Resilience Assessment of Cloud Video Transcoding Services

Priscila Silva
psilva4@umassd.edu
University of Massachusetts Dartmouth

Karen da Mata
kalvesdamata@umassd.edu
University of Massachusetts Dartmouth

Fatemeh Salboukh
fsalboukh@umassd.edu
University of Massachusetts Dartmouth

Jamilson Dantas
jrd@cin.ufpe.br
Federal University of Pernambuco

Lance Fiondella
lfiondella@umassd.edu
University of Massachusetts Dartmouth

Abstract

In the digital era, cloud-based video transcoding services are essential for platforms like Netflix and YouTube, relying on virtual machines for efficient processing. While prior research has primarily explored performance evaluation, limited attention has been given to the resilience of these services against emerging disruptions. This paper aims to predict the processing rate of video transcoding services to evaluate the resilience of small, medium, and large VMs within a cloud environment by applying four regression models. Findings reveal that a mixed regression model incorporating covariates, their interactions, and polynomial terms provides the most accurate prediction of performance and resilience. Among the VMs, the small VM exhibits the highest resilience, sustaining robust performance across varying conditions, with mean time to failure identified as the key factor in its sustained operation. These insights underscore the significance of predictive models for forecasting resilience, supporting proactive decisions for preventive maintenance and fault-tolerant design to reduce cloud service downtime.

Keywords

Resilience, Cloud Computing, Video Transcoding Services, Virtual Machines, Predictive Models

1 Introduction

Cloud computing [6] is essential for modern video transcoding services, allowing users to seamlessly convert multimedia files into various formats for streaming platforms, social media, and other applications. Companies like Netflix, YouTube, and Amazon Prime rely on these cloud-based solutions to efficiently manage the large volumes of content processed daily. Virtual machines (VMs)[17] are extensively employed in cloud environments for video transcoding due to their flexibility and scalability, allowing them to effectively meet dynamic demands. However, VMs are susceptible to a variety of failures[14], such as hardware breakdowns, software glitches, network disruptions, and cyberattacks. In the event of such failures, the mean time to repair (MTTR) remains a crucial factor in maintaining overall service resilience, as prolonged repair times can lead to significant delays and disruptions in transcoding operations.

Previous research on video transcoding services has explored the influence of various factors such as cloud platforms [10, 18], network conditions [16], CPU bottlenecks [4], and service demand [3] on response time. More comprehensive analyses have also been conducted using predictive models for load balancing [5] or for reliability and availability optimization [8] to enhance transcoding time. While these studies have contributed significantly to cloud computing evaluation, they often overlook the combined effects of factors such as VM variability, dynamic arrival times, and the mean time to failure (MTTF) and MTTR on system performance. Moreover, resilience assessments in existing research lack predictive capabilities to dynamically forecast performance preserved and lost under adversarial conditions. This gap underscores the need for a more holistic approach that integrates dynamic resilience evaluation to better anticipate and mitigate potential service disruptions.

This paper addresses this gap by applying regression models to predict the performance of video transcoding services across three distinct VM types (small, medium, and large) under varying conditions, evaluating their resilience in adversarial scenarios. We simulate diverse service request arrival times and alter MTTF and MTTR values to emulate realistic cloud computing environments, estimating the processing rate for each scenario. To model service performance, we utilize multiple linear regression (MLR), multiple linear regression with interaction (MLRI), polynomial regression (PR), and mixture regression (MR). These models are employed to calculate resilience metrics, enabling resilience forecasting that can guide decision-making for optimal resource allocation and enhanced service availability.

The remainder of this paper is organized as follows: Section 2 gives a background on cloud-based video transcoding services. Section 3 describes resilience engineering and common resilience metrics. Section 4 presents the regression models to predict system performance. Section 5 illustrates the proposed modeling approaches. Section 6 discusses the findings and future research plans.

