfold cross-validation. The approach that we apply is certainly great compared with other approaches. There are two tasks that we are performing in this approach. The first task is that we are going to select the whole dataset and will perform training on those particular sets. And, in the second step, we make use of binary classification to improve the classification accuracy. This binary classification that we have done has gone through a deep learning process. The approach we have applied is truly simple and up to the logic that can bring the desired result. By using the different languages, we can easily implement this approach. This has eliminated most of the big data problems on a short time note. By this process, we can also resolve the multi-classification problem.

The evaluation measures are as follows:

The bigger the throughput, the more expense of handling time.

The more hubs we use, the less the preparing time.

If enough hubs are utilized, even the size of throughput is large, and the presentation can be close to the ideal one.

Spark ML with dispersed deep learning is a decent decision to manage real-time issues. Specifically, by utilizing more hubs, we can resolve enormous information issues, especially those identified with the continuous forecast in the services field.

Figure 6 describes the data collection process. Design a fusion deep learning neural system for Fog computing: in this phase, BPNN dependent on ensemble algorithms is to be created that it will run viably on broad databases of IoT Applications [19]. Furthermore, create FBEBDBA; likewise, it can manage many data variables without variable removal. Training and experimentation on datasets: the IoT Application model will be prepared on the sensor dataset to forecast honestly and build reinforcement learning. Deployment and examination on the real-time condition: the readied and attempted expectation model will be sent in a genuine circumstance made by the human masters and will be used for extra improvement in the procedure and will follow the above plan.

6. Result Evaluation

The difference between existing and proposed models according to time evaluation: this clearly can be observed according to Figure 1, which is a comprehensive representation of our model. In general, consistency is enough to evaluate the model to classify. Since we aim to identify if the sample is odd, accuracy and recall are critical metrics to test

Input: algorithm, β, λ, training dataset, T, L, Q, P, Z = store the classified records.
Output: classified dataset labeled dataset
Labeled dataset LD, pre-eminent model
For completely dataset L occurrence $\in$ novel dataset L event LD to do
Z $\longleftarrow$ { smart city IoT Sensor Applications datasets (parking, transportation, pollution, Security and sensor IoT dataset) dataset type$_1$..., IoT Sensor Applications (Usage Datasets) dataset type}
The condition of {|Z| = I}
Z $\longleftarrow$ $Z \cup$T perform the analysis in terms of prediction for recommendation and use the backpropagation learning
$M = M - \xi\nabla(X, Z)\nabla M$,
$b_Y = b_Y - \xi\nabla(X, Z)bY$,
$b_Z = b_Z - \xi\nabla(X, Z)bz$.
LD $\longleftarrow$Random sample consensus (Z, perfect α, β, λ) {predict LD}
New LD $\longleftarrow$ LD \ { IoT Sensor Applications (Usage Datasets) dataset classification event}
Perfect $\longleftarrow$ FLD (New LD)
Preeminent model LD $\longleftarrow$ preeminent model LD$\cup$ New LD
The end for return preeminent model LD

Algorithm 2: FBEBDBA-based analysis algorithm.

