symmetry

MDPI

Article

Generative Adversarial Networks (GAN) and HDFS-Based Realtime Traffic Forecasting System Using CCTV Surveillance

Praveen Devadhas Sujakumari [1,*] and Paulraj Dassan [2]

[1] Department of Computer Science and Engineering, R.M.K. College of Engineering and Technology, Puduvoyal 601206, India
[2] Department of Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai 601206, India
* Correspondence: praveencse37@gmail.com

Abstract: The most crucial component of any smart city traffic management system is traffic flow prediction. It can assist a driver in selecting the most efficient route to their destination. The digitalization of closed-circuit television (CCTV) systems has resulted in more effective and capable surveillance imaging systems for security applications. The number of automobiles on the world's highways has steadily increased in recent decades. However, road capacity has not developed at the same rate, resulting in significantly increasing congestion. The model learning mechanism cannot be guided or improved by prior domain knowledge of real-world problems. In reality, symmetrical features are common in many real-world research objects. To mitigate this severe situation, the researchers chose adaptive traffic management to make intelligent and efficient use of the current infrastructure. Data grow exponentially and become a complex item that must be managed. Unstructured data are a subset of big data that are difficult to process and have volatile properties. CCTV cameras are used in traffic management to monitor a specific point on the roadway. CCTV generates unstructured data in the form of images and videos. Because of the data's intricacy, these data are challenging to process. This study proposes using big data analytics to transform real-time unstructured data from CCTV into information that can be shown on a web dashboard. As a Hadoop-based architectural stack that can serve as the ICT backbone for managing unstructured data efficiently, the Hadoop Distributed File System (HDFS) stores several sorts of data using the Hadoop file storage system, a high-performance integrated virtual environment (HIVE) tables, and non-relational storage. Traditional computer vision algorithms are incapable of processing such massive amounts of visual data collected in real-time. However, the inferiority of traffic data and the quality of unit information are always symmetrical phenomena. As a result, there is a need for big data analytics with machine learning, which entails processing and analyzing vast amounts of visual data, such as photographs or videos, to uncover semantic patterns that may be interpreted. As a result, smart cities require a more accurate traffic flow prediction system. In comparison to other recent methods applied to the dataset, the proposed method achieved the highest accuracy of 98.21%. In this study, we look at the construction of a secure CCTV strategy that predicts traffic from CCTV surveillance using real-time traffic prediction analysis with generative adversarial networks (GAN) and HDFS.

Keywords: Intelligent Transportation System (ITS); image synthesis; generative adversarial networks (GAN); traffic management system; Hadoop Distributed File System (HDFS)

Citation: Devadhas Sujakumari, P.; Dassan, P. Generative Adversarial Networks (GAN) and HDFS-Based Realtime Traffic Forecasting System Using CCTV Surveillance. *Symmetry* **2023**, *15*, 779. https://doi.org/10.3390/sym15040779

Academic Editor: Silvio Pardi

Received: 25 January 2023
Revised: 28 February 2023
Accepted: 6 March 2023
Published: 23 March 2023

1. Introduction

Deep learning networks have proven to be effective tools in a variety of difficult fields, including computer vision, healthcare, business, and finance, among others. For example, machine learning technologies are being employed in a growing number of Internet of Things (IoT) devices for botnet detection and network traffic analysis (28.1 billion devices in 2020; trillions in 2025) [1]. Modern society is characterized fundamentally by urbanization

and an increase in building density. This way of living not only benefits the economy but also offers new difficulties for local government. Effective traffic management and analysis are some of these problems. Numerous personal cars, an increase in the number of freight vehicles used to transport products and commodities, and clogged pedestrian traffic are all effects of high population density. Suboptimal algorithms cannot be employed to address transportation issues based on scant manually obtained statistics. To make effective judgments, forecast their effects, and evaluate their outcomes, authorities need an automated system for analyzing citywide traffic flow.

The rate of data growth is accelerating due to the expansion of computers, smartphones, the internet, and sensory devices. Big information is generally understood to refer to data collections that are too massive or complicated for conventional relational databases to efficiently acquire, maintain, and handle. Symmetric IoT applications will eventually become a significant new data source with the aid of machine learning (ML) techniques. The five Vs are used to summarize the characteristics of big data [2]. These five Vs are volume, velocity, variety, veracity, and value. A lot of data are gathered by mobile devices, social media, the Internet of Things (IoT), and other sources.

The most common problems when evaluating real-time data from city cameras are low counting accuracy, classifying a small number of vehicle types, and following an object while recognizing the speed and driving direction in all sections entering the functional zone of the crossroads. Little research has been done to gather and analyze traffic flow speed and movement patterns utilizing survey street cameras, despite the obvious compensations of building such organizations [3]. Large volumes of video data have been successfully gathered, interpreted, and analyzed using artificial neural networks [4].

Many deep learning-based techniques that analyze time-series traffic data have been created recently to forecast traffic volume [5]. Since there is no evident order dependence for traffic predictions, it is crucial to consider all the data of a period identically to create a series of upcoming trip times using modeling of the dispersion of previous periods, which time-series techniques cannot manage. The basis of deep learning is therefore thought to be generative adversarial networks (GANs). Because they offer segments based on the predicted probability density function of a challenging dataset, they are potential picture-generating algorithms.

Many devices today create data anywhere and at any time. Data grow exponentially and becomes a complex item that must be managed. Unstructured data are a subset of big data that are difficult to process and have volatile properties. CCTV cameras are used in traffic management to monitor a specific point on the roadway. CCTV generates unstructured data in the form of images and videos. However, the inferiority of traffic data and the quality of unit information are always symmetrical phenomena. Unlabeled data are widely available and easy to obtain, so their number is enormous. However, because there is a lack of subjective evaluation information, evaluation results based on unlabeled data frequently differ to reflect public opinion. Because of the data's intricacy, these data are challenging to process. The generative adversarial network (GAN) is one such efficient technology that has piqued the curiosity of the scientific community. This study describes the creation of a traffic congestion dataset that has been enhanced via GAN-based augmentation. The GAN-enhanced traffic congestion dataset is then utilized to train artificial intelligence (AI)-based algorithms [6].

Numerous studies have relied heavily on in-ground or side induction sensing devices while utilizing typical traffic sensor infrastructure for traffic prediction [7]. The global navigation satellite systems (GNSS) integrated into cellphones [8] and those mounted on taxis [9] are two more ubiquitous traffic sensors. However, traffic control organizations may find it costly to deploy the sensor infrastructure at various points throughout a city or in hundreds of cabs. CCTV systems, which are extensively used for traffic surveillance, object tracking (such as automatic number plate recognition (ANPR) systems [10], tracking, and event detection applications [11], can replace accessible and affordable sensor infrastructure. Recent work on prediction utilizing CCTV datasets has received less attention than that

using loop detectors and GNSS sensors (e.g., in [12]). Only one third of the literature, according to a recent article in [13], is concentrated on urban arterials, and more research has been done on highways or motorways than on forecasting urban traffic. Thanks to initiatives such as the Urban Observation (UO) project in the Northeast of England, run by York University, CCTV datasets from several local authorities in the UK have lately become publicly accessible.

The big data ecosystem most widely recognized is Hadoop. After the release of Yet Additional Resource Negotiator (YARN) [14], processing Hadoop distributed file organization data using batch-oriented MapReduce is no longer the only option (HDFS). There are several Hadoop processes and harvests that really can run on Hadoop that support several computer languages, such as Python, Java, and Scala. Additionally, they provide simple access to the countless machine-learning algorithms needed for effective smart city management. Spark is one such tool which also supports Python, Java, and Scala. Spark supports Python, Java, and Scala. The primary benefit of the Hadoop construction stack is that it is exposed basis and cost-free to utilize. The biggest drawbacks of this product are its limited support and the well-known security flaws that affect most NoSQL databases and products in the Hadoop ecosystem. These problems can be resolved by using client-to-node encryption, using third-party products like Kerberos, and other techniques. However, it takes highly trained professionals to bring these and the rest of the infrastructure into operation.

This paper's primary contributions are as follows:

- This study suggests using big data analytics to transform unstructured real-time CCTV data into knowledge shown in a web dashboard.
- An architectural framework based on Hadoop that can serve as the ICT backbone for effectively managing these unstructured data. The Hadoop distributed file scheme (HDFS) stores a variety of data types using the Hadoop file storage system, HIVE databases, and non-relational storage.
- Based on current developments in vehicle recognition and tracking tasks, we have suggested and applied a unique traffic flow prediction organization.

We carefully examined our system and presented empirical proof that the suggested solution is accurate enough to serve as the basis for further high-level models.

The remainder of this article is structured as surveys: in Section 2, we examine research on traffic forecasting techniques. After that, in Section 3, the approach for the generative adversarial connection is explained. It entails the chosen architecture, rolling time-domain training and testing, a perfect fusion mechanism, and multi-dimensional indicators. The experimental findings are examined and contrasted with benchmark techniques in Section 4. The contributions of this study and future work are discussed in Section 5's conclusion.

2. Literature Survey

A deep overlay unidirectional, bidirectional LSTM neural network topology was presented by Cui et al. [15]. The network traffic velocity could be predicted thanks to the time series data's forward and backward correlation. Since the algorithm makes predictions based on historical traffic data, it can anticipate traffic speed on motorways and compound urban traffic networks. Deep learning-based GANs can overcome these limitations. The complicated and nonlinear interplay of instant traffic flow sequences has led to the presentation of a more hybrid prediction framework integrating parametric and non-parametric approaches.