2 Cloud-Based Video Transcoding Services

Video streaming [20] has become a fundamental part of modern internet usage, powering platforms like YouTube, Netflix, Amazon Prime, and many others to deliver high-quality media content to billions of users worldwide. To ensure smooth playback across a wide range of devices—including smartphones, tablets, smart TVs, and desktops—and to adapt to varying network conditions, a process known as video transcoding is employed. This process is crucial for converting raw video files into multiple formats, resolutions, and bit rates to meet the technical specifications of each device and ensure optimal viewing quality. Given the immense volume of video content generated daily, from user-created videos to feature-length films, performing transcoding locally is often impractical and inefficient. Cloud computing [6] provides an ideal solution to this challenge, offering scalable and flexible infrastructure capable of managing the intensive workloads involved in video transcoding.

One of the key components of cloud computing infrastructure is [17] the VMs, which are isolated virtualized environments that run on shared physical hardware, offering the flexibility to perform various computing tasks. In video transcoding, VMs are assigned to process the video streams by encoding them into the necessary formats. These VMs come in small, medium, and large sizes, each with varying computational power, memory, and storage capabilities. A small VM might be sufficient for transcoding a short, low-resolution video, whereas a large VM would be required to process high-definition or 4K content. Once a service request for video transcoding is submitted, the cloud provider allocates a suitable VM based on the workload, and the VM transcodes the video into the requested format. Upon completion, the video is sent back to the user for streaming or download. The performance of this process depends heavily on the VM's capabilities and the speed at which it can process and deliver the transcoded video, which determines the overall response time.

However, like any computing system, VMs are susceptible to failures, which can arise from hardware malfunctions, software crashes, or network interruptions, leading to disruptions or delays in the video transcoding process. Moreover, the MTTF and MTTR are critical factors that affect the availability and reliability of transcoding services. Prolonged downtime can lead to significant delays, negatively impacting the user experience. Therefore, it is crucial to assess the resilience of VMs, focusing on how quickly they can recover from failures and continue delivering content.

3 Resilience Engineering

Resilience [7] is the ability of a system to continue providing services during degraded operations and to rapidly recover high performance after experiencing such degradation. The concept of performance varies depending on the domain, but it is typically understood as the system's ability to meet its goals. For example, in the context of video transcoding services, resilience can be measured in terms of response time, which is the time it takes for the system to process and return a video. When a system experiences failures such as hardware issues or network slowdowns, the response time may increase, indicating a degradation in performance. Resilience may then be measured by how well the system can minimize this increase in response time during the failure and how quickly it can restore normal processing speeds once the issue is resolved.

Figure 1 illustrates changes in performance over time, including a canonical resilience curve (solid line) with four key stages [11]: (i) preparing to preserve performance in case a disruption occurs at time t_d, (ii) absorbing the disruption to maintain functionality until reaching a performance minimum at time t_{min}, (iii) recovering to a new steady-state performance at time t_r, and (iv) adapting processes and designs to enhance future resilience. Additionally, Figure 1 highlights two other trends commonly seen in real-world scenarios: improved performance (dashed line) and more complex degraded performance (dotted line) resulting from multiple shocks. In systems such as cloud computing, multiple shocks frequently occur, where disruptions might cascade across components and amplify the negative impact on overall performance.

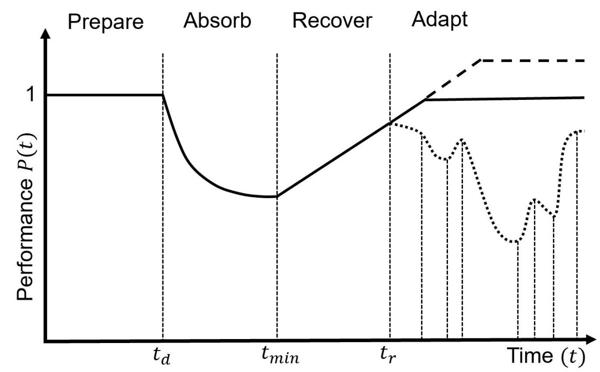

Figure 1: Conceptual resilience curve.

