Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing the Quality of IoT-Bigdata to Improve Data Streaming Efficiency Using Artificial Intelligence Techniques

Gara Jaya Raju[1], Dr G Samuel Vara Prasada Raju[2]

[1]Research Scholar, Dept. of CSE, JNTUK, Kakinada, India.
[1]Sr. Assistant Professor, Dept. of CSE, Aditya University, Surampalem, AP, India.
[2]Professor of Computer Science, School of Distance Education, Andhra University, Visakhapatnam, AP, India.
Email-ID: [1]gjrraju2008@gmail.com , [2]gsvpraju2011@yahoo.com
Orcid-ID: [1]https://orcid.org/0000-0002-9692-6646

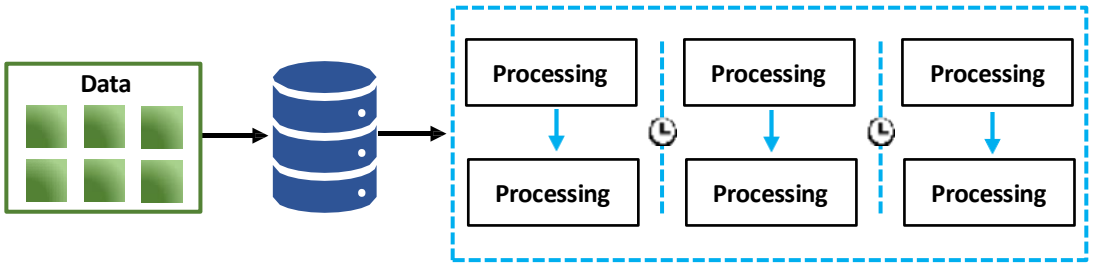
Many real-time cutting-edge applications prefer Internet of Things devices used for primary data generation because they use special sensors and software to generate and transmit data from one end to the other over the Internet. Based on the applications' necessity, the IoT-bigdata must be transmitted speedily, continuously, incrementally, and with low latency. One of the effective methods of IoT-big data is data-streaming technology, which can continuously transmit large amounts of data, permitting users to fetch the content immediately instead of waiting to download it. Once the big data is transmitted to the server, the big data analytics model is enabled to analyze the data for predicting and forecasting. However, the raw IoT big data might have problems regarding missing elements, duplicate data, mismatched data, and data overfitting, which interrupts the computational process of the analytics model and stops the process. It degrades the performance of the analytical model and the streaming process. Hence, IoT-Bigdata must be preprocessed and normalized to improve streaming, analyzing, and forecasting efficiency. This problem is considered a major research problem, and this paper has aimed at designing and implementing Exploratory Data Analytics, an Artificial Intelligence technique for analyzing and preprocessing the raw IoT-Bigdata and improving its quality, which can enhance the efficiency of data streaming, analyzing, and forecasting concerning the application where the IoT devices are deployed. The analytical model is implemented and experimented with Python and datasets taken from Kaggle.com, and the results are verified. The results obtained were compared with similar methods, and it was found that the proposed work obtained 99.89% accuracy in data preprocessing.

Keywords: IoT-Bigdata, Bigdata Streaming, IoT Data, Data Analytics, Environmental Forecasting.

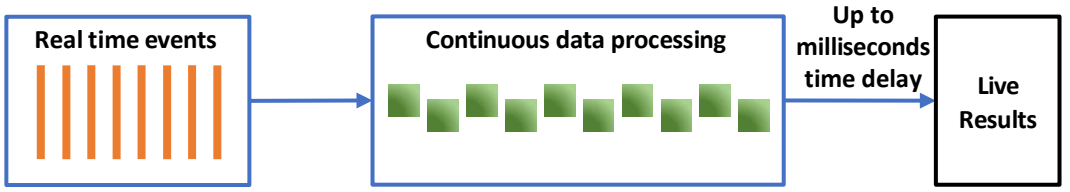
Introduction

Every day begins with a novel idea in the digital world, resulting in a technical revolution. State-of-the-art technologies in mobile networks and internet connectivity drive human life efficiently. Data transmission is vital in using different kinds of data for various applications. Based on processing, data is typically classified into two categories: bounded and unbounded. Starting and ending are clearly defined in bounded categories; conversely, unbounded data is

not predetermined and has no end in either or both directions. Unbounded data transmission over the internet can be referred to as data streaming. Real-time and near real-time outcomes are possible since input data is suddenly and continuously processed in such applications—low latency and flexibility. Scalability and continuous data reprocessing are the merits of data streaming rather than batch processing. Figure 1 and Figure 2 illustrate the batch processing and data streaming, respectively. Data will be stored in data sets over different intervals in batch processing, and continuous data processing will be taken in stream processing.



(a)



(b)

Figure-1 (a) Batch processing (b) continuous data processing

In addition to continuous flow and unbounded data, data streaming has some unique features such as heterogeneous, high-volume, and time-sensitive Data analysis, financial analysis, online education, real-time recommendations, media, and IoT are a few instances of streaming data. Most of the data streaming process over the Internet is carried out by the Internet of Things (IoT), which can be defined as a network of various devices based on Internet connectivity driven by technologies. Surveillance cameras, weather forecast devices, vehicle sensors, and agricultural monitoring devices are examples of IoT devices. In various scenarios, dynamic and distinct data are continuously processed and sent to a streaming application. Massive data is constantly generated from devices or sensors, resulting in bigdata consisting of various data sets at an ever-increasing rate. The volume of information, velocity, and variety are the parameters referred to as the "3Vs" of big data. It's categorized into two types: structured and unstructured. Structured data is generally numeric from an existing database management system, whereas unscheduled and unorganized data are called

unstructured. Additionally, IoT Big data cannot be limited to either of these categories. Real-time decisions, analytics, and insights are also considered to determine its categorization.

In any real-time applications, sensor devices, fog devices, edge devices, and IoT devices are deployed to monitor and record the data about the environment and transmit it to the storage devices or the server with the help of the broker (Data hub). Since the above-said devices generate and transmit the data continuously, it becomes a streaming process. Thus, a streaming processor is connected with the broker, which streams data in a pipeline for a processor or storage. The data processing is performed based on the action defined in the software. A simple process of IoT data streaming process is illustrated in Figure 2. When the data size is increased drastically and continuously and produces a massive amount of data, it is called big data. The continuous data cannot be transmitted simultaneously; it takes more time and is transmitted through a stream processing engine (Figure 3).

