International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 747-759
https://doi.org/10.70917/ijcisim-2026-2791
Article

Diagnose, Forecast, and Auto-Recover from Failures Using Machine Learning Algorithms for Univariate and Multivariate Metrics in Cloud Data Migration

Ramya M C[1*], Thriveni J.[2]

[1]Department of Computer Science and Engineering University of visveswaraiah college of engineering Bangalore, India ramyamc.sjce@gmail.com
[2]Department of Computer Science and Engineering University of visveswaraiah college of engineering Bangalore, India drthrivenij@gmail.com

Abstract—Data migration to the cloud often presents significant challenges, necessitating robust diagnostic techniques to identify and resolve issues. This study explores the application of Hotelling's T^2 method and MYT (Myth) decomposition for diagnosing problems in data migration processes. By leveraging both univariate and multivariate data, we aim to provide a comprehensive approach to detect anomalies and ensure the
integrity and performance of the migrated data. Our results demonstrate the effectiveness of these statistical methods in pinpointing migration issues, thereby facilitating more reliable and efficient cloud data migration strategies.
This paper presents an in-depth analysis of network anomaly detection techniques tailored for cloud computing environments. We explore the challenges associated with detecting anomalies in dynamic and scalable cloud networks and examine the implications of these anomalies on cloud service performance and security. Furthermore, we discuss emerging technologies and innovative approaches, including machine learning algorithms and statistical methods, that can improve the precision and effectiveness of detecting anomalies in cloud-based networks.
Keywords: Data Migration, Cloud Computing, Anamoly detection, Security.

I. INTRODUCTION

Data migration to the cloud is a crucial process for organizations looking to improve data management, increase access, and use powerful cloud analytics. But this migration comes with its own set of challenges, such as data integrity problems, performance issues, and possible disruptions to business operations. It is crucial to accurately and quickly resolve these issues to ensure an easy transition and reliability of the migrated data.

Traditional diagnosis tools are often inadequate in dealing with complexity and multidimensionality of data migration issues. To tackle this, Hotelling's T^2 method and MYT (Myers and Tyler) decomposition are explored. These statistical tools can be powerful tools in uncovering anomalies and help to understand the root causes of migration challenges. We hope to use these approaches for both univariate and multivariate data to build a comprehensive toolset for identifying problems in cloud data transfers.

The Hotelling's T^2 technique is particularly useful for multivariate data analysis, where it can be used to detect anomalies and assess the uniformity of the data across multiple dimensions. On the other hand, MYT decomposition offers a methodical process for breaking down the overall variability in the data, supporting the detection of particular error sources.

We show here how these methods can be applied to a scenario for simulated data migration and their usefulness in the diagnosis and resolution of migration problems. The results of our work highlight the need for using advanced analytical methods to improve cloud data transmission process's reliability and efficiency.

II. LITERATURE SURVEY

Data migration to cloud computing technologies has already been discussed extensively, and many recommendations have been provided for overcoming challenges involved and ensuring the correctness and efficiency of the process. This paper provides an overview of scientific research and methods aimed at detecting data migration problems, highlighting the use of statistical methods such as Hotelling's T^2 and MYT decomposition in particular

Data Migration Challenges

Many studies have pointed to the technical and operational difficulties in moving data to the cloud. Some of

these are data loss, data corruption, latency, and compatibility issues [**?**], [**?**]. Common tools for data accuracy are often not enough to cope with complex cloud environments, which require more sophisticated diagnosis tools.

A. Statistical Methods in Data Migration

In recent years, the use of statistical methods for diagnosing data migration problems has become popular and one of the most successful process control methods of this type is the "multivariate statistical" method T^2, known as "Hotelling's $T^{2"}$ [?]. The method is especially good for detecting abnormalities when data migration involves more than one variable.

Another statistical method, called the MYT decomposition, is capable of decomposing complex multivariate data structures into more manageable ones [?]. This method helps to identify the sources of variation, thus pinpointing the causes of migration problems.

B. Applications of Hotelling's T^2 *Method*

This "Hotelling's $T^{2"}$ technique is widely used in other areas such as quality control, finance and biostatis-tics [?] [?]. To date, its use has been mainly limited to data migration, with a promising application. It has been proven to be effective in data integrity check and anomaly detection during migration process [?] and [?].

C. MYT Decomposition in Multivariate Analysis

The decomposition of MYT has been used in different fields to make the analysis of multivariate data easier [?], [?]. The use of this tool to diagnose data migration problems is becoming a valuable method because it breaks data down into independent components for more accurate detection of anomalies and the cause of those anomalies.

D. Combined Use of Hotelling's T^2 *and MYT Decomposition*

Hotelling's T^2 method is complemented by MYT decomposition and can be used to provide a powerful diagnostic tool in data migration problems. Hotelling's T^2 method gives an overall indication of anomalies in the data, but MYT decomposition can help to understand the individual variables that cause anomalies. This combined approach has the potential to enhance the accuracy and reliability of diagnostics in data migration [?], [?].

The Although the traditional approach to diagnosing data migration problems is still widely used in the literature, more and more advanced statistical methods, such as Hotelling's T^2 method and MYT decomposition, are gaining traction.

The above techniques provide a strong foundation for anomaly detection and analysis in typical complex multivariate data sets encountered in cloud migrations. Further studies are needed to better understand, develop, and verify these methods to enhance the accuracy and efficiency of data transfers.