To quantify system resilience and summarize performance over time, resilience metrics [1] have been proposed to evaluate how a system performed during past disruptions. These metrics are usually computed using historical data, which helps engineers and decision-makers to (i) improve future designs by addressing known vulnerabilities, (ii) develop strategies for likely disruptions, (iii) set resilience goals and policies, and (iv) enhance communication and coordination for faster responses. The most common metric used for resilience assessment is the performance preserved (PP) [2]

$$PP = \int_{t_d}^{t_r} P(t)dt \tag{1}$$

which measures the area under the curve between the disruptive time t_d and the recovery time t_r, computing the cumulative performance P over time, where higher values are preferred. The performance lost (PL) is also commonly computed to measure the area above the curve [19] as

$$PL = P(t_d)(t_r - t_d) - \int_{t_d}^{t_r} P(t)dt \tag{2}$$

which represents the accumulated performance lost during recovery periods, and therefore, lower values are preferred.

In the context of cloud-based video transcoding service, these metrics can be applied predictively by setting t_d as the expected time for a system disruption, such as a VM failure or network issue. Similarly, t_r is defined as the estimated time when the system is expected to fully recover and resume normal video transcoding operations. This allows for proactive resilience assessment, predicting performance loss from shocks of varying intensity and estimating performance recovery based on specific actions taken, which helps to minimize downtime and ensure continuous service availability.

4 Predictive Models

Regression models [9] are statistical techniques well-suited for predicting the performance of cloud-based video transcoding by capturing the relationship between input variables, such as workload characteristics, and output performance metrics, such as the processing rate reflecting how quickly a single service request can be completed by a VM. Models based on regression techniques applied in resilience assessment [13] include (i) multiple linear regression, (ii) multiple linear regression with interaction, (iii) polynomial regression, and (iv) a mixture of these three techniques, where the residuals are assumed to be normally distributed.

Multiple linear regression (MLR) predicts the performance P of VMs based on its linear relationship with explanatory factors as

$$\hat{P}(i)_{MLR} = \beta_0 + \sum_{j=1}^{m} \beta_j X_j(i) \tag{3}$$

where β_0 is the baseline change in performance, $X_1(i)$, $X_2(i)$,..., $X_m(i)$ are the m covariates representing the factors impacting VMs performance such as MTTF, MTTR, and arrival time, and β_1, β_2,..., β_m are coefficients characterizing the intensity of these covariates.

Multiple linear regression with interaction (MLRI) models the performance of VMs based on the interaction between covariates

$$\hat{P}(i)_{MLRI} = \beta_0 + \sum_{j=1}^{m} \beta_j X_j(i) + \sum_{j=1}^{m} \sum_{k=j+1}^{m} \beta_{j(m+k)} X_j(i) X_k(i) \tag{4}$$

where $\beta_{j(m+k)}$ is the coefficient characterizing the intensity of the interaction between covariates X_j and X_k.

Polynomial regression (PR) characterizes the performance of VMs based on their linear relationship with covariate X of degree j

$$\hat{P}(i)_{PR} = \beta_0 + \sum_{j=1}^{\omega} \sum_{k=1}^{m} \beta_{j(j+k)} X_k(i)^j \tag{5}$$

where ω is the maximum degree of the polynomial and $\beta_{j(j+k)}$ is the impact of the covariate on performance.

Mixture regression (MR) models the performance of VMs by combining the other three regression methods discussed above

$$\hat{P}(i)_{MR} = \beta_0 + \sum_{j=1}^{m} \sum_{k=j+1}^{m} \beta_{j(m+k)} X_j(i) X_k(i) + \sum_{j=1}^{\omega} \sum_{k=1}^{m} \beta_{j(j+k)} X_k(i)^j \tag{6}$$

which accounts for the individual effects of covariates, their interactions, and polynomial degrees.

To estimate the numerical values of β parameters in the models, least squares estimation (LSE) [12] is used. This method identifies parameter values that minimize the difference between the actual performance P data and the predicted $\hat{P}$ data in each time interval i, using a portion of the actual data for model fitting. To validate and compare the model's performance on a given dataset, goodness-of-fit measures [15] are computed. Common measures include (i) root mean squared error (RMSE), which measures the average magnitude of prediction errors during training, with lower values indicating better model accuracy, (ii) predictive root mean squared error (PRMSE), which evaluates the model's prediction accuracy on unseen data during testing to assess its generalizability, and the (iii) adjusted coefficient of determination (r^2_{adj}), which accounts for the number of predictors in the model to provide a more accurate measure of how well the model explains the variability in the data during training. In practice, no single model consistently outperforms others across all metrics. As a result, model selection often depends on subjective judgment, with preference given to models that demonstrate lower errors and higher r^2_{adj}.