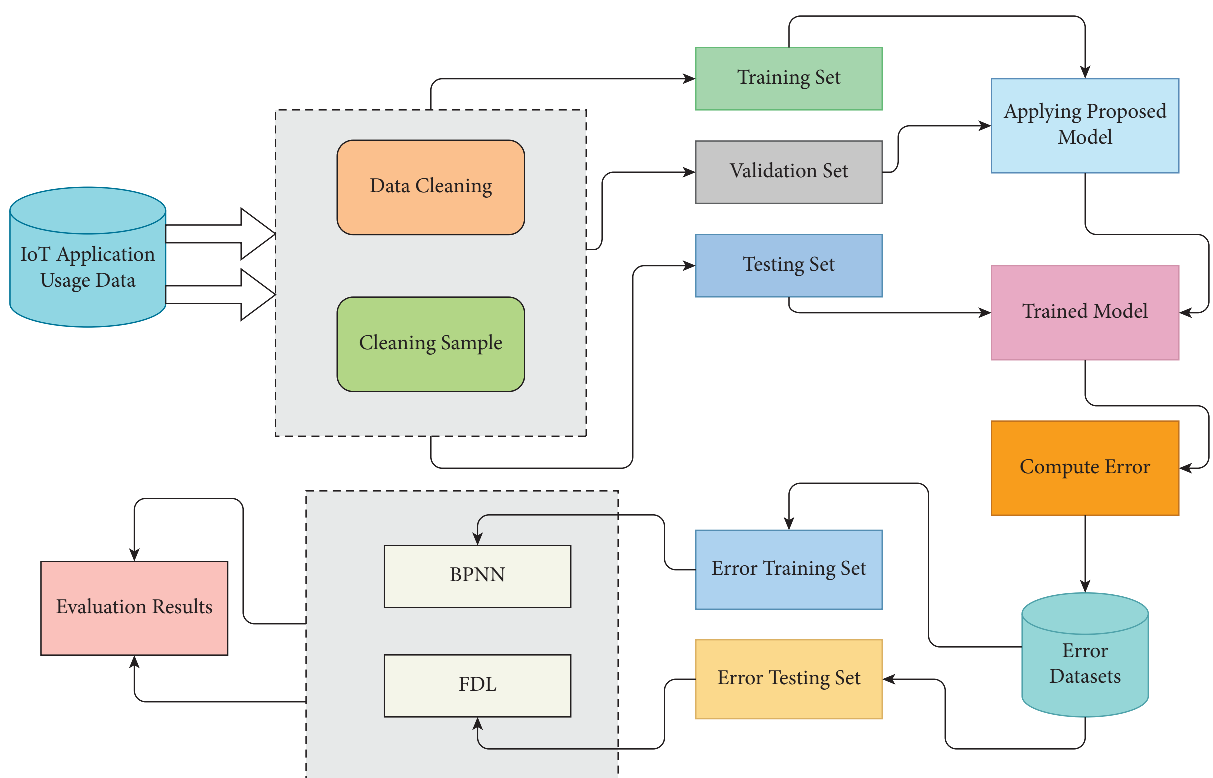

Figure 6: Data collection process.

our model. Precision rate is primarily used to judge if the classifier will accurately classify the anomaly, i.e., it focuses mainly on reported irregular samples in which how many such samples are truly abnormal. Also, the recall rate primarily judges if the classifier will recognize all odd samples. $F\beta$ index is a mixture of the two preceding metrics; if β is less than 1, the recall rate is more significant. The precision rate, on the opposite, has a larger effect on model quality evaluation. This experiment uses F1 markers since we consider precision and recall rates equally relevant for this article in the data collection. The performance of each model under four classification indicators (Accuracy, Precision, Recall, and F1) in three different datasets is evaluated. Traditional approaches are limited to have good efficiency. This research work proposes row dataset classification through FDL and uses MapReduce-based model in Hadoop. Increased accuracy, reduced error rate, memory consumption, and time consumption can be achieved by featuring space regression

Table 3: Comparative analysis between fusion deep learning algorithms (parking, transportation, pollution, security, and sensor IoT dataset accuracy in %).

Category-IoT	Dataset source	Fusion deep learning algorithm (parking, transportation, pollution, security, and sensor IoT dataset accuracy in %)				
		SVM	SVMG-RBF	BPNN	S3VM	FDL
Parking datasets	[39]	74.25	88.25	91.80	91.80	92.33
Smart car datasets	[40]	74.34	74.83	79.83	85.00	92.83
Smart car datasets	[41]	72.58	74.58	87.23	86.97	88.58
IoT network datasets	[43]	71.00	82.00	83.91	85.12	89.00
IoT network datasets	[44]	72.33	75.44	80.35	87.01	90.32

