Neural Network-Based Prediction of Short-Term Cloud Data Transfer Throughput in Distributed Systems

[1]D. V. H. Venu Kumar, [2]Mr.Y.V.Ramesh,
Assistant Professor , Department of Computer Science & Engineering, Geethanjali Institute Of Science And Technology, Nellore, India

Abstract
Accurate prediction of cloud data transfer throughput is crucial for optimizing system performance, minimizing latency, and ensuring efficient resource utilization in distributed computing environments. However, traditional forecasting methods often fall short in capturing the complex, multivariate factors influencing throughput variations across distributed cloud systems. This project proposes a neural network-based predictive model, trained on a novel dataset collected from real-world file transfers across various Amazon Web Services (AWS) regions. The dataset includes multivariate features such as disk I/O bandwidth, CPU utilization, and network performance metrics, allowing for a comprehensive analysis of short-term throughput variations. Experimental results demonstrate that the proposed model achieves low prediction error rates—3.7% for network throughput and 6.1% for disk throughput—outperforming traditional univariate and ARIMA-based models. The improved forecasting capability enables dynamic resource management tasks such as autoscaling and load balancing, leading to cost-effective operations and enhanced reliability in cloud data transfer processes.
Keywords:

Introduction

In cloud computing, efficient data transfer is critical for services like auto-scaling, load balancing, and replica placement. However, predicting short-term variations in end-to-end throughput is challenging due to dynamic network behavior and overlooked system-level factors. Traditional prediction models typically use only network throughput data, ignoring variables such as CPU load, disk speed, or dataset size, which significantly impact transfer performance. This project introduces a neural network-based prediction framework using a hybrid multivariate approach. It integrates network metrics (e.g., latency, bandwidth) with engineered system metrics (e.g., disk I/O, CPU utilization) collected from both source and destination nodes in real AWS transfers. Unlike models using synthetic traffic tools like iperf, this method leverages real-time monitoring data and feature engineering to capture system behavior more effectively. Models like LSTM, RNN, and other multivariate neural networks are trained to predict oneminute- ahead throughput with high accuracy. The system enhances cloud performance forecasting, enabling service providers to make proactive decisions for workload distribution and infrastructure planning. With an average error rate of just 3.7% for network throughput prediction, this AI-driven solution offers a scalable and accurate tool for managing cloud traffic variability, improving efficiency, and reducing operational risks.

Motivation

Cloud computing powers a vast range of modern applications, from video streaming and big data analytics to real-time collaboration tools. At the heart of these services lies the need for efficient, high-speed data transfer between geographically distributed systems. Even brief dips in throughput can lead to latency, delayed processing, or service degradation, especially in timesensitive environments. While throughput prediction has been explored in previous research, most models are limited in scope. They typically rely on network-

layer data alone, such as bandwidth or latency, and often ignore system-level and dataset-specific characteristics. In real-world scenarios, factors like CPU usage, disk I/O speed, and file size significantly influence the actual transfer perform- -ance, making single-source predictions insufficient.

Moreover, many existing studies use synthetic datasets or controlled memory-to-memory transfers, which fail to represent the complexities of actual cloud operations. Without real file transfers or multivariate time series data collected from source and destination systems, models lack the robustness needed to perform in dynamic environments like AWS or Azure. This limits their practical application in industry settings. This project is motivated by the need for a more realistic and comprehensive approach to throughput prediction. By leveraging real cloud monitoring data and combining both network and end-system features, it aims to capture a fuller picture of transfer dynamics. This hybrid approach is expected to improve prediction accuracy, support proactive resource management, and help cloud service providers deliver more consistent performance to users.

Objective

The primary objective of this study is to accurately predict short-term variations in end-toend cloud data transfer throughput using neural network models. Existing methods often rely on limited or univariate data sources, neglecting important parameters such as disk I/O performance, CPU usage, and dataset characteristics. This research addresses that gap by introducing a more comprehensive, multivariate approach that includes end-system metrics and network indicators collected from real-world cloud environments.To achieve this, the study involves collecting time series data through actual file transfers within Amazon Web Services (AWS), capturing metrics from both source and destination systems. The objective is not only to model the throughput behavior using deep learning but also to compare performance against traditional methods like univariate models and least-correlated multivariate models.. By developing a predictive system that integrates hard metrics and behavioral patterns, this research aims to support better cloud resource planning, load balancing, and performance tuning. The work also contributes a publicly available dataset and modeling framework to the research community, setting the stage for future studies in intelligent cloud infrastructure management.

Literature Review

This chapter reviews prior work on predicting short-term cloud data transfer throughput in distributed systems. Accurate forecasting is essential for enhancing performance, reducing latency, and optimizing resource use. Researchers have explored univariate models, ARIMA, and machine learning techniques. These methods aim to model cloud dynamics and ensure reliable transfers. Traditional approaches, however, often struggle with multivariate and nonlinear patterns. This review highlights strengths and gaps that inform our proposed neural network-based model.

