Volume 11, Issue 5, May – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

A COREM-Based Framework for Cross-Provider Cloud Outage Analysis and Resilience Evaluation

Muvva Venkatesh[1]; Mohammed Taha[2]; Dr. K. Rajitha[3*]; Dr. K. Sreekala[4]

(23261A05A3)[1]; (23261A05A2)[2]
[1,2]Under Graduate Student; [3]Professor; [4]Assisstant Professor

[1;2;3;4]Department of Computer Science & Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India

Corresponding Author: Dr. K. Rajitha[3*]

Publication Date: 2026/05/30

Abstract: Cloud computing platforms such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare provide critical infrastructure for modern applications. Despite high availability guarantees, service outages continue to occur due to various factors including infrastructure failures, network disruptions, and configuration errors. These incidents significantly impact service reliability and user experience. This paper proposes a multi-cloud incident analysis framework based on a normalized dataset constructed using a standardized 14-column reliability schema. The framework introduces a COREM (Cloud Outage Risk Evaluation Model) weighted scoring algorithm to quantify and compare the risk associated with incidents across different cloud providers. The model evaluates incidents based on severity, duration, impact, and frequency. The system enables cross-provider comparative analysis and visualization of outage trends, helping to identify reliability patterns and high-risk services. The proposed approach improves transparency in cloud reliability assessment and supports better decision-making for multi-cloud deployment strategies.

Keywords: *Cloud Computing, Incident Analysis, Reliability Assessment, Multi-Cloud, COREM Model, Risk Scoring, Outage Analysis Calibration Explainable Artificial Intelligence.*

How to Cite: Muvva Venkatesh; Mohammed Taha; Dr. K. Rajitha; Dr. K. Sreekala (2026) A COREM-Based Framework for Cross-Provider Cloud Outage Analysis and Resilience Evaluation. *International Journal of Innovative Science and Research Technology*, 11(5), 2391-2401. https://doi.org/10.38124/ijisrt/26may1147

I. INTRODUCTION

Cloud computing has become a fundamental component of modern digital systems, supporting applications across industries such as finance, healthcare, and e-commerce. Organizations increasingly adopt multi-cloud strategies to improve availability, scalability, and fault tolerance. However, despite these advantages, cloud platforms continue to experience service outages that can disrupt operations and cause significant losses.

Cloud incidents may arise due to hardware failures, software bugs, network congestion, or misconfigurations. Each cloud provider maintains its own incident reporting format, making it difficult to perform direct comparisons across platforms. This lack of standardization limits the ability to analyze reliability trends effectively.

Traditional approaches to incident analysis often focus on individual providers and do not provide a unified framework for cross-provider comparison. Furthermore, most analyses do not quantify risk in a structured manner, making it challenging to evaluate the overall reliability of cloud services. To address these challenges, this project proposes a Multi-Cloud Incident Analysis Framework that:

- Normalizes incident data across multiple providers
- Uses a standardized reliability schema
- Applies a weighted risk scoring model (COREM)
- Enables comparative reliability analysis

The proposed system aims to provide a comprehensive and data-driven approach to evaluating cloud reliability.

Another major challenge in cloud computing environments is the lack of standardized incident reporting and reliability evaluation across different providers. Each cloud platform follows its own format for reporting outages, making it difficult to perform consistent cross-provider

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

comparisons. Additionally, incident data often varies in terms of detail, terminology, and severity classification, which can lead to inconsistencies in analysis. This lack of uniformity limits the ability to accurately assess overall system reliability and identify common failure patterns.

Recent advancements in data analytics and cloud monitoring techniques provide effective solutions to these challenges. By applying structured data processing and normalization methods, incident data from multiple providers can be transformed into a unified format. This enables meaningful comparison and deeper analysis of outage characteristics such as duration, impact, and frequency. Furthermore, analytical models allow for better interpretation of incident trends and service reliability.

In addition to data standardization, modern cloud reliability analysis requires a quantitative approach to evaluate and compare risks across providers. Simple observation of incident logs is insufficient for understanding the true impact of outages. A systematic risk evaluation mechanism is necessary to incorporate multiple factors such as severity, duration, and frequency into a single measurable metric.

This paper proposes a Multi-Cloud Incident Analysis System that integrates data normalization techniques with a weighted risk evaluation model known as COREM (Cloud Outage Risk Evaluation Model). The system processes incident data from multiple cloud providers, standardizes it using a predefined schema, and computes risk scores based on key incident attributes. To evaluate the effectiveness of the approach, comparative analysis is performed across providers using normalized datasets.

Additionally, the system introduces a structured risk scoring mechanism that ensures consistent evaluation of incidents under varying conditions. This enables identification of high-risk services and patterns of recurring failures. The framework also supports visualization of reliability trends, improving interpretability and aiding decision- making.