By Du et al. [16], among the UAV applications that require real-time object detection are substructure inspection, search and rescue, and reconnaissance and investigation. A brand new method for estimating generative models that Goodfellow et al., have proposed makes use of adversarial networks. In the proposed approach, they simultaneously train two models: (a) a reproductive model that collects data distributions; and (b) a judicial model that calculates the likelihood of the model's presence using exercise data. This method does not require any unrolled approximate inference networks or Markov chains during

training or sample creation. The experimental findings represented a significant advance in this area and inspired the researchers to apply GAN networks to more relevant issues.

Active generative adversarial networks were described by Kong et al. [17] for picture categorization. The active learning strategy is recommended because it facilitates the collection of annotations by choosing samples with a high possibility of performance improvement. Since labeling information is difficult to obtain exclusively, active learning is preferred. In this study, GAN networks are combined with only an active learning technique to develop strong candidates. Each sample receives a unique reward to quantify the level of ambiguity, which then prompts the CGAN to generate illuminating examples for a given label.

To predict traffic density shortly, Li et al. [18] proposed a gradient-boosting method in addition to the hierarchical reconciliation. Three aspects of traffic flow that drew a lot of attention were the spatial and temporal models, their relationships, and the dynamics of traffic density across various spatial at aggregate levels. Utilizing three different datasets, a comparison with SARIMA, a Kalman filter model, and RF approaches, the presentation of the planned context was assessed. By using the gradient-boosting method, which provides automated and extremely flexible ways to learn, information in large datasets is educated. The traffic forecasting accuracy in a large road network will benefit greatly from this information. No pollution data were included, even though they are projected over a longer time horizon.

Tang et al. [19] demonstrated that handmade features such as the histogram of directed gradients and color histograms can be used in the absence of training data. Computing visual attributes for every tracked object considerably increases the computational workload, particularly when there are a lot of monitored objects. The cumulative performance is rarely close to real-time when combined with detector processing time.

Lin et al. [20] created a multi-quantity based on a GAN that could improve the accuracy of a traffic prediction for actual values by using the joint probabilities of accuracy to minimize the disparity in various traffic distribution situations. The combined dispersion of relative entropy was used to reduce the discrepancy between the various traffic dispersal conditions. This objective was achieved through the reduction of the discrepancy between the different traffic distribution scenarios.

To reduce the time-consuming training, superior communication costs, and data privacy risks of global GCNs as the count of data increases, Xia et al. [21] present a short-term traffic flow predictive approach that combines community detection-based federated learning with a graph convolutional network (GCN). The Federated Community GCN, or FCGCN, seeks to produce a traffic state prediction that is precise, fast, and safe based on a period of extensive traffic information. This is crucial for the smooth running of intelligent transport organizations.

Haryana et al., cover the use of the Indian Navigation Satellite (INS) and the global positioning system (GPS) [22] to estimate location and velocity. The Kalman filter is used to integrate the signals from the INS and GPS sensors. The dilution of Precision (DOP) technique is used to choose a selection of satellites to be used as range data. There are two types of Kalman filters used: feedforward and feedback. The experiment demonstrates how the chosen satellites affect the measurements. The autonomous UAV was constructed and tested using the method and procedures described in this article.

Wang et al. [23] created an efficient and accurate malware detection system through merging convolutional neural networks (CNNs) with generative adversarial networks (GANs). First, they constructed a code visualization approach and used GAN to generate more examples of harmful code variants for data augmentation. Then, CNN created the lightweight AlexNet to classify malware families.

Wang et al. [24] suggests an algorithm for jointly detecting and locating multiple beams-stealing attackers based on RSSI (received signal strength indicator) maps without the training procedure associated with deep learning-based approaches. The first step is to create an RSSI map using interpolation of the raw RSSI data to enable high-resolution

localization while lowering monitoring costs. To detect and locate many attackers without having any prior information on the attackers, three image processing processes, including edge detection and segmentation, are carried out on the created RSSI map.

Jilani et al. [25] describes the creation of a traffic congestion dataset that has been enhanced via GAN-based augmentation. The GAN-enhanced traffic congestion dataset is then utilized to train AI-based traffic congestion models. In this paper, a five-layered convolutional neural network (CNN) deep learning model for traffic congestion classification is developed. The suggested model's performance is compared to that of several other well-known pretrained models, including ResNet-50 and DenseNet-121.

Wang et al. [26] proposed that traffic photos from the present surveillance system in Shaanxi Province are recovered and pre-processed to create a proper training dataset, including diverse illumination, weather circumstances, and vast scenarios. To detect traffic congestion, a network topology based on residual learning is developed, and may be pre-trained and fine-tuned. The network is then moved to the traffic application and retrained using a self-created training dataset to produce the TrafficNet.

Khazukov et al. [27] aimed to create a full and high-quality system for collecting real-time data, including average vehicle speed, driving directions, and traffic flow intensity. At the same time, data are gathered throughout the whole functional area of intersections and surrounding road segments that are covered by the angle of the street video surveillance camera. The existing survey for traffic flow prediction is shown in Table 1.

Table 1. Literature survey for traffic flow prediction.

Author	Proposed System	Contribution	Comparison	Dataset Used
Zang et al. [28]	STGI-ResNet	Large-scale traffic prediction using a novel spatiotemporal DL architecture	Only 1-month data are considered	Chengdu, China's Didi GAIA initiative traffic data set
Cui et al. [15]	LSTM-BDLSTM	Observed spatiotemporal linkages between time's past and future	Only one model was contrasted	LOOP-SEA Dataset, PeMS-BAY Dataset
Guo et al. [29]	CNN+RNN+C3D	A project involving large-scale traffic flow prediction	Compared to only 2 models	Dataset on Shenzhen, China, traffic flow
Cheng et al. [30]	STKNN	Proposed method to take traffic spatial variation into account	Only focused on predicting short-term traffic	traffic information from PeMS and Beijing floating car speed information
Kumar et al. [31]	GCNN	Spatiotemporal video prediction using transfer learning	Only used video information	Indian Driving Dataset (IDD)
Ketabi et al. [32]	Numerous-variable RNN	Superior outcomes compared to other camera-based studies	Only used camera-generated data	Images from London's traffic cameras
Zang et al. [33]	CNN-STFSA	Considered the temporal and spatial characteristics of traffic flow	High error rate and low sample size	Washington State Transportation Department data set
Lu et al. [34]	LSTM-MDC	For traffic prediction, it is advised to employ a unique DL-based network	Fewer traits were employed	Information on traffic from PeMS
Rahman F. I [35]	KNN SVM ANN	Weather-informed short-term TFP using machine learning's KNN, SVM, and ANN	Short term prediction	Transportation Infrastructure Ireland (TII)

3. Proposed System

Current traffic information, historical traffic information, and any other relevant statistical data are utilized in the development of an appropriate mathematical method that is used to develop a traffic flow prediction shown in Figure 1. It is possible to use a method of intellectual computation to produce more accurate projections of the amount of traffic that will occur in the years to come. The findings may provide a plausible basis for vehicle dynamic guidance and an urban circulation regulator if they are implemented.

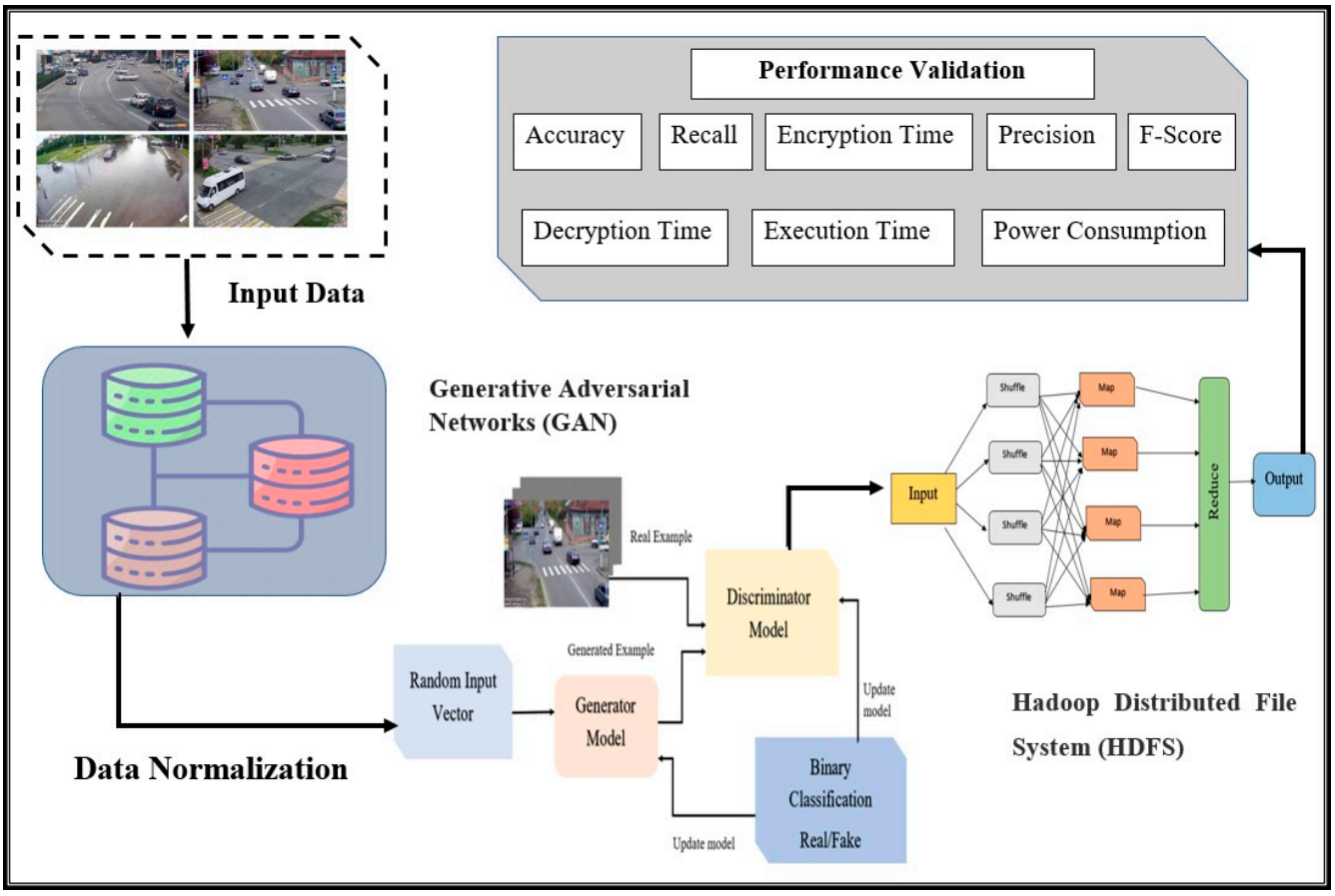

Figure 1. The proposed method of GAN-HDFS.

