International Journal of Advance and Applied Research
www.ijaar.co.in
ISSN – 2347-7075 Impact Factor – 8.141
Peer Reviewed Bi-Monthly
Vol. 7 No. 5 January – February - 2026

A Study on Decision Outcome Effectiveness in Cloud-Based Decision Support Systems Considering Availability Context

Mrs. Sandhya Nilesh Jaykar[1] & Dr. Milind Shankar Godase[2]
[1]*Assistant Professor, Dept. Computer,*
Karmaveer Bhaurao Patil Institute of Management Studies & Research Varye, Satara
[2]*SIOM, Pune.*
DOI - 10.5281/zenodo.18898027

Abstract:

In the era of cloud computing, mobile environments, and wireless networks, maintaining ultra-high availability of decision support computing resources (e.g., five-nines or 99.999% availability) is increasingly difficult. At the same time, organizational reliance on decision support systems continues to grow. Consequently, understanding the likelihood of obtaining accurate and reliable decision recommendations under varying availability conditions has become critically important. This study proposes a probabilistic model that establishes a relationship between the availability of decision support resources and the probability of achieving correct decision outcomes. Drawing on principles of system reliability theory, the proposed probability functions are formally defined and analytically developed. The effectiveness of the model is evaluated using a simulated decision-making scenario, and experimental outcomes are assessed through goodness-of-fit analysis and Analysis of Variance (ANOVA) testing.

Keywords - Cloud Computing, Decision Support Systems (DSS), Resource Availability, System Reliability, Probabilistic Modelling, Decision Outcome Accuracy.

Introduction:

The widespread availability of pervasive networking, low-cost storage, and high-performance computing has established a strong foundation for realizing the potential of cloud computing. Cloud computing may be defined as an environment in which computing resources are hosted, accessed, and utilized through a distributed Internet infrastructure. It extends traditional desktop computing by enabling scalable and virtualized access to distributed processing servers. Within this paradigm, applications are constructed by integrating resources from multiple services and geographically dispersed locations, most commonly delivered through a software-as-a-service (SaaS) model. This approach significantly reduces the burden of software maintenance, updates, and infrastructure management for organizations and end users.

A major advantage of cloud computing lies in its ability to dynamically scale resources and introduce new capabilities on demand, without requiring investments in hardware, specialized personnel, or software licensing. Beyond the cost benefits of the pay-as-you-go model, cloud computing offers reduced implementation risk and faster time-to-value when compared to traditional in-house systems. From the user's perspective, implementation details and underlying technologies are largely transparent; instead, the primary concern is reliable and timely access to services that meet functional requirements (Buyya et al., 2008). Consequently, system reliability and availability have emerged as critical challenges in cloud environments.

Despite advances in infrastructure and service management, achieving complete availability remains unrealistic. While a cloud application may be highly reliable, user access to the service is often constrained by external factors such as network conditions, particularly in mobile and wireless environments. For example, a mobile user interacting with a cloud-based decision support system (DSS) while traveling may experience temporary connectivity loss when passing through areas with limited network coverage, such as tunnels. If users were aware of the likelihood and duration of such interruptions, they could proactively adjust their interaction with the DSS, for instance by caching data locally or postponing critical decision activities. Although exact interruption timing may be unpredictable, probabilistic estimates of availability can provide valuable insight.

Prior research by Russell et al. (2008) demonstrates that awareness of computing resource availability—such as network connectivity, data access, or application services—can significantly influence both system support and user decision-making behavior. Their findings emphasize the importance of exposing availability information to users, rather than confining it solely to internal system algorithms. Building upon this foundation, the present study extends availability awareness to cloud-based decision support systems.

The primary objective of this research is to develop a probabilistic model that incorporates availability context information to assess decision outcome accuracy. Specifically, the study proposes a model that describes the relationship between decision support resource availability and the likelihood of correct decision outcomes. The proposed model is evaluated using a simulation-based experimental approach. Section 2 reviews existing reliability and availability-related technologies and introduces a probabilistic reliability framework for DSS. Section 2.3 presents the proposed model linking DSS availability to decision outcomes. Section 3 details the simulation experiment, Section 4 discusses the experimental results, and Section 5 concludes the paper.

