IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

Integrating Generative Adversarial Networks (GAN) with Hadoop Distributed File System (HDFS) for Real-Time Traffic Forecasting using CCTV Surveillance

Sudeep G[1], Prof. Rajeshwari N[2]

MCA Student, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[1]

Assistant Professor, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[2]

ABSTRACT: Predicting traffic flow is the most crucial part of a smart city's traffic management system. It helps drivers choose the best route to their destination. CCTV cameras have improved security surveillance, but the increasing number of cars on the road has caused congestion. Road capacity hasn't kept up, and traffic management needs a smart solution. To tackle this, researchers turned to adaptive traffic management in order to create the most of existing infrastructure. However, the enormous volumes of data from CCTV cameras, in the form of images and videos, are unstructured and difficult to process. This project aims to identify a way to oversee and use this data effectively to improve traffic flow.

KEY WORDS: CCTV, Generative Adversarial Networks (GAN)

I. INTRODUCTION

The quick expansion of urban populations and the increasing demand for efficient transportation systems have made traffic forecasting a crucial task for urban planners and transportation authorities. Accurate traffic forecasting can help alleviate traffic congestion, reduce travel times, and improve air quality. Traditional traffic forecasting methods rely on historical data and statistical models, which may not capture the complexity and dynamics of real-time traffic flow.

GANs have shown remarkable success in generating realistic synthetic data, which can augment real data and improve the accuracy of traffic forecasting models. Our approach uses GANs to produce fictitious traffic information that mimics real-world traffic patterns are subsequently applied to train a traffic forecasting model. The HDFS-based architecture enables efficient storage and processing of large volumes of video data from multiple CCTV cameras, allowing for real-time traffic forecasting and monitoring.

This paper presents a comprehensive overview of our proposed system, including the GAN-based data generation module, the HDFS-based data storage and processing module, and the traffic forecasting module. We evaluate the performance of our system using real-world CCTV surveillance footage and demonstrate its effectiveness in improving traffic forecasting accuracy and efficiency.

II. PROPOSED SYSTEM

Current traffic information, historical traffic informations well as any more pertinent statistics are utilized in the creation of an appropriate mathematical method that is used to develop a traffic flow prediction shown in Figure 1. It is feasible to use a method of intellectual computation to produce more accurate projections of the amount of traffic that will transpire in the years to come. The findings may provide a plausible basis for vehicle dynamic guidance and an urban circulation regulator Should they be implemented.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