Predicting Throughput of Cloud Network Infrastructure Using Neural Networks:
D. Phanekham, S. Nair, N. Rao, and M. Truty et al. (2021)

This paper introduces a neural network-based approach for predicting throughput in cloud network infrastructure. The authors leverage multivariate data collected from cloud platforms to train models that can anticipate short-term fluctuations in network performance. Their experiments show improved accuracy compared to classical time series models. This predictive capability supports better decision-making in dynamic cloud environments. The research underlines the growing importance of machine learning for real-time network optimization.

Intelligent Hybrid Model to Enhance Time Series Models for Predicting Network Traffic:
T. H. H. Aldhyani et al. (2020)

The paper introduces an intelligent hybrid model that combines neural networks with traditional time series techniques. By integrating ARIMA and deep learning models, it captures both linear trends and nonlinear patterns in network traffic. The hybrid architecture increases forecasting precision and stability in high-variance environments. It is particularly useful for short-term predictions in cloud networks. The system is evaluated using real network traffic datasets and performs better than standalone methods. This hybrid method aids in cloud resource management and bandwidth optimization.

Time Series Analysis to Predict End-to-End Quality of Wireless Community Networks:

P. Millan, C. Aliagas, C. Molina, E. Dimogerontakis, and R. Meseguer et al. (2019)

This research applies time series forecasting techniques to evaluate end-to-end network quality in wireless mesh networks. Although focused on community networks, the techniques have relevance for cloud and edge computing systems. The model analyzes traffic load variations and link reliability over time to estimate performance metrics. It helps in understanding throughput variability, crucial for maintaining QoS in distributed architectures. The approach uses opensource data from operational networks for validation. Its insights support efficient routing and bandwidth management. The work lays the groundwork for extending similar forecasting to hybrid cloud scenarios.

Throughput Prediction Using Recurrent Neural Network Model:

B. Wei, M. Okano, K. Kanai, W. Kawakami, and J. Katto et al. (2018)

This study proposes a recurrent neural network (RNN) model to predict short-term data throughput in networked environments. RNNs are well-suited to sequence data, making them ideal for capturing temporal dependencies in network performance. The model is trained on historical throughput values and responds well to rapid fluctuations. It outperforms traditional regression and ARIMA models in both accuracy and adaptability. It helps in anticipating performance bottlenecks and planning transfers efficiently. The approach is applicable in real-time network optimization systems.

Forecasting Short-Term Data Center Network Traffic Load with Convolutional Neural Networks:

A. Mozo, B. Ordozgoiti, and S. Gómez-Canaval et al. (2018)

This paper explores the use of CNNs for forecasting traffic load in data center networks. Unlike RNNs, CNNs detect spatial and temporal patterns in multivariate traffic data. The proposed model is lightweight, fast, and effective for short-term load predictions. It is tested on real data center logs and achieves higher accuracy than baseline statistical models. The model aids in detecting anomalies and managing traffic congestion proactively. It supports intelligent

The authors of this study offer a model for the purpose of making reliable predictions about the onset of foot ulcers. Model training, feature extraction, and preprocessing are all steps in a sequential process [15]. Using mask-based segmentation, preprocessing improves picture quality and removes noise from the scanned raw RGB images.

Proposed Model

The proposed system is designed to enhance the prediction of short-term variations in cloud

data transfer throughput for distributed systems. Traditional forecasting methods such as ARIMA and Support Vector Machines (SVM) often struggle with capturing the non-linear and complex relationships inherent in cloud environments. To overcome these limitations, the system employs deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, or Convolutional Neural Networks (CNN), which are well-suited for modeling time-series data.

These models can capture both short-term fluctuations and long-term dependencies in cloud traffic, providing more accurate predictions.

The system operates by analyzing historical cloud traffic data, extracting relevant features

such as network latency, traffic volume, and server load. By processing vast amounts of real-time and historical data, the model learns to identify patterns in throughput behavior, allowing it to predict future fluctuations with high precision. The use of deep learning enables the system to adapt to the dynamic nature of cloud environments, where traffic patterns can change rapidly.

Once trained, the system offers real-time predictions, which are crucial for dynamically

optimizing resource allocation and minimizing latency in distributed systems. The ability to mak accurate, timely predictions enables cloud providers to efficiently manage resources, ensuring that throughput is maximized while delays are minimized. This approach provides a robust solution for improving cloud data transfer, enhancing system performance, and contributing to more efficient and reliable distributed computing environments.

Key components of the system include:

Data Collection and Preprocessing: Gathering historical cloud traffic data and performing necessary cleaning, normalization, and encoding to ensure high-quality inputs for the predictive models.

Feature Engineering: Creating relevant features from raw data, such as traffic volume, latency, server load, and network congestion, to capture key patterns that influence throughput.

Model Selection and Training: Employing advanced machine learning models like LSTM and CNN to train the system, with the ability to handle time-series data effectively.

Performance Evaluation: Assessing model performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared, ensuring accurate and reliable throughput predictions.