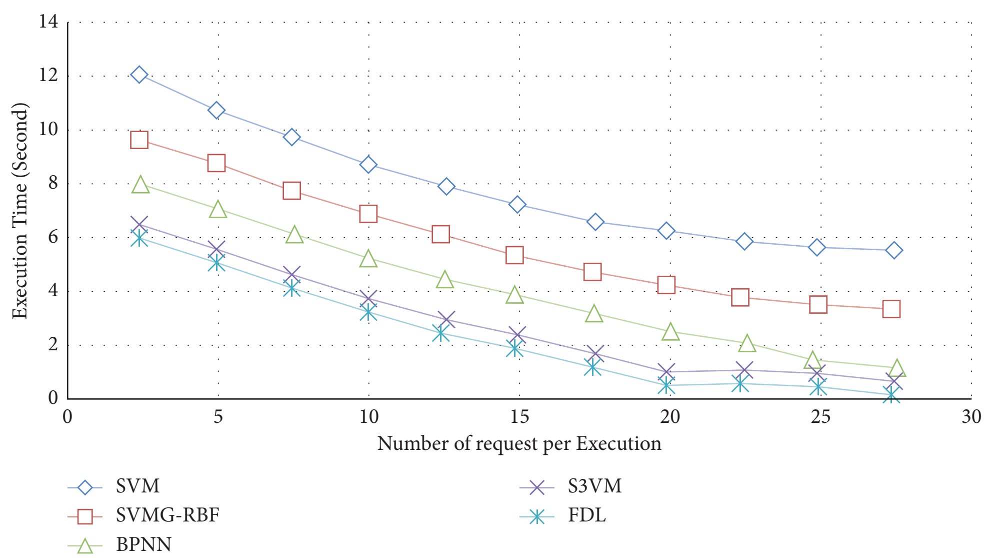

Figure 7: The result in different algorithms' ex ecution perspective.

and parameter alteration. Also, MapReduce is used to proliferate the execution speed. Previous research studies enhanced the execution speed through the linear kernel function. Table 3 and Figures 7–10 show comparative analysis of different deep learning and machine learning algorithms: support vector machine (SVM), radial basis function kernel, BPNN, and semisupervised SVM (S^3VM).

The time consumption is the time necessary to classify the entire dataset. The formula is as follows:

$$\text{time consumed} = \text{end time} - \text{start time}. \quad (5)$$

Figure 8 shows the comparative analysis of different algorithms in terms of time consumption and number of experiments. The X axis indicates the number of code executions, while the Y axis represents the memory usage of the proposed and existing model during system running (KB). As per the results, the proposed FDL consumes moderate system space when data are executed. The suggested FDL model uses coefficient correlation to produce an effective and efficient classification rate. So, we determine that it is the objective of space complexity.

The classifier's error rate indicates the number of incorrectly classified data. The classifier error rate can be calculated using the following formula.

$$\text{error rate} = \frac{\text{misclassified samples}}{\text{total samples to classify}} \times 100. \quad (6)$$

Figure 9 shows the error rate of the fusion deep learning model (as shown in Figure 11). The Y axis indicates the error rate percentage for both procedures, while the X axis provides the individual experimental results. SVM, SVG-RBF, BPNN, and S3VM classifiers with a low coefficient correlation based on our established fusion deep learning model produce a significant percentage of error rate in the preceding example. It has a lower error rate and misclassified less data than the basic approach. This technique improves categorization and distinguishes between normal and IoT application-based data.

Figure 10 shows the comparative analysis of Fog big data analysis methods. We calculate computation time in seconds and perform a test on the category of IoT application datasets.

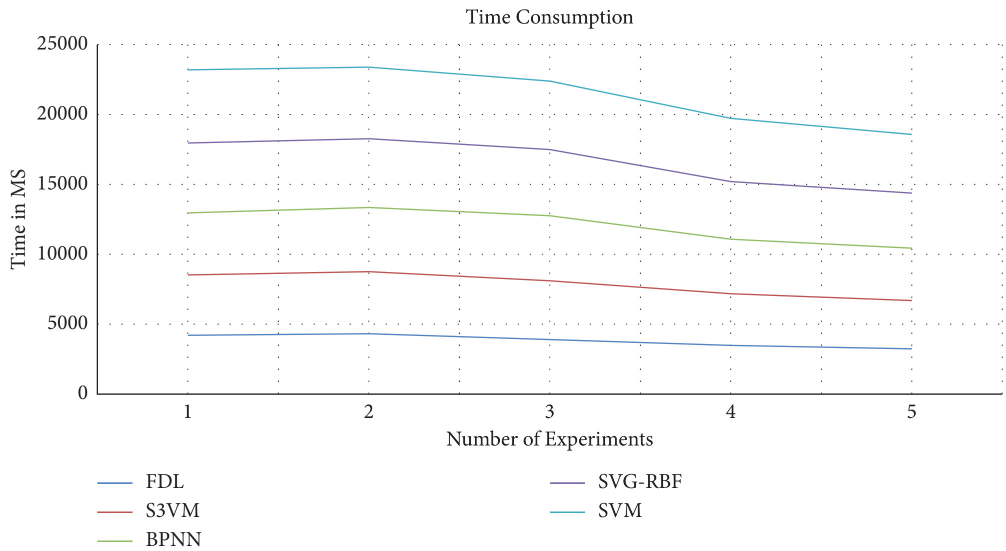

Figure 8: Comparative analysis of different algorithms in terms of time consumption and number of experiments.