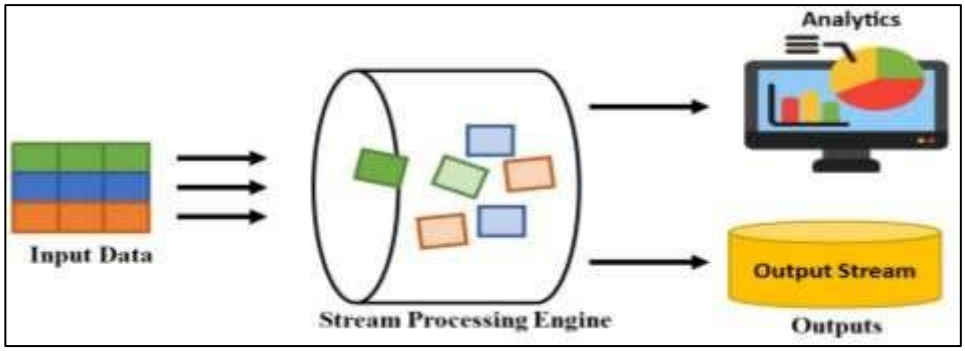


Figure-2. Standard Data Streaming

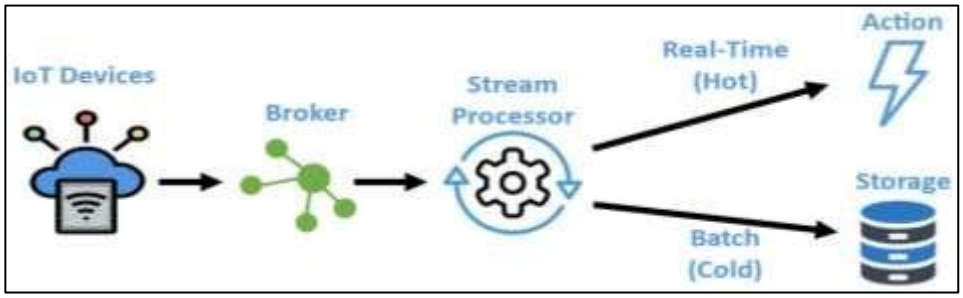


Figure-3. IoT-Data Streaming

Considering the high-level architecture of IoT-Bigdata streaming, multiple devices are interconnected by the Internet, creating an IoT network. The data generated from various devices of the IoT network are aggregated and transmitted through a hub as a data queue. The data obtained from the devices are sent to AWS IoT, Azure IoT hub, cloud IoT core, and Hive MQ, the standard and trusted solution providers for IoT messaging and IoT deployments. They connect, communicate, and control the data and send it to visualize, analyze, and store it. These services stream the data with the help of streaming services such as Kafka, Amazon Kinesis, cloud, and Azure Event Hub and manage the storage with the help of Google Cloud Storage,

Amazon S3, and Azure Blob Storage. The overall high-level architecture of IoT-Bigdata streaming is illustrated in Figure-3.

This paper contributes the following to solve the research problem better.

- A detailed explanation of the problem statement is given with mathematical expressions and pictorial representation.
- It explains clearly how the IoT-bigdata is generated and the necessity of preprocessing to improve streaming efficiency.
- The artificial intelligence model, with its detailed architecture and layer functionalities, is explained for analysis of IoT big data.
- The dataset, experimental setup, and results obtained are explained with the performance evaluation of the proposed model.

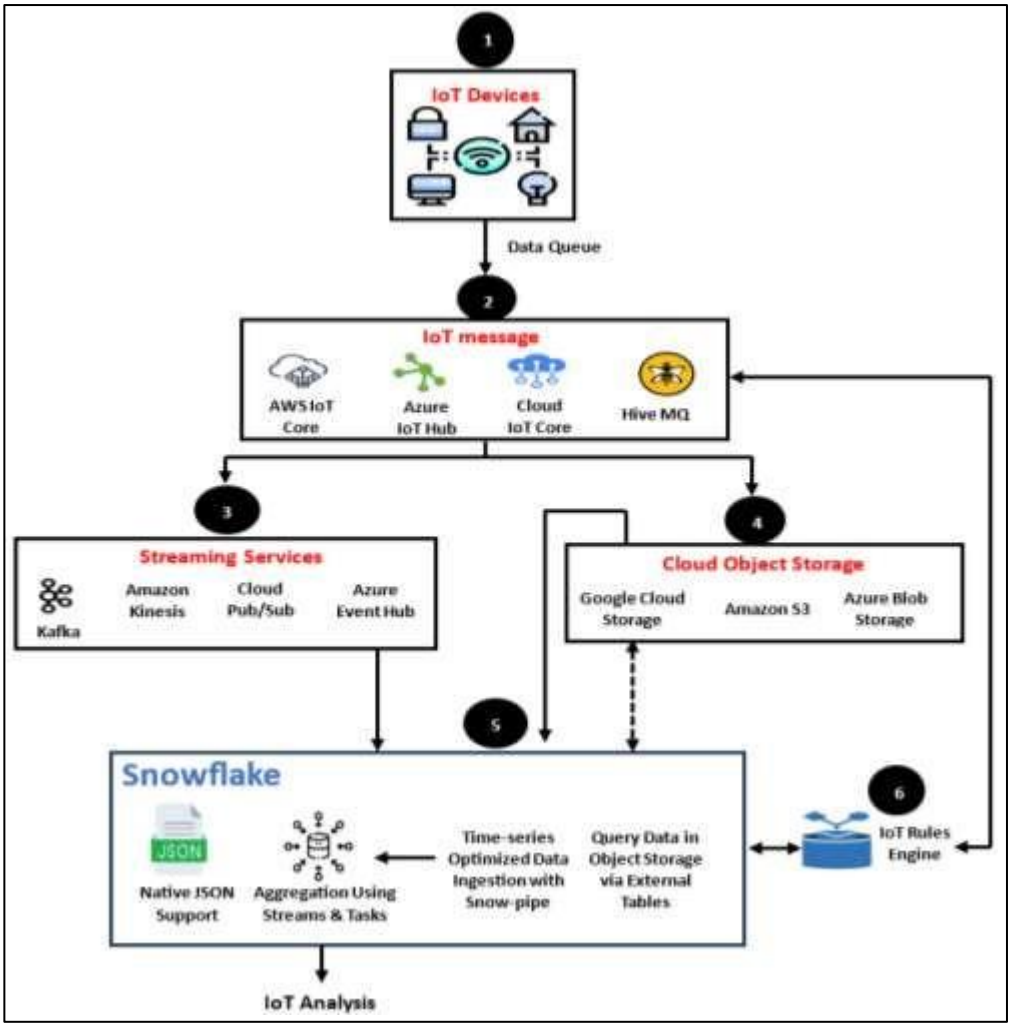


Figure-4. IoT-Bigdata Streaming Architecture

Real-time information is the key factor in solving undefined and complex problems; identifying and troubleshooting such issues can be done faster by utilizing IoT and Big data. For instance, Safety precautions, preventive maintenance, malpractice detection, and security enhancement are possible. Privacy, quality, integration, and security are the main challenges in processing big data; biases, errors, and noise should be filtered out to ensure data reliability, and securing data from unauthentic access is more essential. Data from various sources, different systems, and distinct formats result in complexity and diversity; this is why Integrating data is crucial. In terms of ideal insights, an effective data analysis is needed. To face the challenges and find solutions, data pre-processing is recommended.