By combining data normalization, weighted risk scoring, and comparative analysis, the proposed system aims to provide a reliable, transparent, and scalable approach for multi-cloud reliability assessment and outage analysis.

II. LITERATURE SURVEY

Unyi et al. (2025) proposed an explainable Graph Neural Network (GNN)-based approach for fault forecasting in cloud service debugging. Their work utilized service dependency graphs along with explainable AI techniques to improve fault prediction accuracy. The study demonstrated that incorporating explainability helps in understanding failure patterns more effectively. However, the approach relies heavily on internal cloud logs, which limits its applicability in public multi-cloud analysis scenarios. [1]

Cinque et al. (2019) introduced resilience evaluation metrics for cloud services. Their work quantified service robustness using formal metrics and benchmarking techniques. The study provided a structured approach to measuring resilience. However, the evaluation is largely abstract and lacks direct correlation with real-world incident data. [2]

Zhang et al. (2018) analyzed failures in largescale cloud systems through root cause analysis. Their work identified cascading failures as a major issue in distributed cloud environments. The study provided detailed insights into failure propagation mechanisms. However, the use of proprietary production logs restricts reproducibility and broader applicability. [3]

Gupta et al. (2017) performed an empirical study of cloud outages using statistical analysis of public incident reports. The study highlighted that network and power failures are among the most dominant causes of outages. Although the research provides real-world insights, it is limited by the number of providers included in the analysis. [4]

Baset et al. (2016) analyzed cloud Service Level Agreements (SLAs) and their relation to reliability during outages. Their research showed that SLA violations often occur during major incidents, highlighting the importance of reliability guarantees. While the study is relevant from a business perspective, it lacks transparency due to limited access to detailed provider data. [5]

Dean (2014) discussed system design principles for building resilient cloud infrastructure. The study emphasized redundancy, fault tolerance, and distributed system design to improve reliability. While the work provides valuable practical insights, it does not include a quantitative model for evaluating incident risks. [6]

Bailis et al. (2014) explored coordination avoidance techniques in distributed systems to reduce failure propagation. Their theoretical approach demonstrated how minimizing coordination can improve system scalability and reliability. However, the study is not specifically focused on outage analysis or incident data evaluation. [7]

Birke et al. (2014) proposed predictive modeling techniques for failure prediction in cloud systems. Their research demonstrated that early fault signals can be detected using monitoring data, enabling proactive failure management. While effective, the approach involves complex machine learning models that may be difficult to implement at scale. [8]

Kandula et al. (2009) conducted a foundational study on characterizing cloud failures using failure log analysis. Their research identified common causes of failures in data center environments, providing early insights into reliability challenges in cloud systems. While the study is highly influential, the dataset used is outdated and may not reflect modern cloud architectures. [9]

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

Fox et al. (2009) presented a comprehensive architectural view of cloud computing and identified key risks associated with cloud services. The study laid the foundation for understanding cloud reliability challenges. However, it is largely conceptual and does not include empirical analysis of incidents. [10]

III. DESIGN OF THE SYSTEM

- *Data Collection Interface:*

The system provides an interface for collecting incident data from multiple cloud service providers such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare. This module enables the aggregation of incident reports, including details such as timestamps, affected services, and incident descriptions.

- *Data Preprocessing Module:*

The collected incident data is processed to remove inconsistencies, missing values, and redundant information. This module ensures that raw data from different providers is cleaned and formatted for further analysis.

- *Data Normalization Module:*

A normalization module converts incident data from different providers into a standardized 14-column schema. This ensures uniformity in attributes such as severity, duration, impact, and region, enabling meaningful cross-provider comparison.

- *COREM Scoring Module:*

The system implements the COREM (Cloud Outage Risk Evaluation Model) to calculate a risk score for each incident. The model evaluates incidents based on weighted factors such as severity, duration, impact, and frequency to produce a unified risk metric.

- *Analysis and Visualization Module:*

This module analyzes the computed risk scores and generates visual representations such as graphs and charts. It helps in identifying trends, comparing provider reliability, and highlighting high-risk services.

- *Evaluation Module:*

The system evaluates reliability by comparing risk scores across providers and analyzing incident patterns over time. This module ensures consistency and supports data-driven insights for multi-cloud decision-making.

IV. BLOCK DIAGRAM

The block diagram of the proposed Multi-Cloud Incident Analysis System illustrates the overall workflow of the system from data collection to reliability assessment and reporting. The process begins when the user or analyst initiates data collection from public cloud data sources such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare. Incident data is gathered from sources including status pages, RSS feeds, and official outage reports.

The collected data then undergoes preprocessing in the data processing layer, where it is cleaned, filtered, and structured to remove inconsistencies and missing values. This stage ensures that raw incident data from different providers is transformed into a consistent and usable format. The processed data is then passed to the normalization module, where it is converted into a standardized 14-column schema to enable uniform comparison across multiple cloud platforms.