Optimization and Tuning: Fine-tuning model parameters and employing techniques like cross-validation to enhance the model's generalization ability and accuracy across different traffic scenarios.

This study is carried out to evaluate the effectiveness and feasibility of implementing a predictive model for cloud data transfer throughput in distributed systems. The system must efficiently analyze traffic trends and provide accurate short-term forecasts to support real-time resource allocation. The model's development considered constraints such as computational complexity, data availability, and system scalability. The majority of the tools and technologies used, including machine learning frameworks and data processing libraries, are open-source and readily accessible. This significantly reduces development costs while maintaining performance standards. Only specific infrastructure resources, such as cloud storage or GPU instances for model training, incurred minimal cost. Overall, the system is both technically sound and economically viable.

SYSTEM ARCHITECTURE

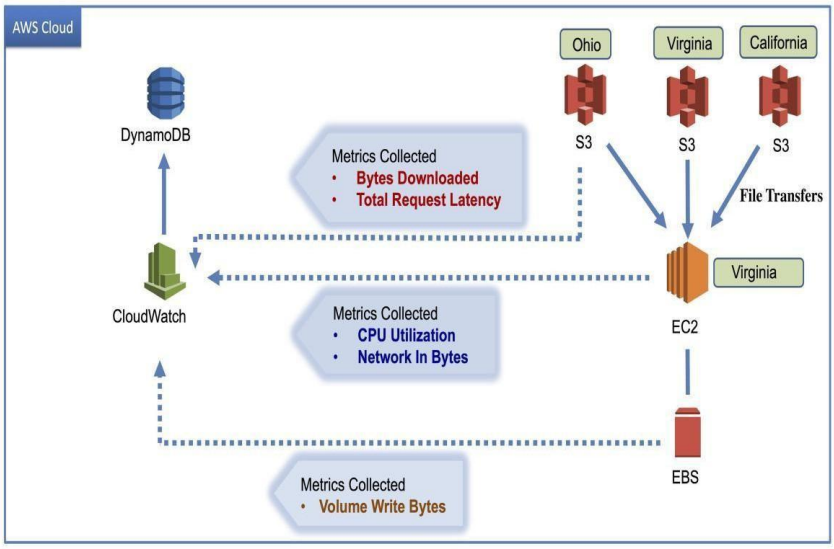


Fig.1. System Architecture

It shows the service architecture that we used in the AWS cloud to collect multivariate time series data about end-to-end data transfers. We conducted file transfers between S3 Object Storage System buckets and EC2 instances.

DATA COLLECTION

Table 1 shows the source and destination regions.The EC2 instances are created in *useast- 1* region (Virginia) while the files can reside in different regions throughout the US.

AWS CloudWatch is a monitoring service that collects vast amounts of metrics from all services.

We collected metrics from S3, EC2, and EBS services. Since EC2 instances usually do not have storage, they are connected to EBS block storage via a storage area network.

We collected different types of metrics from CloudWatch and transformed and stored them in a NoSQL database service DynamoDB to be extracted later in JSON format.

TABLE 1. Source and destination regions.

S3 Source Region	EC2 Destination Region
Virginia (us-east-1)	Virginia (us-east-1)
Ohio (us-east-2)	Virginia (us-east-1)
California (us-west-1)	Virginia (us-east-1)

TABLE 2. File and dataset characteristics.

Dataset Size	Average File Size
10GB	1MB
50GB	100MB
50GB	1GB

TABLE 3. Instance characteristics.

Instance Type	# of vcpus	Network Performance
t2.small	1	Low to moderate
t2.xlarge	4	Moderate
t3.small	2	Up to 5Gbps

Fig.2. Experimental setup showing regions, dataset characteristics, and instance specifications.

This behavioral difference has a direct effect on the achievable throughput. Therefore, we created files in different average sizes (1MB, 100MB, and 1GB).
Cloud providers do not exactly give out the maximum bandwidth of their networks and Instance types are usually listed with *Low*, *Moderate*, or *Upto X* GBPS network performance.
Considering the source regions we used, which allowed us to have different latencies, we had 27 different settings (3 source regions x 3 instance types x 3 average file sizes) in our file transfers.

DATA TRANSFORMATION

In this step, raw data is cleaned and made suitable for model input:
Handling Missing Values: Missing entries are identified and addressed using imputation techniques or data removal.
Scaling and Encoding: Numerical features are scaled, and categorical variables are encoded using methods like one-hot encoding to ensure uniformity across the dataset.