III. Migration Monitoring Overview

Migration flow monitoring is an essential part of data migration to the cloud, where it is crucial to ensuring a smooth and efficient migration process. It includes ongoing monitoring and tracking of data transfers to identify and address problems in real-time. With effective monitoring, you can gain a real-time view of the migration process to detect bottlenecks, performance problems, and data inconsistencies. Monitoring tools and dashboards can provide organizations with information on the migration's progress, the effectiveness of data transfers, and the ability to make informed decisions to optimize the migration.

Establishing a comprehensive system for tracking migration flows is crucial for maintaining data integrity, reducing downtime, and meeting regulatory standards. Good practices involve creating alerts to suspicious activity, regular health checks and relying on automated tools to monitor data transfers. Key metrics to monitor include transfer speed, error rates, and system performance. In addition, every incident response plan should be clearly defined to deal with potential incidents that may occur throughout migration. Monitoring the migration flow proactively can enable organisations to ensure a smooth move to the cloud, reduce risks and drive the project towards a successful outcome.

As shown in Fig.1, we are visualizing the data movement from a wide variety of sources, such as cloud environments, on-premises data center, VMware systems, etc., to the cloud. This comprehensive view allows us to track and manage the intricate process of data migration, ensuring all aspects are covered and any potential issues are promptly addressed.

This is a key element where our monitoring system is important as it monitors a number of key factors throughout. It keeps track of security risks to ensure that data is kept secure and follows security procedures, giving immediate warnings and alerts to help prevent any potential security breaches. Loss of packets is carefully monitored to see if the data is being transmitted reliably and efficiently, to reduce the chance of damage to the data or loss of data during the transmission process.

The system also closely monitors key performance parameters, besides security and data integrity. Monitoring CPU usage to ensure that the processors have their use optimised, without causing bottlenecks or any drop in performance. Memory utilization is monitored to avoid overloading, and if they are being used, they are kept to a high performance during data migration.

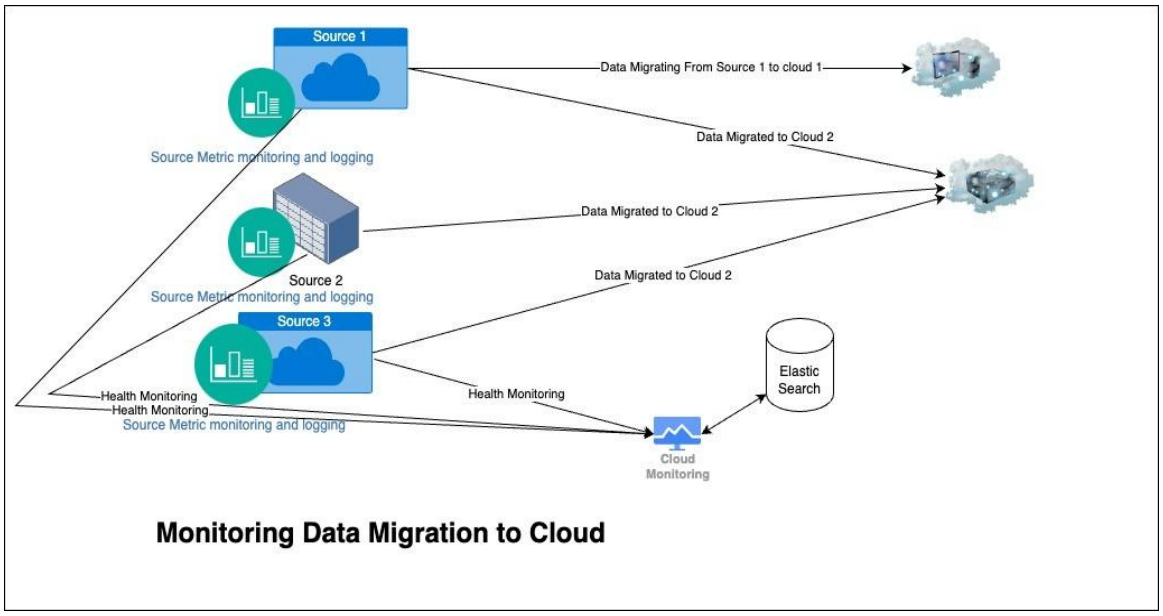

Fig. 1. Migration Monitoring Overview

I/O performance is monitored to ensure that data is being moved to and from the device at the optimal rate for read/write operations. The activities of the network of cells are also closely monitored to control the bandwidth and process of stable, efficient data transfer.

These are the key features that are visualised and monitored to ensure a smooth, secure and efficient data migration from cloud environments, data centers, and VMware systems to the cloud. This holistic approach helps reduce risks and improve the overall effectiveness and trustworthiness of the transmission process.

Baselines for metrics data are being developed from monitoring data collected and analysed. This gives us a good idea about how things will turn out and what problems might occur before they do. In case of error conditions, our system will propose possible troubleshooting in case of hardware failures, which will ensure a rapid and efficient resolution. The system is also capable of generating events automatically and implementing resolutions without manual intervention for software-related issues, reducing downtime and ensuring smooth operations.

It is a proactive way of monitoring and analysis, continuously collecting data on “key performance indicators(KPI)” like “CPU usage”, “memory utilization”, “I/O performance”, and “network activity”. This information is used to determine an overall baseline of our systems in normal operation. Any variation from this baseline can signify potential issues and help us to take proactive steps.