Next, the normalized dataset is fed into the COREM engine, which performs risk scoring and classification. The COREM (Cloud Outage Risk Evaluation Model) calculates a risk score for each incident based on key factors such as severity, duration, impact, and frequency. These factors are assigned appropriate weights, allowing the system to generate a unified and quantitative measure of incident risk.

The analyzed data is then forwarded to the analysis and visualization module, where patterns, trends, and provider-wise comparisons are generated using graphical representations. This stage helps in identifying high-risk services, recurring failure patterns, and reliability differences across cloud providers.

Finally, the system produces results and reports, including risk levels, comparative insights, and actionable recommendations. These outputs assist users in understanding cloud reliability and making informed decisions regarding multi-cloud deployment and risk management strategies.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

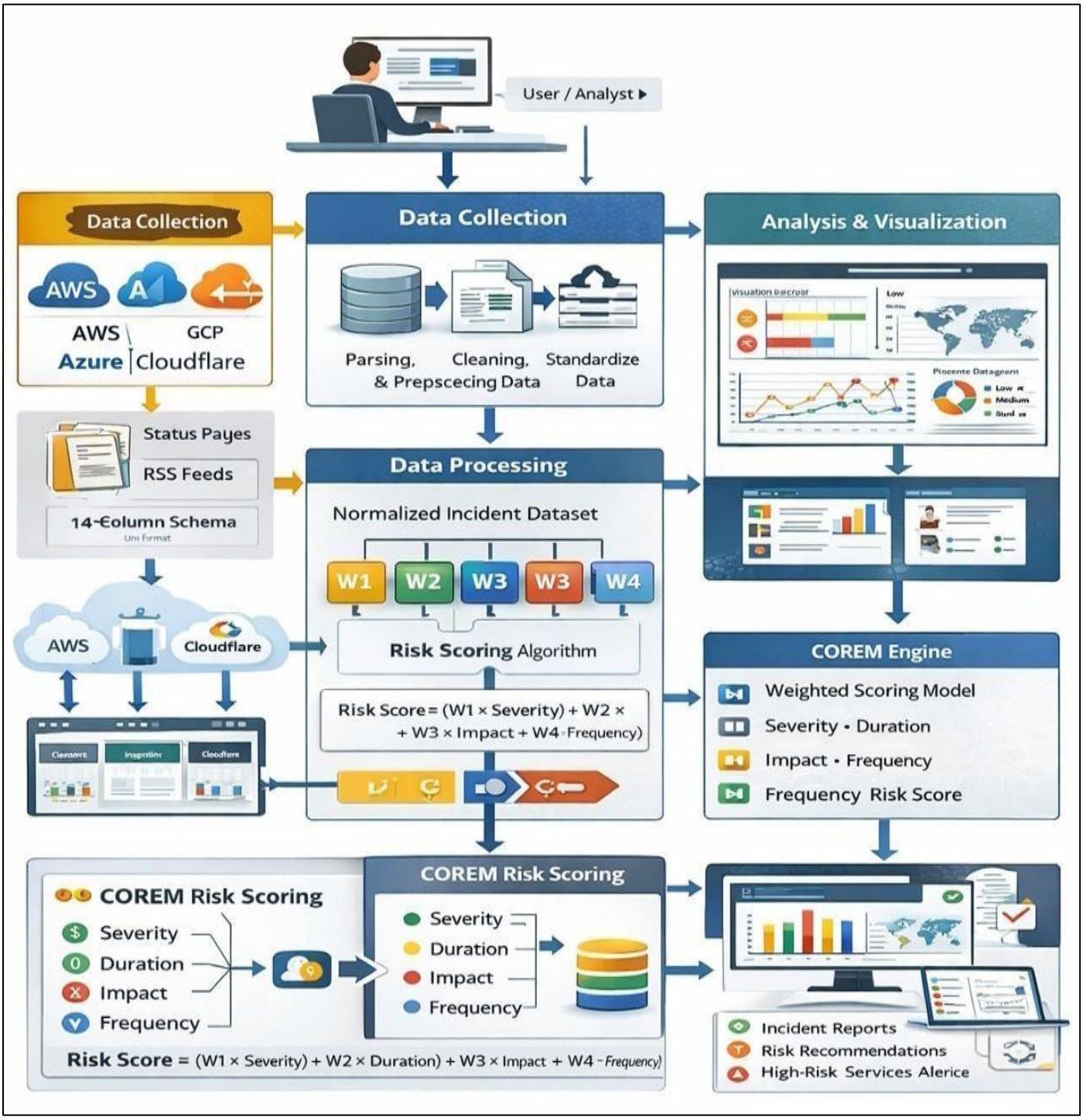

Fig 1 Block Diagram (Architecture) of the Multi-Cloud Incident Analysis System Using COREM

V. FLOW DIAGRAM OF MULTI-CLOUD INCIDENT ANALYSIS SYSTEM

The flow chart illustrates the working process of the Multi-Cloud Incident Analysis System. The process begins when the user or analyst collects incident data from public cloud providers such as AWS, Azure, GCP, and Cloudflare. The collected data first undergoes preprocessing, including cleaning and structuring.