5 Illustrations

This section evaluates the predictive models presented in Section 4 by comparing their ability to forecast the performance of cloud-based video transcoding services using goodness-of-fit measures. The regression models provide accurate predictions of individual VM performance under varying conditions, enabling a detailed analysis of response times specific to each machine type. This approach allows the assessment of the resilience of VMs, which might guide the optimization of performance and resource allocation to enhance overall video transcoding resilience.

Experiments were conducted by sending 50 individual service requests, one at a time, to three distinct VMs (small, medium, and large) while varying key factors such as request arrival time (AT) between 5-15 seconds, MTTF, and MTTR. These factors served as the respective covariates X_1, X_2, and X_3 in the predictive models and were adjusted according to rates found in the literature [14] to enable a comprehensive analysis of how each machine responds under different conditions. The response time (Resp. T) of each VM was recorded, and the processing rate (Proc. R), computed as the inverse of the response time, was used as the performance variable P to be predicted by the models since it reflects how quickly each VM can handle a service request. Table 1 provides an example of how this data was structured to serve as input for the predictive models. To optimize the regression models, the dataset was normalized by dividing all values of each variable by its highest recorded value across all intervals. Once model parameters were estimated using the experimental data, forecasting the future performance of a cloud-based video transcoding service became straightforward: with knowledge of the MTTF, MTTR, and request arrival time for the selected VM, the expected processing rate could be confidently predicted using the model.

5.1 Example I: Model Fitting and Validation

Forward and backward stepwise selection procedures [9] were employed to identify the most relevant subset of covariates that best characterized the data. This process involved iteratively adding or removing covariates from a base model, as well as incorporating interactions between covariates and varying polynomial degrees, to improve prediction accuracy. The primary measure of improvement was the PRMSE criterion, ensuring that each modification to the model enhanced its predictive capability. The model development followed these algorithmic steps: (i) Initially, regression models with no covariates (or all covariates) were constructed. (ii) LSE was then applied using random 50% samples of the data for training models and estimate parameters, followed by computing goodness-of-fit measures for model validation. (iii) The models were tested to predict the remaining 50% of the data not used for model fitting to assess the model's predictive accuracy. (iv) Covariates were progressively added or removed until the minimum PRMSE value was reached. This process was repeated until no further improvement in PRMSE could be achieved by adjusting the covariates.

Table 2 presents the covariates (X), interactions between covariates (denoted by an asterisk '*'), and polynomial degrees included in each model that minimized the PRMSE. For example, the optimal mixture regression (MR) model that achieved the lowest PRMSE for the small VM incorporated the three individual covariates X_1

Table 1: Input data for predictive models

		Small VM				Medium VM				Large VM			
Exp. (i)	**AT (s)**	**MTTF (h)**	**MTTR (h)**	**Resp. T (s)**	**Proc. R (s)**	**MTTF (h)**	**MTTR (h)**	**Resp. T (s)**	**Proc. R (s)**	**MTTF (h)**	**MTTR (h)**	**Resp. T (s)**	**Proc. R (s)**
1	5	3.45677	0.01282	177.14490	0.00565	0.45550	0.02388	204.95417	0.00488	2.58310	0.14334	622.00740	0.00161
2	10	2.06384	0.01355	179.77843	0.00556	3.47076	0.14991	658.66319	0.00152	1.78315	0.24518	988.66555	0.00101
3	15	0.31848	0.02271	212.76847	0.00470	0.31108	0.14398	637.32746	0.00157	2.94786	0.05802	314.87727	0.00318
...	...	...	...	...	...	...	...	...	...	...	...	...	...
50	5	3.75535	0.15389	685.01747	0.00146	5.02563	0.07665	394.93395	0.00253	1.61163	0.33190	1300.83098	0.00077