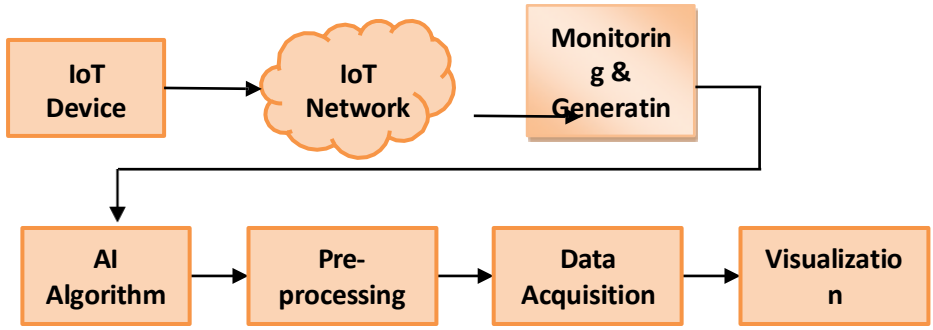


Figure-5. Workflow of Data pre-processing

Cleaning, verifying, and validating techniques are taken over for data quality. Encrypting, authenticating, and auditing methods are involved in data security. Consent management and anonymization play the main role in data privacy. Mapping, transformation, federating, and aggregating methods achieve the integration. Mining, visualizing, and modeling are carried out during the data analysis. Figure 3 illustrates the detailed IoT Big data processing process up to visualization. To generate pre-processing data among big data, AI algorithms can be used to carry out the required techniques and methods. IoT Network consists of numerous IoT devices, and various data types are continuously monitored and generated from different sources. Hence, the scenario makes the challenges. To provide solutions for the challenges, data pre-processing is essential, and the task is executed using Artificial Intelligence (AI) techniques; after that, data acquisition and visualization become more efficient and result-oriented. The continuous research and development projects in AI and Machine learning (ML) enable automation, real-time oriented insights analysis, and better performance. This paper describes utilizing the appropriate and effective AI techniques to ensure the quality of IoT Big data to provide efficient streaming data.

Literature Survey

The problem statement can be identified by learning about earlier research work. This section surveys earlier research methods similar to uncertain data streaming, big data, IoT big data, and data analytics. For example, Segura-Garcia et al. [1] used Factor analysis, Multinomial Linear Regression, and Artificial Neural Networks to assess video streaming quality through

Mobile Networks. The combination of IoT and Big Data was used for real-time continuous monitoring and processing by Malek et al. [2]. The problems encountered while storing the IoT Big data in the Cloud have resulted in better accessibility and variable resource management. Cai. H et al. [3] discussed the increased data processing efficiency and the uniqueness of the various IoT-based applications—another survey by Feng. C et al. [4] stated that Fog-based cloud computing was discussed for IoT Big healthcare data to enhance energy efficiency. Data tracing is also discussed; while processing the IoT data through the cloud, we can easily track down the current data scenario. Wongthongtham discussed the challenges of IoT-based big data analytics, P et al. [5] to improve overall big data processing. The author listed the difficulties in IoT-based Bigdata transformations: data management in size, data processing related to data acquisition, analysis of unstructured data, data semantics, and visualization. The above-listed challenges must be addressed to enhance the data quality in the bigdata transformations. Janani Arthanari and Baskaran. R [6] have researched data analytics of live traffic video streaming through cluster computing. In that research analysis, a few streaming applications and software were experimented with to analyze the data of the live video stream. It was concluded that Apache Kafka looks promising, among others. Wei-Chen et al. [7] worked on the diverse data of multimedia devices that share big data-based video streaming for the holistic experience by the integrated distribution system. However, the research lacks an upgraded distribution network, video coding, and communication for multiple.

Hung Cao and Monica Wachowicz [8] have developed an architecture using fog, edge, and cloud computing to help the data analytics of streaming IoT. They have used this particular architecture to analyze the real-time parking data alone. The shortcomings of this architecture are that it is suitable only for handling parking-based streaming data. To overcome this, we need to develop an AI-based architecture. Kumari et al. [9] have discussed big data streaming security as the data requires secure analysis, processing, and storage. Multi IoT devices are streaming the data, which uses the Big Data storage, which has more and more data. So, the security of big data is essential. The authors propose using an advanced architecture level to ensure Big Data security. The above research has become the base for the security of streaming Big Data and the proper applications used. Mehmood E and Anees T [10] discussed in their survey the implementation of data warehousing and cloud computing to differentiate rational and non-rational data. They also discussed the data structure and shape of the data, which is available in various IoT-based platforms that need to stream the entire domain used, which is considered a challenge. Mirzaie et al. [11] have stated that accessing the data quality plays a vital role in segregating the meaningful data and increasing the streaming quality.

Thus, to improve the streaming quality of the big data, an algorithm must process, analyze, and segregate the required or meaningful data among all other data available from the connected IoT devices. Also, it is essential to transfer the data in the format necessary to preserve it while in storage. These above actions can be done with the help of an Artificial Intelligence (AI) algorithm to deliver the overall data quality. The algorithm will process, segregate, preserve, and provide the needed data. The survey analyzes the existing issues and challenges to understand the problem statement and decide how to design the research method to solve the problem.

Limitation and Motivation

The literature survey identified that several earlier methods have focused on improving the data streaming efficiency concerning time, data streaming rate, error, data loss, and forecasting accuracy individually. Some methods have focused only on data streaming, and others have focused only on data analytics. One of the common challenges of data streaming is managing and processing a large volume of big data parallelly, which needs a special kind of architecture and system that can manage and process the ongoing data flow obtained from multiple sources. It also requires adequate storage for unstructured data forms and formats, which traditional databases cannot store. The analytics model should face the main challenges of volume, value, velocity, volatility, and variety, which make computational and time complexities. However, the raw data generated from the IoT devices have missing, redundant, overfitting, and wrong and mismatched elements that must be preprocessed before streaming, analysis, and forecasting. This paper has motivated me to apply artificial intelligence algorithms to preprocess the IoT-big data generated from the IoT network.

Problem Statement

Recently, most real-time applications use multi-purpose sensors or sensors-based IoT devices to monitor, record, and transmit data from one end to the other. Generally, the Internet of Things (IoT) refers to a network of physical devices. These devices mainly use sensors that can collect data from the environment and transfer data from one device to another without the involvement of humans. For example, in an agricultural environment, let's consider a scenario with **N** number of IOT devices deployed in a matrix format shown in Figure-6. These devices are said to be IoT networks that connect all the devices in a network and can monitor the whole environment and generate data. The IoT network comprises **K** types of devices installed in the single network to monitor and record multiple kinds of data. The IoT devices deployed in the environment are in a square format, depicted in Figure 6.

