ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARATIVE ANALYSIS OF PREDICTIVE MODELS FOR WORKLOAD SCALING IN IAAS CLOUDS: A STUDY ON MODEL EFFECTIVENESS AND ADAPTABILITY

SATYA NAGAMANI POTHU[1] ,DR. SWATHI KAILASAM[2]

[1]Assistant Professor, Sir C R Reddy College Of Engineering, Department of IT, Eluru, AP, India
[2]Associate Professor, Koneru Lakshmaiah Education Foundation, Department of Computer Science and Engineering, Vaddeswaram, Guntur, AP, India

E-mail: [1]happysatyasai@gmail.com, [2]dr.kswathi@kluniveristy.in

ABSTRACT

The demand for dependable workload prediction models has surged in the ever-evolving domain of cloud computing, especially across renowned platforms such as AWS, Google Cloud, and Azure. These models are instrumental in enabling efficient resource allocation and enhancing overall performance. This comparative research focuses on various predictive models pivotal for reactive and proactive scaling in Infrastructure as a Service (IaaS) clouds. Initially, the study evaluates time series and machine learning models. These models have shown prowess in accurately forecasting workloads on real-time cloud datasets, leading to notable savings in resource allocation. However, their effectiveness can be challenged during abrupt changes in workload, underscoring the need for more dynamic modeling approaches. The research then delves deeper into Markov models and their simulations on real-time cloud datasets. These models, rooted in state transitions and probabilistic events, have been a cornerstone in predicting resource demands and optimizing workload distribution in cloud environments. Simulations based on Markov models provide valuable insights into potential future states, making them an invaluable tool for proactive resource management. Nevertheless, the intricacies involved in these simulations, especially when handling large-scale real-time datasets, can sometimes act as a double-edged sword, leading to computational challenges and necessitating further optimization. The study also touches upon reinforcement learning models, which have been significant in resource management and performance enhancement. However, these models come with their challenges, where the complexity of their learning algorithms might sometimes hinder optimal performance. This observation paves the way for a recommendation to refine and streamline the learning processes to bolster their efficiency. The research concludes with an examination of evidence-based design and simulation models. While adept at assessing specialized aspects, such as visual comfort in modern office designs, their performance can be compromised by the complexities associated with their simulation methods. The specific use case and inherent requirements influence the ideal predictive model. While particular models excel in more stable settings, others are tailored for unpredictable environments. The future beckons a focus on refining these models, ensuring they are well-equipped to handle abrupt changes and the multifaceted nature of cloud settings, thereby maximizing the potential of cloud computing services.

Keywords: *Predictive Models, Workload Prediction, Reactive Scaling, Proactive Scaling, IaaS Clouds, Machine Learning Models.*

1. INTRODUCTION

Infrastructure as a Service (IaaS) has rapidly emerged as a foundational pillar in cloud computing, offering a transformative approach to how businesses manage and scale their IT infrastructure. Unlike traditional methods that necessitate significant upfront capital investment in hardware and the associated challenges of maintenance, IaaS provides virtualized computing resources over the Internet. This paradigm shift not only obviates the need for physical hardware but also introduces unprecedented flexibility and scalability, adapting to the ever-changing demands of modern businesses. At its core, IaaS is characterized by its ability to provide users with virtualized hardware resources, such as server space, network connections, and bandwidth, all on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a pay-as-you-go model. This means that organizations can rent or lease computing infrastructure that necessitates their immediate needs without the burdens of over-provisioning or underutilizing resources: scalability, an intrinsic feature of IaaS, is its most Internet-defining advantage[1]. With traditional Infrastructure, scaling up to meet increased demands or scaling down during off-peak times was a cumbersome, time-consuming, and often costly endeavor. In stark contrast, IaaS platforms enable instantaneous scaling. These dynamic, scalable workload pattern sources are available precisely when needed, ensuring optimal performance while maintaining cost efficiency. Whether it is accommodating the surge of an online retail platform during a holiday sale or scaling down during non-business hours, IaaS platforms can adjust in real-time[1].