LOCAL AREA THROUGHPUT TRANSFORMATION

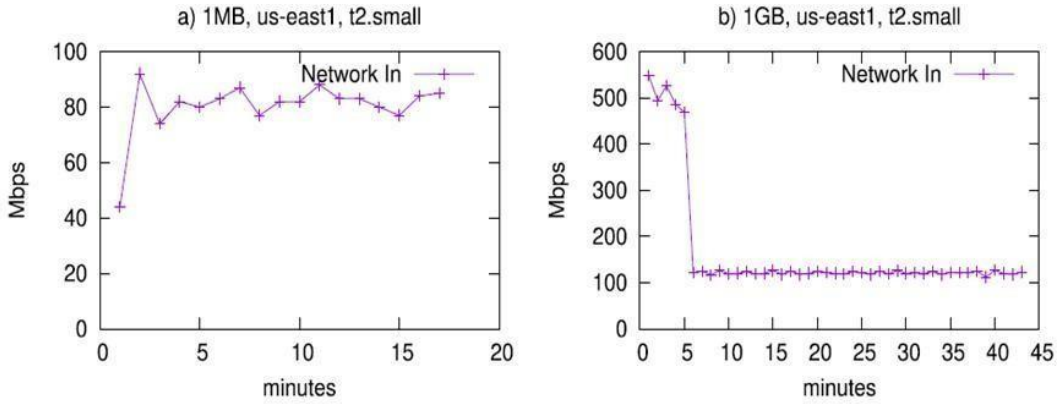


Fig.3. Network In for 1MB and 1GB files on t2.small in us-east-1

While conducting the file transfers we knew that there would be different parameters that will affect the achievable throughput.
Based on the size of the file, the TCP protocol might show different behavior:
A file transfer can spend much of the time in *slow start* phase if it is small.
A file transfer can spend much of the time in *congestion avoidance phase* if it is large.
When the file size is small, the throughput remains under the 100 Mbps limit in local area transfers.

WIDE AREA THROUGHPUT TRANSFORMATION

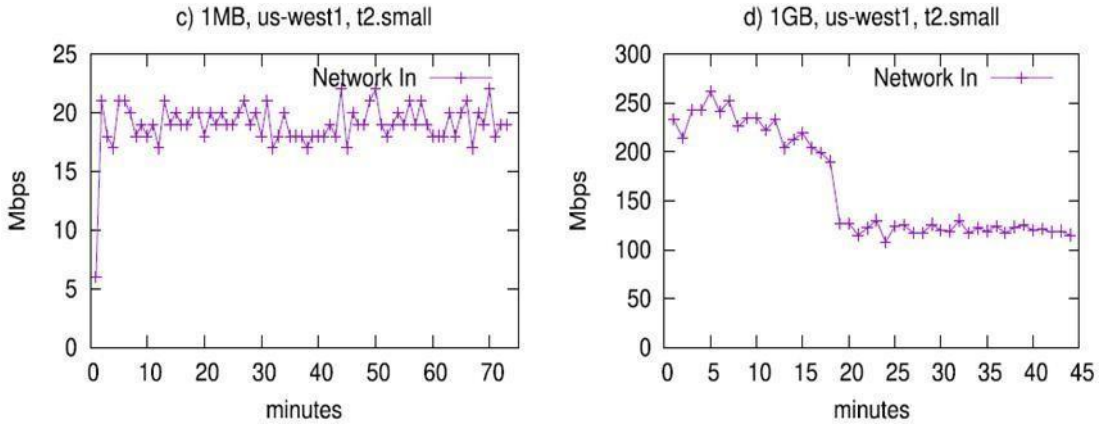


Fig.4. Network In for 1MB and 1GB files on t2.small in us-west-1

The wide-area throughput varies more compared to local area throughput over time.
It only reaches 20 Mbps for wide-area transfers due to latency.
When we look at large file transfers, they have a higher throughput since they spend most of the transfer time in the congestion avoidance phase but after some point, the throughput drops down dramatically to 100Mbps band.
We suspect this dropdown is due to bandwidth-shaping strategies applied by the cloud provider.
It takes longer for the shaping strategy to kick in for wide-area transfers (20 minutes into transfer) than local-area transfers 5 minutes into transfer).
This phenomenon might be due to the strategy being applied after a certain number of bytes are transferred.

MODEL EVALUATION METRICS

Metrics provided by cloud systems are good indicators of the load, especially for the black box storage systems like S3. We collected 5 different metrics on the end-to-end data transfer path:

Bytes Downloaded (BD): The sum of the number of bytes downloaded for requests made to an Amazon S3 bucket, where the response includes a body over a period.

Total Request Latency (TRL): The sum of the elapsed per-request time from the first byte received to the last byte sent to an Amazon S3 bucket.

NetworkIn (NI): The number of bytes received by the instance on all network interfaces.
This metric identifies the volume of incoming network traffic to a single instance.

CPU Utilization (CU): The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application on a selected instance.

Volume Write Bytes (VWB): The total number of bytes written on EBS volumed.

$$\text{MAE} = \frac{\sum |y - \hat{y}|}{n}$$

$$\text{MAPE} = \frac{100}{n} \sum \frac{|y - \hat{y}|}{|y|}$$

We decided to use MAE and MAPE to measure the accuracy of our models. Although MAE provided better results as a loss function compared to MAPE (training and validation curves converged easily for MAE), it is not a good performance metric for our models for two reasons.
First, Our throughput values come from very different types of networks and settings so
their ranges are quite different. Second, MAE value can differ for different datasets so it cannot be compared against the results of other studies. So, in our study, we used MAPE to compare the performance of our models but in some cases also provided MAE results as well because certain trends were more visible in MAE.