If there are any hardware failures, the system will give comprehensive troubleshooting information, helping technicians to correctly diagnose and solve the failure. This involves detecting the problem, recommending replacement components and offering detailed repair procedures. This will allow hardware problems to be resolved quickly and efficiently without affecting the overall system performance.

When it comes to software problems, the system has a more automated approach. If an anomaly is found, an event is created that initiates some predefined corrective measures. These can involve service restarts, patching or changing configuration. Software problem resolution can be automated to quickly get back to normal operation and without manual intervention. Overall, our approach to baselining and predictive analysis, combined with automated troubleshooting and resolution, enhances the reliability and efficiency of our data migration process. This ensures that potential issues are identified and addressed promptly, maintaining the integrity and performance of our systems during the transition to the cloud.

IV. Design Overview

In Fig. 2, we are working with the metrics coming from monitoring system, that is used to monitor KPI. All these metrics are captured and saved in the database on an ongoing basis, so that they can be monitored and analyzed in real time as well as in the past. The data is then processed based on pre-defined Service Level Agreements (SLAs) and any deviation from these SLAs is dealt with appropriately expected performance thresholds are quickly identified. After the metrics have been processed, the system compares them to determine if there are any issues. When these limits are breached (such as CPU or memory usage limits, or high I/O wait time), an alarm is automatically triggered and the relevant operations and support teams are alerted and notified to take action.

If the problem is detected, it decides if it is a hardware or software problem. On the other hand, if a hardware failure is detected (e.g. a server component failure, or the availability of a critical hardware resource is determined to be

unavailable) a ticket is created to facilitate further investigation and resolution of the failure by the proper teams. The automation system, on the other hand, takes a more proactive approach when the problem is software and is determined to be solvable via automation. It first tries to identify the root cause from the error log, system behavior and troubleshooting of possible configuration problems. Upon detection, the system can perform an auto-correction process, such as service restarts, configuration changes or software patches, without manual involvement. This automated process ensures that software-related issues are swiftly addressed, minimizing downtime and improving system reliability.

This is a description of all the core components of a complete fault identification & diagnostic architecture for virtualized data hubs, and how they fit in with other parts of the system. The monitored framework is at the heart of the process as it allows to monitor key system parameters during the data migration process to the cloud like “CPU utilization”, “network I/O”, “memory” and “disk I/O”. These parameters and their timestamps are kept in an archive of “time-series”. The prediction module analyzes and decodes the trend of resource consumption of applications, using this archive.

This element is used in conjunction with the calibration module to transform network utilization data from the present physical host into ones that can be compared to a benchmark physical system. Elements that detect anomalies verify resource usage statistics with the projections from the forecasting module. The malfunction identification unit communicates with these anomaly detection elements to detect irregularities in the application, and alerts a warning sign to take action prior to serious loss of application performance. If something was detected, the fault diagnosis module analyzes data from the fault detection system to determine the exact cause of the fault. It then passes this to the governance framework based on rules to create a direction, and this direction is then fed into the allocation system to solve the problem.

If abnormalities are not deteced by the fault detection module,the module informs the policy-driven system, which subsequently assesses various time-based policies. As per these guidelines, the allocation system allocates resources at particular times. The surveillance system can only store any recorded metrics of the various system variables in the temporal database in the absence of any notifications created by the anomaly identification system.

Fig.1 We are also proactively keeping track of the health of both data and network metrics on our switches with an extensive monitoring solution. This system will allow us to collect real-time information on different performance indicators, so that, we can quickly detect any anomalies or possible problems. Through the evaluation of metrics like; bandwidth utilization, latency and error rates we can ensure that we maintain optimal performance and reliability within our network infrastructure. This proactive plan does not only help in the smooth running but also enhances our ability to respond promptly to any kind of disruption which in the long run helps in enhancing the overall performance of our network environment.

Cloud monitoring systems are the necessary components of cloud computing setups, which make sure that cloud services and applications operate, perform, and are safe. They provide real-time updates on different measures, enabling businesses to optimize resource utilization, with fast detection of problems and enhancing the overall user experience.

V. SOLUTION DESCRIPTION

The dstat tool in this case tracks the mode of reception and transmission of packets to and through the clouds. It also collects I/O process information to determine any running processes during data migration that can cause network latency problems. Additionally, we are collecting performance metrics such as memory and CPU usage.

Once the third-party tool has been used to collect its data, it will be processed as shown in Fig 2. The major parts used in the computation of CPU, memory are the following:

A. Monitoring system with Threshold Component definition

The main idea behind this component is to collect and record essential configuration information, defining key aspects like hardware specifics, socket details and switch con-configurations. It also captures hypothetical CPU, memory and network requirements that are relevant to the systems and cloud environments involved in the migration process. This involves detailing the setups of machines where applications function over their entire lifespan, recognizing that an application might reside on diverse machines, each with its distinct specifica-tions. Initially, forecasting is conducted using hypothetical parameters before being relayed to the fault detection system. This component establishes threshold values for each metric associated with every component participating in the migration process.