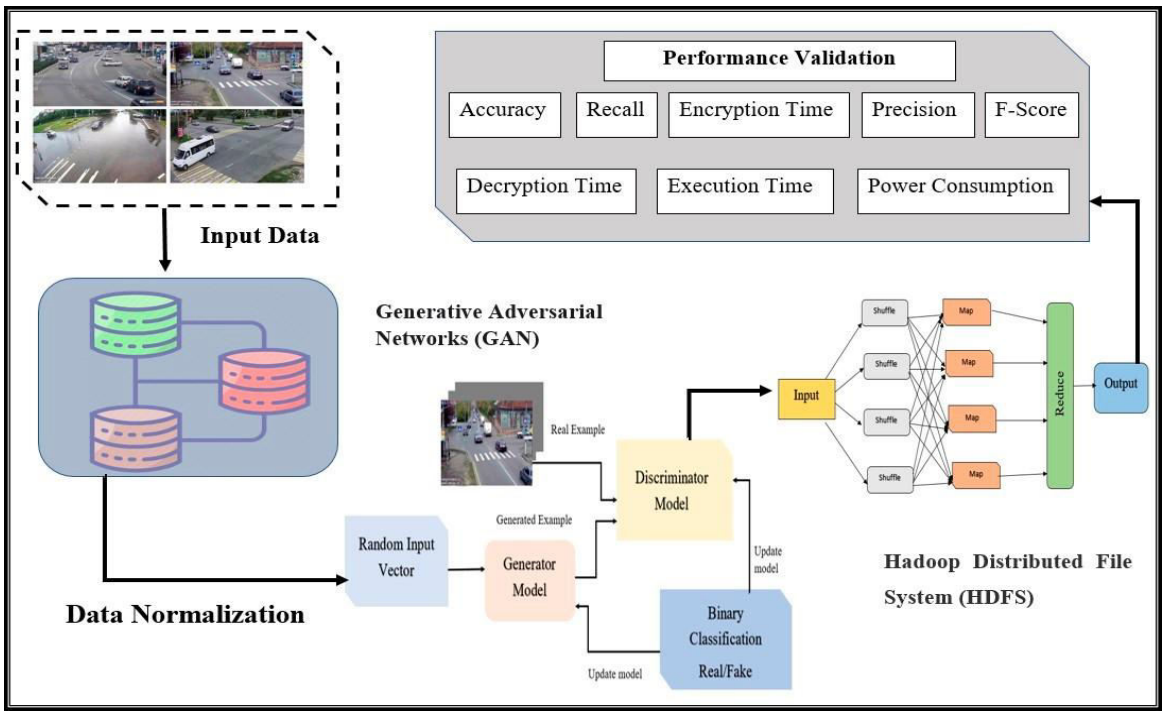

Figure 1. The proposed method of GAN-HDFS.

2.1. Data Collection

This research makes use of the Hive project, a data warehousing system built on the Hadoop framework. HiveQL, its query language, supports the majority of the SQL-92 standard, but with a key distinction: it employs the schema-on-read approach. This technique enables flexible data ingestion and storage, allowing for the creation of tables on top of the information as needed to facilitate querying.The information is stored in a database, text files, and RC Files (with Apache HBase as the default storage engine). Metadata is stored implicitly using Apache Derby, although other relational databases like MySQL can be used as alternatives.

2.2. 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 − xmin

$$\overline{\text{xScale} = \text{xmax} - \text{xmin}} \qquad (1)$$

2.3. Traffic Flow Prediction Using GAN-HDFS

Deep learning algorithms rely on three crucial factors: database availability, algorithm selection and application, and the algorithm itself. Given the limited scope of the traffic database, this study emphasizes the importance of data augmentation. To address traffic congestion, we explored a novel approach called sequential Generative Adversarial Networks (GAN-HDFS) [37]. This method accurately models temporal dynamics on roadways, generating realistic traffic scenarios.

2.3.1. Generative Adversarial Networks (GAN)

Utilizing data augmentation and image modification techniques, the GAN perfect is accustomed to create synthetic imageries that look very comparable 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.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

Hadoop stores all data files in HDFS, a crucial component of distributed computing, which enables efficient data management and processing. The Primary Name Node plays a vital role in managing file access, namespace, and metadata, while the Secondary Name Node serves as a backup to ensure data reliability. Data is stored in blocks across multiple Data Nodes, which regularly report to the Name Node, and file chunking improves efficiency by dividing files into smaller pieces. Multiple Data Nodes manage memory connected to the network, enabling parallel processing. Map-reduce programming is accustomed to process data, requiring writing map and reduce functions. The map function handles data input and returns intermediate figures, which are subsequently processed by the reduce function to produce the final output. This process parallels calculations, following the design by Dean and Ghemawat, enabling efficient data processing and analysis.

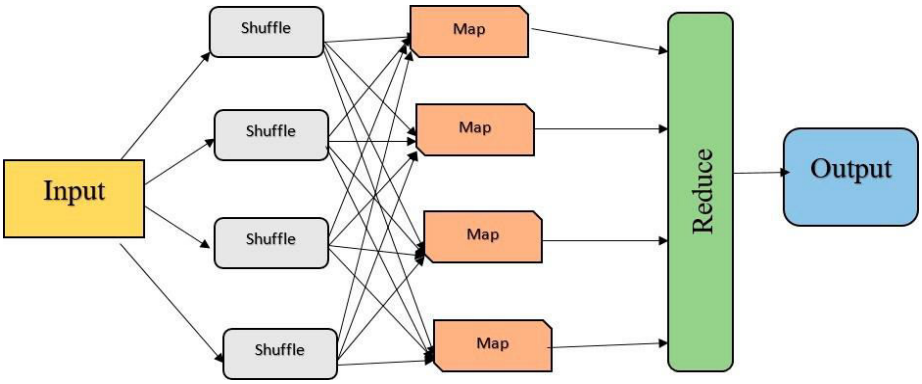

Figure 2. Map-Reduce Framework.