MODEL COMPARISION

We are interested in predicting end-toend data transfer throughput for which the transfers start from the disk of the source storage system and end again at the disk volume of the destination system. Therefore we focus on predicting the one-step-ahead *NetworkIn* and *VolumeWriteBytes* metric to have an idea about how the destination system observes the data transfer throughput.
Phanekham et al. claim that only the least-correlated variables benefit the prediction model most, therefore they analyze the correlations among their metrics and select three: network throughput, latency, and buffer size. While the former two are time series data, the last one is not. The best MAPE result they were able to achieve is 29% for predicting daily traffic.

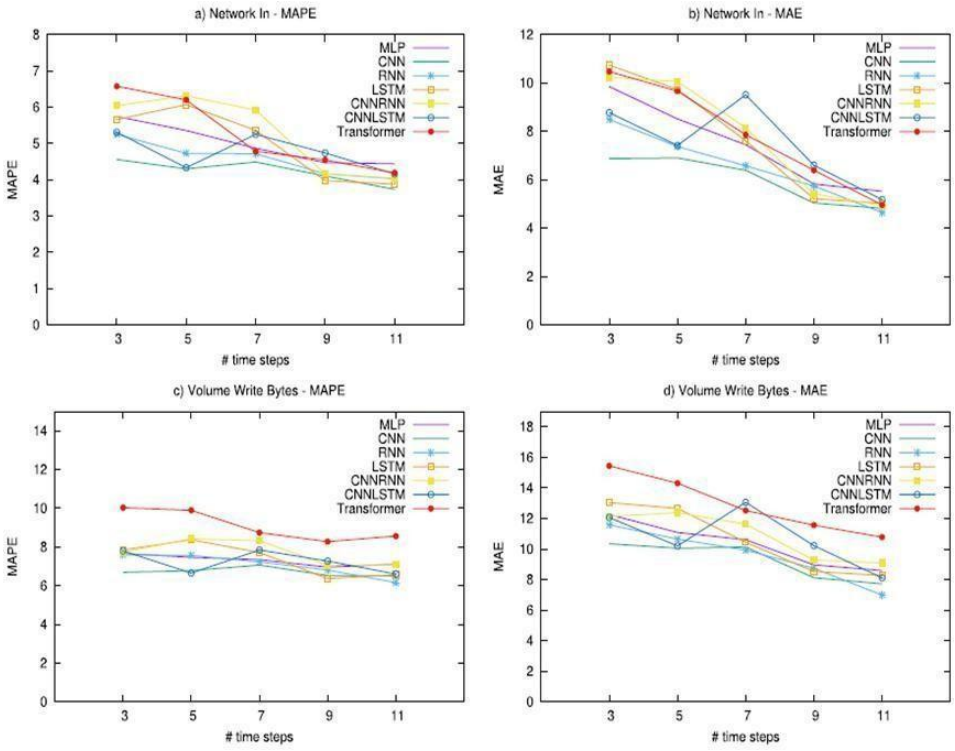


Fig.5. Error Rate of Prediction Models vs Time Steps

The bold values represent the best values among 5-metric multivariate, 2-metric multivariate, and univariate models for NI and VWB metrics. The 2-metric multivariate MLP is best among other MLP variations for the prediction of NI metric, the 2-metric multivariate Transformer is best among other Transformer variations for the prediction of VWB metric, and the univariate Transformer model is best among other Transformer variations for the prediction of NI metric.

According to this result, our 5-metric multivariate model outperforms all other models with **3.73%** MAPE for the prediction of NI metric with a CNN model and with **6.16%** MAPE for the prediction of VWB metric with an RNN model.

SYSTEM IMPLEMENTATION

SYSTEM WORKFLOW

The proposed system architecture consists of multiple interconnected modules, each responsible for a specific stage of the throughput prediction pipeline. The architecture ensures modularity, scalability, and accurate short-term prediction of data transfer throughput in cloud environments.Each module—from data collection to model prediction—operates sequentially to transform raw cloud metrics into actionable insights. This workflow supports intelligent automation for performance tuning, resource optimization, and proactive decision-making in realtime cloud operations.