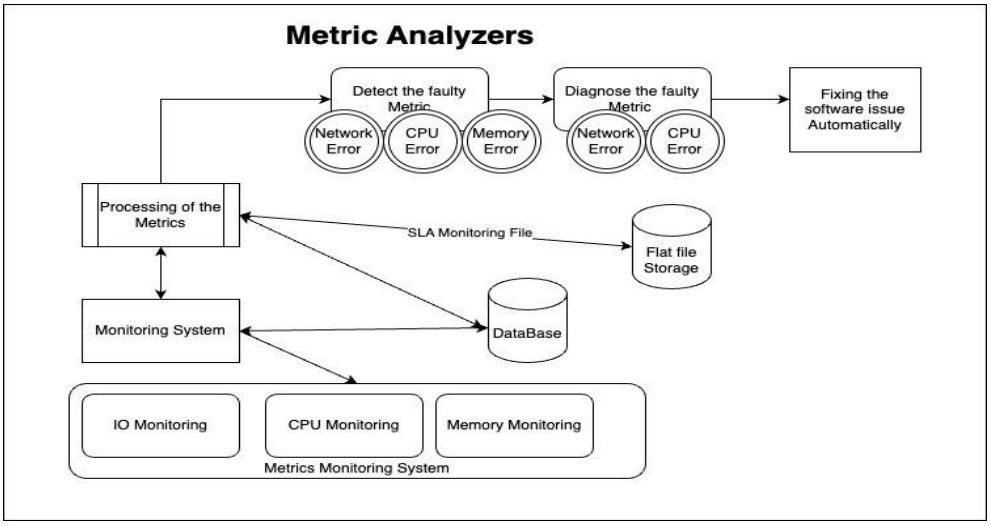

Fig. 2. Design Overview of the Monitoring System

B. Fault Detection

Hotelling's T^2 algorithm is a statistical method used for multivariate analysis. It's particularly useful in network analysis where there are multiple variables to consider simultaneously. This algorithm helps in understanding patterns, correlations, and anomalies within complex network datasets.

During the data migration process, network flaws can disrupt real-time data flow or migration, occasionally causing issues in the data copying phase, resulting in inconsistent data copies or data loss.

Forecasting or predicting network issues based on hypothetical or historical network data analysis assists in

maintaining consistency in time and data integrity.

The objective of the prediction module is to deconstruct the past utilization data of each computing asset into its distinct elements, which are elaborated upon in Table 1. We utilize the multiplicative model, which has the generalized mathematical formulation in the following form:

$X_t=(T_t*S_t*C_t)*(E_t)$

HOTELLING'S T^2 OUTLIER DETECTION

Data Preparation

1) Organize your multivariate time-series dataset into a matrix, where each row corresponds to a specific time

TABLE I HOTELLINGS T^2 FORMULA ANNOTATION

Notation	Representation	Details
(X_t)	hoteling value at time	Hoteling value at time
(T_t)	Trend Component	Temperature at time
(S_t)	Seasonal Component	Service quality at time
(C_t)	Cyclical Component	Periodic quality at time
(E_t)	Error Component	Error satisfaction at time

point, and each column represents a distinct variable.

2) Calculate the mean vector ($\bar{\mathbf{X}}$) and the covariance matrix

(**S**) of your data.

Hotelling's T^2 Calculation

For each time point t, calculate the Hotelling's T^2 statistic using the formula:

$$T^2 = (\mathbf{X}_t - \bar{\mathbf{X}})^T \mathbf{S}^{-1} (\mathbf{X}_t - \bar{\mathbf{X}})$$

Here, $\mathbf{X}_t$ is the vector of observations at time t, $\bar{\mathbf{X}}$ is the mean

vector, and **S** is the covariance matrix.

Outlier Detection

Compare the calculated T^2 values against a critical thresh-old derived from the Hotelling's T^2 distribution. If T^2 exceeds the critical threshold, you may consider the corresponding time point t as an outlier.

Algorithm 1 outlining the Hotelling's T^2 Outlier Detection

Interpretation

Investigate the time points identified as outliers to discern the nature of anomalies or deviations in your data.

Algorithm 1 Hotelling's T^2 Outlier Detection

Input: Multivariate time series data matrix X, Significance level α
Calculate mean vector $\bar{X}$ and covariance matrix S from X
for each time point t in X **do**
    Calculate T_t^2 using the formula:

$$T_t^2 = (\mathbf{X}_t - \bar{\mathbf{X}})^T \mathbf{S}^{-1} (\mathbf{X}_t - \bar{\mathbf{X}})$$

    Compare T_t^2 with critical threshold from Hotelling's T^2 distribution
    if T_t^2 > critical threshold **then**
        Print: Anomaly detected at time point t
    end if
end for
Output: Detected anomalies

Time Series Decomposition Algorithm

To forecast future utilization of the resource, we will first find the cyclical component, later will find the trend, seasonal component. Each component forecasts are combined in future to find the final forecast using multiplicative model

Algorithm 2 outlining the Hotelling's T^2 Time Series Decomposition

VI. Steps Involved in Preprocessing Data using

Htelling's T^2 method

Hotelling's T^2 is a multivariate statistical method used to detect patterns or anomalies in network latency data over time. Here's how you can apply it:

1) **Data Collection:** Gather network latency data at regular intervals over a specific period of time. Record latency values for each time instance.

2) **Data Preparation:** Organize the collected latency data into a array where each line corresponds to a time in-stance, and every column denotes a different dimension of latency.