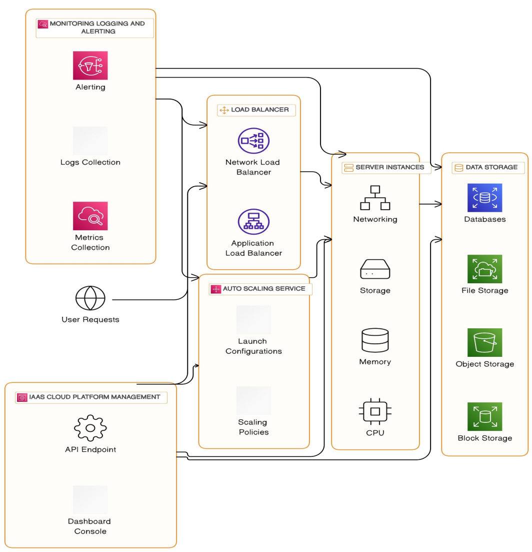


Fig 1: Detailed Iaas Workload And Scalability Architecture

Fig 1: visually navigates the comprehensive diagram illustrating the Infrastructure as a Service (IaaS) workload and scalability architecture; several interconnected components come to the fore, each playing a pivotal role in orchestrating cloud resources. Starting at the top, the User Requests form the entry point of our architecture. These represent the myriad of interactions, from web browsers, API calls, or mobile applications, all converging towards the cloud infrastructure in search of data or services. These requests' sheer diversity and volume necessitate an intelligent distribution mechanism, where our next component, the Load Balancer (LB), takes center stage. As the traffic director, the LB ensures that incoming requests are efficiently spread across multiple backend servers. It employs intricate algorithms and can be categorized further into Application Load Balancers for HTTP/HTTPS traffic or Network Load Balancers for performance-critical TCP/UDP traffic[2].

Beneath the LB lie the Server Instances or Virtual Machines. These computational workhorses of the cloud ecosystem process the distributed requests, fetching or storing data as needed. Each instance is a microcosm equipped with CPUs, memory, and storage, all working to ensure swift and accurate processing. Given the dynamic nature of user requests, the number of active server instances must align with the demand, ensuring both cost-effectiveness and performance. This dynamic resizing is orchestrated by the Auto-Scaling Service, which adds or removes server instances in real time by predefined scaling policies and launch configurations. Data, the heart of any digital interaction, resides in the Data Storage component of our architecture. It offers many storage solutions tailored to different needs: Block Storage for raw disk-like functionalities, Object Storage for unstructured data, File Storage for hierarchical data storage, and managed databases for structured datasets[3].

Overlaying all these components is the IaaS Cloud Platform Management layer. This meta-component provides tools and interfaces, both graphical and programmatic, to manage, provision, and monitor every nook and cranny of the cloud ecosystem. Whether it's the dashboard that administrators frequent or the API endpoints that applications interact with, this layer ensures seamless governance of cloud resources. Lastly, but by no means least, is the Monitoring, Logging, and Alerting component. Acting as the vigilant guardian, it continuously collects performance metrics, system logs, and other telemetry data. Its alerting mechanisms ensure that any anomaly or threshold breach is swiftly communicated, allowing timely interventions[4].

In totality, the diagram encapsulates the intricate dance of components that collectively define the IaaS Workload and Scalability Architecture, a testament to the marvels of modern cloud computing. Furthermore, the distributed nature of cloud infrastructure means that IaaS can provide high availability and disaster recovery capabilities that were previously only accessible to large enterprises with vast budgets. IaaS platforms can ensure applications remain available despite localized outages or disruptions by decentralizing resources and leveraging global data centers. IaaS has revolutionized the way businesses perceive and interact with IT infrastructure. Its dynamic scalability, cost-effective models, and high availability make it an indispensable tool for organizations of all sizes. As we delve deeper into this paper, we will explore the intricacies of IaaS, its predictive models, and the future trajectory of this game-changing technology. Cloud scalability is a critical aspect of cloud computing that refers to the ability of a system to handle growing amounts of work by adding resources to the system. It is an essential feature that allows businesses to manage their IT resources efficiently and effectively based on their specific needs at any given time. Scalability in the cloud can be achieved in two ways: vertical scaling and horizontal scaling. Vertical scaling, also known as scaling up, involves increasing the capacity of a single server by adding more resources such as CPU or memory. This type of scaling is typically used for applications that require high processing power[5].