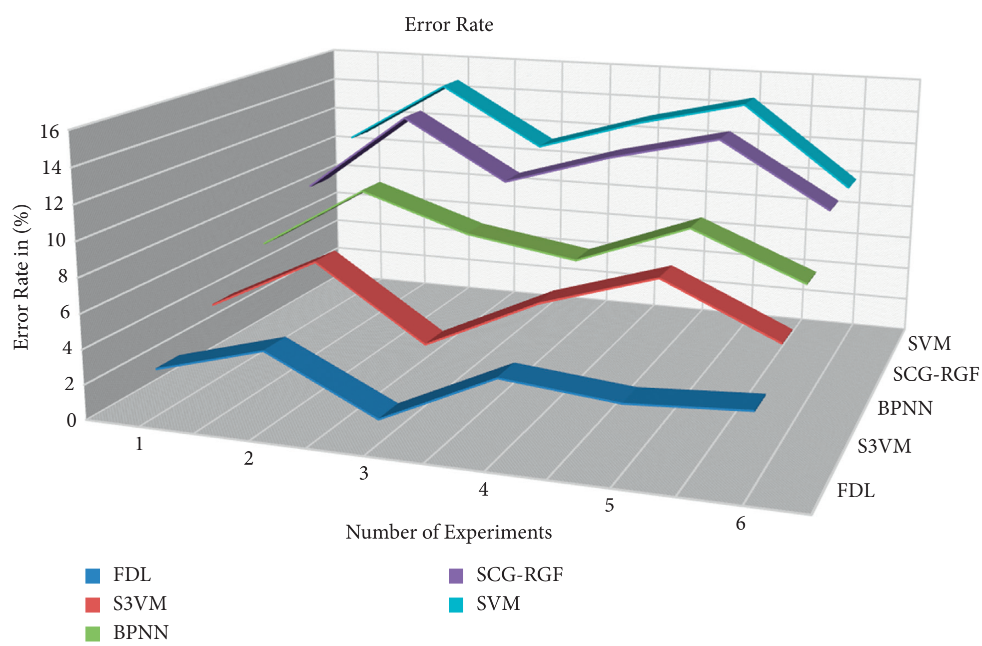

Figure 9: Comparison of error rate.

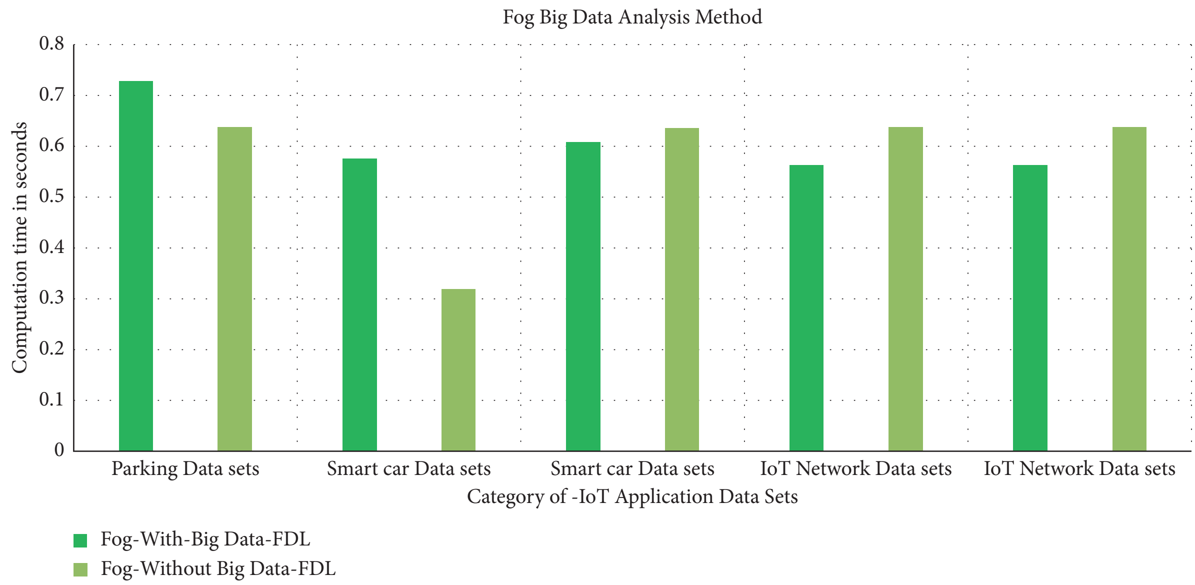

Figure 10: Comparative analysis of Fog big data analysis methods.