System Reliability, Availability, and Decision Making

Many studies tend to use the terms *system reliability* and *system availability* interchangeably; however, these concepts represent distinct dimensions of system performance. Before examining availability in detail, it is essential to clarify this distinction. Although reliability and availability are most often discussed in the context of hardware or physical equipment, both concepts are equally applicable to software systems. Reliability generally refers to the probability that a system or process will perform its intended function without failure over a specified period of time, whereas availability represents the extent to which a system is capable of delivering its intended function at a given moment (Bhagwan et al., 2003). As a relative measure, availability encompasses not only system failures but also performance-related factors such as delays, congestion, and resource loading.

For example, system reliability may be quantified by the frequency of failures and the duration between them. A system that operates successfully only half of the time would be considered to have 50% reliability and would also be unavailable 50% of the time. However, unavailability does not necessarily imply system failure. A system may be fully operational yet temporarily unable to accept additional requests due to excessive workload or processing delays. In this sense, availability subsumes reliability by accounting for both failure-based and performance-related interruptions. Given the growing dependence of organizations on computing systems, substantial research efforts have focused on measuring and reducing outages that affect reliability and availability.

Much of the existing literature has concentrated on hardware-centric perspectives. Reussner et al. (2003) employed rich architecture definition language (RADL) to predict component reliability through compositional analysis of usage profiles and environmental factors. Mikic-Rakic et al. (2005) proposed an approximate solution for assessing how different deployment environments—such as wired, mobile, and grid systems—affect software availability. Henson (2006) introduced a method to improve hard disk reliability by partitioning file systems into independently repairable fault-isolation domains. Similarly, Dai et al. (2003) developed a model for evaluating service reliability in heterogeneous distributed systems, defining reliability as the probability of successfully delivering a service in a distributed environment. While these studies provide valuable insight, their primary focus remains on system behavior rather than on the consequences of reduced availability.

This hardware-oriented emphasis is characteristic of much of the reliability and availability research and has often resulted in limited consideration of software service availability. With the growing adoption of web services and service-oriented architectures, renewed attention has been given to software availability. In web service environments, hardware is treated as an underlying platform upon which loosely coupled software services operate and are composed to support complex processes. Consequently, software availability becomes critical for effective service composition and execution. Nevertheless, even within this domain, research continues to emphasize infrastructure-level solutions. For instance, Salas et al. (2006) proposed replicating web services across geographically distributed hardware, while Sung et al. (2007) introduced dynamic cluster configurations to improve service availability. Other studies have leveraged contextual information, such as physical location (Ibach et al., 2004) and network bandwidth reservation (Xu et al., 2003), to enhance availability.

Despite these advances, research on web service availability has largely focused on ensuring the successful composition and execution of services, with limited attention to how availability affects the outcomes derived from system usage. The approach presented in this study addresses two key limitations of prior work: the predominant focus on hardware-centric availability and the lack of consideration of decision-level outcome impact. While existing research offers a strong foundation for quantifying and improving system reliability and availability, it does not explicitly connect these metrics to decision-making performance. By mapping system availability to decision outcomes, this study enables users of cloud-based decision support systems to assess the likelihood of achieving successful results within the constraints of a given decision opportunity.

Existing Availability-Related Technologies:

In a decision support context, availability should be considered not only from a systems perspective but also in terms of decision-related resources. These resources may include models, data, services, agents, processing units, output devices, other decision makers, or even the decision maker requesting support. While not all DSS scenarios require external or distributed resources, most modern systems utilize the capabilities offered by computer networks. The introduction of networking and distributed resources adds an additional dimension to the concept of availability. Therefore, availability extends beyond a simple on/off or operating/failed status and reflects a broader notion of resource readiness.