Next, the system performs data normalization by converting the processed data into a standardized format for consistent analysis. The normalized data is then passed to the pattern analysis module, where failure trends and root causes are identified. These features are further analyzed using the COREM model, which evaluates incidents based on factors such as severity, duration, impact, and frequency to generate risk scores and classifications. Finally, the system outputs incident reports along with reliability insights and recommendations, supporting effective multi-cloud decision-making.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

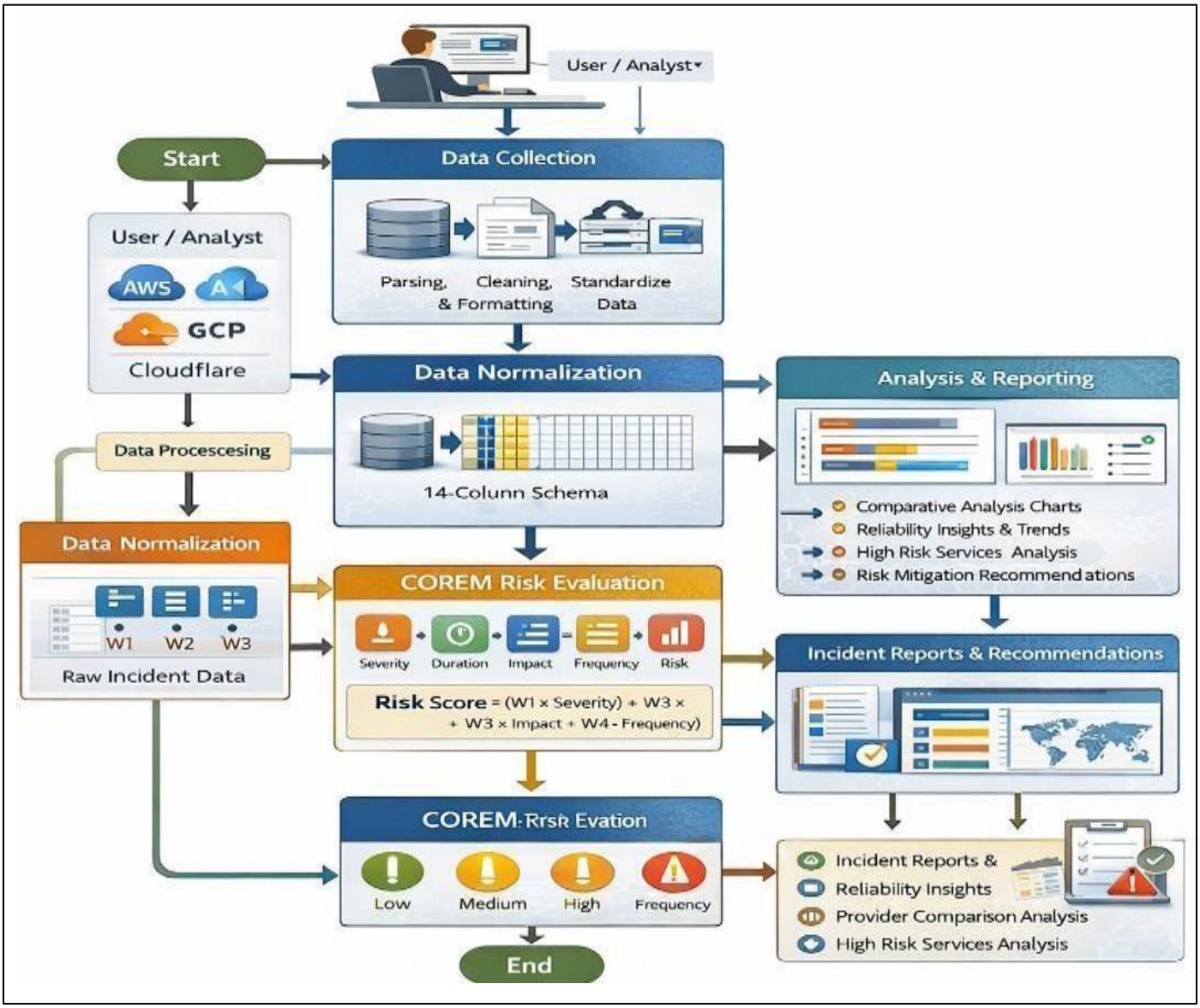

Fig 2 Workflow of Multi-Cloud Incident Analysis and Risk Classification System Schema Framework

VI. METHODOLOGY

The proposed system integrates data preprocessing, normalization techniques, and a weighted risk evaluation model (COREM) to perform multi-cloud incident analysis and reliability assessment. The methodology is divided into several stages to ensure accurate and consistent evaluation of cloud service incidents.