(arrival time), X_2 (MTTF), and X_3 (MTTR), as well as interactions between $X_2 * X_3$ and $X_1 * X_2 * X_3$, along with quadratic terms for X_2 and X_3 (i.e., X_2^2 and X_3^2). Table 2 also provides the number of parameters in each model and the corresponding goodness-of-fit values, which are penalized for the number of parameters to adjust for model complexity. The best-performing model, highlighted in bold, demonstrates the lowest errors (RMSE and PRMSE) and the highest predictive capability (r^2_{adj}). The results suggest that all models performed well, achieving low errors and high r^2_{adj} values, which indicates that the selected covariates are highly correlated with VM performance. However, mixture regression models that included both individual covariates, their interactions, and higher-order polynomials consistently outperformed other models, demonstrating superior goodness-of-fit measures and achieving $r^2_{adj} > 0.85$ across all cases, which indicates that MR effectively captures the observed performance trends of VMs.

To visually assess the best model fit on the performance dataset, Figure 2 compares the empirical data with the MR-fitted model for each VM, as MR proved to be the best model overall across all three cases. The black line represents the actual performance of each VM, showing the normalized processing rate dataset, while the gray line represents the predictions made by the model. Bullet markers indicate the data points used for training, and asterisk markers denote the testing samples used to compute the predictive accuracy measures. Figure 2 shows that for the small VM, one predicted data point at interval 37 deviates significantly from the actual data, indicating a potential outlier or a region where the model struggles to capture the true performance dynamics. A similar deviation is observed in interval 43 for the large VM, which could indicate a sensitivity in the model to certain conditions, such as unexpected variations in MTTF or MTTR. In contrast, all predicted data points of the medium VM aligned closely with the actual performance, suggesting that the model predictions are accurate, capturing the underlying trends in the data. Overall, the performance of the MR model is robust, with occasional minor deviations, reinforcing its ability to predict VM performance while highlighting areas where further refinement could improve accuracy.

5.2 Example II: Resilience Metrics

To validate how well the MR model assesses the resilience of the VMs based on the predicted video transcoding processing rates over time, the resilience metrics outlined in Section 4, such as performance preserved (PP), Equation (1), and performance lost (PL), Equation (2), were computed by discretizing the integrals into summations over the time intervals. These metrics are calculated both using the actual data to reflect the true resilience of the system and in a predictive manner using the MR model's forecasts, allowing for a comparison between the model's predicted resilience and the system's actual resilience.

Table 3 presents the relative error of the predictive resilience metrics for the MR model. The results indicate that the MR model effectively predicted the performance preserved (PP) and performance lost (PL) for the cloud-based video transcoding services analyzed, with a relative error of less than 0.16 across all three VMs studied. This highlights the robustness of mixture regression models in resilience engineering, positioning them as an accurate method for anticipating disruptions, tracking recovery trends, and predicting future system resilience. By offering reliable forecasts of resilience metrics, the MR model enables a proactive approach to system management, facilitating the early identification of vulnerabilities and better preparation for potential disruptions. Furthermore, for the experiments conducted, results show that the small VM seems to be more resilient compared to the medium and large VMs. This conclusion is drawn from the higher performance preserved and lower performance lost observed for the small VM, indicating its ability to maintain a more stable processing rate under the conditions considered. The small VM resilience suggests it is better equipped to absorb disruptions and recover quickly, making it a potentially more reliable option in scenarios where maintaining continuous service is critical. This emphasizes the value of resilience metrics in comparing system performance across different resources, aiding in the selection of optimal configurations for cloud-based services. Moreover, applying regression models to predict performance ensures that resilience is not only evaluated retrospectively but also proactively integrated into future operations, helping to minimize downtime and enhance overall system robustness.

5.3 Example III: Sensitivity Analysis

Now that the model is built, we can adjust the parameters to assess resilience for other scenarios. For example, we focus on the small VM, as it offers the best performance, and set the arrival time (AT) to 1s, indicating that service requests will arrive more frequently than during model fitting. We then perform a sensitivity analysis by varying the MTTF from 1 to 10 hours and the MTTR from 0 to 1 hour, maintaining values consistent with those used for model fitting and testing, as illustrated in Table 1. Using these inputs, we predict the processing rate for transcoding videos when AT is 1s, and assess system resilience by calculating the performance preserved and lost. Figure 3 illustrates this process.