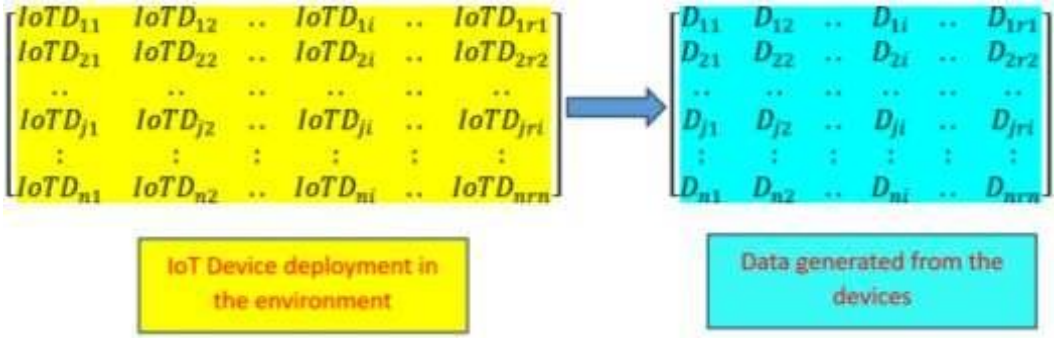


Figure-6. IoT Bigdata Generation

All devices are categorized into K types, each with its behavior and usage. For example, some devices monitor only temperature, and some may generate humidity. All the **K** types of devices are arranged with a sequence number, subject ID, and state of the devices to indicate the nature of the current process. For example, the '**K'** devices, which monitor

temperature, humidity, air, and water levels, were deployed in the agricultural environment. Each device generates the data $\boldsymbol{d_0}$ to $\boldsymbol{d_r}$ from $\boldsymbol{t_0}$ to $\boldsymbol{t_r}$ in the size of (L X L) environment.

$$\{\boldsymbol{t_0}, \boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_i}, \dots, \boldsymbol{t_r}\}$$

$$\{\boldsymbol{d_0}, \boldsymbol{d_1}, \boldsymbol{d_2}, \dots, \boldsymbol{d_i}, \dots, \boldsymbol{d_n}\}$$

The total time to fetch data from various IoT devices is 7 hours daily. At the t_0, the d_0 data is produced from the IoT device daily. The time interval between $\boldsymbol{t_0}$ and $\boldsymbol{t_r}$ is not constant; data is produced randomly every time. This is how the data is generated from one IoT device (Matrix). These various '**N**' numbers of IoT devices in the environment (L X L) communicate with each other and produce a massive amount of data in the form of L X L X N concerning. $\boldsymbol{K}$ at the $\boldsymbol{t_0}$ to $\boldsymbol{t_0}$ from all the devices collectively known as Big Data, which can stream every second over the internet. At $\boldsymbol{t_0}$, big data produced in the form of $\frac{LXLXN}{K}$.

Big Data was generated from the '**K**' type of devices, such as temperature sensors, humidity sensors, water level sensors, and air-quality sensors, which were deployed in the agriculture field. Raw data is the primary source of inaccuracy in IoT devices, resulting in quality issues. Quality problems often arise from the sensor IoT devices, creating data analytics difficulties. Quality problems are due to network problems, sensor malfunctioning, external interferences, and interruptions in the power of sensors. Any faults made while capturing or processing the data stream are clearly shown by error. For example, malfunctions in the sensor's installation or use outside the approved activities result in errors.

The quality of the data plays a crucial role in IoT Analytics. Some quality aspects that impact IoT analytics are accuracy, consistency, completeness, and time frame. Accuracy measures whether the data produced from every sensor in the IoT network is uniform and acceptable. Environmental conditions can impact the consistency of the data. The completeness parameter determines the missing information or the sensor readings data gap. Since the sensor produces data in a time series, it is important to note that it is produced within the acceptable time frame. Since the data is produced from the '**N'** number of devices and wide types of sensors, it is essential to note that the data is synchronized. This is the reason for Data Quality (DQ) errors. Regarding the sensor dataset, the errors affecting the Data Quality (DQ) are outliers, missing elements, empty data, mismatched data, overfitting, and duplicate data.

Outliers are anomalies or spikes, unexpected values deviating from most data. These values are either fall below or exceed the threshold (normal value of data). These kinds of errors are hard to find and very rare. The missing/empty data happened in the transmitted data from various sensors for reasons such as unstable wireless connection, environment interferences, malicious attacks, errors in recording, etc. Data analytics are stopped due to the abrupt data loss generated by IoT devices. This leads to noisy data and missing values. Noisy signals are caused by unknown changes in the signal during the processing, capturing, storing, and transmission of data. If the dataset contains unnecessary rows and columns while integrating big data from various IoT devices, it leads to duplicate data. Multiple correlations can result from data duplication, underlying relationships, or accidental correlation. In this error, values under numerous columns in a data set may be correlated and cause quality issues.

Redundancy appears when there are multiple readings from the same sensor or a few readings from the other sensors. The readings from the non-existent devices are also considered the source of uncertainty. Datasets typically store different data types, such as strings, integers, and floats. If the number (integer) form of data is stored in the place of string, this leads to the incorrect data type error.

According to the researchers, real-world experience, and various studies, missing values and outliers are the most frequent sensor quality issues that cause data generation interruptions, impacting IoT analytics. In terms of video big data from IoT devices, videos are collected as shorts and frames or images. For example, in 2 minutes of video, it has 120 frames. To avoid quality issues like defocus, video blurring, part occlusion, etc. Noise removal, color conversion from RGB to GS (greyscale), cropping, rotating, color enhancement, and registration of images in the axis are the processes that can occur in the pre-processing stage. With the introduction of a pre-processing method to get efficient streaming, there is a need to produce good-quality data. Data collected by sensors should be of good quality with a good data flow and be developed with the help of pre-processing. However, pre-processing big data is a tough task since the data is huge. And it can be achieved quickly with the implementation of AI algorithms.

Data pre-processing

Various steps in the proposed model improve streaming by enhancing the input IoT big data quality. The proposed model includes Three phases: data acquisition, preprocessing, and data streaming, as shown in Figure-5. Data acquisition is the process of organizing and arranging the input raw data. The data pre-processing step includes data reduction, transformation, cleaning, and redundancy removal. After pre-processing the input data, the AI-based approach is deployed to perform the continuous streaming over the IoT devices. As mentioned, many IoT-based sensors or data analysis devices are deployed to collect large amounts of data. However, all the data are not of the same quality, meaning they have diverse changes in size, length, redundancy, etc. This will reduce the quality of the data streaming process. Therefore, a data-preprocessing method is applied to reduce and transform the unnecessary data.