SYSTEM MODULES

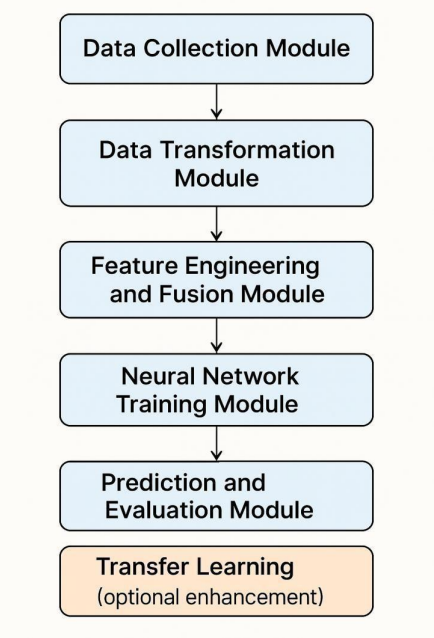


Fig.6. Workflow Diagram

DATA COLLECTION MODULE

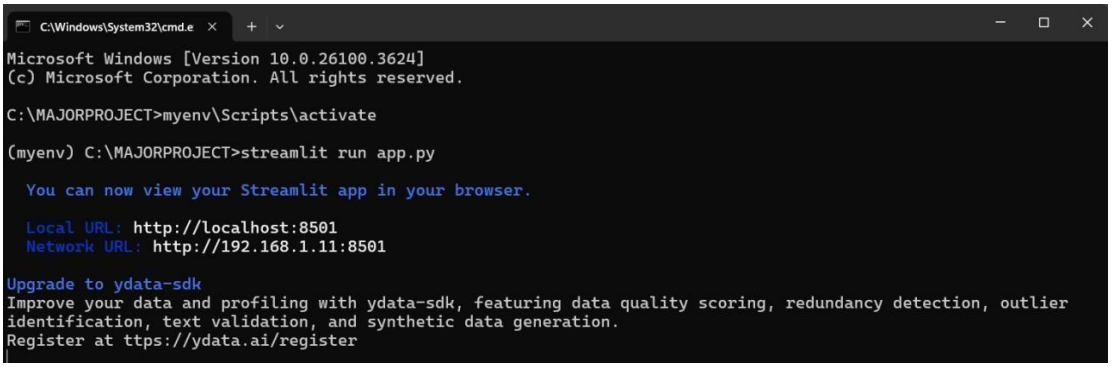
This module is responsible for gathering raw system metrics from cloud resources during active data transfers. It collects real-time performance data from Amazon EC2 instances and services like Cloud Watch. Both sender and receiver machines are monitored, ensuring that data is captured from multiple perspectives. The collected metrics include CPU utilization, memory usage, disk write throughput, network packets, and latency values. Purpose: To provide a comprehensive, time-synchronized dataset that reflects the state of the system during transfers.

DATA TRANSFORMATION MODULE

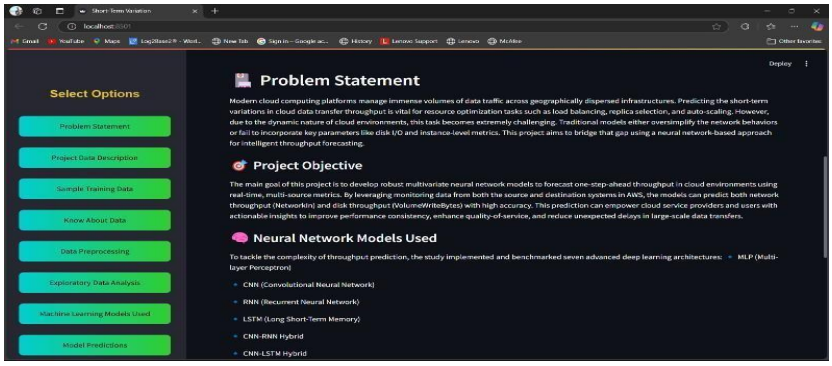
Once collected, the raw data must be cleaned, structured, and prepared for further processing.
This module handles synchronization of timestamps between sender and receiver.
It removes anomalies such as failed transfers or inconsistent logs.
Metrics are normalized, converted into a consistent format (e.g., Mbps), and stored in a structured database like DynamoDB.
Purpose: To prepare clean, standardized data that is ready for feature engineering and modeling

Results

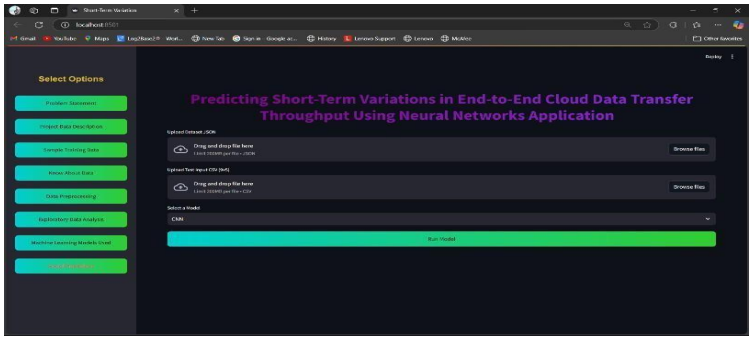
The execution of the process will be explained clearly with the help of continuous screenshots
Step1:Hosting the website in a browser.