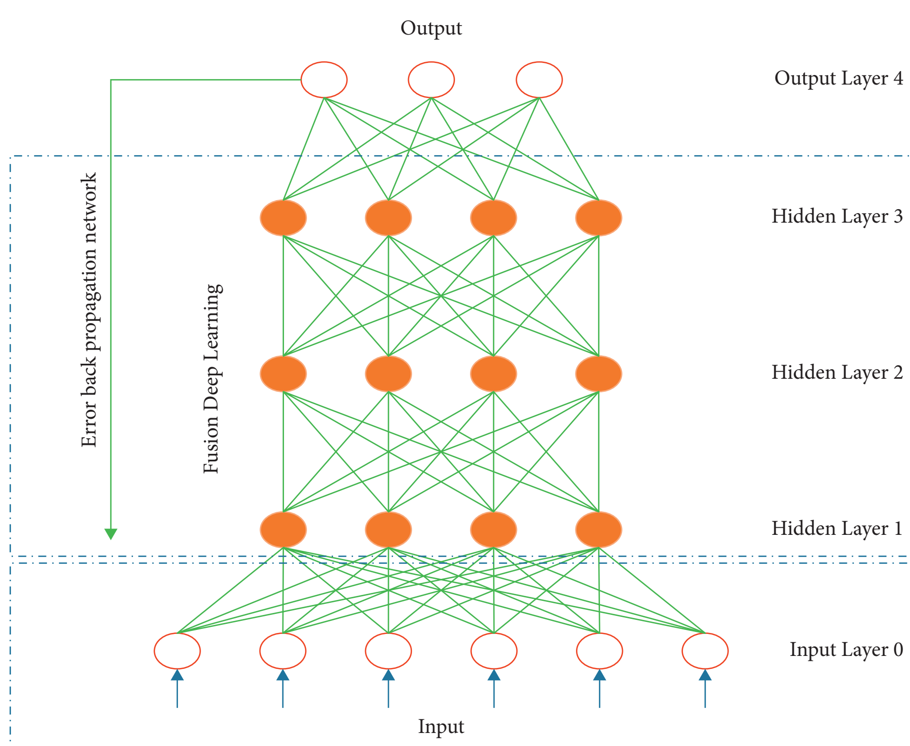

Figure 11: Explanation of a coordinated by a statistical relationship showing the layer. It goes through six layers.

7. Conclusion

IoT application generated data have been part of every smart city. However, IoT systems are limited in computational and connectivity capacity, which are the bottlenecks in creating scalable, intelligent machine-learning techniques. While advancements in technology and platform upgrades pave the way for a future involving rapid IoT expansion, device rollout, and strong high-volume IoT application usage data analytics, we argued that it was challenging to combine smart technologies from various domains. This article justified that our proposed deep learning and information exploration system for IoT paves way to implement adaptive learning strategies locally, at the edge, across Fog, or in the cloud. Consequently, in the broader sense of things, the ability of FDL should be completely used to deliver value and advantages to IoT users. Our proposed Fog big data analysis for IoT sensor application using fusion deep learning poses new obstacles for potential machine-to-machine communication practices. In cryptography, the analysis may involve researching the impact of malicious machines on other computers, which may contribute to system compromise. Computers must be designed to spread resources seamlessly (hyperconvergence), enabling the scalability of linked IoT devices. Therefore, as a federation of edge machines converges, the concurrent developments in deep learning technologies are realizable in resource-constrained IoT networks, thereby understanding the future of IoT-enabled human lives.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they do not have any conflicts of interest.

Acknowledgments

The authors extend their gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through a research group program under grant number R.G.P. 1/77/42.

References

[1] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for iot: review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, 2018.

[2] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier fog computing with large-scale IoT data analytics for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, 2018.

[3] O. Osanaiye, S. Chen, and Y. Zheng, "From cloud to fog computing: a review and a conceptual live VM migration framework," pp. 169–3536, 2016, http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

[4] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities," *ACM Computing Surveys*, vol. 50, no. 3, pp. 1–43, 2017.