III. RESULT AND DISCUSSION

3.1. Experimental Setup

The system that was created 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.

3.2. 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 were used to evaluate the model's presentation.

3.2.1 Precision analysis

A comparison of precision of the GAN-HDFS approach and other existing techniques is shown in Fig 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 values 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.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

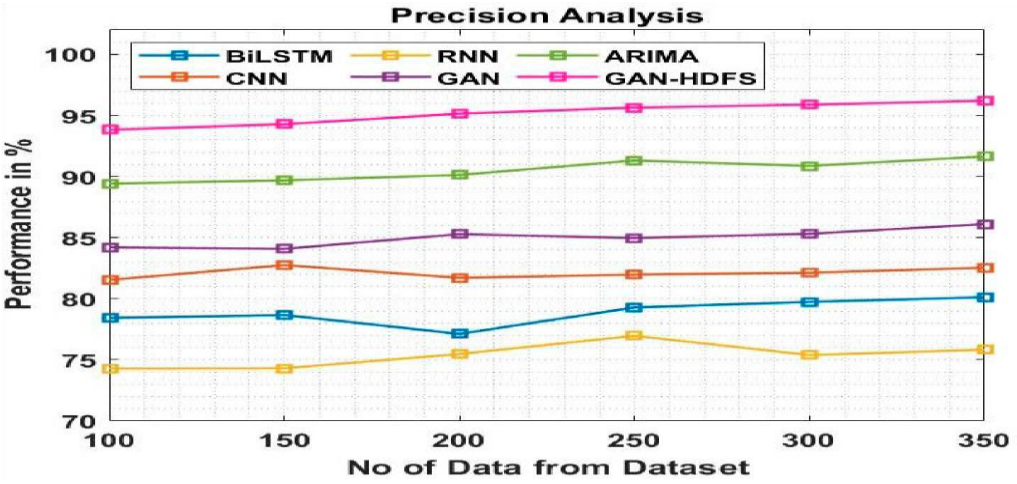

Figure 3. Precision analysis for GAN-HDFS with existing systems.

Table 1. 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

3.2.2. Recall

Comparative recall testing between the GAN-HDFS strategy and other approaches is shown in Fig 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.

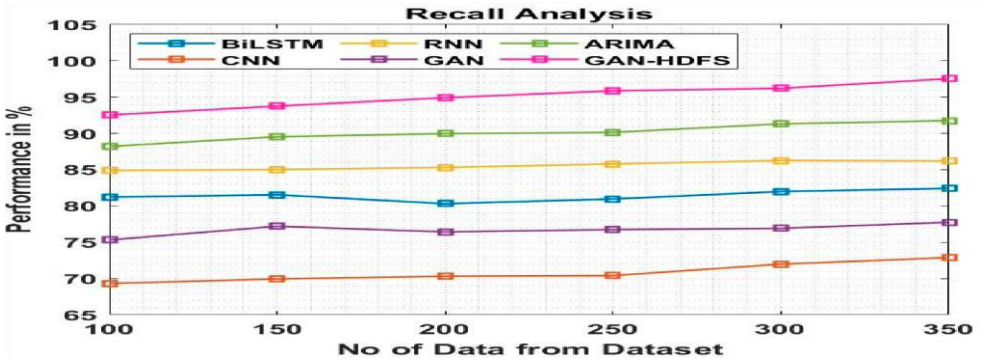

Figure 4. Recall analysis for GAN-HDFS with existing systems.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

Table 2. 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

3.2.3. F-Score

A comparison of F-Score examination of the GAN-HDFS methodology with other available approaches is shown in Fig 7 and Table 4. The figure demonstrates that the machine learning approach has led to an improved F-Score performance. For instance, the GANHDFS 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 values 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.