3.1. Data Collection

A data warehouse called the Hive project runs on the Hadoop framework. Its query language, HiveQL, implements most of the SQL-92 standard; however, unlike conventional relational databases, it supports the schema-on-read technique. Using this technique, the information may be taken in and saved, and if necessary, tables can be constructed on top of the information to facilitate data searching. A database, as well as text files and RC Files, is used to store the data (by default, Apache HBase). Apache Derby is used to implicitly store the metadata; however, other relational databases, such as MySQL, may be used instead. Spark [36], MapReduce, and Tez are the other three options for Hive to use as its execution engine. Sample images from CCTV are shown in Figure 2.

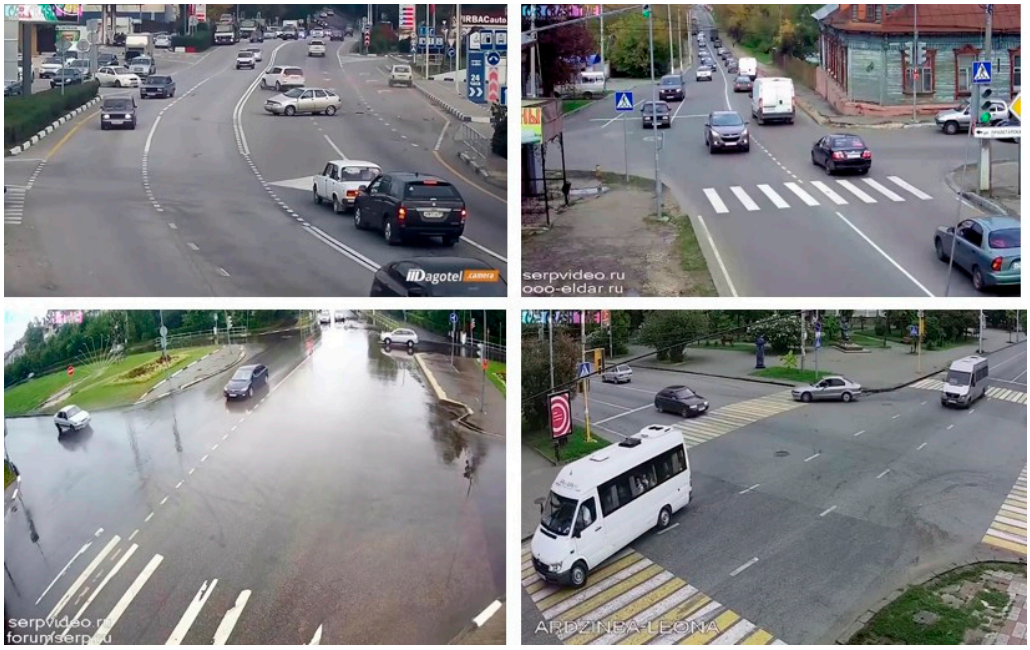

Figure 2. Sample images from CCTV.

3.2. Data Normalization

The presented GAN-HDFS model uses min–max normalization to begin the process of traffic data normalization at the beginning level. The min–max method applies to this piece of work provided that the result of the linear alteration procedure applied to the initial information is a range that falls within the boundaries [0,1].

$$x_{Scale} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

If $x_{\min}$ and $x_{\max}$ were the attributes' minimal and maximal values, respectively, and x_{Scale} denotes the measure that is close to the characteristic matching processes.

3.3. Traffic Flow Prediction Using GAN-HDFS

When it comes to deep learning algorithms, the most important aspects to consider are the availability of a database, the algorithm's selection and application, and the algorithm itself. This instruction places a lot of emphasis on the usage of information augmentation because of the traffic database's limited reach. As a result, a study of a novel approach known as sequential generative adversarial networks GAN-HDFS was carried out to address the issue of traffic congestion [37]. The temporal dynamics of roadways would be accurately modeled using this technique to produce realistic traffic scenarios. In a recent paper [38,39], researchers developed a novel approach for forecasting traffic flow that analyses real-time traffic datasets using generative adversarial networks. Comparing this strategy to the baseline method, it made noticeable progress. An inventive deep learning framework has been presented for traffic state estimation because the traffic states that are readily available in the real world are frequently insufficient. Real-time production of traffic road statuses will be possible using this system.

3.3.1. Generative Adversarial Networks (GAN)

Utilizing data augmentation and image modification techniques, the GAN perfect is used to create synthetic imageries that look very similar to the originals. In essence, GAN makes use of a min–max game and a value purpose called V (DR, GR)), where DR is the discriminator and GR is the generator. Exploiting the cost for the producer while minimizing the gain for the discriminator are the two competing objectives that are at stake.

The next loss function can be lowered or increased, and this is the aim of both the generator (GR) and the discriminator (DR). To generate samples GR (rn), the GAN generator is run with a random noise P(rn) devoted to the early input information x. Here, (rn) stands for the random noise variable. The GAN generator creates these samples, as shown in Figure 3.

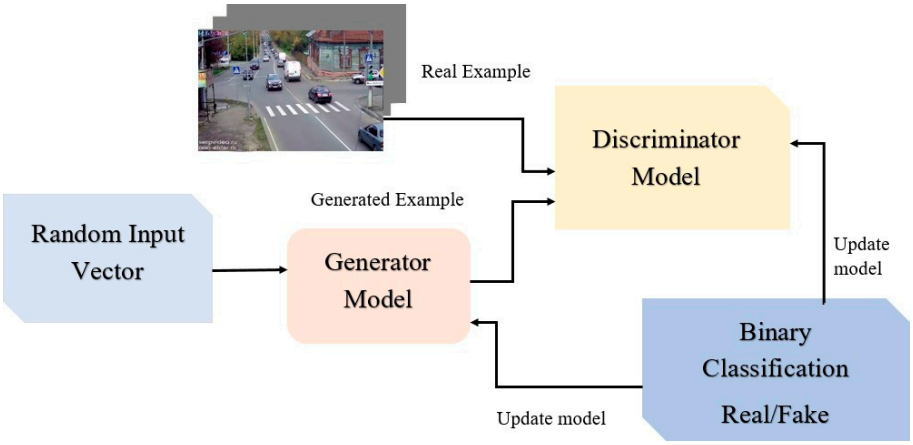

Figure 3. Representation of the GAN architecture.

One could consider the noise to be a latent demonstration of the information that the producer is trying to study. Every element of the "noise" vector can be thought of as a feature that is given to the producer; the more valuable a feature, the better the image that is produced. A case x may or may not be present in data distribution *Pd*, depending on how a discriminator assesses the probability. The discriminator uses a mathematical formula known as *DR (GR(rn))* to determine whether a specific instance is false or real. The generator tries to generate images that are almost perfect to deceive the discriminator. Even though it is impossible to tell the difference between actual and fraudulent samples, the discriminator attempts to improve its performance by doing so [40–44]. The discriminator wants to make the next loss function bigger, whereas the generator wants to make it smaller. After receiving the input x, the GAN generator adds the random noise component *P(rn)* to the equations to produce samples *GR(rn)* [45–48]. The discriminator's parameter DR represents the assessment of the probability that an actual case of x occurs from the entire data distribution *Pd(x)* [49–52]. The encoder must first comprehend the representation feature to achieve its main objective, and the discriminator must look for the discriminating characteristic [53–55].

$$\min[\max[V(DR, GR)]] = -\mathbb{E}_{\mathcal{S}\sim pd(x)}[\log\ DR(x)] + \mathbb{E}_{rn\sim prn}[\log(1 - DR(GR(rn)))] \tag{2}$$

$$\mathcal{J}^{pixel}_{recons} = \mathbb{E}_{\coprod\sim DR_{Encoder}\ (s), s\sim I_{real}}[||\boldsymbol{K}(q) - \tau(\boldsymbol{\mathcal{S}})||] \tag{3}$$

$$\begin{bmatrix} p' \\ q' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \tag{4}$$

$$p' = -p,\ \ q' = \ \ q \tag{5}$$

$$\begin{bmatrix} p' \\ q' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \tag{6}$$

$$p' = p,\ \ q' = -q \tag{7}$$

To augment data with high-quality images and improve segmentation performance for the resulting dataset, the GAN is used.