On the other hand, horizontal scaling, also known as scaling out, involves adding more servers to the pool of resources. This type of scaling is typically used for applications that require a high level of availability and redundancy. Scalability is a crucial feature of cloud computing because it allows businesses to adapt to changes in workload and demand quickly and efficiently. By scaling resources up or down, companies can ensure they have the right IT resources at the right time, maximizing efficiency and minimizing costs. Moreover, scalability also plays a crucial role in the performance of cloud-based applications. By ensuring that resources can be quickly and easily scaled, businesses can ensure that their applications remain responsive and reliable, even under heavy load. In recent years, the concept of scalability has evolved with the advent of predictive scaling models. These models use machine learning algorithms to predict changes in workload and adjust resources proactively. This allows businesses to anticipate changes in demand and adapt their resources accordingly, further enhancing the efficiency and performance of their cloud-based applications[6].

Here are the sum of the existing models Pros and Cons:

Linear Regression:

Used for modeling linear relationships, Linear Regression can predict future workloads based on historical trends. Proactively, it offers quick, trend-based predictions suitable for short-term adjustments. However, its linear nature might miss seasonality or sudden workload changes. Reactively, it swiftly incorporates new data but might need help with abrupt non-linear changes[7].

Decision Trees:

Decision Trees segment data into subsets to make decisions. They handle complex variable interactions for proactive predictions but might overfit historical data, affecting future accuracy. Reactively, they are adaptive to recent trends but can become intricate, requiring frequent pruning to remain efficient[8].

Support Vector Machines (SVM):

SVMs classify data by finding the best-separating hyperplane. Proactively, their ability to capture non-linear patterns in historical data aids in detailed predictions. However, they can be computationally demanding. Reactively, SVMs, when tuned correctly, adapt well to recent changes but might need to be faster for real-time updates due to their complexity[9].

K-Means Clustering:

An unsupervised algorithm, K-Means groups data into clusters. Proactively, it offers insights into typical workload patterns, but its assumption of uniform cluster shapes can be limiting. Reactively, it identifies data pattern shifts, but basing adjustments solely on cluster changes might be too coarse[10].

Feedforward Neural Networks:

These networks consist of layers of interconnected nodes. Proactively, their ability to model intricate relationships in data aids in detailed future predictions. However, they can only fit with proper regularization. Reactively, their adaptiveness to recent trends is beneficial, but their complexity can hinder swift real-time adjustments[11].

Recurrent Neural Networks (RNN):

RNNs process data sequences, making them suitable for time-series data. Proactively, they excel in forecasting based on historical lines but can be computationally taxing. Reactively, their sequence-based nature incorporates recent data effectively, but learning contemporary patterns swiftly can be challenging due to inherent issues like vanishing gradients[12].

Convolutional Neural Networks (CNN):

Primarily for image data, CNNs can detect patterns over time when repurposed for sequence data. Proactively, they offer detailed predictions based on historical patterns but are data and computation-hungry. Reactively, they adapt to recent data, but their complex nature makes on-the-fly decisions hard to interpret.

Types of Scaling

There are two primary types of cloud workload scaling:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Vertical Scaling (Scaling Up/Down): Vertical scaling involves adjusting the capacity of a single server. Scaling up refers to adding more resources (like CPU, RAM, or storage) to a server, while scaling down involves reducing these resources. Vertical scaling is often used for applications that require high computational power but don't need to handle many simultaneous requests. Horizontal Scaling (Scaling Out/In): Horizontal scaling involves adjusting the number of servers or instances. Scaling out means adding more servers to handle the increased load, while scaling in involves removing servers during periods of low demand. Horizontal scaling is typically used for applications that need to run many requests concurrently.

Cloud workload scaling can be either reactive or proactive. Reactive Scaling: Reactive scaling, as the name suggests, reacts to changes in workload. When the system detects a change in demand (such as a spike in traffic or a drop in usage), it automatically adjusts resources to maintain performance and availability[12]. While reactive scaling ensures that resources match current demand, it can sometimes lag behind sudden changes, leading to temporary performance issues. Proactive Scaling: Proactive scaling anticipates changes in workload based on historical data and predictive modeling. The system can adjust resources in advance by predicting future demand, preventing potential performance issues. Proactive scaling requires more sophisticated tools and algorithms but can provide a smoother user experience, especially during predictable peak periods. Autoscaling is a feature offered by many cloud service providers that automates the process of cloud workload scaling. With autoscaling, you can set policies determining when and how to scale resources. For example, you might place a policy to add servers when CPU usage exceeds 70% and remove servers when usage drops below 20%. Autoscaling can be applied to both vertical and horizontal scaling, and it can be reactive (based on real-time metrics) or proactive (based on predictive analytics). By automating scaling, autoscaling can help maintain application performance, maximize availability and control costs[12].