Figure 3 shows that performance preserved decreases with higher MTTR and lower MTTF, while performance lost increases. However, when MTTF is high, performance remains stable even with longer repair times. This suggests that for the scenarios considered, MTTF has a greater impact on resilience than MTTR, highlighting

Table 2: Models validation on data from the three types of VMs

VM	Model	Covariates	Parameters	RMSE	PRMSE	r^2_{adj}
Small	MLR	X_1, X_2, X_3	4	0.2047	0.2856	0.7086
	MLRI	$X_1, X_2, X_3, X_1 * X_2, X_1 * X_3, X_1 * X_2 * X_3$	7	0.1657	0.2900	0.7733
	PR	$X_1, X_2, X_3, X_2^2, X_2^3$	6	0.0936	0.2885	0.9314
	MR	$X_1, X_2, X_3, X_2 * X_3, X_1 * X_2 * X_3, X_1^2, X_2^2, X_3^2$	9	**0.0821**	**0.2202**	**0.9384**
Medium	MLR	X_1, X_2, X_3	4	0.1664	0.1778	0.5339
	MLRI	$X_1, X_2, X_3, X_1 * X_2, X_1 * X_3, X_1 * X_2 * X_3$	7	0.1089	0.2395	0.7754
	PR	$X_1, X_2, X_3, X_1^2, X_2^2, X_2^3$	7	0.0899	0.1531	0.8471
	MR	$X_1, X_2, X_3, X_1 * X_2, X_1 * X_3, X_2 * X_3, X_2^2, X_3^2$	9	**0.0770**	**0.1249**	**0.8690**
Large	MLR	X_1, X_2, X_3	4	0.1532	0.2635	0.7089
	MLRI	$X_1, X_2, X_3, X_1 * X_2, X_1 * X_3, X_2 * X_3, X_1 * X_2 * X_3$	8	0.1281	0.2524	0.7711
	PR	$X_1, X_2, X_3, X_1^2, X_2^2, X_2^2$	7	0.1340	0.2366	0.7634
	MR	$X_1, X_2, X_3, X_1 * X_3, X_2 * X_3, X_1 * X_2 * X_3, X_2^2, X_3^2$	9	**0.0673**	**0.1902**	**0.9262**

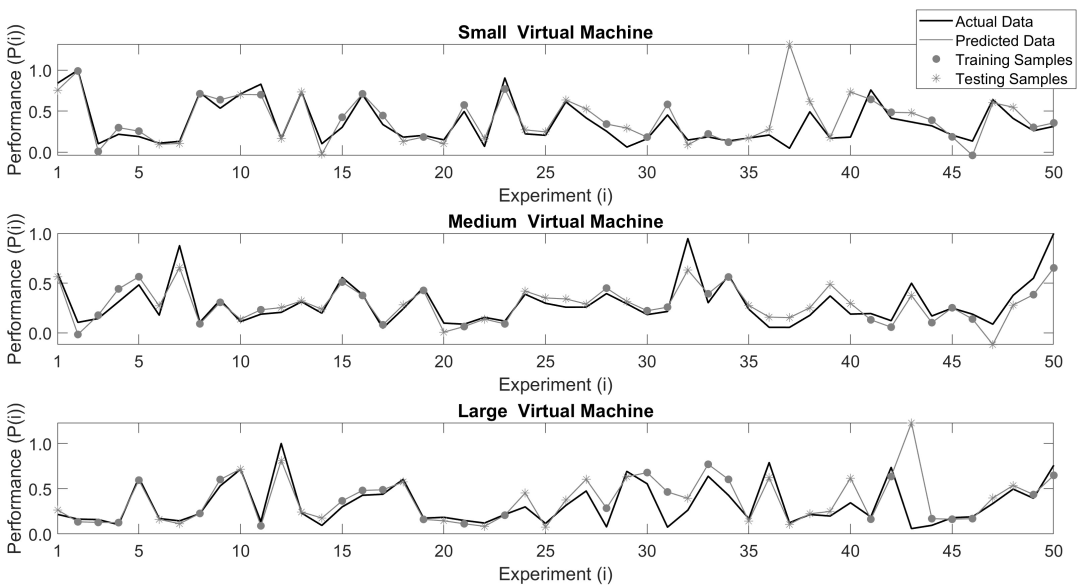

Figure 2: Mixture regression model fit using a random 50% of the dataset from each VM for training