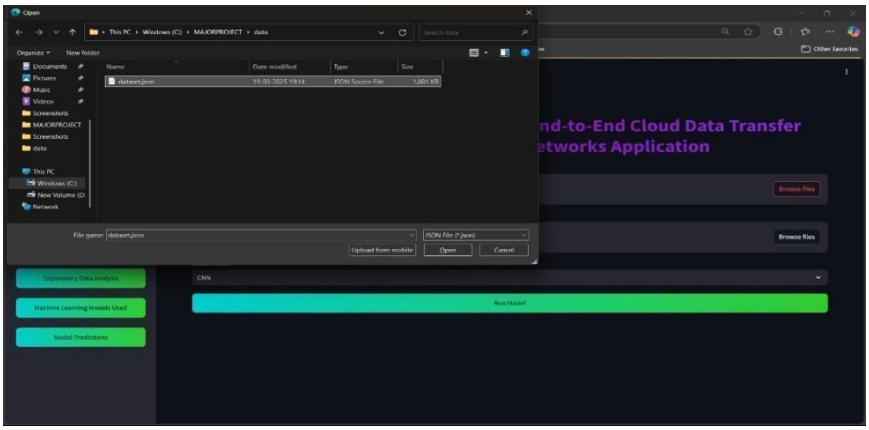
Step2: Arriving at Home Page.



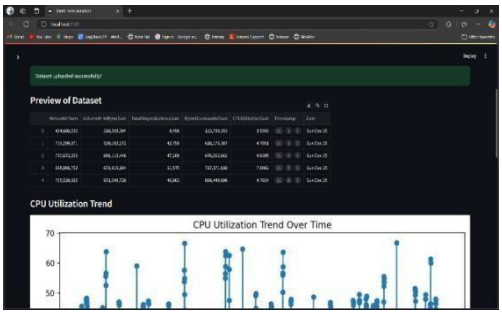
Step3: Selecting the Model Predictions



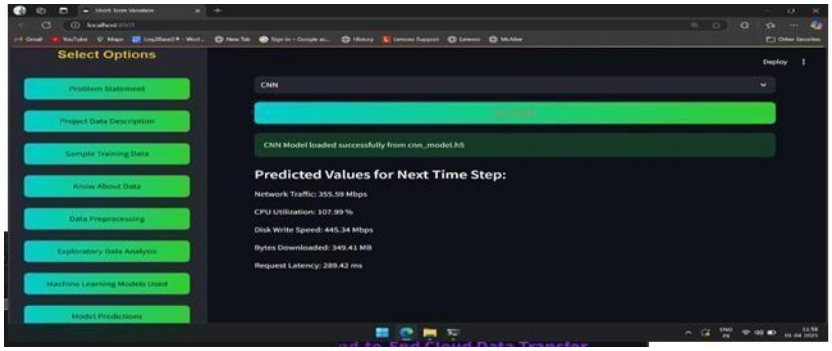
Step4:Uploading the Dataset in Model Predictions.



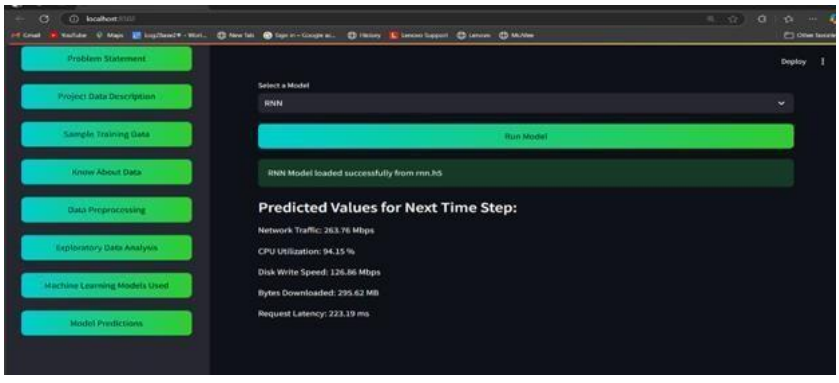
Step5: Dataset Preview and Visualization.



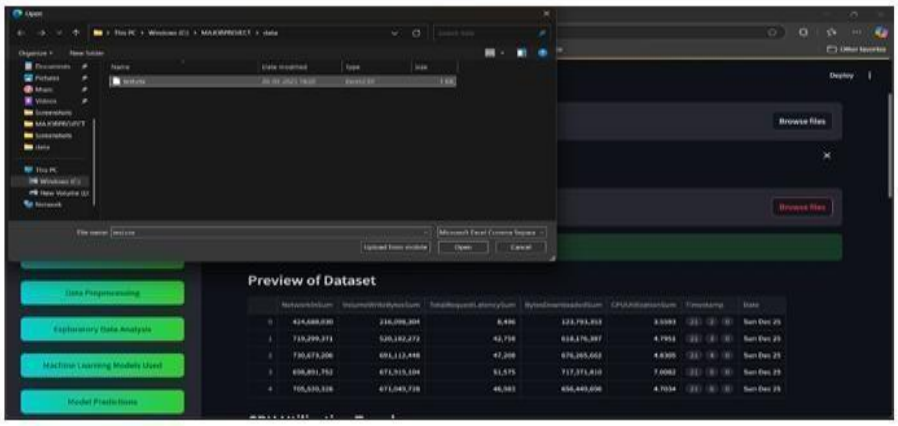
Step6: Uploading the Test.csv in Model Predictions.