3) **Calculation of Mean Vector:** Calculate the mean vector $\bar{X}$ of the latency data. This represents the average la-tency values across all dimensions at each time instance.

4) **Calculation of Covariance Matrix:** Calculate the co-variance matrix S of the latency data. The covariance matrix represents the relationships between different dimensions of latency.

5) **Calculation of Hotelling's T^2 Statistic:** Compute the Hotelling's T^2 statistic using the formula:

$$T^2 = n(\bar{X} - \mu)^T S^{-1} (\bar{X} - \mu)$$

Where n is the number of observations (time instances), $\bar{X}$ is the mean vector, μ represents the expected mean vector, and S^{-1} is the inverse of the covariance matrix S.

Algorithm 2 Time Series Decomposition

```
Input: Time series data X with length n, Seasonality length L
Output: Trend X_t, Cyclical component X_t, Seasonal factors, Random error

procedure COMPUTETOTAL(X, L)
    Compute total for each L-period
end procedure
procedure CALCULATEMOVINGAVERAGE(X, L)
    Calculate moving average for each L-period
end procedure
procedure CALCULATECMA(X, L)
    Calculate centered moving average for each L-period
end procedure

procedure DECOMPOSETIMESERIES(X, L)
    CMA = CALCULATECMA(X, L)
    DistinguishTrendCyclical(X, CMA)                ▷ Step 2
    SeasonalFactors = CALCULATESEASONALFACTORS(CMA, L)
    DeseasonalizedData = SEASONALADJUSTMENT(X, SeasonalFactors)
    Trend = ANALYZEDESEASONALIZEDDATA(DeseasonalizedData)
    CyclicalComponent = CALCULATECYCLICALCOMPONENT(X, Trend)
    RandomError = COMPUTERANDOMERROR(X, Trend, CyclicalComponent, SeasonalFactors)
end procedure
```

6) **Set Threshold for Anomaly Detection:** Determine a threshold for Hotelling's T^2 statistic. Values exceeding this threshold indicate significant deviations from the expected latency patterns.

7) **Anomaly Detection:** Compare calculated Hotelling T^2 statistic with the threshold. When the number of statis-tic exceeds the threshold, it indicates a deviation or abnormal performance of the network latency at that moment.

8) **Visualization (Optional):** Visualize the Hotelling T^2 statistic (optionally) with time to see when the network latency is not behaving as expected.

VII. EXPERIMENTAL RESULTS

This experiment involved a time-based migration of data in which we migrated both data and virtual machines to the cloud. We employed the dstat tool for monitoring within the cloud environment. Dstat can fetch statistics from diverse system components, including network connections and IO devices, and analyze network traffic on dedicated lines. For our cloud configuration, we allocated dedicated bandwidth: Cloud 1 with 500 GB, Cloud 2 with 200 GB, and Cloud 3 with its allocated bandwidth, all adhering to their respective

Service Level Agreements (SLAs).

During data transfer process, we encountered issues with Cloud 3 due to its low bandwidth. We collected network traffic data from all clouds using the installed dstat tool. The data was analyzed through a monitoring system equipped with service integrators and duty service analysis tools, which monitored the dedicated network for latency. Disturb, installed on all clouds, gathered information, enabling the detection of issues using the "Hotelling's T^2 square" method.

During the data transfer phase, we faced challenges with Cloud 3 due to its limited bandwidth. We gathered network traffic data from all clouds using the dstat tool installed. This data underwent analysis via a monitoring system equipped with service integrators and duty service analysis tools, overseeing the dedicated network for latency. Disturb, deployed across all clouds, collected information, facilitating issue detection using Hotelling's T^2 statistic method.

We conducted experiments in our existing cloud environment, modifying the workload and transaction categories through JMeter. The application layer was deployed on a machine with a 'multi-core' processor. We recorded the average response time of transactions, along with the typical network and input/output consumption of the application servers executing the business logic components, on an hourly basis.

Fig 3 and Table II and Table V display the network and I/O utilization for a day, captured at 60-minute intervals.

Tables III and IV, along with Figures 4 and 5, present the network and I/O utilization errors over a week at identical intervals. The data clearly shows that higher I/O levels lead to evident network latency.

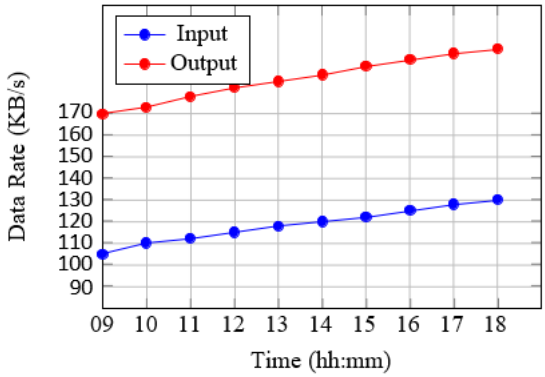

Fig. 3. Healthy I/O Data for 24 Hours

[**?**] focused on utilizing both unhealthy and healthy real-time datasets to monitor the behavior of virtual machines. Their analysis specifically incorporated metrics related