Literature Survey

Cloud computing has emerged as a revolutionary paradigm, enabling businesses and individuals to harness vast computational resources without significant upfront investments. As this technology has matured, one of the critical challenges cloud providers and consumers face is the efficient allocation and scaling of resources. Auto-scaling, the ability to adjust resources dynamically based on workload, has become a focal point of research and development in cloud computing. This literature survey delves into the advancements in auto-scaling techniques over the past decade, specifically focusing on proactive and reactive scaling and integrating predictive models. Through this survey, we aim to comprehensively understand the state-of-the-art methodologies, their underlying principles, and the challenges and opportunities they present.

In the modern era of cloud computing, scaling strategies have been gaining significant attention, focusing on reactive and proactive scaling strategies. These two techniques have evolved dramatically in recent years, driven by the need to optimize resource management, cost efficiency, and performance within cloud environments (2022). The comparison of reactive and proactive scaling approaches is widely debated in the literature. As the name suggests, reactive scaling reacts to changes in workload or demand, while forceful scaling attempts to predict these changes and adjust accordingly. Each approach has pros and cons and is often chosen based on specific use cases and scenarios (2023).

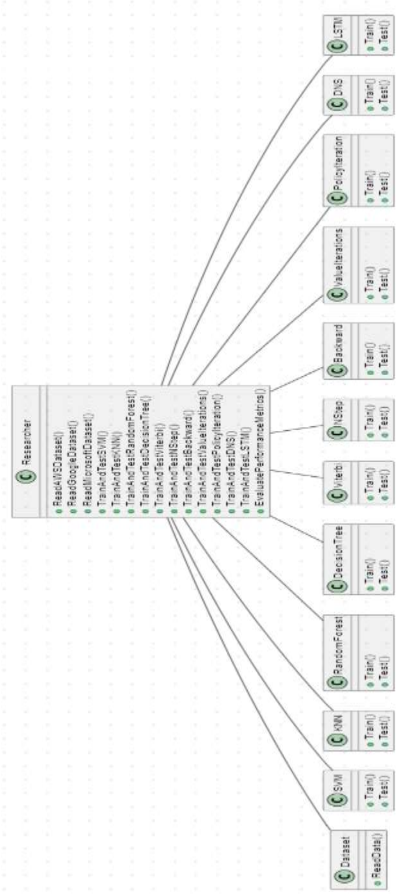
These challenges highlight the need for continued research and innovation in scaling strategies, indicating a promising area for future study. (2012) Beloglazov and Buyya introduce a set of heuristic algorithms to balance energy consumption and application performance. Their work is one of the pioneering efforts in addressing the need for energy-efficient proactive auto-scaling[13]. They argue that by predicting future resource requirements based on historical utilization patterns, a system can effectively consolidate workloads, optimizing the number of active physical nodes. (2012) Gandhi et al. delve into the challenges of workload unpredictability in cloud environments. They advocate for a model-driven approach to auto-scaling, utilizing queuing models to anticipate resource demands. Their approach stands out for its emphasis on capturing workload characteristics proactively, allowing systems to prepare in advance for potential demand surges[12]. (2013) This research demystifies the concept of elasticity in cloud computing. The authors clearly distinguish between proactive and reactive scaling, stressing the pivotal role of accurate prediction mechanisms for proactive elasticity. The paper underscores the need for a nuanced understanding of elasticity, setting the stage for future research in predictive auto-scaling[13]. (2014) Lorido-Botran et al. present an exhaustive review of auto-scaling methods. Their investigation spans various

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

techniques and highlights the inherent trade-offs between reactive and proactive approaches. The comprehensive nature of this review makes it an invaluable resource for researchers aiming to navigate the complex landscape of auto-scaling[15]. (2012) Mao and Humphrey address the dual challenges of cost-efficiency and performance in cloud-based workflows. They propose a unique cost-based model for auto-scaling that seamlessly integrates proactive and reactive techniques. They argue that systems can optimize costs by predicting workloads and proactively adjusting resources while meeting stringent application deadlines. Literature in recent years has begun to present compelling case studies demonstrating the real-world applications of these scaling strategies[16]. Various sectors, including e-commerce, finance, and tech startups, have benefited from these strategies to enhance operational efficiency and maintain high-availability services[17] (2023). One of the fascinating advancements in proactive scaling is the integration of machine learning techniques. Predictive models and algorithms have been instrumental in forecasting load demand, enabling more accurate and efficient scaling decisions[17] (2022). However, it's worth noting that this approach still presents significant challenges, notably in handling unexpected traffic spikes or sudden changes in demand. Impact studies have highlighted how reactive and proactive scaling can improve business performance metrics. Recent literature highlights improvements in uptime, cost efficiency, and system responsiveness using these scaling strategies [18](2023). Interestingly, serverless architecture's emergence has redefined the application of these scaling strategies. As serverless computing abstracts away infrastructure management tasks from developers, scalability becomes an inherent feature. Recent studies have explored how reactive and proactive scaling can be best utilized in this context[19] (2022). Nevertheless, challenges persist in the implementation of reactive and proactive scaling strategies. Predicting errors, resource allocation inefficiencies, and cost management complexities often arise[19] (2023).