A natural question arises: how can detailed information about resource availability be obtained? Research from multiple domains provides guidance:

High-Availability Computing:

This domain focuses on monitoring and evaluating hardware and system components. Methods have been developed to assess the status of power (Chakraborty et al., 2006; Rahmati et al., 2007), network performance (Roughan et al., 2004; Shahram et al., 2006), computer components (Brown et al., 1999; Weatherspoon et al., 2005), processing loads (Zhoujun et al., 2007), and storage systems (Blake et al., 2003).

Software Services and Web-Based Resources:

Software services provide an abstraction layer for programmable functions and data. Research on web service composition and quality of service (QoS) provides solutions to quantify the likelihood of service interruptions. QoS is typically defined as the probability that a network or service fulfills a defined service-level agreement. This probability can be used to predict potential outages and estimate their duration. Extensive research exists on QoS monitoring, adaptation to dynamic system conditions, and web service composition (Ali et al., 2004; Loyall et al., 1998; Menasce, 2004; Thio et al., 2005; Peer, 2005; Pistore et al., 2004). Web service composition is especially important for scheduling and execution planning, where resource availability information is critical.

Human Users as Decision Resources:

From the perspective of “users as a resource,” human-computer interaction research has focused on detecting user presence and availability (Begole et al., 2004; Danninger et al., 2006; Muhlenbrock et al., 2004). While most studies focus on determining whether a user is online, probabilistic models can forecast user presence and availability. For instance, Horvitz et al. (2002) developed a service to support collaboration and communication by building predictive models using data on user activity, device proximity, and calendar information.

These three research domains provide effective methods for obtaining quantitative availability information for decision-related resources, including hardware, network infrastructure, software services (e.g., web servers, databases, business logic applications), and human collaborators. Given these existing solutions, the probability model developed in this work does not address how availability data is collected. Instead, the model builds on system reliability theory, extends it to decision-related resource availability, and focuses on how the probability of a correct decision outcome may be influenced by the availability of these resources.

Decision-related Computing Resource Availability as a System Reliability Problem:

In the context of Decision Support Systems (DSS), the availability of decision-related resources can be viewed as analogous to system reliability. A decision resource may consist of the DSS itself, another system, data from storage, a network communication medium, an output device (e.g., monitors, printers), collaborating system users, or even the decision maker requesting support.

One way to conceptualize a decision resource is as a hierarchical structure of interacting functions. This hierarchy can have 1 to n levels, and in the case of a computing device, these levels may include hardware, firmware, and software. The fundamental idea is that a decision resource is composed of multiple sub-resources at lower hierarchical levels. Functions at these levels interact to provide the overall functionality of the resource at the top of the hierarchy. Upper-level functions depend on the proper operation of lower-level functions to maintain reliability, as illustrated in **Figure 1**, which presents a generalized hierarchy.

Not every level is required for all resources, but the dependencies between levels are evident. For instance, consider a spatial DSS that provides guidance through a graphical map. This system may rely on data from a remote website that converts street addresses to latitude and longitude. Such a website or data service is itself a resource, dependent on a hierarchy of underlying functions to operate effectively. Examining **Figure 1**, if power is unavailable, connectivity and all higher-level functions fail. Without connectivity, storage cannot deliver data; without storage, processing cannot occur; and without processing, software cannot operate. Similarly, if the software (such as a web server, address-to-lat/long converter, or host operating system) fails, the resource cannot respond to DSS requests. The dependency flows implicitly from bottom to top, although higher levels do not affect lower levels—for example, power may still be available even if connectivity fails.

Typically, component, software, and system availability is expressed as the likelihood that the resource is available, rather than the likelihood it is unavailable. In reliability engineering, the concept of "five nines" refers to systems with 99.999% uptime, which corresponds to approximately 31.5 seconds of downtime per year. Calculating availability becomes more complex in systems with hierarchical dependencies, as is the case for decision resources.