Data reduction

The enormous amount of data requires more time to analyze and is difficult to evaluate. At this stage, the data reduction step is performed, which reduces the depiction in the raw input dataset. This process will increase the accuracy of the proposed model's analytical results. It is applied to all the data in the input raw dataset; the main goal of this step is to clear the traffic in the network based on its data or variables.

$$\boldsymbol{DC_1 \rightarrow D_1, D_2, \ldots . D_n}$$

$$\boldsymbol{DC_2 \rightarrow D_1, D_2, \ldots . D_n}$$

$$\boldsymbol{DC_n \rightarrow D_1, D_2, \ldots . D_n}$$

In the above equation, $\boldsymbol{DC}$ and $\boldsymbol{D}$ represents the data cluster and data generated from **N** number of IoT device implemented within the region. Further, the collected datasets are divided into

blocks (BL). The blocks are separated based on the medium to capture the input data. The data collected from different sensors are aggregated using the following equation.

$$DC = \sum_{i=1}^{n} BL_i$$

$$BL = \sum_{i=1}^{n} device_i$$

Based on the device ID, the data reduction process is performed at different time intervals, $t_0, t_1, \ldots . t_m$. Following this, a data transformation technique is performed.

Data transformation

This step is performed to limit the range of the input data within a threshold range. One of the most common methods used for data transformation is the min-max technique. The main motive of this step is to convert the input raw data into a suitable format for data analysis and modeling. That is, every minimum value is transformed into 0, every maximum value is transformed into 1, and the remaining data are normalized between **0** and **1**.

$$y = \frac{X - Xmin}{Xmax - Xmin} \times (n - m) + m$$

The data transformation process is performed using the above equation. This formula scales the value of features between **0** and **1**.

Data Cleaning using EDA

This paper used the Exploratory Data Analytics model to preprocess data (data cleaning) under various circumstances. Data cleaning is fixing or replacing incomplete, duplicate, or irrelevant input data. This step makes the model more efficient in decision-making and improves data quality, model accuracy, and consistency in decision-making. In that sense, in this paper, a data cleaning step is performed to fill or replace the missing values and remove noisy data.

Data analytics is based on the main characteristics, and the data visualization method is called exploratory data analysis (EDA). EDA determines the data manipulation methods to ease the process of pattern recognition, anomaly detection, examining a hypothesis, and investigating other conditions. EDA mainly focuses on hypothesis testing to understand the data and the relationship among the data entities. It uses statistical methodologies and mathematical expressions to analyze the data appropriately.

The main objective of EDA is to examine the data before applying data processing methods. It helps identify errors, data patterns, outliers, and anomalies and obtain the data relationship. Thus, this paper uses EDA, an artificial intelligence technique, to preprocess the proposed dataset. These statistical functions that can be performed using EDA are clustering, dimensionality reduction, univariate, bivariate, and multivariate visualization. One of the most common data science tools for developing EDA is Python. The exploratory data analytics

model calculates the dataset's mean value using the following mathematical expressions. Three different kinds of mean values can be calculated based on the dataset, size, and dynamic behavior of the dataset.

$$Mean = \bar{x} = \frac{\sum_i^n x_i}{n} \quad (1)$$

$$Weighted\ Mean = \bar{x}_w = \frac{\sum_{i=1}^n w_i x_i}{\sum_i^n w} \quad (1.2)$$

$$Truncated\ Mean = \bar{x}_{tr} = \frac{\sum_{i=p+1}^{n-p} x_i}{n-2p} \quad (1.3)$$

The median value of the dataset is obtained by choosing the middle element of the data after sorting in ascending or descending order. For the data $x_1, x_2, \dots, x_n$, the median value is obtained using the following formula:

$$\text{if n is odd} \rightarrow \text{Median} = x_{\frac{n+1}{2}}$$

$$\text{if n is even} \rightarrow \text{Median} = \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n+1}{2}}\right)$$

$$\text{Weighted Median} = x_k$$

$$\text{where} \sum_{i=1}^{n} w_i = 1 \text{ and } \sum_{i=k+1}^{n} w_i \leq \frac{1}{2} \text{ and } \sum_{i=1}^{k-1} w_i \leq \frac{1}{2}$$

Similarly, the MAD value, variance, and standard deviation are calculated for the input dataset below.

$$\text{Mean absolute deviation} = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

$$\text{Variance} = s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

$$\text{Standard Deviation} = s = \sqrt{\text{Variance}}$$

$\text{Median absolute deviation} = \text{Median}(|x_1 - m|, |x_2 - m|, \dots, |x_n - m|)$,
where m is the median.

Missing Values

Missing values occur for various reasons, such as detection errors, physical object detection errors, issues with continuous data accessing, etc. They are more common in raw input data and make the model more complex for analysing and managing it. The missing values are handled by ignoring the tuples or filling in the missing values. The following steps are used to identify the missing data.

Function_ME(D data)

{

Input D = $\{x_1, x_2, \ldots, x_i, \ldots, x_n\} \forall i = 0, 1, 2, \ldots, n$;

Count=0;

For I = 1 to size (D)

count=count +1;

End I

M=mean (d) * count;

M= M - $\sum_{i=0}^{size(D)} d(i)$

}

In other words, the average values are calculated for the entire dataset and preprocessing.

Noisy Data

This data type occurs due to an entry error or fault during collection. It is handled using three methods: binning, regression, and clustering. The binning method on noisy data is used to smoothen the input data. This method divides the overall data into multiple segments of equal size. Each segmented data is handled separately, which will reduce the noise in the input dataset. Second, the regression method is used, which fits the noisy data using the regression function. The regression function has single or multiple independent variables. Third, the clustering method is used, which clusters the group with similar data in the raw input dataset. The remaining data are eliminated or removed from the cluster. The noisy input data are easily removed, and the quality of the data received from IoT sensors also increases. After pre-processing the input data, feature extraction and selection process. The essential data is continuously streamed from one point to another based on the extracted features. The overall process of IoT-Bigdata generation and preprocessing method is given as pseudocode.

Pseudocode_IoTBGPreProc (S Data)

{

Input: collect all sensor Data **S**;

Output: Pre-processed-**S**;

For I = 0 to 15

D(i) = S(i).data;

Allocate the memory;

```
End I
For I=1 to size (D)
    Check_For_Missing_Element(D);
nTrD1 = elimated_Missing_Electment(D);
    Check_For_typeMisMatch(D);
nTrD2 = elimated_typeMisMatch(D);
    Check_For_typeWrongElements(D);
nTrD3 = elimated_typeWrongElements(D);
end I
return (TrD = nTrD1 + nTrD2 + nTrD3);
}
```

The pseudo-code reads all the raw data and preprocesses it with respect to missing elements, type mismatches, and wrong elements. After removing and rectifying the data element-based mistakes, the final data TrD is fed to the AI algorithm for prediction. The above pseudocode is implemented in Python with a time series dataset, and the outputs are verified. The performance of the proposed preprocessing methods is evaluated by comparing the output with the output of similar methods.