➢ *Data Collection*

The system begins with the collection of incident data from public cloud providers such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare. The data is gathered from sources such as status pages, RSS feeds, and official outage reports. This collected data forms the input for further processing and analysis.

➢ *Data Preprocessing*

The collected incident data undergoes preprocessing to remove inconsistencies, duplicate entries, and missing values. This stage includes data cleaning, formatting, and structuring to convert raw incident reports into a usable format for analysis.

➢ *Data Normalization*

To ensure uniformity across different cloud providers, the system converts the processed data into a standardized 14-column schema. This normalization step enables consistent representation of attributes such as severity, duration, impact, and region, allowing meaningful cross-provider comparison.

➢ *Feature Extraction*

After normalization, key features such as incident severity, duration, impact level, frequency, and affected services are extracted from the dataset. These features are essential for analyzing incident patterns and evaluating system reliability.

➢ *Risk Evaluation Using COREM Model*

The extracted features are passed to the COREM (Cloud Outage Risk Evaluation Model), which computes a weighted risk score for each incident. The model evaluates incidents

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

based on factors such as severity, duration, impact, and frequency, enabling a quantitative assessment of risk.

➢ *Risk Classification and Analysis*

Based on the computed risk scores, incidents are classified into different categories such as low, medium, and high risk. The system also performs pattern analysis to identify recurring failures and service dependencies across cloud providers.

➢ *Result Generation and Visualization*

Finally, the system generates incident reports, comparative analysis, and visual representations of reliability trends. These results include risk levels, high-risk service identification, and recommendations, helping users make informed decisions regarding multi-cloud deployment and reliability improvement.

VII. IMPLEMENTATION

➢ *Data Collection Interface*

The system is implemented using Google Colab, where incident data is uploaded in the form of structured datasets (Excel files). The interface allows users to input multi-cloud incident data collected from providers such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare.

➢ *Data Preprocessing*

Once the dataset is loaded, preprocessing is performed to clean and standardize the data. This includes handling missing values, removing inconsistencies, formatting text fields, and converting numerical attributes such as incident duration into a consistent format. Outlier handling is also applied to ensure stable analysis.

➢ *Data Normalization*

The processed data is converted into a standardized 14-column schema to ensure uniform representation across all cloud providers. This step enables consistent comparison of attributes such as severity, duration, blast radius, and root cause.

➢ *Feature Engineering*

Key features are extracted and transformed into numerical representations. This includes mapping categorical values such as severity and blast radius into scores, converting binary attributes (e.g., cascading failure, retry storm) into numerical form, and applying logarithmic scaling to incident duration for normalization.

➢ *COREM Risk Scoring Model*

The COREM (Cloud Outage Risk Evaluation Model) is implemented as a weighted scoring function. The model calculates a risk score for each incident using factors such as duration, severity, blast radius, root cause, and cascading effects. The scoring incorporates calibrated weights and logarithmic scaling to ensure realistic and balanced risk evaluation.

➢ *Pattern Analysis*

The system performs descriptive statistical analysis to identify patterns in the dataset. This includes analyzing average outage duration by root cause, cascading failure rates across providers, and failover success rates. These insights help in understanding reliability trends and failure behavior.

➢ *Visualization Dashboard*

A comprehensive multi-panel dashboard is generated using Matplotlib and Seaborn. The dashboard includes visualizations such as provider- wise outage duration, cascading failure rates, retry storm distribution, root cause heatmaps, MTTR trends, and risk level distribution.

➢ *Provider Resilience Ranking*

The system computes a resilience score for each cloud provider based on COREM scores and incident characteristics. Providers are ranked according to their ability to handle outages, enabling comparative reliability assessment.

➢ *Interactive Risk Evaluation*

An interactive evaluation module allows users to input custom incident parameters such as duration, severity, and root cause. The system calculates the corresponding COREM risk score and risk level in real time.

➢ *Recommendation Generation*

Based on observed patterns in the dataset, the system generates actionable recommendations for mitigating incidents. These recommendations vary depending on factors such as root cause, severity, and provider characteristics.

➢ *Result Generation and Reporting*

Finally, the system outputs structured results including risk scores, classifications, visual insights, and recommendations. These outputs support data-driven decision-making for improving cloud reliability and multi-cloud deployment strategies.

VIII. KEY FUNCTIONALITIES

➢ *Multi-Cloud Data Collection*

The system enables users to collect and upload incident data from multiple cloud providers such as AWS, Microsoft Azure, Google Cloud Platform (GCP), and Cloudflare. The data is obtained from status pages, outage reports, and structured datasets for further analysis.