The hierarchical structure implies that the probability of a resource being available can be determined as the product of the availability probabilities of all levels in the hierarchy. Each level's function is independent in terms of its availability but dependent on the lower levels for overall functionality. Therefore, if the availability of each hierarchical level is expressed in probabilistic terms, probability theory can be applied. Because the function at each level is independent in terms of its availability but reliant on lower levels for failure, the multiplicative rule of probability is appropriate. This principle is formalized in **Equation (1)**.

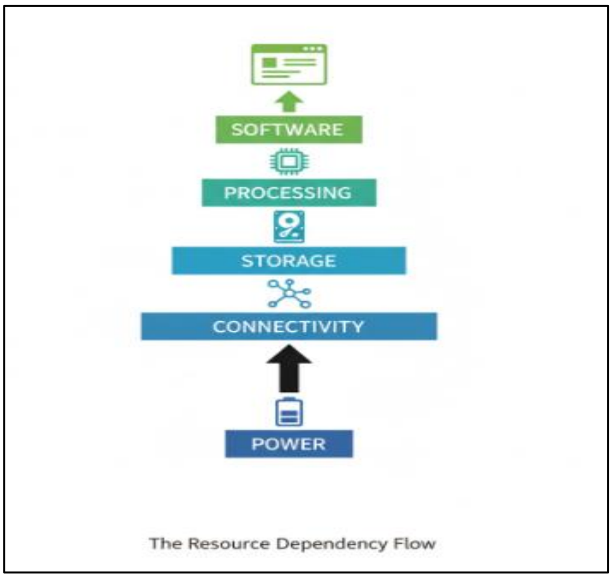

Fig.1 Decision Resource Hierarchy

$P(B \cap A) = P(B) \cdot P(A)$

Applying this to Figure 1, let W = Power, C = Connectivity, G = Storage, O = Processing, and T = Software. Table 1 shows the probability of availability for each level. The overall availability of the decision resource corresponds to the topmost level:

Level	Function	Function Probability	Level Probability
1	Power	P(W)	P(W)
2	Connectivity	P(C)	P(W) * P(C)
3	Storage	P(G)	P(W) * P(C) * P(G)
4	Processing	P(O)	P(W) * P(C) * P(G) * P(O)
5	Software	P(T)	P(W) * P(C) * P(G) * P(O) * P(T)

$$P(R) = \prod_{i=1}^{n} P(F_i)$$

Extending this model, all resources necessary for decision support can be treated as a set of dependent resources. The overall DSS availability is given by Equation (3), where P(S) denotes the availability of all decision resources and P(F) is the availability of each hierarchy level i = 1 to n for each resource r = 1 to t:

$$P(S) = \prod_{r=1}^{t} \prod_{i=1}^{n} P(F_{i,r})$$

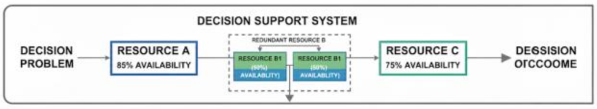

Fig. 2 A DSS with dependent resources

For example, Resource A represents data in a data warehouse, Resource B is the address-to-lat/long web service, and Resource C is the local processor providing graphical mapping. The DSS availability for these dependent resources is:
0.85×0.50×0.75=0.319 or 31.9%

However, Figure 2 does not account for redundant resources. Redundant resources duplicate functionality and generally improve overall system availability by distributing dependencies. The availability of redundant resources is given in Equation (4), where P(RR) represents the probability of a redundant resource for r = 1 to t:

$$P(RR) = 1 - \prod_{r=1}^{t} (1 - P(R_r))$$

Equation (5) generalizes DSS availability for systems with both redundant (P(X)) and non-redundant resources (P(R)), applying the multiplicative rule of probability for all resources:

$$P(S) = \prod_{i=1}^{t} P(R_i) \cdot \prod_{j=1}^{v} \left(1 - \prod_{q=1}^{u} (1 - P(X_q))\right)$$