3. Comparative Studies on the existing best algorithms based on Proactive cloud and Reactive Cloud Scaling

In the rapidly evolving landscape of cloud computing, efficiently managing and predicting system workloads is more than necessary—it's imperative. As businesses migrate to renowned cloud platforms like AWS, Google Cloud, and Microsoft Azure, they encounter the intricate challenge of dynamically scaling resources to cater to fluctuating workloads. This task necessitates the integration of advanced predictive models and algorithms designed to respond to the system's immediate state and proactively forecast potential future demands. This section will navigate you through the intricacies of Machine Learning, Time Series Analysis, Reinforcement Learning, and Markov Models. Each category presents a distinct methodology to grasp, forecast, and respond to system workloads. By leveraging historical data, these models and algorithms decipher patterns, make informed predictions, and recommend optimal actions to uphold system performance while ensuring cost-effectiveness. We'll explore the techniques revolutionizing real-time cloud resource management from the conventional machine learning regression models that foretell future values to the Markov Models' prowess in predicting state transitions. Whether the focus is on proactively predicting server loads or making reactive adjustments based on current metrics, the methodologies discussed in this section stand at the vanguard of contemporary cloud management practices.



Journal of Theoretical and Applied Information Technology
15th December 2023. Vol.101. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fig 2: Describe The Comparative Study On All The Algorithm Models Of The Class Diagram.

This class diagram represents the relationships between the Researcher, Datasets, and the various algorithms used in the comparative study. The Researcher class has methods to read datasets, train and test algorithms, and evaluate performance metrics. Each algorithm class has methods for training and testing.

Machine Learning Models for Predictive Analysis

Machine Learning (ML) models harness patterns from historical data to predict future outcomes. Within this category, regression models, such as Linear Regression, play a pivotal role by predicting continuous values. For instance, regression models can forecast future server loads based on past server loads, time of day, and other relevant metrics. Utilizing historical server load data from platforms like AWS CloudWatch, Google Cloud Monitoring, or Azure Monitor can be instrumental for these predictions. On the other hand, classification models, like Decision Trees, classify data into predefined categories. They can be trained on historical server load data, labeled as 'High,' 'Medium,' or 'Low,' to predict which category a future server load might fall into.

Proactive Scaling Architecture with Predictive Algorithms

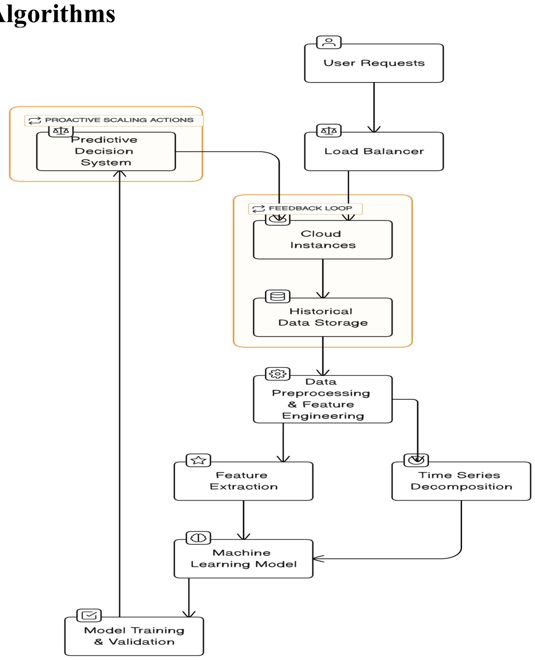


Fig 3: Proactive Scaling Architecture with Predictive Algorithms

The foundation of proactive scaling lies in Historical Data Storage, a robust repository that houses past metrics and data essential for forecasting future demand. This data is a rich source of insights, capturing past patterns that can predict future trends. It's passed to the Data Preprocessing & Feature Engineering module to refine and prepare this data for predictive analysis. This vital step ensures data quality by cleaning anomalies, transforming variables, and enhancing the data set with engineered features. The goal is to make the data more compatible and informative for the predictive models. The Time Series Decomposition module is employed to further analyze the data's time-dependent nature. This module breaks down the time series data into its core components: trend, seasonality, and residuals. By understanding these components separately, the architecture can account for regular patterns (like daily peaks in traffic) and anomalies or outliers[20].