3.3.2. Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System, or HDFS, is a distributed file system created to store enormous volumes of data (hundreds of terabytes and even terabytes of data) and offer high-throughput access to these data. Files are redundantly stored across many machines to guarantee their resilience to failure, and are highly available for highly parallelized applications. A software framework called Hadoop Map-Reduce processes large amounts of data concurrently and fault-tolerantly. It is used to process enormous datasets because of its high efficiency and scalability. Master/slave architecture is used by HDFS. An HDFS cluster typically includes an Effectively Set, secondary Name Nodes, a single Name Node, and at minimum three Data Nodes. The Master Server panels client entrance to files and manages the file scheme namespace.

All the data files in the Hadoop cluster are stored in HDFS, the essential component of distributed computing. The file state's administrator, Primary Name Node, oversees managing user document access, the namespace, and the descriptive metadata for the file system. The secondary Name Node is a name for Name Node's backup. By dividing the regional disc into many storage blocks, each Data Storage block stores information in this manner. Periodically, the Data Node must report to the Name Node. Additionally, chunking of files improves throughput. A variety of Data Nodes, normally three ends in the cluster, oversee managing the memory that is connected to the network that they run on. Writing map operators and a reduce function are required for map-reduce programming.

The map function returns a list of the intermediate values together with the key after taking a key and a couple of intermediate levels. The chart purpose, which is constructed in a method that permits several map functions to operate concurrently, is the component of the computer that splits tasks shown in Figure 4. The output of the map functions is processed in some way when the reduce function is used—typically by joining values—to obtain the wanted result in a production folder. Thus, the Map Decrease closely follows the design and nomenclature provided by Dean and Ghemawat [56–59] to parallelize calculations.

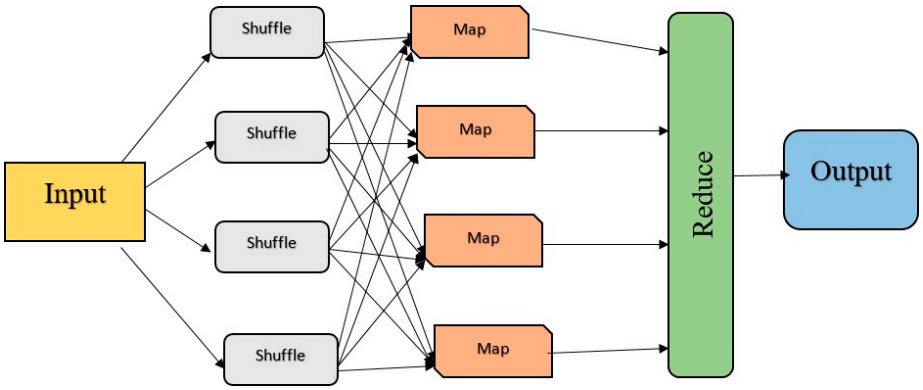

Figure 4. Map-Reduce Framework.

4. Result and Discussion

4.1. Experimental Setup

The system that was designed was evaluated using two metrics. We first trained and assessed the detection network and all suggested upgrades using our new dataset. The outcomes of the traffic flow estimation problem were then assessed. We processed surveillance data using the technology during peak hours to achieve this. The number of cars moving through each of the four popular congestion directions was compared with the actual data gathered by on-the-ground observers. All tests were run on the Ubuntu 18.04 operating system with an Nvidia RTX 2080 Ti GPU, 12 Intel Core i7-8700 CPU cores operating at 3.20 GHz, and 32 GB of RAM. On top of Fb Detectron's official realization [60–62], we built our solution.

4.2. Performance Metrics

Five different presentation parameters were used to check the precision of the outputs that the trained model generated. Accuracy, recall, precision, F1 score, and energy consumption [63–68] were used to assess the model's presentation.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{9}$$

$$F\text{-}Score = 2 * \frac{precision * recall}{precision + recall} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\begin{aligned} &Energy\ Consumption \\ &\mu = mean(x) \\ &\mu(X) = \frac{\sum (xi - \mu)^2}{N - 1} \end{aligned} \tag{12}$$

4.2.1. Precision

A comparison of precision of the GAN-HDFS approach and other existing techniques is shown in Figure 5 and Table 2. The graph demonstrates how the machine learning strategy has an improved performance with precision. For instance, the GAN-HDFS model has a precision value of 93.83% for data 100, compared to the precision values of 78.43%, 81.54%, 74.29%, 84.21%, and 89.43% for BiLSTM, CNN, RNN, GAN, and ARIMA models, respectively. The GAN-HDFS model has nonetheless demonstrated its best performance with various data sizes. The precision value of the GAN-HDFS is 96.21% under 350 data, compared to 80.13%, 82.54%, 75.83%, 86.11%, and 91.65% for BiLSTM, CNN, RNN, GAN, and ARIMA models, respectively.

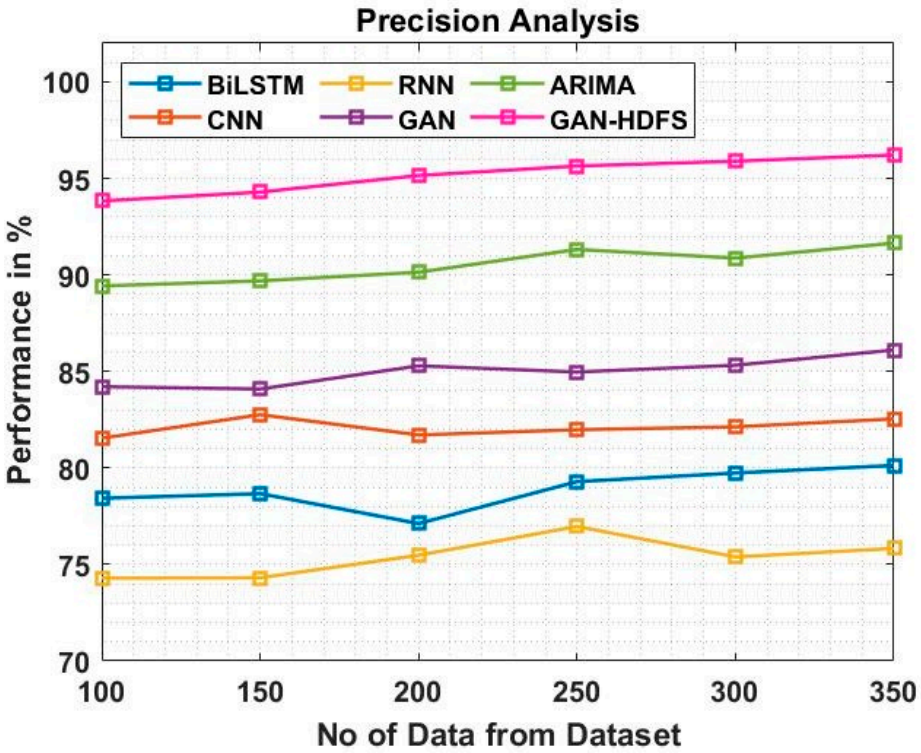

Figure 5. Precision analysis for GAN-HDFS with existing systems.

Table 2. Precision analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	78.43	81.54	74.29	84.21	89.43	93.83
150	78.67	82.76	74.31	84.09	89.69	94.29
200	77.12	81.71	75.48	85.29	90.14	95.15
250	79.29	81.98	76.97	84.97	91.32	95.64
300	79.74	82.13	75.39	85.32	90.87	95.89
350	80.13	82.54	75.83	86.11	91.65	96.21

4.2.2. Recall

Comparative recall testing between the GAN-HDFS strategy and other approaches is shown in Figure 6 and Table 3. The chart demonstrates that the machine learning strategy led to an improved recall performance. For instance, the recall value for the GAN-HDFS model with data 100 is 92.54%, while the recall values for BiLSTM, CNN, RNN, GAN, and ARIMA models are 81.23%, 69.32%, 84.90%, 75.32%, and 88.21%, respectively. The GE-DBN model has nonetheless demonstrated its best performance with various data sizes. Similarly, the recall value of the GAN-HDFS is 97.54% under 350 data, whereas the recall values of BiLSTM, CNN, RNN, GAN, and ARIMA models are 82.43%, 72.89%, 86.19%, 77.76%, and 91.76%, respectively.

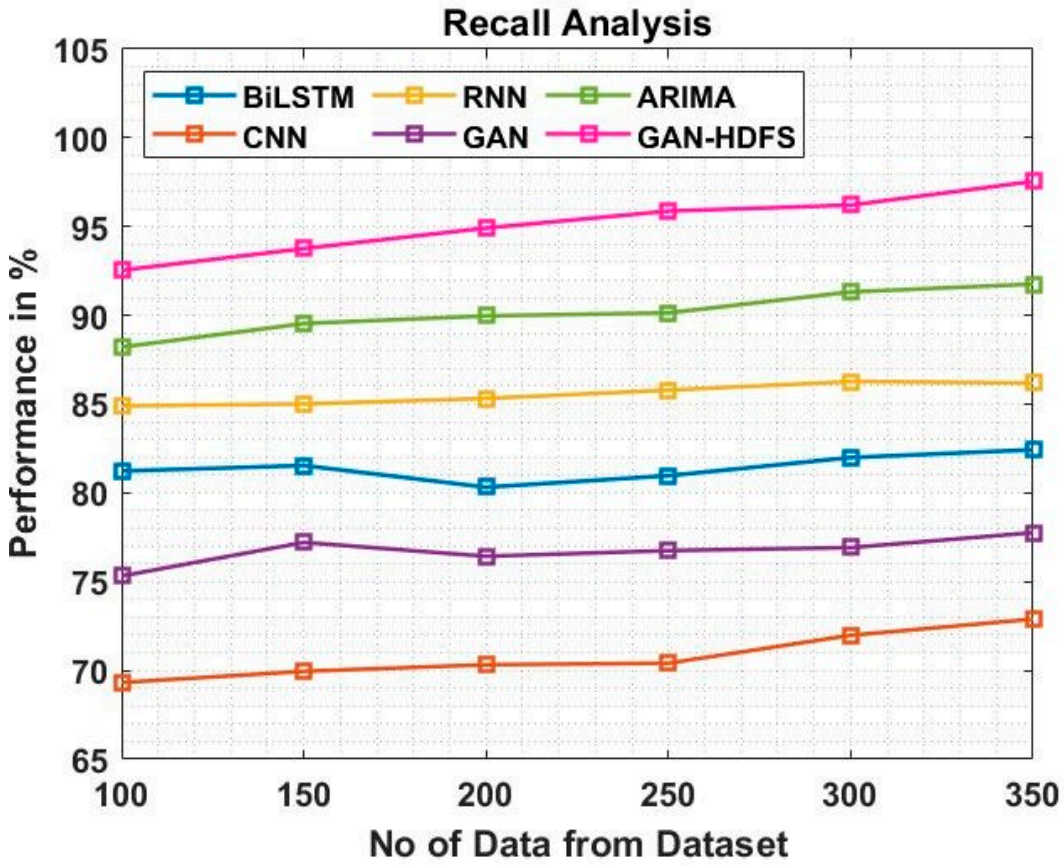

Figure 6. Recall analysis for GAN-HDFS with existing systems.