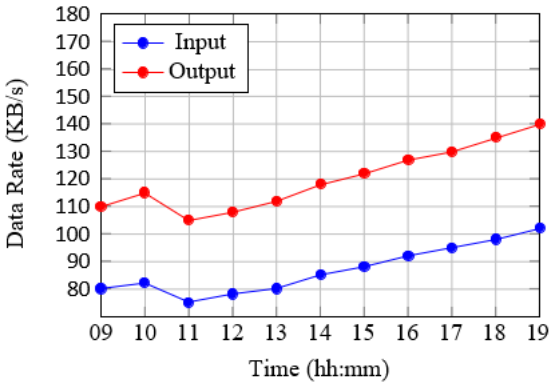

Fig. 4. Unhealthy I/O Data for 24 Hours (Due to Network Latency)

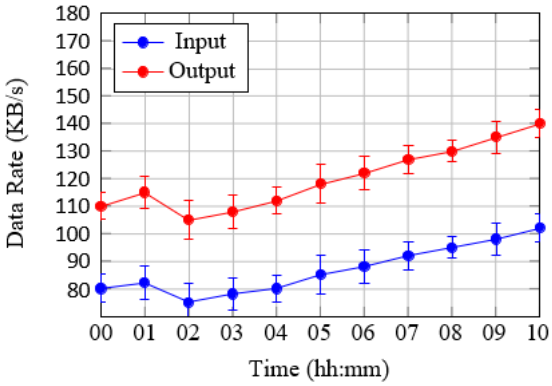

Fig. 5. Unhealthy I/O Data for 24 Hours with Error Bars (Due to Network Latency)

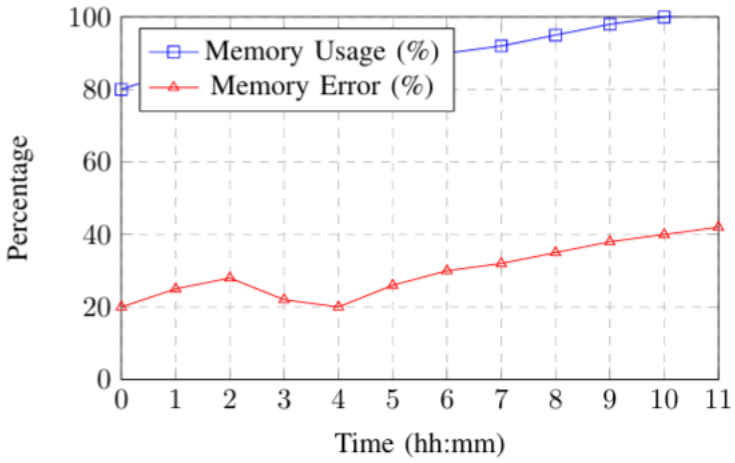

Fig. 6. Memory Usage and Memory Error over Time

TABLE II HEALTHY NETWORK DATA

Time (hh:mm)	Latency (ms)	Bandwidth Utilization (%)	Packet Loss Rate (%)	Uptime (%)
00:00	5	20	0.1	99.9
01:00	6	22	0.2	99.8
02:00	5	18	0.1	99.9
03:00	4	21	0.1	99.9
04:00	6	25	0.2	99.8
05:00	5	23	0.1	99.9
06:00	4	20	0.1	99.9
07:00	5	22	0.1	99.9
08:00	6	24	0.2	99.8
09:00	5	21	0.1	99.9
10:00	6	22	0.2	99.8
11:00	5	18	0.1	99.9
12:00	4	21	0.1	99.9
13:00	6	25	0.2	99.8
14:00	5	23	0.1	99.9
15:00	4	20	0.1	99.9
16:00	5	22	0.1	99.9
17:00	6	24	0.2	99.8
18:00	5	21	0.1	99.9
19:00	6	22	0.2	99.8
20:00	5	18	0.1	99.9
21:00	4	21	0.1	99.9
22:00	6	25	0.2	99.8
23:00	5	23	0.1	99.9

TABLE III UNHEALTHY NETWORK DATA FOR SELECTED HOURS

Time (hh:mm)	Latency (ms)	Bandwidth Utilization (%)	Packet Loss Rate (%)	Uptime (%)
09:00	200	80	5.2	95.0
10:00	250	85	6.5	92.3
11:00	180	75	4.8	96.2
12:00	300	90	7.2	89.6
13:00	220	82	6.0	94.0
14:00	280	88	6.8	91.7
15:00	210	79	5.6	93.4
16:00	270	87	6.6	90.1
17:00	230	81	5.9	92.1
18:00	260	86	6.4	89.8

TABLE IV UNHEALTHY I/O DATA FOR SELECTED HOURS (DUE TO NETWORK LATENCY)

Time (hh:mm)	Input (KB/s)	Output (KB/s)
09:00	98	135
10:00	102	140
11:00	105	144
12:00	108	148
13:00	110	150
14:00	112	152
15:00	115	155
16:00	118	158
17:00	120	160
18:00	122	162

TABLE V HEALTHY I/O DATA FOR 24 HOURS (DUE TO NETWORK LATENCY)

Time (hh:mm)	Input (KB/s)	Output (KB/s)
00:00	90	140
01:00	92	145
02:00	95	148
03:00	88	135
04:00	87	133
05:00	91	142
06:00	96	150
07:00	98	155
08:00	102	160
09:00	105	165
10:00	110	170
11:00	112	173
12:00	115	178
13:00	118	182
14:00	120	185
15:00	122	188
16:00	125	192
17:00	128	195
18:00	130	198
19:00	132	200
20:00	135	203
21:00	138	206
22:00	140	208
23:00	142	210

TABLE VIHEALTHY I/O DATA FOR 24 HOURS WITH ERROR (DUE TO NETWORK LATENCY)

Time (hh:mm)	Input (KB/s)	Input Error (KB/s)	Output (KB/s)	Output Error (KB/s)
09:00	105	6	165	6
10:00	110	5	170	5
11:00	112	6	173	6
12:00	115	5	178	5
13:00	118	6	182	6
14:00	120	5	185	5
15:00	122	6	188	6
16:00	125	5	192	5
17:00	128	6	195	6
18:00	130	5	198	5

to Memory and CPU. In contrast, our approach involves calculating multivariate data to assess network latency. Unlike the conventional use of a single variable for latency calculations, our method employs a multivariate approach to enhance the accuracy of results.