Step 7: Model Prediction Output Using CNN.



Step 8: Model Prediction Output Using LSTM.



Step 9: Model Prediction Output Using RNN.

Conclusion

This project successfully developed a predictive system to estimate short-term variations in end-to-end cloud data transfer throughput using multivariate neural network models. By collectingreal-timemetricsfromAWSservicesandapplyingdeeplearningtechniques, the system achieved high accuracy, with models like CNN and RNN outperforming traditional baselines in predictingbothnetworkanddiskthroughput.Themodulararchitecturecomprisingdatacollection, transformation, feature engineering, and model training ensured scalability and reliability across different cloud configurations. Testing validated the system's performance under varying conditions such as instance types, file sizes, and regional transfers. Additionally, transfer learning was explored to enhance performance in data-limited scenarios. The proposed system provides an efficient and scalable solution for throughput prediction in cloud environments and lays the groundwork for future enhancements in auto-scaling, load balancing, and replica selection.

Future Scope

While the proposed model delivers accurate short-term throughput predictions, there is ample scope for further development. Extending the model to support long-term forecasting would enable more strategic planning for large-scale data transfers. Integrating the system with real-time cloud orchestration tools could allow dynamic tuning of transfer parameters like concurrency and parallelism during active operations. The predictive output can also support intelligent replica selection and load balancing to optimize resource usage. Expanding compatibility across cloud platforms such as Azure and Google Cloud would improve its versatility, while incorporating model explainability techniques like SHAP or LIME could enhance transparency. Finally, applying optimization methods like pruning and quantization would make the system more efficient for real-world deployment in resource-constrained environments.

References

1. C. So-In. (2009). A Survey of Network Traffic Monitoring and Analysis Tool. St. Louis,MO,USA.[Online].Available:https://scholar.google.com/scholar?as_q=A+Survey+ of+Network+Traffic+Monitoring+and+Analysis+Tool&as_occt=title&hl=en&as_sdt=0 %2C31,http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.7759&rep=rep1& type=pd

2. T. H. H. Aldhyani and M. R. Joshi, ‘‘Intelligent time series model to predict bandwidth utilization,’’ Int. J. Comput. Sci. Appl., vol. 14, no. 2, pp. 130–141, 2017.
3. A. Azari, M. Ozger, and C. Cavdar, ‘‘Risk-aware resource allocation for URLLC: Challenges and strategies with machine learning,’’ IEEE Commun. Mag., vol. 57, no. 3, pp. 42–48, Mar. 2019.
4. C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, ‘‘Spatio-temporal wireless traffic prediction with recurrent neural network,’’ IEEE Wireless Commun. Lett., vol. 7, no. 4, pp. 554–557, Aug. 2018.
5. M. Chen, ‘‘Machine learning for wireless networks with artificial intelli- gence: A tutorial on neural networks,’’ 2017, arXiv:1710.02913. [Online]. Available: https://arxiv.org/abs/1710.02913
6. H. D. Trinh, L. Giupponi, and P. Dini, ‘‘Mobile traffic prediction from raw data using LSTM networks,’’ in Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC), Sep. 2018, pp. 1827–1832.
7. Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, ‘‘A hybrid deep learning based traffic flow prediction method and its understanding,’’ Transp. Res. C, Emerg. Technol., vol. 90, pp. 166–180, May 2018.
8. M. A. Shoorehdeli, M. Teshnehlab, and A. K. Sedigh, ‘‘Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter,’’ Fuzzy Sets Syst., vol. 160, no. 7, pp. 922–948, Apr. 2009.
9. J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, ‘‘Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach,’’ in Proc. IEEE Conf. Comput. Commun. (INFOCOM), May 2017, pp. 1–9.
10. H. K. Choi, ‘‘Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model,’’2018,arXiv:1808.01560.[Online].Available: http://arxiv.org/abs/1808.01560.
11. C.-W. Huang, C.-T. Chiang, and Q. Li, ‘‘A study of deep learning networks on mobile traffic forecasting,’’ in Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC), Oct. 2017, pp. 1–6.
12. R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, ‘‘The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice,’’ IEEE Commun. Mag., vol. 52, no. 6, pp. 234–240, Jun. 2014.
13. G. Lai, W.-C. Chang, Y. Yang, and H. Liu, ‘‘Modeling long- and short-term temporal patterns with deep neural networks,’’ in Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Jun. 2018, pp. 95–104.
14. A. Azzouni and G. Pujolle, ‘‘NeuTM: A neural network-based framework for traffic matrix prediction in SDN,’’ in Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS), Apr. 2018, pp. 1–5.
15. Komperla, Ramesh Chandra Aditya, et al. "A Novel Approach to Diabetic Foot Ulcer Prediction: Pedographic Classification Using ELM-PSO." *2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC)*. IEEE, 2024.2024-05-02