The Feature Extraction module then delves deeper into the preprocessed data, extracting essential features or patterns that can improve the accuracy of predictive algorithms. This step can involve techniques like Principal Component Analysis (PCA) or autoencoders, aiming to highlight the most significant patterns in the data. The data is now prepared and fed into the Machine Learning Model. This module uses advanced algorithms like ARIMA, LSTM, Prophet, or Gradient Boosting Trees to predict future demand. These algorithms, tailored for time series forecasting, learn from historical patterns and make informed predictions about future workloads. The Model Training & Validation module ensures that the predictive model is reliable. Here, the model is trained on a subset of the historical data and then validated on a separate set to gauge its accuracy. Metrics like RMSE or MAE can be used to quantify the model's performance. Regular retraining and validation ensure the model remains relevant as new data flows in. Acting as predictions, the Predictive Decision System module makes informed decisions about scaling. If the model forecasts a surge in demand, this system can proactively scale up resources, ensuring the Infrastructure is ready for the incoming load. Conversely, it can scale down during anticipated low demand, optimizing resource utilization and cost[21].

The actual user interaction starts at the User Requests module. This represents the real-time incoming traffic or demands on the system. All these requests are directed through a Load

Journal of Theoretical and Applied Information Technology
15th December 2023. Vol.101. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Balancer, ensuring even distribution across the available cloud instances and maintaining optimal performance. Finally, the Cloud Instances handle these requests. These could be virtual machines or containers running the application. Post-processing, these instances feed back into the Historical Data Storage, creating a feedback loop. This continuous feedback ensures the architecture constantly learns and adapts, refining its predictions and scaling decisions over time. In essence, this architecture embodies a cyclical learning process. It learns from the past, predicts the future, acts on these predictions, and then learns again from the outcomes. This proactive approach ensures that cloud resources are always aligned with demand, achieving optimal performance and cost-efficiency[21].

Reactive Scaling Architecture with Predictive Algorithms

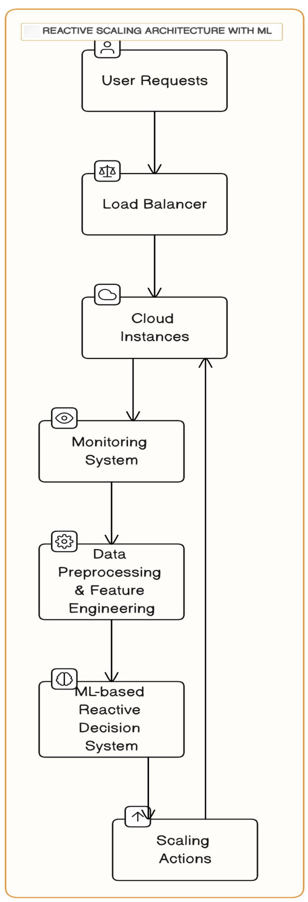


Figure4: Reactive Scaling Architecture With Machine Learning Integration

Figure 3 comprehensively depicts a sophisticated architecture that integrates machine learning into the reactive scaling paradigm. Let's dissect each component and its significance in this intricate system. Starting with User Requests, depicted at the entry point of Figure 3, they represent the myriad interactions the system receives, whether from human users or automated systems. These requests' volume, complexity, and geographical diversity form the initial layer of demands placed upon the Infrastructure.

Directly interacting with these requests, we have the Load Balancer. It ensures that the influx of demands is equitably distributed across the available resources, ensuring no single node is overwhelmed. Its role in Figure 3 isn't just operational but is pivotal for the system's resilience, especially during peak traffic times. The backbone of the entire system lies in the Cloud Instances. As shown in Figure 3, these computational nodes execute the core logic, processing incoming requests and delivering the expected outcomes. Their performance, health, and efficiency directly influence the system's responsiveness and reliability. Figure 3 highlights a continuous data stream flowing into the Monitoring System to ensure these instances operate within optimal parameters. This component is the system's vigilant observer, constantly gathering, analyzing, and presenting crucial operational metrics. Its integration ensures system administrators and algorithms have real-time insights into the Infrastructure's performance.