➢ *Data Preprocessing*

The system performs data cleaning, formatting, and structuring to remove inconsistencies, missing values, and redundant entries. It also includes outlier handling to ensure stability and reliability in analysis.

➢ *Data Normalization*

A normalization module converts incident data into a standardized 14-column schema, ensuring uniform representation of attributes across providers and enabling accurate cross-provider comparison.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

➢ *Feature Engineering*

The system extracts and transforms key features such as severity, duration, blast radius, root cause, and failure indicators. Categorical and binary attributes are converted into numerical representations, and logarithmic scaling is applied to normalize duration values.

➢ *COREM-Based Risk Scoring*

The COREM model evaluates each incident by computing a weighted risk score based on multiple factors. The scoring mechanism incorporates calibrated weights and scaling techniques to provide a realistic and quantitative measure of risk.

➢ *Risk Classification*

Based on computed scores, incidents are classified into categories such as low, moderate, high, and critical risk levels, allowing quick identification of severe incidents.

➢ *Pattern Analysis*

The system performs descriptive statistical analysis to identify trends such as root cause impact, cascading failure rates, failover success rates, and outage duration patterns across providers.

➢ *Visualization Dashboard*

A comprehensive multi-panel dashboard is generated using graphical visualizations such as bar charts, heatmaps, and trend lines. This helps in understanding reliability patterns and comparing cloud providers effectively.

➢ *Provider Resilience Ranking*

The system calculates a resilience score for each cloud provider based on COREM scores and incident characteristics. Providers are ranked to reflect their ability to handle outages efficiently.

➢ *Interactive Risk Evaluation*

An interactive module allows users to input custom incident parameters and obtain real-time COREM risk scores and classifications, enabling scenario-based analysis.

➢ *Recommendation Generation*

The system generates actionable recommendations based on observed patterns in the dataset. These recommendations help in mitigating risks and improving cloud service reliability.

➢ *Reporting and Insights*

The final output includes structured reports, risk insights, and analytical summaries that support informed decision-making in multi-cloud environments.

IX. SOFTWARE SYSTEM CONFIGURATION

➢ *Development Environment*

- Programming Language: Python 3.8 or newer
- Platform: Google Colab / Jupyter Notebook
- IDE / Text Editor: Visual Studio Code, PyCharm (optional)
- Operating System: Windows / Linux / macOS Virtual Environment: Recommended for dependency management

➢ *Required Python Libraries the Proposed System uses the Following Python Libraries for Data Processing, Analysis, and Visualization:*

- *Pandas:* Used for data loading, cleaning, and manipulation of multi-cloud incident datasets
- *NumPy:* Provides numerical computation support, including logarithmic scaling and statistical calculations
- Matplotlib: Used for generating visualizations such as bar charts, line graphs, and analytical dashboards
- Seaborn: Used for advanced statistical visualizations such as heatmaps and distribution analysis
- Scikit-learn: Used for normalization techniques such as MinMax scaling and analytical support
- Google Colab Utilities (google.colab): Used for file upload and execution in a cloud-based environment
- Warnings Module: Used for handling runtime warnings and ensuring clean execution

➢ *System Requirements*

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB recommended
- Storage: At least 10 GB available Internet Connection: Required for dataset upload and cloud-based execution

X. RESULTS AND DISCUSSION

The proposed Multi-Cloud Incident Analysis System was evaluated to examine its effectiveness in incident risk assessment, reliability comparison, and pattern identification across cloud providers.

Using structured data processing and the COREM model, the system analyzed key features such as incident duration, severity, blast radius, and root cause to compute risk scores and classifications. The analysis revealed significant variations in outage characteristics across different cloud providers. Incidents involving cascading failures and retry storms were observed to have higher durations and impact levels, indicating their strong influence on overall system reliability. Root cause analysis further showed that software defects and configuration errors contributed significantly to high-risk incidents.

The system's effectiveness was demonstrated through the COREM risk scoring model, which provided a consistent and quantitative evaluation of incidents. The classification of incidents into low, moderate, high, and critical risk levels enabled clear identification of critical events. Additionally, the provider resilience ranking offered comparative insights into the outage handling capabilities of different cloud platforms.

The visualization dashboard provided clear representation of trends such as average outage duration,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

failure rates, and root cause impact, improving interpretability of results. The interactive evaluation module further demonstrated the system's ability to assess hypothetical scenarios and generate real-time risk scores and recommendations. Overall, the results indicate that the proposed framework provides a reliable and scalable approach for multi-cloud incident analysis, enabling improved understanding of cloud reliability patterns and supporting data-driven decision-making.