In Table VI and Table VII there is a clear evident that the CPU and Memory were in healthy state and as soon as the threshold value changes the unhealthy data started reporting as shown in Table VIII.

MYT decomposition involves breaking down time series data into three components:

- **Multiplicative Seasonal Effects:** Seasonal variations within shorter time intervals. The data is created using

a multiplicative model for seasonality and trend: The multiplicative model for seasonality can be represented as:

TABLE VII HEALTHY CPU AND MEMORY DATA FOR 24 HOURS

Time (hh:mm)	CPU Usage (%)	Memory Usage (%)
00:00	20	80
01:00	25	85
02:00	28	88
03:00	22	82
04:00	20	80
05:00	26	85
06:00	30	90
07:00	32	92
08:00	35	95
09:00	38	98
10:00	40	100
11:00	42	102
12:00	45	105
13:00	48	108
14:00	50	110
15:00	52	112
16:00	55	115
17:00	58	118
18:00	60	120
19:00	62	122
20:00	65	125
21:00	68	128
22:00	70	130
23:00	72	132

TABLE VIII UNHEALTHY MEMORY AND CPU DATA FOR 24 HOURS WITH ERROR

Time (hh:mm)	Memory Usage (%)	Memory Error (%)	CPU Usage (%)	CPU Error (%)
00:00	80	± 3	20	± 2
01:00	85	± 4	25	± 3
02:00	88	± 5	28	± 4
03:00	82	± 4	22	± 3
04:00	80	± 3	20	± 2
05:00	85	± 5	26	± 3
06:00	90	± 4	30	± 4
07:00	92	± 3	32	± 2
08:00	95	± 2	35	± 1
09:00	98	± 4	38	± 3
10:00	100	± 3	40	± 2
11:00	102	± 4	42	± 3
12:00	105	± 3	45	± 2
13:00	108	± 4	48	± 3
14:00	110	± 3	50	± 2
15:00	112	± 4	52	± 3
16:00	115	± 3	55	± 2
17:00	118	± 4	58	± 3
18:00	120	± 3	60	± 2
19:00	122	± 4	62	± 3
20:00	125	± 3	65	± 2
21:00	128	± 4	68	± 3
22:00	130	± 3	70	± 2
23:00	132	± 4	72	± 3

$$y(t) = A(t) \times (1 + S(t))$$

Where:

- y(t) observed value at time t.
- A(t) trend component at time t.
- S(t) seasonal effect at time t, with seasonal varia-tions that repeat over time.

- **Yearly Patterns:** Annual fluctuations.
- **Trend:** Long-term direction in the data. Trend Compo-nent: The trend component A(t) is modeled as an expo-nentially increasing function (e.g., 10*(1.05xˆ)), meaning the data increases over time.
- **Seasonal Effect**: S(t) is the seasonal effect, modeled using a sine wave (sin(2*pi*x/12)) to represent the yearly cycle. The trend is multiplied by the seasonal effect, so that the fluctuations become more pronounced in certain months (e.g., higher in winter or summer, depending on the business cycle).

The seasonal effect is modeled as a sine wave (sin(2*pi*x/12)) with an amplitude of 0.3, representing annual variations (positive and negative fluctuations). The sine function introduces cyclic fluctuations, repeating every year.

The graph shows how the value of the data varies over time due to seasonal effects (the sine wave), long-term trends (the exponential growth), and yearly patterns (repeating fluctua-tions).

The trend gradually increases, while the seasonal pattern fluctuates every year, reflecting a typical seasonal trend in many time series data sets like retail sales, weather patterns, etc.

MULTIPLICATIVE SEASONAL EFFECTS, YEARLY PATTERNS, AND TREND

IN THIS GRAPH FIG 7, WE WILL ILLUSTRATE A TIME SERIES WITH:

- MULTIPLICATIVE SEASONAL EFFECT: VARIATIONS THAT CHANGE ACROSS DIFFERENT SEASONS.
- YEARLY PATTERNS: REPEATING FLUCTUATIONS EVERY YEAR.
- TREND: A LONG-TERM DIRECTION (UPWARD OR DOWNWARD) IN THE DATA.

IX. SERVER RESPONSE TIME AND LATENCY BEFORE VS AFTER MIGRATION

IN FIG. 8 WE COMPARE THE SERVER RESPONSE TIME AND LATENCY before and after migrating to the cloud.

X. CONCLUSION

In this study, we have demonstrated how machine learning (ML) algorithms can be effectively applied to diagnose, forecast, and auto-recover from failures during cloud data migration, particularly focusing on both univariate and multivariate metrics. The complex nature of cloud migration, with its high variability and dependency on numerous system metrics, makes it essential to employ predictive and Multiplicative Seasonal Effects, Yearly Patterns, and Trend and reliable migration of large-scale data operations.