Table 3. Recall analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	81.23	69.32	84.90	75.32	88.21	92.54
150	81.54	69.95	85.01	77.21	89.54	93.76
200	80.32	70.32	85.32	76.43	89.97	94.92
250	80.97	70.41	85.79	76.75	90.14	95.87
300	81.99	71.98	86.27	76.92	91.32	96.21
350	82.43	72.89	86.19	77.76	91.76	97.54

4.2.3. F-Score

A comparison of F-Score analysis of the GAN-HDFS methodology with other available approaches is shown in Figure 7 and Table 4. The figure demonstrates that the machine learning approach has led to an improved F-Score performance. For instance, the GAN-HDFS model's F-Score for data 100 is 93.01%, while the corresponding values for the BiLSTM, CNN, RNN, GAN, and ARIMA models are 76.80%, 74.28%, 78.90%, 81.25%, and 88.43%, respectively. The GAN-HDFS model has nonetheless demonstrated its best performance with various data sizes. The F-Score value of the GAN-HDFS model is 96.32% under 350 data, compared to 80.12%, 78.97%, 83.41%, 86.28%, and 91.76% for the BiLSTM, CNN, RNN, GAN, and ARIMA models, respectively.

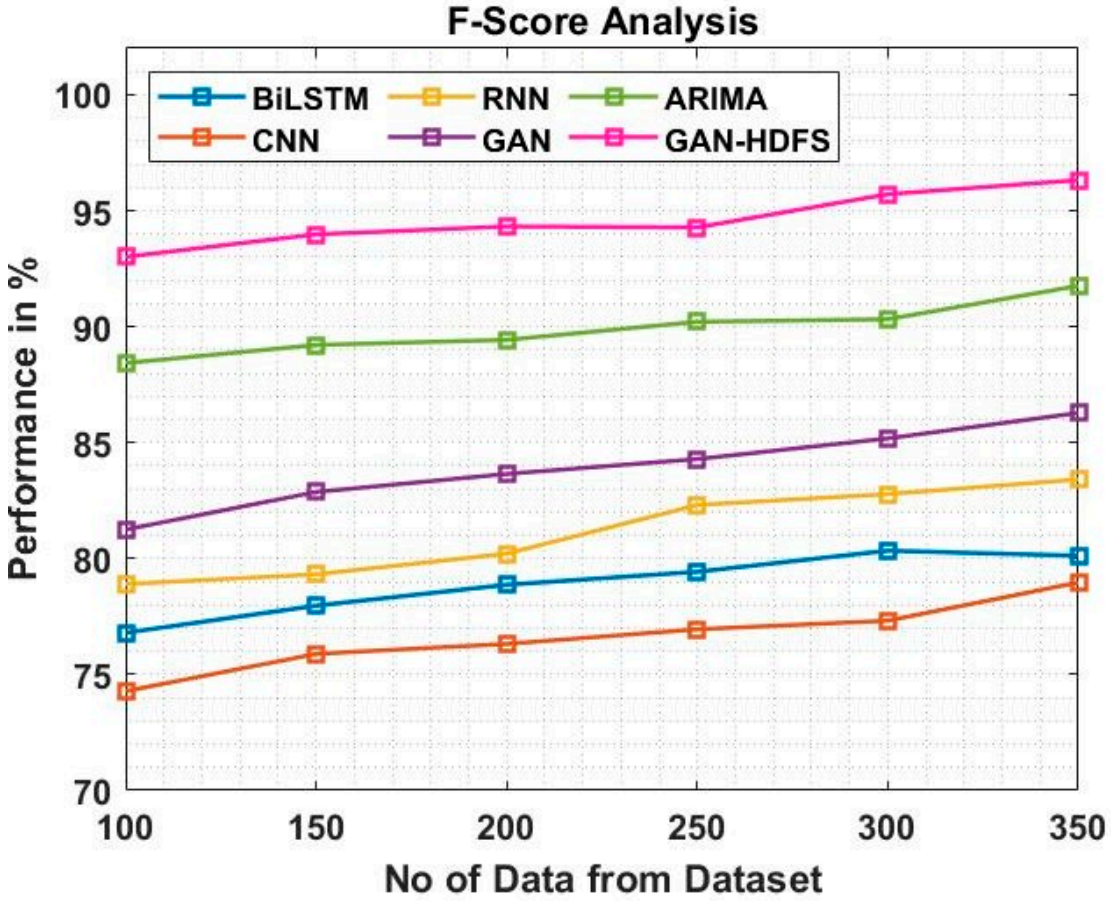

Figure 7. F-Score analysis for GAN-HDFS with existing systems.

Table 4. F-Score analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	76.80	74.28	78.90	81.25	88.43	93.01
150	77.97	75.89	79.32	82.87	89.21	93.98
200	78.87	76.32	80.21	83.65	89.43	94.32
250	79.43	76.95	82.31	84.29	90.21	94.28
300	80.34	77.32	82.78	85.18	90.32	95.71
350	80.12	78.97	83.41	86.28	91.76	96.32

4.2.4. Accuracy

A comparison of accuracy of the GAN-HDFS approach and other existing techniques is shown in Figure 8 and Table 5. The graph demonstrates that the machine learning strategy has an improved performance in terms of accuracy. For instance, the accuracy for the GAN-HDFS model with data 100 is 96.52%, whereas the accuracy values for the BiLSTM, CNN, RNN, GAN, and ARIMA models are 81.90%, 77.21%, 73.87%, 85.43%, and 89.31%, respectively. The GE-DBN model has nonetheless demonstrated its best performance with various data sizes. The accuracy value of the GAN-HDFS model is 98.21% under 350 data, compared to 82.67%, 79.73%, 75.43%, 87.32%, and 92.43% for the BiLSTM, CNN, RNN, GAN, and ARIMA models, respectively.

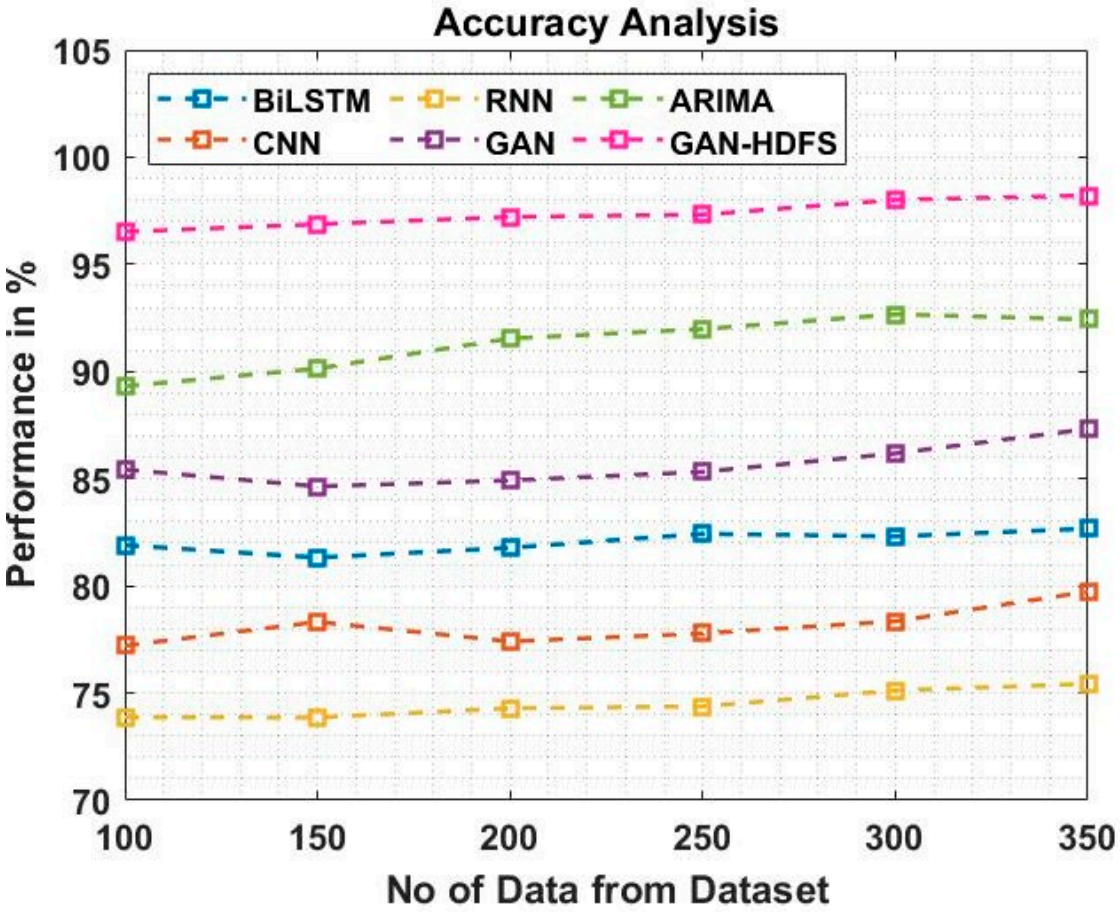

Figure 8. Accuracy analysis for GAN-HDFS with existing systems.