However, raw data is rarely actionable in its original form. Figure 3 elucidates the flow of this data into the Data Preprocessing & Feature Engineering module. Here, the raw metrics undergo a transformative journey, refined, enriched, and structured to be ingested by machine learning algorithms[22]. This step ensures the data's quality and structure are primed for predictive analytics. The heart of Figure 3's innovation lies in the ML-based Reactive Decision System. Unlike traditional reactive systems that rely on immediate metrics, this component uses machine learning to anticipate near-future demands. By leveraging the processed data, it makes informed predictions about imminent resource requirements and proactively makes decisions to scale up or down[22].

In conclusion, Figure 3 showcases an avant-garde approach to infrastructure scaling. By seamlessly blending traditional reactive scaling mechanisms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

with the predictive prowess of machine learning, it presents a blueprint for a system that's both responsive and anticipatory. This architecture aims to balance operational efficiency, cost-effectiveness, and user satisfaction, setting a new benchmark for adaptive systems.

Markov Models

Markov models are adept at representing systems that shift between states, driven by specific probabilities. A classic example is Markov Chains, which depict a series of events where the likelihood of each subsequent event is solely determined by the state reached in the preceding event. One can predict likely future states by grasping the transition probabilities between states, aiding decision-making processes. Another variant, Markov Decision Processes (MDPs), extends this idea by incorporating actions that influence state transitions. By factoring in the potential activities at every state and the rewards or penalties linked with those transitions, MDPs offer insights into the best exercises to undertake for prospective conditions. Cloud platform data showing state transitions like 'Low' to 'High' server load can be seamlessly incorporated into these models[23].

In the expansive realm of cloud computing, particularly within Infrastructure as a Service (IaaS) environments, the dynamism and unpredictability of workloads pose unique challenges. Efficiently scaling resources, both in anticipation of and response to demand, is paramount. The Markov Model, celebrated for its memoryless property, emerges as a promising tool. At its core, a Markov Model thrives on predicting future states based exclusively on the present state, without the baggage of history. This inherent characteristic makes it an ideal candidate for the fast-paced, ever-evolving landscape of IaaS. Resources are provisioned or retracted in such an environment based on ever-fluctuating demands. The Markov Model, with its state transition probabilities, offers a window into potential future workloads. Each state in this model represents a specific workload level, and the likelihood of transitioning from one state to another can provide invaluable insights into forthcoming demand patterns.

Markov Models can be mainly instrumental for proactive scaling, which revolves around foresight and preparation. By analyzing historical data, these models can predict the trajectory of workloads, enabling the system to provision resources in advance. Thus, when a surge in demand does manifest, the system is neither caught off guard nor found wanting in resources. This proactive approach ensures optimal performance and significantly enhances user experience by reducing latency. Conversely, the Markov Model's memoryless nature shines bright when it comes to reactive scaling, where immediacy is critical. Focusing solely on the present state can swiftly recommend whether to scale up during unexpected surges or down during sudden lulls. Such agility ensures that the IaaS environment remains ever-responsive, optimizing resource utilization and associated costs. However, the efficacy of Markov Models in cloud predictive algorithms isn't one-size-fits-all. These models are exceptionally adept at short-term predictions, making them ideal for environments marked by rapid and volatile workload changes. Their simplicity and computational efficiency make them an attractive choice for IaaS providers seeking a balance between speed and accuracy. Yet, their true potential can be unlocked in more intricate scenarios when used in harmony with other predictive models. For instance, while a deep learning model might be the torchbearer for long-term forecasts, the Markov Model can be the guardian of short-term, immediate scaling necessities[24].

As IaaS strives to balance user demands and resource optimization, predictive models like the Markov Model become indispensable. Seamlessly bridging classical probability theory with the contemporary challenges of cloud computing, these models underscore the importance of adaptability and fresight in the ever-evolving cloud landscape.