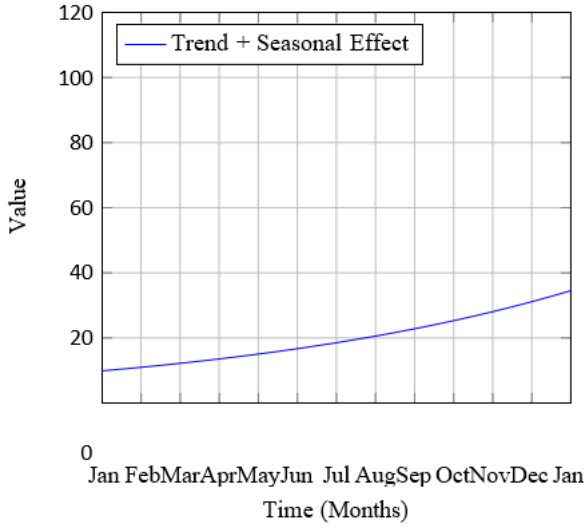

Fig. 7. Time Series with Trend, Seasonal Effects, and Yearly Patterns

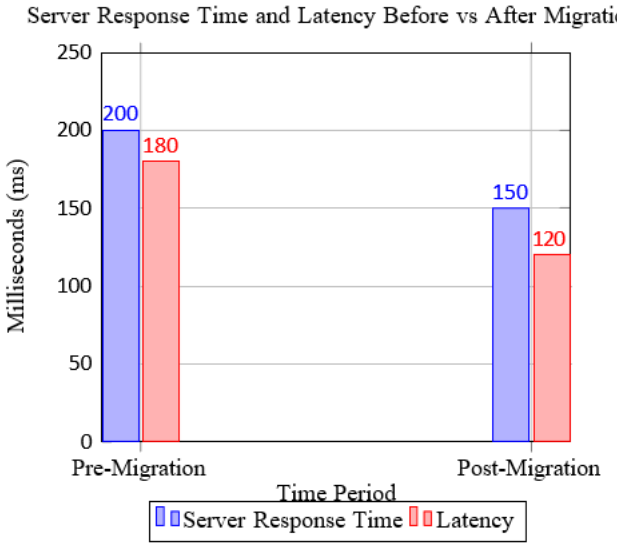

Fig. 8. Comparison of Server Response Time and Latency Before and After Cloud Migration

adaptive strategies to maintain the stability and success of the migration process.

Ultimately, this research highlights the importance of machine learning in modernizing and streamlining cloud data migration, making it more efficient, resilient, and cost-effective. Future work could further enhance these methods by incorporating additional data sources, improving model interpretability, and integrating feedback loops for continuous improvement in failure prediction and recovery strategies. As cloud adoption continues to grow, these predictive and adaptive techniques will be pivotal in ensuring the smooth

- Competing Interests - Network algorithm and security practises in cloud, migration

- Funding Information -Not Applicable

- Author contribution - Author has referred to the existing algorithm and reused it.

- Data Availability Statement -It is real time data used for the validation of the results.

- Research Involving Human and /or Animals -Not Applicable

- Informed Consent -Yes

References

1. T.W. Anderson. Introduction to Multivariate Statistical Analysis. WileyInterscience, 2003.
2. Jun Chen, Lei Zhao, and Hongwei Zhang. Applications of multivariate statistical methods in data migration monitoring. Journal of Data Science, 13(2):215–229, 2015.
3. R. Gnanadesikan. Methods for Statistical Data Analysis of Multivariate Observations. John Wiley & Sons, 1997.
4. J. Edward Jackson. A User's Guide to Principal Components. John Wiley & Sons, 1991.
5. Richard Arnold Johnson and Dean W Wichern. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, 2007.
6. Ali Khajeh-Hosseini, David Greenwood, and Ian Sommerville. Cloud computing: A survey. The University of St Andrews, St Andrews, UK, Tech. Rep., 2010.
7. Hyunsoo Lee and Bong-Keun Yoo. Combined application of hotelling's t 2 and myt decomposition for cloud data migration. Journal of Cloud Computing, 7(1):1–14, 2018.
8. Robert L Mason and John C Young. Multivariate Statistical Process Control with Industrial Applications. SIAM, 2002.
9. Douglas C Montgomery. Introduction to Statistical Quality Control. John Wiley & Sons, 2008.
10. Jin Park and Soo-Young Kim. Integration of myt decomposition and hotelling's t 2 method for effective cloud data migration diagnostics. International Journal of Data Science, 5(2):99–115, 2019.
11. M C Ramya, Sumit Kumar Bose, Michael Salsburg, Venkat Shivaram, and Shrisha Rao. Detecting and diagnosing application misbehaviors in 'on-demand' virtual computing infrastructures. China International Electrical and Energy Conference (CIEEC), pages 198–203, 2011.
12. Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A survey and analysis of infrastructure as a service (iaas) in cloud computing. Journal of Systems and Software, 82(11):1889–1899, 2009.
13. Wei Zhang, Min Li, and Jun Chen. Application of hotelling's t 2 control chart for cloud data migration monitoring. Computers & Industrial Engineering, 106:405–413, 2017.