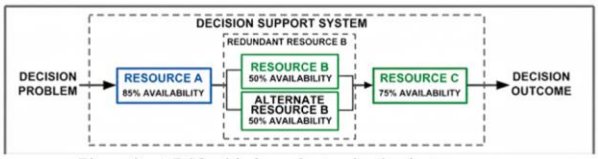

Figure 3. A DSS with dependent and redundant resources

In Figure 3, Resource B has multiple providers for the address-to-lat/long service. The availability of the redundant resource B is:
1−((1−0.50)×(1−0.50))=0.75
Using this value, the overall DSS availability becomes: 0.85×0.75×0.75=0.478 or 47.8%

Thus, the overall resource availability P(S) provides a deterministic measure of the likelihood that the DSS can supply guidance. In data-oriented DSS contexts, where the guidance is assumed to be correct based on available data, P(S) can be directly mapped to the probability of accurate decision outcomes.

Decision Outcome as a Function of Resource Availability: A Probabilistic Model:

The equation presented in the previous section is a key component for predicting the likelihood of a successful decision outcome. Specifically, Equation (5) estimates the probability of decision resource availability, assuming that if the resource is available, the DSS will provide a correct and precise solution to the decision problem.

However, a decision maker may still make the correct choice independently, without DSS support. Therefore, the overall probability of a successful outcome must also account for the probability that the decision maker selects the correct option without assistance.
Equation (6) formalizes this concept:

P(C)=P(S)+P(A)P(C) = P(S) + P(A)P(C)=P(S)+P(A)

where P(C)P(C)P(C) is the probability of a correct decision outcome, P(S)P(S)P(S) is the probability that decision resources are available, and P(A)P(A)P(A) is the probability that the decision maker independently chooses the correct answer from the available alternatives.

This probabilistic model allows the decision maker to quantify uncertainty and anticipate the likelihood of success before interacting with the DSS. An additional advantage arises in time-constrained scenarios: if a resource is temporarily unavailable, it introduces uncertainty because the decision maker does not know when—or if—the resource will become available. Using Equation (6), the decision maker can make a more informed "wait or proceed" decision based on the likelihood of resource availability. Furthermore, providing the probability for specific resources, rather than a general overall probability, can further reduce uncertainty and enhance decision-making accuracy.

Experimental Evaluation:

In today's environment, decision makers increasingly rely on data to support and justify their decisions, generally assuming that the data is accurate and that the Decision Support System (DSS) will provide correct guidance (Amaro et al., 2005; Covin et al., 2001). With the growing prevalence of mobile and cloud computing, and the increasing distribution of system resources through service-oriented architectures, computing grids, and distributed databases, research on resource availability becomes increasingly relevant.

This study addresses the following research question: *Does the availability of decision-related resources map to decision outcomes in a probabilistic manner?* Based on this question and the probabilistic model presented in Section 2.3, the alternate hypothesis is formulated as:

$$
\begin{aligned}
P(C) &= P(R)+P(A)P(C) \\
&= P(R) + P(A)P(C) \\
&= P(R)+P(A)
\end{aligned}
$$

where P(C)P(C)P(C) is the probability of a correct decision outcome, P(R)P(R)P(R) is the probability that decision-related resources are available, and P(A)P(A)P(A) is the probability that the decision maker independently chooses the correct outcome without DSS support.

To evaluate this model, a scenario was required where the DSS provides deterministic guidance (e.g., go/no-go, yes/no, or a singular correct answer) and the decision involves selecting from data alternatives that may be unavailable. A stock trading simulation was chosen for this purpose because it is representative of these decision opportunities and is easily understood.

The decision problem involved purchasing a stock from the Standard & Poor's 500 (S&P 500) list. The decision maker used a simple strategy based on the principle that trading volume precedes price movements (Fontanills et al., 2001). The DSS identifies the stock with the highest volume during the purchase period, providing the correct advice if the relevant resource is available. When the DSS advice was unavailable, the decision maker randomly selected a stock from the S&P 500.