Table 1 Comparison of Proposed Multi-Cloud COREM-Based System with Existing Cloud Monitoring and Analysis Approach

Features	Proposed COREM-Based System	Traditional Monitoring Tools	Basic Incident Analysis	Cloud Provider Dashboards
Data Integration	Multi-cloud unified dataset (AWS, Azure, GCP, Cloudflare)	Single-provider focus	Limited dataset scope	Provider-specific only
Data Normalization	Standardized 14-column schema	No standardization	Partial standardization	No cross-provider normalization
Risk Evaluation	Weighted COREM risk scoring model	Threshold-based alerts	Basic severity tagging	Limited severity indicators
Duration Handling	Logarithmic scaling for accurate evaluation	Raw duration values	Basic averaging	Raw metrics only
Root Cause Analysis	Integrated root cause scoring	Limited analysis	Manual identification	Descriptive only
Pattern Analysis	Statistical trend and failure pattern detection	Minimal trend analysis	Limited analysis	Basic visual trends
Visualization	Multi-panel analytical dashboard	Basic charts	Minimal visualization	Standard dashboards
Risk Classification	Multi-level (Low, Moderate, High, Critical)	Alert-based classification	Simple categorization	Limited classification
Provider Comparison	Cross-provider reliability comparison	Not supported	Not supported	Not supported
Resilience Ranking	Provider-level resilience scoring	Not available	Not available	Not available
Interactive Evaluation	Real-time incident risk evaluation	Not available	Not available	Not available
Recommendation System	Actionable mitigation recommendations	Limited alerts	Manual analysis	No recommendations
Transparency	Interpretable scoring model	Partial transparency	Low transparency	Black-box metrics
Scalability	Supports multi-cloud environments	Limited to single cloud	Limited scope	Provider-bound

➢ *Testing*

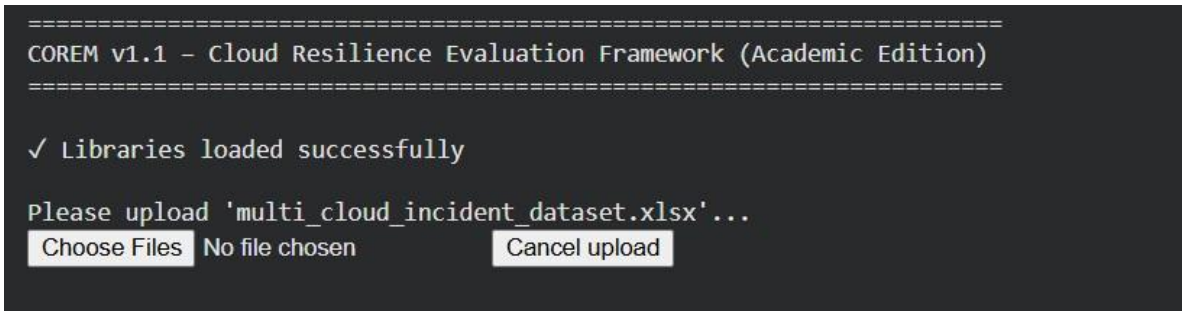

Fig 3 Initial Interface of COREM System

The Fig 3 shows the initial interface of the COREM system implemented in a Python-based environment. It prompts the user to upload the dataset file (multi_cloud_incident_dataset.xlsx) andconfirms successful loading of required libraries. This interface ensures proper initialization of the system before data processing begins. It provides a simple and controlled entry point for executing the analysis pipeline.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

```
✓ Total incidents analyzed: 109
✓ Providers covered: AWS, Azure, Cloudflare, GCP
✓ Root cause categories: 15

✓ Data cleaning completed
✓ Date range: 2022-06-21 00:00:00 to 2026-02-10 00:00:00

✓ COREM scoring applied to all incidents

[                ] Pattern Analysis (Descriptive Statistics)

Top root causes by average duration:
                                mean  count
root_cause_category
Cooling_Infrastructure_Failure  1452.0      1
Hardware_Failure                 870.0      2
External_Network_Failure         667.0      3
Deployment_Process_Gap           414.9      7
Power_Infrastructure_Failure     396.0      4

Cascading failure rates by provider:
          cascading_rate  incident_count
provider
Azure              1.000             25
AWS                0.618             34
Cloudflare         0.520             25
GCP                0.520             25

Auto-failover success rates by root cause (lowest performing):
                  mean  count
root_cause_category
Network_Failure   0.00      5
Software_Defect   0.22     41
Hardware_Failure  0.25      2

✓ Pattern analysis completed (all statistics computed from dataset)
```

Fig 4 Data Upload and Preprocessing Status

The Fig 4 displays the output after dataset upload and preprocessing. It shows key dataset statistics such as total incidents analyzed, number of providers covered, and root cause categories identified. The system performs data cleaning, normalization, and conversion of attributes such as duration, severity, and blast radius into numerical formats. It also converts date values and handles missing or inconsistent data. This step ensures that the dataset is structured and ready for accurate risk.