Table 3: Actual and predicted resilience metrics

VM	Model	Data	PP	PL
Small	MR	Actual	17.6973	31.3027
		Predicted	20.4067	28.5933
		Relative Error	0.1531	0.0866
Medium	MR	Actual	14.9083	34.0917
		Predicted	14.3022	34.6978
		Relative Error	0.0406	0.0177
Large	MR	Actual	16.3321	32.6679
		Predicted	18.8499	30.1500
		Relative Error	0.1541	0.0770

the importance of maximizing MTTF to maintain system performance despite potential delays in repair. Maximizing MTTF can be achieved through preventive maintenance, fault-tolerant design, automated monitoring, efficient load balancing, and regular system updates to extend the time between failures and enhance system resilience. Once again, this is where predictive models prove valuable by identifying trends in system performance, enabling proactive actions to be taken before disruptions arise.

6 Conclusion

This paper presented regression models to predict the performance and resilience of cloud-based video transcoding services, focusing on metrics like performance preserved and lost. We explored multiple linear regression (MLR), multiple linear regression with interaction (MLRI), polynomial regression (PR), and a mixture regression (MR) model that combines these techniques. The models were validated through 50 experiments on small, medium, and large

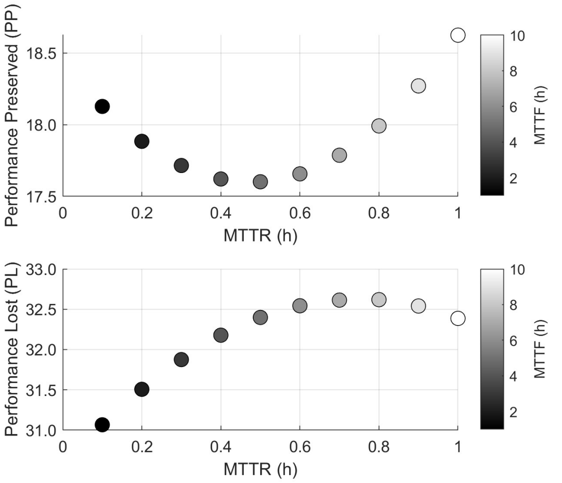

Figure 3: Sensitivy analysis of resilience metrics

virtual machines (VMs) by varying service arrival times, MTTF, and MTTR. Results showed that the MR model performed best, effectively incorporating covariates, interactions, and polynomial terms to predict processing rates and resilience metrics. The small VM proved more resilient, maintaining stable performance under varying conditions, with sensitivity analysis revealing MTTF as the most significant factor impacting its performance. This highlights the significant value of predictive models in forecasting resilience, enabling proactive decisions to optimize MTTF through preventive maintenance and fault-tolerant design, minimizing downtime, and improving resource allocation for cloud services.

Future research will explore more complex VM configurations and operational conditions to improve model generalization, integrate real-time monitoring and dynamic resource allocation to optimize resilience, and develop hybrid models combining machine learning and traditional techniques to enhance prediction accuracy.

Acknowledgment

This research was supported by the Homeland Security Community of Best Practices (HS CoBP) through the U.S. Department of the Air Force under contract FA8075-18-D-0002/FA8075-21-F-0074. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of the Air Force.

References

[1] Bilal M. Ayyub. 2015. Practical Resilience Metrics for Planning, Design, and Decision Making. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 1, 3 (2015), 04015008.

[2] Michel Bruneau and Andrei Reinhorn. 2007. Exploring the concept of seismic resilience for acute care facilities. *Earthquake Spectra* 23, 1 (2007), 41–62.