The simulation was programmed in Matlab, providing a robust environment for modeling and evaluating the decision problem. The resource enabling high-volume stock selection was coded with five hierarchy levels (Figure 1), each assigned a probability between 0 and 100%, determined randomly at runtime. Equation (5) was implemented to calculate the overall resource availability probability. A randomly generated "outage" variable was then compared to this probability; if the outage value exceeded the resource probability, the resource was considered unavailable.

The simulated decision maker always followed DSS advice when available and selected randomly when it was not. Each simulation run generated availability probabilities, the outage value, and a single decision outcome, recording the resource availability, DSS advice, correct stock, and stock selected. Multiple executions of the simulation were conducted with varying run sizes ranging from 100 to 1,000,000 runs

More than ever, decision makers depend on data to make and justify their decisions and the general assumption is that the data is accurate and the DSS will provide a correct answer (Amaro et al. 2005; Covin et al. 2001). As mobile and cloud computing environments increasingly become the norm and system resources become more likely to be distributed (e.g. service oriented architectures, computing grids, and distributed databases) the relevance of the research in resource availability will be progressively more significant. As a result, this study raises the research question: does the availability of decision-related resources map to decision outcomes in a probabilistic manner? Based on the above discussion and the probabilistic model presented in Section 2.3, the following alternate hypothesis is formulated:

Decision-related resource availability maps to accurate decision outcomes according to the following probability:

$$P(C) = P(R) + P(A),$$

where P(C) is the probability of correct decision outcome,

P(R) is the probability decision-resources are available,

and P(A) is the probability that the decision maker chooses the correct answer without the decision support system.

To evaluate the probabilistic model, it is desirable to have a scenario where the DSS provides deterministic (go/no-go, yes/no, or singular answer) guidance in support of a decision opportunity. Further, the decision needs to be based on selection from possible data alternatives, where the data resource may be unavailable. For purposes of evaluating the above hypothesis, a simulation of a stock trading decision was modeled. Stock purchasing was chosen because it is representative of the decision opportunities identified above and easy to understand for a broad range of audiences.

A decision problem was constructed where a stock is purchased from the list of Standard & Poor's 500 stocks (S&P 500). The decision maker has a simple strategy for deciding which stock to purchase. In equity trading, there is a concept that volume precedes price (Fontanills et al. 2001) and this is the purchasing strategy that the decision maker employs. The DSS has the capability to identify from the list of 500 which stock has the highest volume for the time of purchase and this is the correct advice provided. To provide this advice, the DSS requires a resource that specifically identifies the stock with the highest volume for the purchase period. To choose a stock the decision maker requests the highest volume stock for the DSS and always takes the provided advice, if available. If the DSS is unable to provide advice, the decision maker selects a stock from the list of 500 stocks.

A precise and explicit model of the decision problem and simulation was programmed in Matlab. This software provided a robust programming environment where the simulation could be created and evaluated. The resource that provided the high-volume stock selection was coded with 5 hierarchy levels according to Figure 1. Each of these levels was coded with a probability between 0 and 100% that would be determined randomly at run time. The equation shown in (5) was coded to determine the overall resource availability probability. This probability was compared to an "outage" variable whose value was also set randomly. If the outage variable value was lower than the resource availability probability, the resource was considered unavailable. The decision maker was also coded as part of the simulation and always took the advice offered by the DSS. When the advice was not available, the simulated decision maker chose a stock randomly from the list of 500.

A run of the simulation consisted of the generating the availability probabilities, the outage value, and a single decision outcome. For each run, the availability status of the resource and subsequently the DSS advice, was recorded, with the correct stock and the stock selected by the decision maker. Several executions of the simulation were made of varying run sizes from one hundred to one million.

Result:

The collected data were analyzed using SPSS, with correct decision outcomes coded as 1 and incorrect outcomes as 0. Similarly, resource availability was coded as 1 for available and 0 for unavailable. For each run-size, the probability of resource availability was calculated and applied to the probability model to forecast the expected decision outcome accuracy.