Table 5. Accuracy analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	81.90	77.21	73.87	85.43	89.31	96.52
150	81.32	78.32	73.86	84.64	90.14	96.87
200	81.78	77.41	74.29	84.92	91.54	97.21
250	82.43	77.78	74.38	85.32	91.98	97.32
300	82.31	78.35	75.12	86.18	92.67	98.01
350	82.67	79.73	75.43	87.32	92.43	98.21

4.2.5. Encryption Time

The encryption time analysis of the GAN-HDFS approach with existing methods is described in Table 6 and Figure 9. The data clearly demonstrate that the GAN-HDFS method has performed better than the other methods in every way. For example, with 100 data, the GAN-HDFS method has taken only 5.943 ms to encrypt, while the other existing techniques such as BiLSTM, CNN, RNN, GAN, and ARIMA have an encryption time of 12.342 ms, 10.432 ms, 7.219 ms, 8.201 ms, and 9.321 ms, respectively. Symmetrically, for 350 data, the GAN-HDFS method has an encryption time of 6.692 ms, while the other existing techniques such as BiLSTM, CNN, RNN, GAN, and ARIMA require 13.654 ms, 12.782 ms, 7.943 ms, 8.328 ms and 10.372 ms of encryption time, respectively.

Table 6. Encryption time analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	12.342	10.432	7.219	8.201	9.321	5.943
150	12.549	10.893	7.531	8.282	9.943	5.981
200	12.943	11.543	7.092	8.125	10.432	6.210
250	13.043	12.059	7.325	8.783	10.157	6.398
300	13.258	12.143	7.167	8.104	10.231	6.482
350	13.654	12.782	7.943	8.328	10.372	6.692

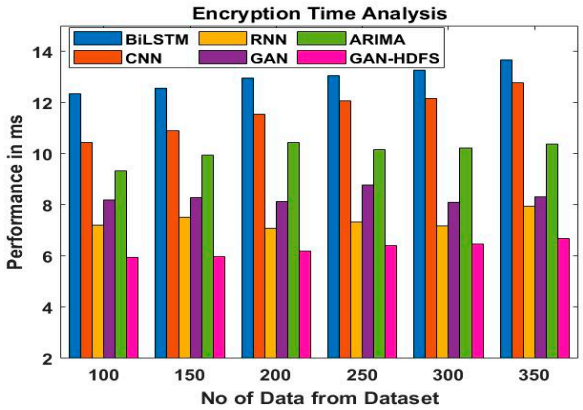

Figure 9. Encryption time analysis for GAN-HDFS with existing systems.

4.2.6. Decryption Time

Table 7 and Figure 10 describe the decryption time analysis of the GAN-HDFS method with existing methods. The data clearly show that the GAN-HDFS method has outperformed the other methods in all aspects. For example, with 100 data, the GAN-HDFS method has taken only 6.721 ms for decryption, while the other existing techniques such as BiLSTM, CNN, RNN, GAN, and ARIMA have a decryption time of 12.894 ms, 10.342 ms, 8.043 ms, 9.210 ms, and 8.743 ms, respectively. Similarly, for the 350 data, the GAN-HDFS method has a decryption time of 7.321 ms while the other existing techniques such as

BiLSTM, CNN, RNN, GAN, and ARIMA require 14.013 ms, 11.854 ms, 8.932 ms, 10.382 ms and 9.710 ms of decryption time, respectively.

Table 7. Decryption time analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	12.894	10.342	8.043	9.210	8.743	6.721
150	12.963	10.541	8.210	9.365	8.932	6.832
200	13.901	10.975	8.428	9.732	8.431	6.942
250	13.453	11.784	8.643	9.843	8.743	7.013
300	13.894	11.652	8.742	9.217	9.419	7.148
350	14.013	11.854	8.932	10.382	9.710	7.321

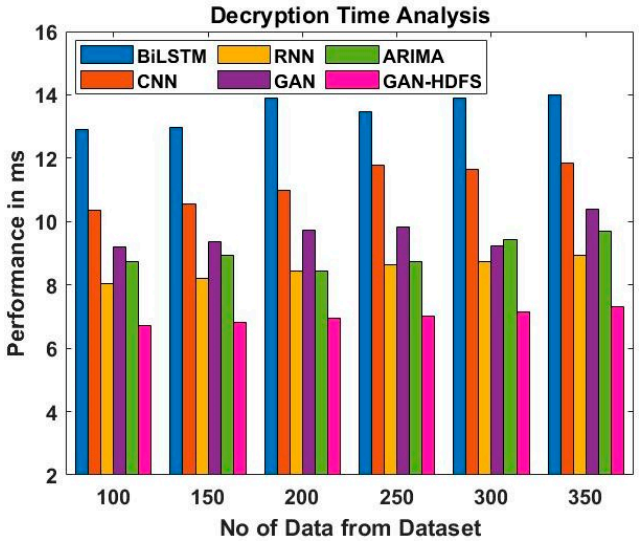

Figure 10. Decryption time analysis for GAN-HDFS with existing systems.

4.2.7. Execution Time

Table 8 and Figure 11 describe the execution time analysis of the GAN-HDFS technique with existing methods. The data clearly show that the GAN-HDFS method has outperformed the other techniques in all aspects. For example, with 100 data, the GAN-HDFS method has taken only 3.143 s to execute, while the other existing techniques such as BiLSTM, CNN, RNN, GAN, and ARIMA have an execution time of 8.320 s, 7.329 s, 5.294 s, 6.312 s, and 4.421 s, respectively. Similarly, for 350 data, the GAN-HDFS method has an execution time of 4.321 s, while the other existing techniques such as BiLSTM, CNN, RNN, GAN, and ARIMA require 9.762 s, 8.129 s, 6.765 s, 7.921 s and 5.154 s of execution time, respectively.

Table 8. Execution time analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	8.320	7.329	5.294	6.312	4.421	3.143
150	8.421	7.431	5.348	6.843	4.589	3.654
200	8.672	7.481	5.914	6.921	4.783	3.781
250	8.721	7.738	6.043	7.317	4.904	3.981
300	9.421	7.956	6.285	7.562	5.087	4.109
350	9.762	8.129	6.765	7.921	5.154	4.321

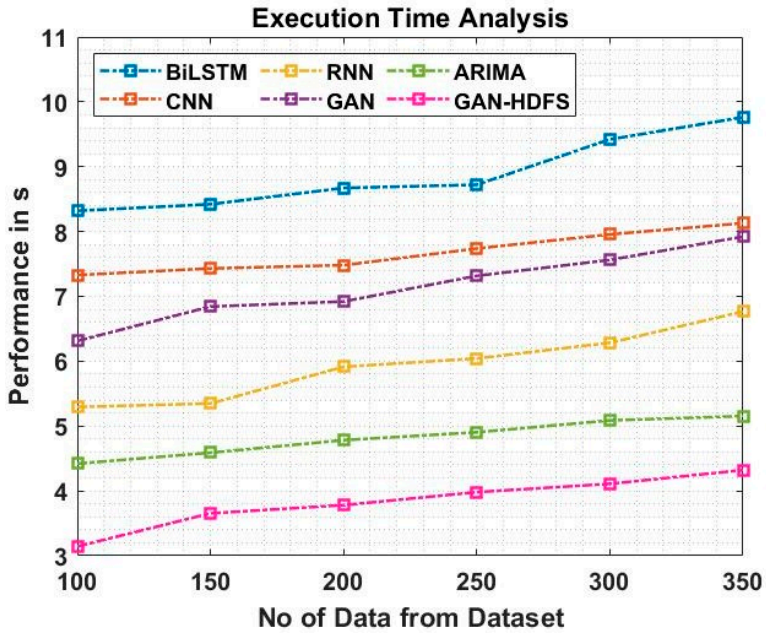

Figure 11. Execution time analysis for GAN-HDFS with existing systems.

4.2.8. Energy Consumption

Table 9 and Figure 12 display the energy consumption analysis of the GAN-HDFS method with existing methods. The proposed method consumes much less energy when compared to the other techniques for any amount of data. For example, with 100 data, the GAN-HDFS method consumes only 30.32 J, while the other methods such as BiLSTM, CNN, RNN, GAN, and ARIMA consume 48.19 J, 52.04 J, 40.64 J, 38.19 J and 45.21 J, respectively. Similarly, with 350 data, the proposed GAN-HDFS method consumes only 32.04 J, whereas the other methods such as BiLSTM, CNN, RNN, GAN, and ARIMA consume 49.74 J, 53.45 J, 43.97 J, 39.75 J and 46.97 J, respectively. The proposed method shows higher performance with less energy consumption.