[3] Syed Muhammad Ammar Hassan Bukhari, Kashif Bilal, Aiman M. Erbad, Amr Mohamed, and Mohsen Guizani. 2022. Video transcoding at the edge: cost and feasibility perspective. *Cluster Computing* 26 (2022), 157–180.

[4] Yuhan Chen, Jingyuan Zhu, Tanvir Ahmed Khan, and Baris Kasikci. 2020. CPU Microarchitectural Performance Characterization of Cloud Video Transcoding. In *2020 IEEE International Symposium on Workload Characterization (IISWC)*. 72–82.

[5] Tewodors Deneke, Habtegebreil Haile, Sébastien Lafond, and Johan Lilius. 2014. Video transcoding time prediction for proactive load balancing. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6.

[6] Adriana Garcia, Hari Kalva, and Borko Furht. 2010. A study of transcoding on cloud environments for video content delivery. In *Proceedings of the 2010 ACM Multimedia Workshop on Mobile Cloud Media Computing* (Firenze, Italy) *(MCMC '10)*. Association for Computing Machinery, New York, NY, USA, 13–18.

[7] Erik Hollnagel, David D Woods, and Nancy Leveson. 2006. *Resilience engineering: Concepts and precepts*. Ashgate Publishing, Ltd.

[8] Ravi Jhawar and Vincenzo Piuri. 2017. Chapter 9 - Fault Tolerance and Resilience in Cloud Computing Environments. In *Computer and Information Security Handbook (Third Edition)* (third edition ed.), John R. Vacca (Ed.). Morgan Kaufmann, Boston, 165–181.

[9] David G. Kleinbaum, Lawrence L. Kupper, Azhar Nizam, and Eli S Rosenberg. 1999. *Applied Regression Analysis and Other Multivariable Methods*. Cengage Learning.

[10] Xiangbo Li, Mohsen Amini Salehi, Yamini Joshi, Mahmoud K. Darwich, Brad Landreneau, and Magdy Bayoumi. 2019. Performance Analysis and Modeling of Video Transcoding Using Heterogeneous Cloud Services. *IEEE Transactions on Parallel and Distributed Systems* 30, 4 (2019), 910–922.

[11] National Research Council. 2012. *Disaster Resilience: A National Imperative*. The National Academies Press, Washington, DC.

[12] Hoang Pham. 1999. Software Reliability. *John Wiley & Sons* (1999).

[13] Priscila Silva, Mariana Hidalgo, Mindy Hotchkiss, Lasitha Dharmasena, Igor Linkov, and Lance Fiondella. 2024. Predictive Resilience Modeling Using Statistical Regression Methods. *Mathematics* 12, 15 (2024).

[14] J.E. Smith and Ravi Nair. 2005. The architecture of virtual machines. *Computer* 38, 5 (2005), 32–38.

[15] Anil K. Srivastava, Virendra K. Srivastava, and Aman Ullah. 1995. The coefficient of determination and its adjusted version in linear regression models. *Econometric Reviews* 14, 2 (1995), 229–240.

[16] Maria Torres Vega, Cristian Perra, and Antonio Liotta. 2018. Resilience of Video Streaming Services to Network Impairments. *IEEE Transactions on Broadcasting* 64, 2 (2018), 220–234.

[17] Zhen Xiao, Weijia Song, and Qi Chen. 2013. Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems* 24, 6 (2013), 1107–1117.

[18] Yulei Xue, Haitao Zhang, and Huadong Ma. 2018. Performance Evaluation of Image and Video Cloud Services. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 733–741.

[19] David Y. Yang and Dan M. Frangopol. 2019. Life-cycle management of deteriorating civil infrastructure considering resilience to lifetime hazards: A general approach based on renewal-reward processes. *Reliability Engineering & System Safety* 183 (2019), 197–212. https://doi.org/10.1016/j.ress.2018.11.016

[20] A. Yaqoob, T. Bi, and G.-M. Muntean. 2020. A Survey on Adaptive 360° Video Streaming: Solutions, Challenges, and Opportunities. *IEEE Communications Surveys Tutorials* 22, 4 (2020), 2801–2838.