For instance, in a run of 100 decisions, the resource was available only 3% of the time. Using Equation (6), this corresponds to an expected 3 correct outcomes, calculated as (3/100)+(1/500)×100=3.2/100(3/100) + (1/500) \times 100 = 3.2/100(3/100)+(1/500)×100=3.2/100.

Since correct outcomes must be integer values, the result was rounded to 3 correct and 97 incorrect outcomes. The same calculation was performed for all run-size sets, and these values were used as inputs for Pearson's Chi-Square Goodness of Fit Test.

The Chi-Square test assesses how closely the observed values match those expected from the model (Chernoff et al., 1954). Table 2 presents the results, with the "Expected" column reflecting the values predicted by the availability model.

Run-Size	% Available	Outcome	Expected	Observed	Residual	Chi-Square	Asymp. Sig.
100	3.00%	Correct	3	1	.344	.558	
		Incorrect	97	96	-1		
1,000	2.80%	Correct	30	32	2	.137	.711
		Incorrect	970	968	-2		
10,000	3.29%	Correct	349	345	4	.048	.827
		Incorrect	9,651	9,655	-4		
50,000	3.08%	Correct	1,639	1,630	9	.051	.821
		Incorrect	48,361	48,370	-9		
100,000	.09%	Correct	3,285	3,254	31	.302	.582
		Incorrect	96,715	96,746	-31		
500,000	.11%	Correct	16,565	16,532	33	.068	.794
		Incorrect	483,435	483,468	-33		
1,000,000	3.12%	Correct	33,158	33,103	55	.094	.759
		Incorrect	966,842	966,897	-55		

In general, a large Chi-Square value (typically >1) suggests a poor fit between observed and expected values. As shown in Table 2, the Chi-Square values are small across all run-sizes, indicating that the model fits the data well.

The Chi-Square test also evaluated the hypothesis: a small significance value indicates that the observed distribution differs from the hypothesized distribution (Plackett, 1983). In all runs, the asymptotic significance exceeded the alpha level of 0.05, indicating no significant difference between observed and expected distributions and supporting the alternate hypothesis.

Additionally, a one-way ANOVA was conducted to analyze the distribution of expected versus observed results. The ANOVA yielded a between-groups sum of squares of 0.038 with an alpha level of 1, corroborating the results of the Chi-Square Goodness of Fit Test.

Conclusion:

This study suggests that cloud computing applications can be enhanced if availability-related context information is accessible. The initial objective was to determine whether system availability can be mapped to system usage and benefits. The findings indicate a probabilistic relationship between the availability of decision support computing resources and the likelihood of correct decision outcomes, assuming structured decisions, accurate data, and deterministic DSS guidance. When availability is uncertain, resource ambiguity introduces additional uncertainty into the decision-making process. By providing users with both the likelihood of correct outcomes and the availability of individual computing resources, informed choices regarding potential support from cloud-based applications can be made.

While the proposed model offers a quantitative tool for estimating outcome correctness, its practical value increases when availability information is conveyed directly to decision makers. Consequently, the model should be integrated into client hardware and software that leverage cloud computing services. Moreover, the model relies on collected availability context data and should be combined with other context-aware technologies, such as location tracking, to enable predictive capabilities. Although this research represents a first step in addressing availability challenges in cloud computing, its current implementation is limited to simulation scenarios. Future work aims to operationalize the model on a mobile client, incorporating network sensing, GPS-based location data, and integration with a real-world cloud computing application.