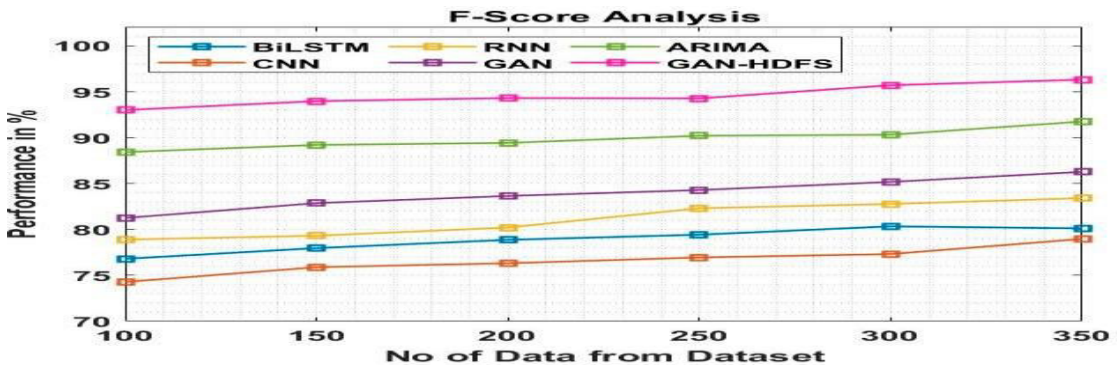

Figure 5. F-Score analysis for GAN-HDFS with existing systems

Table 3. 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

3.2.4. Accuracy

A comparison of precision of the GAN-HDFS approach and other existing techniques is shown in Fig 8 and Table 5. The graph demonstrates that the machine learning strategy has an improved performance With regard to 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%.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

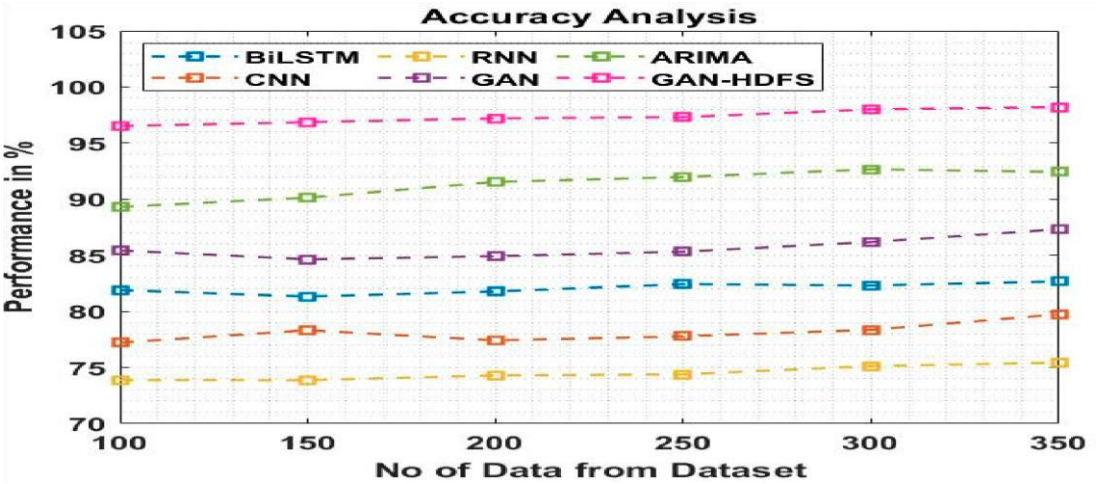

Figure 6. Accuracy analysis for GAN-HDFS with existing systems.

Table 4. 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

3.2.5. Encryption Time

The encryption time analysis of the GAN-HDFS approach with existing methods is described in Table 6 and Fig 9.The data clearly demonstrate that the GAN-HDFS approach has outperformed the alternative approaches 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 5. 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

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

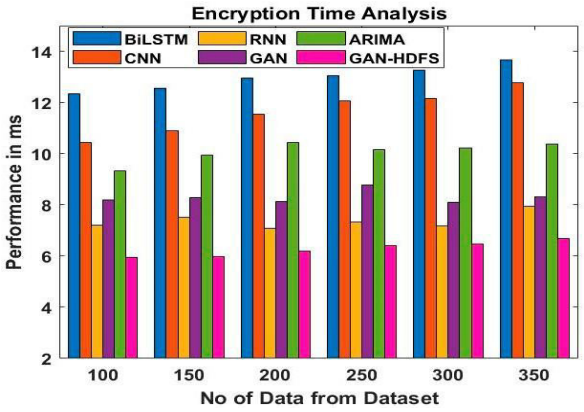

Figure 7. Encryption time analysis for GAN-HDFS with existing systems.