Experimental Setup

The proposed preprocessing method is experimented with Intel Pentium Core i7, 7^{th} generation, 1TB HDD, 16GB RAM, and 2.36GHz processor speed. Python is installed with all essential libraries to enable artificial intelligence algorithms using Kera's model. The dataset (Tabular Play-Ground Series) is a time series dataset generated from 13 sensors. This dataset is collected monthly and stored in tabular form. It comprises 26,000 data sequences with 671 subjects that can be predicted. Each data sequence comprises 60 steps where one step for one second is generated. The extension of Pandas and intel SK Learn extensions are used for data learning. This dataset helps to understand the end-to-end user experience when exploring streaming time series datasets. The dataset is divided into three sets such as training data (train.csv), testing data (test.csv), and final_submission (final_submission.csv), used for training the model, testing the model, and submitting the preprocessed data in a final dataset.

Results and Discussion

Initially, to understand the entire work, a sample dataset is given in Table-1. The sequence number and step number are given, and based on that, the recorded data from the sensors are given. The memory utilization in the training process is reduced by visualizing the data distribution and the number of features extracted. From the visualization, three different data sets are obtained. In the first set of data, the encapsulated training features are stored. It has a

sequence ID related to 60 steps of sensor reading. Each data sequence has a unique subject ID. The second table, called the target table, has the sequence number and class label. Finally, the third table continues from the first table, and the class label is unknown and will be predicted at the classification stage. Then, the feature engineering model is applied to learn and extract the encapsulation of the original sensor data with respect to the sequence number. It is obtained by computing the mean, standard deviation, skewness, median, IQR, kurtosis, and others for all sensor data to check the data quality assessment. One of the AI models, Exploratory Data Analytics, is used to learn and preprocess the dataset for classification. The pre-processing operation is applied after learning and extracting the data features.

Table-1. Sample IoT Data

Sequence	subject	step	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	sensor_08	sensor_09	sensor_10	sensor_11	sensor_12
0	0	47	0	-0.1963	0.1124	1.0	0.3292	-1.0047	-0.1316	-0.1275	0.3687	-0.1	-0.9639	-0.9851	0.5319
1	0	47	1	-0.4474	0.1345	1.0	-0.6584	0.1625	0.3403	-0.2095	-0.8672	0.2	-0.3013	0.0827	-0.2315
2	0	47	2	0.3269	-0.6943	1.0	0.3301	0.4737	1.2805	-0.0947	0.5359	1.4	1.0022	0.4492	-0.5864
3	0	47	3	0.5232	0.7511	1.0	0.9770	-0.5633	-0.7203	0.7933	0.9511	-0.3	-0.9957	-0.4343	1.3447
4	0	47	4	0.2720	1.0746	1.0	-0.1363	0.3986	0.0449	0.5601	-0.5420	0.9	1.0956	0.8126	0.1235

The sensor data set is visualized as a heatmap correlation and identifies the missing elements, redundancy, and wrong and mismatched data. These kinds of data are removed simply since the data is time series data and 90% of the consequent data values are similar and varied only in micro points. The heatmap-based correlation among the dataset is shown in Figure 7. The sensor data correlation is analyzed and shown in Figure-8 for a fixed time interval. It shows that only very few sensors have values different from those of others. Thus, it is understood that the dataset does not provide more information for a small interval of time. More accuracy of missing data and other factors is analyzed by distributing all the sensor data obtained at regular intervals. It is shown in Figure-9.