References:

1. Ali, A.S., Rana, O., and Walker, D.W. "WS-QoC: Measuring Quality of Service Compliance," International Conference on Service Oriented Computing (ICSOC04), New York, NY, USA, 2004.
2. Amaro, H., Blake, S.M., Morrill, A.C., Cranston, K., Logan, J., Conron, K.J., and Dai, J. "HIV Prevention Community Planning: Challenges and Opportunities for Data-Informed Decision-Making," AIDS and Behavior (9:2) 2005, pp. 9-27.
3. Begole, J.B., Matsakis, N.E., and Tang, J.C. "Lilsys: Sensing Unavailability," 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, USA, 2004.
4. Bhagwan, R., Savage, S., and Voelker, G. "Understanding Availability," Peer-to-Peer Systems II (IPTPS 2003), Berkeley, CA, USA, 2003, pp. 256-267.
5. Blake, C., and Rodrigues, R. "High availability, scalable storage, dynamic peer networks: pick two," The 9th Conference on Hot Topics in Operating Systems (HOTOS'03), USENIX Association, Lihue, HI, USA, 2003, pp. 1-1.
6. Brown, A., Oppenheimer, D., Keeton, K., Thomas, R., Kubiatowicz, J., and Patterson,

D.A. "ISTORE: Introspective Storage for Data-Intensive Network Services," The IEEE Seventh Workshop on Hot Topics in Operating Systems, Rio Rico, AZ, USA, 1999, pp. 32-37.

7. Buyya, R., and Sulistio, A. "Service and Utility Oriented Distributed Computing Systems: Challenges and Opportunities for Modeling and Simulation Communities," 41st Annual Simulation Symposium (ANSS-41), Ottawa, Canada, 2008, pp. 68-81.
8. Chakraborty, S., Yau, D.K.Y., Lui, J.C.S., and Dong, Y. "On the Effectiveness of Movement Prediction to Reduce Energy Consumption in Wireless Communication," IEEE Transactions on Mobile Computing (5:2) 2006, pp. 157-169.
9. Chernoff, H., and Lehmann, E.L. "The use of maximum likelihood estimates in $\chi 2$ tests for goodness-of-fit," The Annals of Mathematical Statistics (25) 1954, pp. 579-586.
10. Covin, J.G., Slevin, D.P., and Heeley, M.B. "Strategic decision making in an intuitive vs. technocratic mode: structural and environmental considerations," Journal of Business Research (52:1) 2001, pp. 51-67.
11. Dai, Y.S., Xie, M., Poh, K.L., and Liu, G.Q. "A study of service reliability and availability for distributed systems," Reliability Engineering & System Safety (79:1) 2003, pp. 103-112.
12. Danninger, M., Kluge, T., and Stiefelhagen, R. "MyConnector: Analysis of context cues to predict human availability for communication," 8th International Conference on Multimodal Interfaces, Banff, Alberta, Canada, 2006.
13. Fontanills, G.A., and Gentile, T. The Stock Market Course, John Wiley & Sons, New York, NY, USA, 2001.
14. Henson, V., Ven, A.v.d., Gud, A., and Brown, Z. "Chunkfs: Using divide–and–conquer to improve file system reliability and repair," 2nd Workshop on Hot Topics in System Dependability (HotDep '06), Seattle, WA, USA, 2006.
15. Horvitz, E., Koch, P., Kadie, C.M., and Jacobs, A. "Coordinate: Probabilistic Forecasting of Presence and Availability," The Eighteenth Conference on Uncertainty and Artificial Intelligence, Edmonton, Alberta, Canada, 2002, pp. 224-233.
16. Ibach, P., and Horbank, M. "Highly available location-based services in mobile environments," International Service Availability Symposium, Munich, Germany, 2004.
17. Loyall, J.P., Schantz, R.E., Zinky, J.A., and Bakken, D.E. "Specifying and Measuring Quality of Service in Distributed Object Systems," First International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), Kyoto, Japan, 1998.
18. Menasce, D.A. "Composing Web Services: A QoS View," IEEE Internet Computing (8:06) 2004, pp. 88-90.
19. Mikic-Rakic, M., Malek, S., and Medvidovic, N. "Improving Availability in Large, Distributed Component-Based Systems Via Redeployment," Component Deployment, Springer, Berlin, Germany, 2005, pp. 83-98.
20. Muhlenbrock, M., Brdiczka, O., Snowdon, D., and Meunier, J.L. "Learning to detect user activity and availability from a variety of sensor data," Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2004), Orlando, FL, USA, 2004, pp. 13-22.