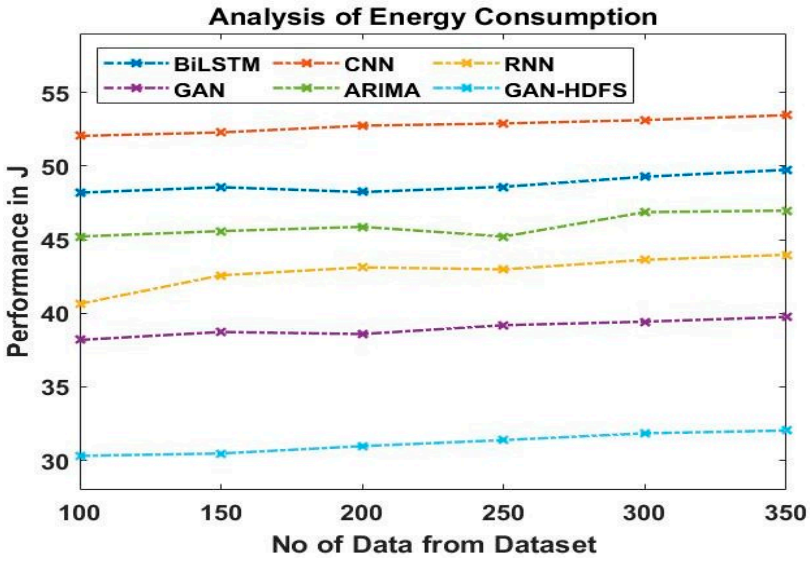

Figure 12. Energy consumption analysis for GAN-HDFS with existing systems.

Table 9. Energy consumption analysis for GAN-HDFS with existing systems.

No. of Data from Dataset	BiLSTM	CNN	RNN	GAN	ARIMA	GAN-HDFS
100	48.19	52.04	40.64	38.19	45.21	30.32
150	48.56	52.28	42.58	38.73	45.58	30.48
200	48.23	52.74	43.12	38.59	45.87	30.98
250	48.59	52.89	42.98	39.19	45.21	31.39
300	49.27	53.12	43.63	39.43	46.87	31.85
350	49.74	53.45	43.97	39.75	46.97	32.04

5. Conclusions

A growing number of methods for gathering, processing, storing, and exploiting data are now accessible thanks to the emergence of big data. This study concentrated on the responsibilities of traffic approximation and prediction and provides a current compilation of readily accessible datasets and tools as a guide for individuals looking for open resources. It is also suggests the use of outside data, such as weather, calendar, and other data. Air pollution emissions and noise levels are important indicators for more accurate estimation and forecasting of traffic conditions. Geographic information systems, data acquisition and processing technologies, along with artificial intelligence technology, are now frequently used in urban traffic systems. Situational awareness in urban traffic is made possible by the state of intelligent identification in urban traffic, which also serves as a theoretical foundation for intelligent control of the urban traffic system. One of the fundamental approaches to relieving urban traffic congestion is short-term traffic flow prediction, which is the foundation and premise of proactive and proactive traffic management. This research aims to solve the classification problem between crowded and uncongested traffic conditions, particularly in developing countries. The lack of traffic datasets is a significant impediment in this scenario. To build the dataset, traffic photos were collected from video recordings captured in developing countries and augmented to provide a larger number of images. This attempted in order to address the problem by exploiting the better deep learning capabilities of a GAN to generate images from the provided dataset. Because the obtained dataset contained fuzzy photos, the adoption of a GAN considerably enhanced image quality. This study utilized GAN-HDFS for a real-time traffic forecasting system using CCTV surveillance and discovered a significant improvement in accuracy. In comparison to other recent methods applied to the dataset, the proposed method achieved the highest accuracy of 98.21%. As a result, the development of intelligent regulation of the urban traffic system will inevitably be intimately related to future research on short-term traffic flow prediction approaches. A fresh approach to solving traffic prediction problems is suggested by the proposed GAN-HDFS and its application. The GAN-HDFS offers a flexible framework for fusing the benefits of several sub-models and an appropriate arena for competition among them. Big information situations (such as traffic forecast and traffic state approximation), which were originally intended for the machine learning environment, can be resolved with GAN-HDFS. However, it is important to investigate how well GAN-HDFS applies to varied datasets, distributions, and statistical characteristics. It is possible that ML technology will one day be a crucial information source because it can be applied to many symmetric IoT applications. In the future, we will create a model that works effectively in both typical and unusual situations, such as special occasions, or other events. Weather conditions as well as information on other incidents such as accidents and road closures can be considered to improve the model. Since missing data make prediction problems even more challenging in real-world applications, we will offer a strategy to handle missing data effectively.

Author Contributions: Conceptualization, P.D.S. and P.D.; methodology, P.D.S.; software, P.D.S.; validation, P.D.S.; formal analysis, P.D.S.; investigation, P.D.; resources, P.D.S.; data curation, P.D.S.; writing—original draft preparation, P.D.S.; writing—review and editing, P.D.; visualization, P.D.S.; supervision, P.D.; project administration, P.D.; funding acquisition, P.D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data will be provided based on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper*; Cisco: San Jose, CA, USA, 2019.
2. The 5 V's of Big Data [Online]. 17 September 2016. Available online: https://www.ibm.com/blogs/watson-health/the-5-vs-ofbig-data/ (accessed on 20 December 2021).
3. Fedorov, A.; Nikolskaia, K.; Ivanov, S.; Shepelev, V.; Minbaleev, A. Trafc fow estimation with data from a video surveillancecamera. *J. Big Data* **2019**, *6*, 1–15. [CrossRef]
4. Li, C.; Dobler, G.; Feng, X.; Wang, Y. TrackNet: Simultaneous object detection and tracking and its application in traffic video analysis. *arXiv* **2019**, arXiv:1902.01466. [CrossRef]
5. Sun, B.; Sun, T.; Zhang, Y.; Jiao, P. Urban traffic flow online prediction based on multi-component attention mechanism. *IET Intell. Transp. Syst.* **2020**, *14*, 1249–1258. [CrossRef]
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
7. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 383–398. [CrossRef]
8. Xie, P.; Li, T.; Liu, J.; Du, S.; Yang, X.; Zhang, J. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Inf. Fusion* **2020**, *59*, 1–12. [CrossRef]
9. Nikodem, M.; Słabicki, M.; Surmacz, T.; Mrówka, P.; Dołega, C. Multi-Camera Vehicle Tracking Using Edge Computing and Low-Power Communication. *Sensors* **2020**, *20*, 3334. [CrossRef]
10. Nguyen, H.; Kieu, L.M.; Wen, T.; Cai, C. Deep learning methods in transportation domain: A review. *IET Intell. Transp. Syst.* **2018**, *12*, 998–1004. [CrossRef]
11. Oh, S.; Kim, Y.J.; Hong, J.S. Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2744–2755. [CrossRef]
12. Do, L.N.N.; Taherifar, N.; Vu, H.L. Survey of neural network-based models for short-term traffic state prediction. *Wiley Interdiscip. Rev.* **2019**, *9*, e1285. [CrossRef]
13. Nair, R.; Dekusar, A. Keep it simple stupid! A non-parametric kernel regression approach to forecast travel speeds. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 269–274. [CrossRef]
14. Hadoop Apache Yarn. Available online: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html (accessed on 25 April 2022).
15. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. *arXiv* **2018**, arXiv:1801.02143. [CrossRef]
16. Du, D.; Qi, Y.; Yu, H. The unmanned aerial vehicle benchmark: Object detection and tracking. *J. Comput. Vis. Pattern Recognit.* **2018**, *23*, 370–386. [CrossRef]
17. Kong, Q.; Tong, B.; Klinkigt, M.; Watanabe, Y.; Akira, N.; Murakami, T. Active generative adversarial network for image classification. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 4090–4097. [CrossRef]
18. Li, Z.; Zheng, Z.; Washington, S. Short-Term Traffic Flow Forecasting: A Component-Wise Gradient Boosting Approach with Hierarchical Reconciliation. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 5060–5072. [CrossRef]
19. Tang, Z.; Wang, G.; Xiao, H.; Zheng, A.; Hwang, J.N. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1080–1087.
20. Lin, Y.; Dai, X.; Li, L.; Wang, F.Y. Pattern sensitive prediction of traffic flow based on generative adversarial framework. *IEEE Trans. Intell. Transp. Syst* **2018**, *20*, 2395–2400. [CrossRef]
21. Xia, M.; Jin, D.; Chen, J. Short-Term Traffic Flow Prediction Based on Graph Convolutional Networks and Federated Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1191–1203. [CrossRef]
22. Sasiadek, J.Z.; Hartana, P. GPS/INS sensor fusion for accurate positioning and navigation based on Kalman filtering. *IFAC Proc. Vol.* **2014**, *37*, 115–120. [CrossRef]
23. Wang, Z.; Wang, W.; Yang, Y.; Han, Z.; Xu, D.; Su, C. CNN- and GAN-based classification of malicious code families: A code visualization approach. *Int. J. Intell. Syst.* **2022**, *37*, 12472–12489. [CrossRef]