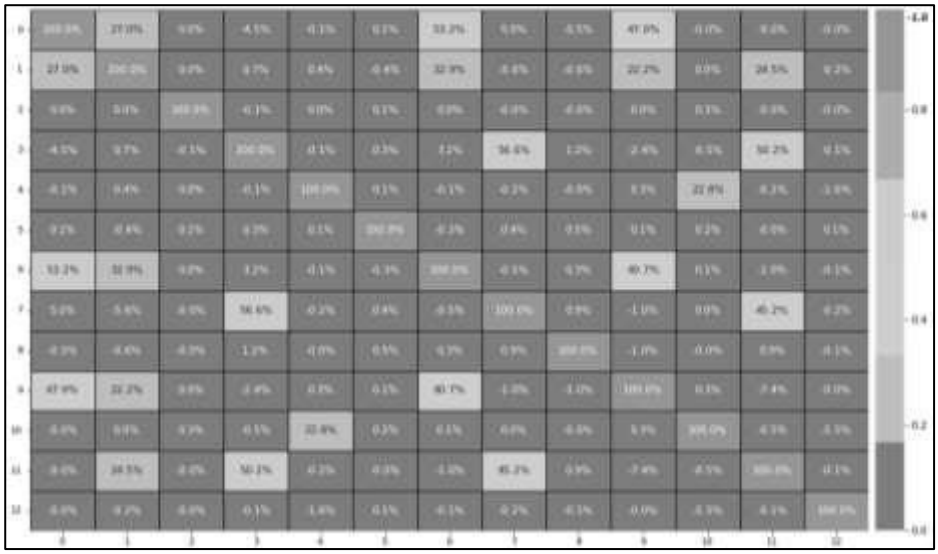


Figure-7. Correlation Heatmap of Training Data

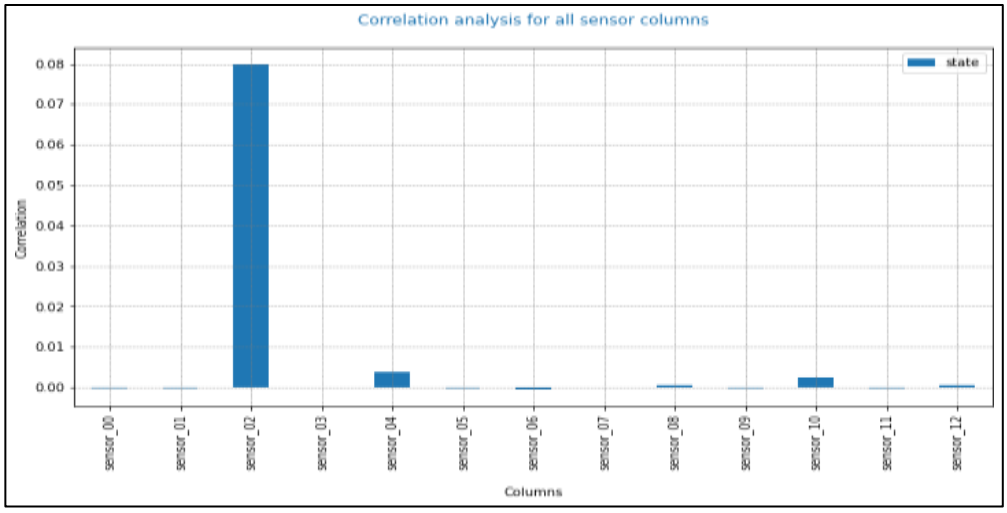


Figure-8. Correlation Analysis of Sensor Data

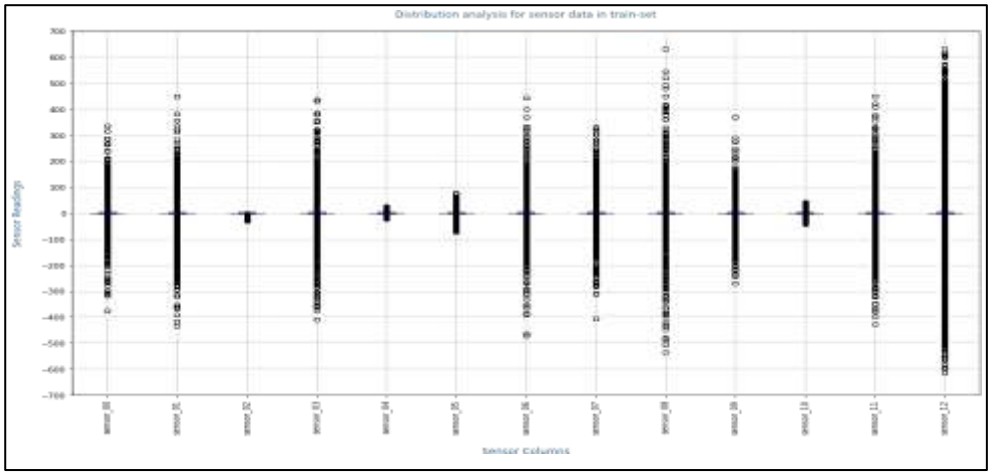


Figure-9. Distribution of Training Data

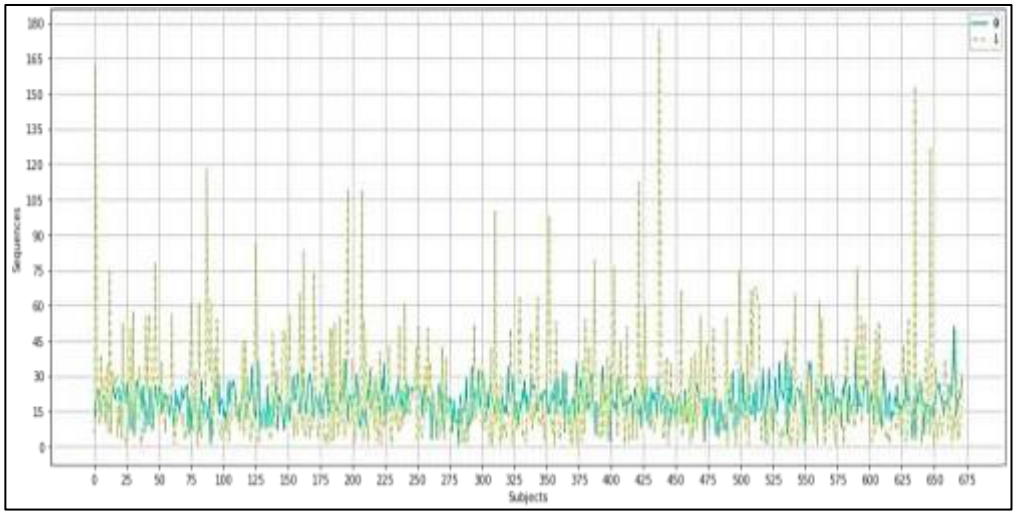


Figure-10. Data Redundancy Identified

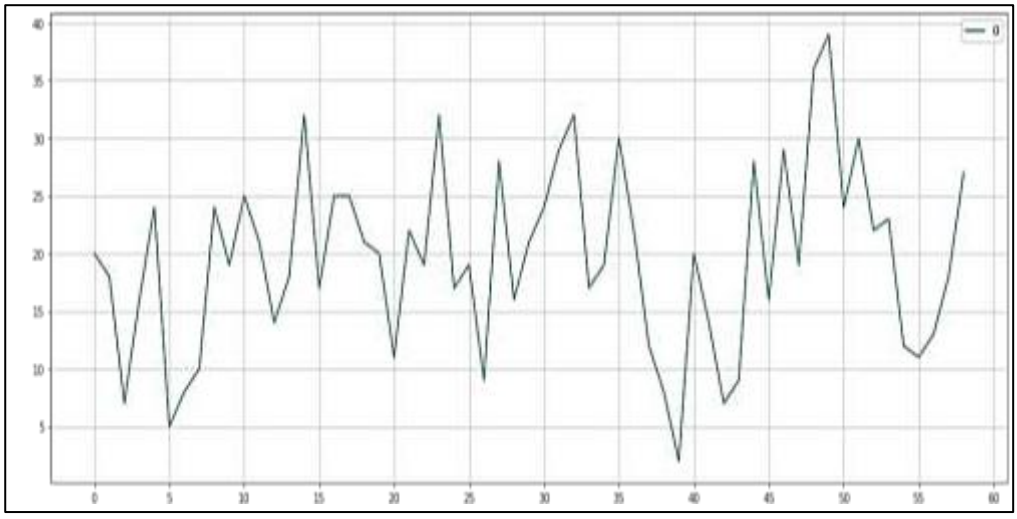


Figure-11. Data Redundancy Eliminated

Figure-9 shows that most of the data reading values are available in a common data range, like -500 to +500. From the experiment, the data sequence is verified by calculating the mean value, the total number of elements present in the data sequence, and the sum of the elements. The sensor data is visualized and associated with subjects, and duplicates will be eliminated. The majority of the sensor data are duplicated with respect to time and subjects, which increases the data density, computational, and time complexity. The sensor data with data redundancy is shown in Figure-10. After duplicate elimination, the output data is shown in Figure-11, which illustrates that no duplicate data is available, reducing the computational complexities. The difference between the data with and without redundancy can be seen in Figure-10 and Figure-11. After eliminating the missing elements, redundancy, and wrong data from the sensor, data is obtained from the experiment, as shown in Figure-12. It shows the preprocessed data for sensor-00 to sensor-12. Figure-12 clearly shows the abnormalities, outliers, and other negative data impacts and will be predicted for forecasting. The performance of the proposed work is evaluated by computing and comparing the learning process efficiency and time complexity of the training and testing phases. The feature size learned with the time complexity of the training and testing phases is given in Table-1.