Fig 5 COREM Risk Scoring Dashboard

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

➢ *The Fig 5 Presents the Comprehensive Visualization Dashboard Generated by the System. It Includes Multiple Analytical Components Such as:*

- Average outage duration across providers
- Cascading failure rates
- Retry storm rates
- Root cause impact heatmap
- MTTR (Mean Time to Recovery) trend analysis
- Auto-failover success rates
- COREM risk level distribution

```
================================================================
 COREM INTERACTIVE INCIDENT EVALUATOR
================================================================

Provider (AWS/Azure/GCP/Cloudflare): AWS
Duration (minutes): 30
Severity (Low/Medium/High/Critical): low
Blast Radius (Regional/Multi-region/Global): regional
Cascading failure? (Yes/No): yes
Retry storm? (Yes/No): yes

Common root causes in dataset:
  1. BGP Routing Error
  2. Capacity Saturation
  3. Configuration Error
  4. Deployment Process Gap
  5. Internal Network Configuration Error
  6. Network Failure
  7. Power Infrastructure Failure
  8. Software Defect
Select root cause (name or number): 7

================================================================
COREM INCIDENT ASSESSMENT REPORT
================================================================
Provider          : AWS
Duration          : 30 minutes
Root Cause        : Power Infrastructure Failure
Severity          : Low
Blast Radius      : Regional
Cascading Failure : Yes
Retry Storm       : Yes
----------------------------------------------------------------
COREM RISK SCORE  : 0.479
RISK LEVEL        : MODERATE
----------------------------------------------------------------
RECOMMENDED ACTIONS:
MONITOR: Increase observability; prepare mitigation runbook
  Mitigation strategies:
    1. Mandate multi-AZ deployment for critical workloads
    2. Verify failover procedures are tested quarterly
  AWS consideration: Validate Route53 health checks; test multi-region failover quarterly
================================================================


================================================================
METHODOLOGICAL LIMITATIONS
================================================================
• Dataset limited to publicly available postmortems (selection bias)
• Small sample sizes for some providers/root causes (n < 5)
• Statistical tests are exploratory; no correction for multiple comparisons
• COREM scores calibrated to this dataset; may require recalibration for
  other cloud environments or time periods
• Recommendations derived from observed patterns, not causal inference
================================================================
```

Fig 6 Interactive Risk Evaluator Output

The Fig 6 illustrates the output of the interactive COREM incident evaluator. Users can input custom parameters such as provider, duration, severity, blast radius, cascading failure, retry storm, and root cause. The system computes a COREM risk score and classifies the incident into risk levels such as Low, Moderate, High, or Critical. It also generates evidence-based recommendations for mitigation strategies. This feature enables scenario-based analysis and supports proactive decision-making in cloud environments.

REFERENCES

[1]. Unyi, A., et al. “Explainable Graph Neural Network-Based Fault Forecasting for Cloud Service Debugging.” 2025 International Conference on Cloud Computing and Artificial Intelligence, 2025.

[2]. Cinque, M., et al. “Resilience Evaluation Metrics for Cloud Services.” Future Generation Computer Systems, vol. 95, 2019, pp. 964–977.

[3]. Zhang, Y., et al. “Root Cause Analysis of Failures in Large-Scale Cloud Systems.” Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2018.

[4]. Gupta, A., et al. “An Empirical Study of Cloud Outages and Their Causes.” IEEE International Conference on Cloud Engineering (IC2E), 2017.

[5]. Baset, S. A., et al. “Towards Understanding Cloud Service Level Agreements and Reliability.” IEEE International Conference on Cloud Computing, 2016.

[6]. Dean, J. “Designs, Lessons and Advice from Building Large Distributed Systems.” Keynote Presentation, ACM Symposium on Operating Systems Principles (SOSP), 2014.

[7]. Bailis, P., et al. “Coordination Avoidance in Database Systems.” Proceedings of the VLDB Endowment, vol. 8, no. 3, 2014, pp. 185–196.

[8]. Birke, R., et al. “Predictive Modeling for Failure Prediction in Cloud Systems.” IEEE Transactions on Cloud Computing, vol. 2, no. 3, 2014, pp. 290–303.

[9]. Kandula, S., et al. “Detailed Analysis of Data Center Failures in Cloud Computing Environments.” Proceedings of the ACM SIGCOMM Conference, 2009.

[10]. Fox, A., et al. “Above the Clouds: A Berkeley View of Cloud Computing.” University of California, Berkeley Technical Report, 2009.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1147

APPENDIX A: DATASET AND SOURCE CODE

➢ *Repository*

The normalized multi-cloud incident dataset used in this project, along with the implementation code for the COREM-based analysis framework, is publicly available in the GitHub repository below. The repository contains the processed datasets, preprocessing scripts, COREM risk scoring implementation, visualization modules, and evaluation outputs used in this study.

➢ *GitHub Repository:*

- https:/github.com/mohammed1819/corem-cloud-dataset.git