24. Yang, Y.; Wei, X.; Xu, R.; Wang, W.; Peng, L.; Wang, Y. Jointly beam stealing attackers detection and localization without training: An image processing viewpoint. *Front. Comput. Sci.* **2022**, *17*, 173704. [CrossRef]
25. Jilani, U.; Asif, M.; Rashid, M.; Siddique, A.A.; Talha, S.M.U.; Aamir, M. Traffic Congestion Classification Using GAN-Based Synthetic Data Augmentation and a Novel 5-Layer Convolutional Neural Network Model. *Electronics* **2022**, *11*, 2290. [CrossRef]
26. Wang, P.; Hao, W.; Sun, Z.; Wang, S.; Tan, E.; Li, L.; Jin, Y. Regional Detection of Traffic Congestion Using in a Large-Scale Surveillance System via Deep Residual TrafficNet. *IEEE Access* **2018**, *6*, 68910–68919. [CrossRef]
27. Khazukov, K.; Shepelev, V.; Karpeta, T.; Shabiev, S.; Slobodin, I.; Charbadze, I.; Alferova, I. Real-time monitoring of traffic parameters. *J. Big Data* **2020**, *7*, 84. [CrossRef]
28. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transp. A Transp. Sci.* **2019**, *15*, 1688–1711. [CrossRef]
29. Guo, J.; Liu, Y.; Yang, Q.; Wang, Y.; Fang, S. GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model. *Transp. A Transp. Sci.* **2020**, *17*, 190–211. [CrossRef]
30. Cheng, S.; Lu, F.; Peng, P.; Wu, S. Short-term traffic forecasting: An adaptive ST-KNN model that considers spatial heterogeneity. *Comput. Environ. Urban Syst.* **2018**, *71*, 186–198. [CrossRef]
31. Kumar, D.T.S. Video based Traffic Forecasting using Convolution Neural Network Model and Transfer Learning Techniques. *J. Innov. Image Process* **2020**, *2*, 128–134. [CrossRef]
32. Ketabi, R.; Al-Qathrady, M.; Alipour, B.; Helmy, A. Vehicular traffic density forecasting through the eyes of traffic cameras; a spatio-temporal machine learning study. In Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, Miami Beach, FL, USA, 25–29 November 2019; pp. 81–88. [CrossRef]
33. Zhang, C.; Yu, J.J.; Liu, Y. Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting. *IEEE Access* **2019**, *7*, 166246–166256. [CrossRef]
34. Lu, H.; Huang, D.; Song, Y.; Jiang, D.; Zhou, T.; Qin, J. St-Trafficnet A Spatial-Temporal Deep Learning Network for Traffic Forecasting. *Electronics* **2020**, *9*, 1474. [CrossRef]
35. Rahman, F.I. Short-Term Tfp Using Machine Learning-KNN, SVM, and ANN With Weather Information. *Int. J. Traffic Transp. Eng.* **2020**, *10*, 371–389. [CrossRef]
36. Panigrahi, S.; Lenka, R.K.; Stitipragyan, A. A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark. *Procedia Comput. Sci.* **2016**, *83*, 1000–1006. [CrossRef]
37. Wu, C.; Chen, L.; Wang, G.; Chai, S.; Jiang, H.; Peng, J.; Hong, Z. Spatiotemporal scenario generation of traffic flow based on lstm-gan. *IEEE Access* **2020**, *8*, 186191–186198. [CrossRef]
38. Zhang, L.; Wu, J.; Shen, J.; Chen, M.; Wang, R.; Zhou, X.; Xu, C.; Yao, Q.; Wu, Q. Satp-Gan: Self-attention based generative adversarial network for traffic flow prediction. *Transp. B Transp. Dyn.* **2021**, *9*, 552–568. [CrossRef]
39. Chatterjee, S.; Hazra, D.; Byun, Y.; Kim, Y.W. Enhancement of Image Classification Using Transfer Learning and GAN-Based Synthetic Data Augmentation. *Mathematics* **2022**, *10*, 1541. [CrossRef]
40. Bogaerts, T.; Masegosa, A.D.; Angarita-Zapata, J.S.; Onieva, E.; Hellinckx, P. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* **2020**, *112*, 62–77. [CrossRef]
41. Ezhumalai, P.; Prakash, M. A deep learning modified neural network (dlmnn) based proficient sentiment analysis technique on twitter data. *J. Exp. Theor. Artif. Intell.* **2021**. [CrossRef]
42. Subramani, N.; Subramanian, M.; Meckanzi, S. Handcrafted deep-feature-based brain tumor detection and classification using mri images. *Electronics* **2022**, *11*, 4178. [CrossRef]
43. Geetha, B.T.; Mary Rexcy Asha, S.; Michaelraj Kingston, R. Artificial humming bird with data science enabled stability prediction model for smart grids. *Sustain. Comput. Inform. Syst.* **2022**, *36*, 100821. [CrossRef]
44. Pokle, S.B. Analysis of ofdm system using dct-pts-slm based approach for multimedia applications. *Clust. Comput.* **2019**, *22*, 4561–4569. [CrossRef]
45. Ravichandran, T. An efficient resource selection and binding model for job scheduling in grid. *Eur. J. Sci. Res.* **2012**, *81*, 450–458.
46. Sayeed, R.F.; Princey, S.; Priyanka, S. Deployment of multicloud environment with avoidance of DDOS attack and secured data privacy. *Int. J. Appl. Eng. Res.* **2015**, *10*, 8121–8124.
47. Satish Kumar, T.; Jothilakshmi, S.; James, B.C.; Arulkumar, N.; Rekha, C. HHO-based vector quantization technique for biomedical image compression in cloud computing. *Int. J. Image Graph.* **2021**, 2240008. [CrossRef]
48. Jaishankar, B.; Vishwakarma, S.; Aditya Kumar, S.P.; Ibrahim, P.; Arulkumar, N. Blockchain for securing healthcare data using squirrel search optimization algorithm. *Intell. Autom. Soft Comput.* **2022**, *32*, 1815–1829. [CrossRef]
49. Kuppuraj, B.; Chellai, S. An enhanced security measure for multimedia images using hadoop cluster. *Int. J. Oper. Res. Inf. Syst.* **2021**, *12*, 1–7. [CrossRef]
50. Gowshika, U.; Ravichandran, T. A smart device integrated with an android for alerting a person's health condition: Internet of Things. *Indian J. Sci. Technol.* **2016**, *9*, 1–6. [CrossRef]
51. Thangavel, R. Resource selection in grid environment based on trust evaluation using feedback and performance. *Am. J. Appl. Sci.* **2013**, *10*, 924. [CrossRef]
52. Hardas, B.M.; Pokle, S.B. Optimization of peak to average power reduction in OFDM. *J. Commun. Technol. Electron.* **2017**, *62*, 1388–1395. [CrossRef]

53. Satpathy, S.; Padthe, A.; Trivedi, M.C.; Goyal, V.; Bhattacharyya, B.K. Method for measuring supercapacitor's fundamental inherent parameters using its own self-discharge behavior: A new steps towards sustainable energy. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102760. [CrossRef]
54. Revanesh, M.; Sridhar, V.; Acken, J.M. CB-ALCA: A cluster-based adaptive lightweight cryptographic algorithm for secure routing in wireless sensor networks. *Int. J. Inf. Comput. Secur.* **2019**, *11*, 637–662. [CrossRef]
55. Subramani, N.; Abbas, M.; Mishra, A.R. A fuzzy logic and DEEC protocol-based clustering routing method for wireless sensor networks. *AIMS Math.* **2023**, *8*, 8310–8331. [CrossRef]
56. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transp. Res. Part C Emerg. Technol.* **2020**, *118*, 102674. [CrossRef]
57. Praveen, D.S.; Raj, D.P. Smart traffic management system in metropolitan cities. *J. Ambient Intell. Human Comput.* **2021**, *12*, 7529–7541. [CrossRef]
58. Balachander, K.; Paulraj, D. ANN and fuzzy based household energy consumption prediction with high accuracy. *J. Ambient Intell. Human Comput.* **2021**, *12*, 7543–7557. [CrossRef]
59. Reshmy, A.K. Data mining of unstructured big data in cloud computing. *Int. J. Bus. Intell. Data Min.* **2021**, *13*, 147–162. [CrossRef]
60. Sermakani, A.M. Effective data storage and dynamic data auditing scheme for providing distributed services in federated cloud. *J. Circuits Syst. Comput.* **2020**, *29*, 2050259. [CrossRef]
61. Hariharan, B.; Paulraj, D. A hybrid framework for Job Scheduling on Cloud using Firefly and BAT algorithm. *Int. J. Bus. Intell. Data Min.* **2020**, *15*, 388–407. [CrossRef]
62. Devi, K.; Muthusenthil, B. Deep learning based security model for cloud based task scheduling. *KSII Trans. Internet Inf.* **2021**, *14*, 3663–3679.
63. Manikandan, S.; Sambit, S.; Sanchali, D. An efficient technique for cloud storage using secured de-duplication algorithm. *J. Intell. Fuzzy Syst.* **2021**, *42*, 2969–2980. [CrossRef]
64. Subbulakshmi, P.; Ramalakshmi, V. Honest auction based spectrum assignment and exploiting spectrum sensing data falsification attack using stochastic game theory in wireless cognitive radio network. *Wirel. Pers. Commun. Int. J.* **2018**, *102*, 799–816. [CrossRef]
65. Rajaram, P.V. Intelligent deep learning based bidirectional long short term memory model for automated reply of e-mail client prototype. *Pattern Recognit. Lett.* **2021**, *152*, 340–347. [CrossRef]
66. Ambeth Kumar, V.D.; Malathi, S.; Abhishek, K.; Kalyana, C.V. Active volume control in smart phones based on user activity and ambient noise. *Sensors* **2020**, *20*, 4117. [CrossRef] [PubMed]
67. Ranjith, C.P.; Hardas, B.M.; Mohideen, M.S.K.; Raj, N.N.; Robert, N.R. Robust deep learning empowered real time object detection for unmanned aerial vehicles based surveillance applications. *J. Mob. Multimed.* **2023**, *19*, 451–476. [CrossRef]
68. Sindhu, V.; Mohan Kumar, P. Energy-efficient task scheduling and resource allocation for improving the performance of a cloud–fog environment. *Symmetry* **2022**, *14*, 2340. [CrossRef]