3.2.6. 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 approach 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 approach 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 6. 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

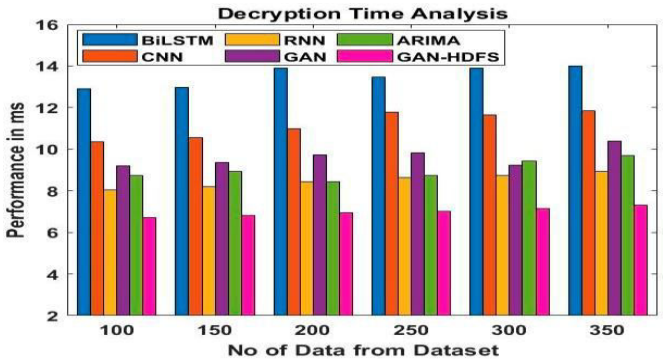

Figure 8. Decryption time analysis for GAN-HDFS with existing systems.

3.2.7. 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 GANHDFS 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 duration 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 duration of 4.321 s, while the other

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

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 7. 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

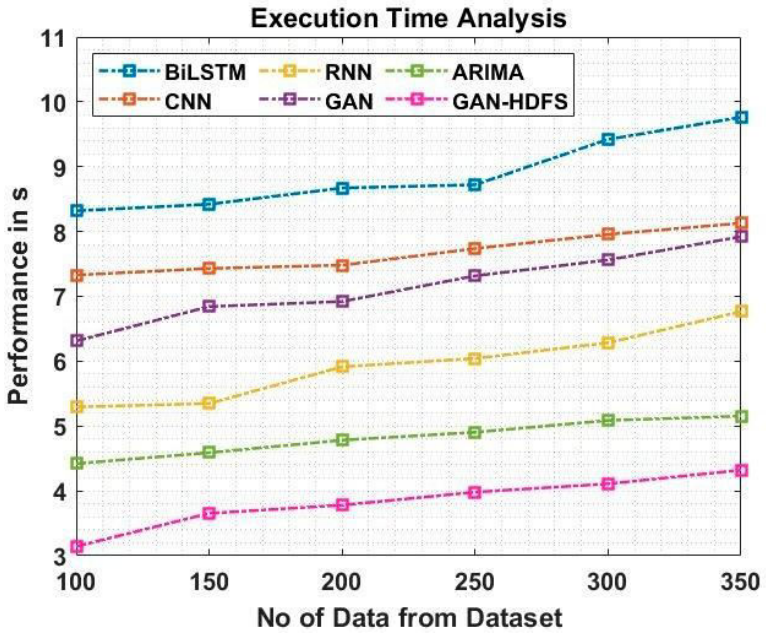

Figure 9. Execution time analysis for GAN-HDFS with existing systems.

IV. CONCLUSION

Finally, this research effectively illustrated the effectiveness of integrating Generative Adversarial Networks (GANs) with a Hadoop Distributed File System (HDFS)-based architecture for real-time traffic forecasting using CCTV surveillance. The proposed system leveraged GANs In order to produce fictitious traffic information, augmenting real-world data and improving forecasting accuracy. The HDFS-based architecture enabled efficient storage and processing of large volumes of video data from multiple CCTV cameras. The outcomes of the experiment demonstrated notable advancements. in forecasting accuracy and efficiency compared to traditional methods. This research aids in the growth of intelligent transportation systems, enabling real-time traffic monitoring and forecasting. Future work includes expanding the system to incorporate additional data sources and exploring applications in other domains.

REFERENCES

1. Subramani, N.; Subramanian, M.; Meckanzi, S. Handcrafted deep-feature-based brain tumor detection and classification using mri images. Electronics 2022, 11, 4178.
2. Pokle, S.B. Analysis of ofdm system using dct-pts-slm based approach for multimedia applications. Clust. Comput. 2019, 22, 4561–4569.
3. 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.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207058 |

4. Balachander, K.; Paulraj, D. ANN and fuzzy based household energy consumption prediction with high accuracy. J. Ambient Intell. Human Comput. 2021, 12, 7543–7557.
5. Reshmy, A.K. Data mining of unstructured big data in cloud computing. Int. J. Bus. Intell. Data Min. 2021, 13, 147–162.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details