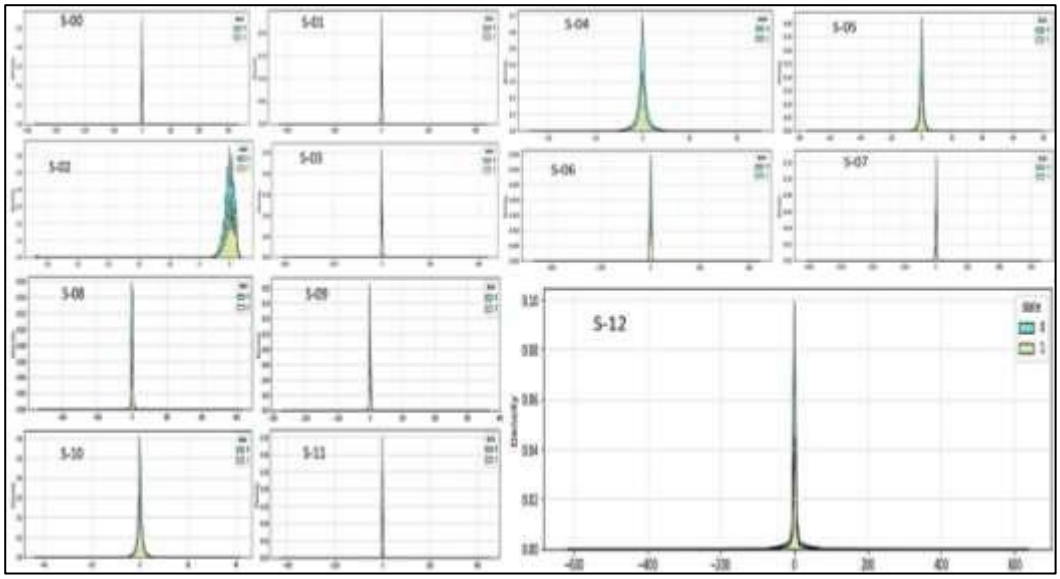


Figure-12. Data Preprocessed for Each Sensor

Table-1. Performance Comparison of Training and Testing Phases

Phase	Before Pre-Processing		After Pre-Processing	
	Feature Size	**Time (sec)**	**Feature Size**	**Time (sec)**
Training	48,498	48	25,968	21
Testing	26,435	27	12,218	13

Table-2. Performance Evaluation W.R.T Accuracy

Preprocessing Methods	Dataset	Accuracy
GDM [12, 13]	Haberman Dataset	95.55%
GDC [12, 13]		97.61%
GDL [12, 13]		96.66%
GDGL [12, 13]		96.71%
GDM [12, 13]	Wine Recognition Dataset	95.11%
GDC [12, 13]		96.90%
GDL [12, 13]		94.32%
GDGL [12, 13]		98.99%

GDM [12, 13]	IRIS Dataset	98.41%
GDC [12, 13]		99.87%
GDL [12, 13]		98.01%
GDGL [12, 13]		99.12%
EDA	**Tabular Play-Ground Series- 2022 year**	**99.89%**

Table 1 shows that the complexity of the testing phase is less than that of the training phase. The computational and time complexity increases for the increased data size, whereas the testing complexity is less than in the training phase. The performance evaluation compares the classification accuracy obtained using the proposed EDA method with similar methods. The accuracy comparison is given in Table-2, showing that EDA's accuracy is higher than others. EDA uses only statistical and mathematical operations and provides high accuracy compared to other methods. The entire dataset is preprocessed and qualified for further data analytics processing. Thus, it is suggested that the EDA method is highly suitable for data preprocessing.

Conclusion

Processing unqualified data degrades the performance of the analytical model and the other data-related processes. So, the data analytics industry requires a preprocessing model to normalize and improve the data quality to enhance the data processing outputs. The IoT data streaming process also needs preprocessing on the data since IoT sensors generate continuous time series data with missing elements, more redundancy, wrong data, and mismatched data in the recorded data. It happens due to sudden climate changes, air pollution, and other natural disasters. This problem is considered a major research problem, and this paper has aimed at designing and implementing Exploratory Data Analytics, an Artificial Intelligence technique for analyzing and preprocessing the raw IoT-Bigdata and improving its quality, which can enhance the efficiency of data streaming, analyzing, and forecasting concerning the application where the IoT devices are deployed. The analytical model is implemented and experimented with Python and datasets taken from Kaggle.com, and the results are verified. The results obtained were compared with similar methods, and it was found that the proposed work obtained 99.89% accuracy in data preprocessing.

Future Work

In the future, the IoT devices and network interruptions will be verified to improve the IoT-Bigdata streaming process.

References:

1. Segura-Garcia, J., Felici-Castell, S., & Garcia-Pineda, M. (2018). Performance evaluation of different techniques to estimate subjective quality in live video streaming applications over LTE-Advance mobile networks. Journal of Network and Computer Applications, 107, 22-37.
2. Malek, Y. N., Kharbouch, A., El Khoukhi, H., Bakhouya, M., De Florio, V., El Ouadghiri, D., ... & Blondia, C. (2017). On the use of IoT and big data technologies for real-time monitoring and data processing. Procedia computer science, 113, 429-434.

3. Cai, H., Xu, B., Jiang, L., & Vasilakos, A. V. (2016). IoT-based big data storage systems in cloud computing: perspectives and challenges. IEEE Internet of Things Journal, 4(1), 75-87.
4. Feng, C., Adnan, M., Ahmad, A., Ullah, A., & Khan, H. U. (2020). Towards an energy-efficient framework for IoT big data healthcare solutions. Scientific Programming, 2020, 1-9.
5. Wongthongtham, P., Kaur, J., Potdar, V., & Das, A. (2017). Big data challenges for the Internet of Things (IoT) paradigm. Connected Environments for the Internet of Things: Challenges and Solutions, 41-62.
6. Arthanari, J., & Baskaran, R. (2019). Enhancement of video streaming analysis using the cluster-computing framework. Cluster Computing, 22(Suppl 2), 3771-3781.
7. Chen, B. W., Ji, W., Jiang, F., & Rho, S. (2015). QoE-enabled big video streaming for large-scale heterogeneous clients and networks in smart cities. IEEE Access, 4, 97-107.
8. Cao, H., & Wachowicz, M. (2019, October). Analytics Everywhere for streaming IoT data. In 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS) (pp. 18-25). IEEE.
9. Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2019). Verification and validation techniques for streaming big data analytics in the Internet of Things environment. IET Networks, 8(3), 155-163.
10. Mehmood, E., & Anees, T. (2020). Challenges and solutions for processing real-time big data stream: a systematic literature review. IEEE Access, 8, 119123-119143.
11. Mirzaie, M., Behkamal, B., Allahbakhsh, M., Paydar, S., & Bertino, E. (2023). State of the art on quality control for data streams: A systematic literature review. Computer Science Review, 48, 10055.
12. Cristianini N., J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge UK, 2000.
13. Nawi, Nazri Mohd, Atomi, Walid Hasen, Rehman, M.Z, (2013), "The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks," Procedia Technology, 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013, Vol. 11, pp.32-39.