[5] T. S. J. Darwish and K. Abu Bakar, "Fog based intelligent transportation big data analytics in the internet of vehicles environment: motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15679–15701, 2018.

[6] A. Kumari, R. Gupta, and S. Tanwar, "Introduction," in *Fog Data Analytics for IoT Applications. Studies in Big Data*, S. Tanwar, Ed., vol. 76Singapore, Springer, 2020.

[7] S. Potluri and K. Subba Rao, "Improved quality of service-based cloud service ranking and recommendation model," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 3, pp. 1252–1258, 2020.

[8] F. Hussain, H. Farahneh, X. Fernando, and A. Ferworn, "VLC enabled foglets assisted road asset reporting," in *Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–6, Sydney, Australia, June 2017.

[9] T. Nam and T. Pardo, "Smart city as urban innovation: focusing on management, policy, and context," *ICEGOV*, vol. 11, 2011.

[10] Y. Simmhan, "Big data and fog computing," in *Encyclopedia of Big Data Technologies*, S. Sakr and A. Zomaya, Eds., Springer International Publishing AG, New York, NY, USA, 2018.

[11] S. Fosso Wamba, M. N. C. Angéla, and E. E. J. Bosco, "Big data, the Internet of Things, and smart city research: a literature review and research agenda," in *EAI International Conference on Technology, Innovation, Entrepreneurship and Education, TIE 2017, Lecture Notes in Electrical Engineering*, A. Reyes-Munoz, P. Zheng, D. Crawford, and V. Callaghan, Eds., vol. 532, Cham, Switzerland, Springer, 2017.

[12] J. Al-Jaroodi and N. Mohamed, "Service-oriented architecture for big data analytics in smart cities," in *Proceedings of the 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Washington, DC, USA, May 2018.

[13] S. S.-D. Xu, C.-H. Chen, and T.-C. Chang, "Design of oneM2M-based fog computing architecture," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9464–9474, 2019.

[14] M. E. Idrissi, O. Elbeqqali, and J. RIFfi, "A review on relationship between iot- cloud computing - fog computing (applications and challenges)," in *Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, Marrakech, Morocco, October 2019.

[15] H. Hou, H. Jin, X. Liao, and D. Zeng, "Stochastic analysis on fog computing empowered mobile crowdsensing with D2D communications," in *2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Guangzhou, China, August 2018.

[16] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of the Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE*, pp. 464–470, Barcelona, Spain, August 2014.

[17] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: enhancing telehealth big data through fog computing," in *Proceedings of the ASE BigData & SocialInformatics 2015*, p. 14, ACM, New York; NY, USA, October 2015.

[18] G. Harish, S. Nagaraju, B. Harish, and M. Shaik, "A review on fog computing and its applications," *International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN*, vol. 8, no. 6C2, pp. 2278–3075, 2019.

[19] M. M. Hussain, M. M. S. Beg, and M. S. Alam, "Fog computing for big data analytics in IoT aided smart grid networks," *Wireless Personal Communications*, vol. 114, no. 4, pp. 3395–3418, 2020.

[20] C. M. Kanaka Sri Shalini, Y. M. Roopa, and J. S. Devi, "Fog computing for smart cities," in *Proceedings of the International Conference on Communication and Electronics*, Coimbatore, India, July 2019.

[21] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, 2016.

[22] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” *IEEE Pervasive Comput*, vol. 8, no. 4, pp. 1536–1268, 2009.

[23] A. Khochare, P. Ravindra, S. P. Reddy, and Y. Simmhan, “Distributed video analytics across edge and cloud using echo,” in *Proceedings of the International Conference on Service-Oriented Computing (ICSOC) Demo*, Malaga, Spain, July 2017.

[24] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: a programming model for large-scale applications on the Internet of things,” in *Proceedings of the ACM SIGCOMM Workshop on Mobile Cloud Computing*, Hong Kong, China, August 2013.

[25] P. Ravindra, A. Khochare, S. P. Reddy, S. Sharma, P. Varshney, and Y. Simmhan, “Echo: an adaptive orchestration platform for hybrid dataflows across cloud and edge,” in *Proceedings of the International Conference on Service-Oriented Computing (ICSOC)*, Dubai, UAE, December 2017.

[26] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, “Towards virtual machine migration in fog computing,” in *Proceedings of the IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, Krakow, Poland, November 2015.

[27] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, “Serendipity: enabling remote computing among intermittently connected mobile devices,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Bangalore, India, July 2012.

[28] F. Ferrández-Pastor, J. García-Chamizo, M. Nieto-Hidalgo, and J. Mora-Martínez, “Precision agriculture design method using a distributed computing architecture on internet of things context,” *Sensors*, vol. 18, no. 6, p. 1731, 2018.

[29] T. Islam and M. M. A. Hashem, “Task scheduling for big data management in fog infrastructure,” in *Proceedings of the 2018 21st International Conference of Computer and Information Technology (ICCIT)*, pp. 1–6, Dhaka, Bangladesh, December 2018.

[30] D. Gil, M. Johnsson, H. Mora, and J. Szymański, “Review of the complexity of managing big data of the internet of things,” *Complexity*, vol. 2019, Article ID 4592902, 12 pages, 2019.

[31] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, “Deep learning sensor fusion for autonomous vehicle perception and localization: a review,” *Sensors*, vol. 20, no. 15, p. 4220, 2020.

[32] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, “An overview of IoT sensor data processing, fusion, and analysis techniques,” *Sensors*, vol. 20, no. 21, p. 6076, 2020.

[33] T. J. Saleem and M. A. Chishti, “Deep learning for internet of things data analytics,” *Procedia Computer Science*, vol. 163, pp. 381–390, 2019.

[34] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: a survey,” *Fourthquarter*, vol. 20, no. 4, pp. 2923–2960, 2018.

[35] M. Shin, W. Paik, B. Kim, and S. Hwang, “An IoT platform with monitoring robot applying CNN-based context-aware learning,” *Sensors*, vol. 19, no. 11, p. 2525, 2019.

[36] A. Singh Rajawat and S. Jain, “Fusion deep learning based on back propagation neural network for personalization,” in *Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA)*, pp. 1–7, Piscataway, NJ, USA, February 2020.

[37] Z. Qi, S. Lyu, B. Zhang, and W. Feng, “Multiactivation pooling method in convolutional neural networks for image recognition,” *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8196906, 15 pages, 2018.

[38] G. Yoon, D. Choi, J. Lee, and H. Choi, “Management of IoT sensor data using a fog computing node,” *Journal of Sensors*, vol. 2019, Article ID 5107457, 9 pages, 2019.

[39] M. R. Anawar, S. Wang, M. Azam Zia, A. Khan Jadoon, U. Akram, and S. Raza, “Fog computing: an overview of big IoT data analytics,” *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7157192, 22 pages, 2018.

[40] M. Kaur and D. Singh, “Fusion of medical images using deep belief networks,” *Cluster Computing*, vol. 23, no. 2, pp. 1439–1453, 2020.

[41] D. Singh, D. Garg, and H. Singh Pannu, “Efficient Landsat image fusion using fuzzy and stationary discrete wavelet transform,,” *The Imaging Science Journal , Taylor and Francis Online*, vol. 65, no. 2, pp. 108–114, 2017.

[42] A. Gupta, D. Singh, and M. Kaur, “An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1309–1324, 2020.

[43] A. Kumar Saxena, S. Sinha, and P. Shukla, “design and development of image security technique by using cryptography and steganography: a combine approach,” *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. 4, pp. 13–21, 2018.

[44] M. Agrawal, A. U. Khan, and P. K. Shukla, “Stock price prediction using technical indicators: a predictive model using optimal deep learning,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 2, ISSN: 2277-3878, 2019.

[45] Y. Pathak, P. K. Shukla, and K. V. Arya, “Deep bidirectional classification model for COVID-19 disease infected patients,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 4, pp. 1234–1241, 2021.

[46] P. Bedi, S. B. Goyal, R. Sharma, D. K. Yadav, and M. Sharma, “Smart model for big data classification using deep learning in wireless body area networks,” in *Micro-Electronics and Telecommunication Engineering. Lecture Notes in Networks and Systems*, D. K. Sharma, L. H. Son, R. Sharma, and K. Cengiz, Eds., vol. 179Singapore, Springer, 2021.

[47] S. B. Goyal, P. Bedi, J. Kumar, and V. Varadarajan, “Deep learning application for sensing available spectrum for cognitive radio: an ECRNN approach,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 3235–3249, 2021.

[48] A. M. Rahmani, T. N. Gia, B. Negash et al., “Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach,” *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[49] G. Jia, G. Han, H. Rao, and L. Shu, “Edge computing-based intelligent manhole cover management system for smart cities,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1648–1656, 2018.