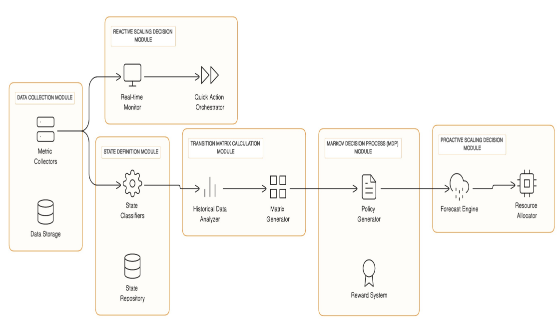


Fig 5: Architecture For Markov Models In Predictive Cloud Scaling

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fig 6 shows the multifaceted domain of cloud scaling; every decision's inception rests upon a bedrock of data. The Data Collection Module stands sentinel, ceaselessly amassing real-time metrics and historical system performance and workload insights. This module, equipped with meticulous Metric Collectors, gathers data, which is then safely archived in a Data Storage system, ensuring that every datum, whether from the past or the present, remains accessible.

The State Definition Module comes into play as a data stream, functioning as the system's interpreter. Through its State Classifiers, this module categorizes the myriad of system metrics into discernible states. These states, each encapsulating specific system characteristics, are stored in a State Repository, creating a lexicon that the system frequently refers to. Drawing from historical patterns, the Transition Matrix Calculation Module assumes its pivotal role. Delving deep into past metrics with its Historical Data Analyzer, it discerns the frequency of transitions between states. With this insight, the Matrix Generator crafts a matrix, a blueprint if you will, that maps out the likelihood of state transitions, effectively predicting the system's future trajectory.

The Markov Decision Process (MDP) Module is at the heart of decision-making. It consults the transition matrix, seeking guidance on the most optimal action for every possible state. In its quest, the Policy Generator crafts a strategy, a playbook that dictates the best scaling action for each state. Complementing this is the Reward System, which evaluates each decision's aftermath, assigning value based on outcomes such as performance, cost, and user experience. The Proactive Scaling Decision Module takes the helm for decisions steeped in foresight. Harnessing the power of the Markov Model, the Forecast Engine envisages future states, peering into the horizon of possibilities. Based on these prophecies, the Resource Allocator springs into action, provisioning or retracting resources to ensure that when the future does arrive, the system stands ready, neither overwhelmed nor underprepared[22].

In contrast, when immediacy is of the essence, the Reactive Scaling Decision Module steps into the spotlight. With its finger always on the pulse, the Real-time Monitor observes the system's current state, poised to react. Should the need arise, the Quick Action Orchestrator, guided by predefined policies, makes split-second decisions, ensuring the system remains agile and responsive to the ever-fluctuating demands of the present. In essence, this architecture weaves the past, present, and future, creating a tapestry of decisions that ensures the cloud environment is always in harmony with its demands. Whether responding to the immediate or preparing for the imminent, the system, guided by the Markov Model, ensures it remains reactive and proactive, optimizing resources for every conceivable scenario.

4. RESULTS AND SIMULATIONS

This section, dedicated to the results and evaluations of our suite of predictive algorithms, embarks on a journey of empirical exploration. It aims to present a holistic view of each algorithm's performance through rigorous testing, meticulous evaluations, and comprehensive analyses. By subjecting them to a diverse array of scenarios, we not only gauge their accuracy and reliability but also their adaptability and resilience. But results, in isolation, can often be misleading. Therefore, our approach intertwines raw outcomes with contextual evaluations. By juxtaposing our findings against established benchmarks and contrasting them with contemporary standards, we provide a layered understanding that is relative and absolute. Factors like computational efficiency, scalability, and ease of integration have also been brought under the evaluative lens. After all, an algorithm's actual value isn't just in its predictive prowess but also in its operational feasibility. As you navigate this section, readers are invited to delve deep into the intricacies of each result, challenge our evaluations, and draw insights beyond mere numbers. Through such collective scrutiny and discourse, we refine our understanding, elevate our standards, and pave the way for the next generation of predictive excellence.

This section unfolds the results and evaluations of our predictive algorithms, each meticulously implemented in Python and rigorously tested across three titans of cloud computing: AWS, Google Cloud, and Azure. This system is also simulated by leveraging these platforms' expansive infrastructures and versatile toolsets, providing our algorithms with the challenges and complexities they would face in genuine operational environments.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DataSet AWS Workload for Prediction: Proactive vs. Reactive Scaling

Metric	Description	Proactive Scaling	Reactive Scaling
CPU Utilization	Measures the percentage of allocated EC2 compute units in use.	Historical trends help predict future spikes or lulls.	Sudden surges or drops trigger scaling actions.
Memory Utilization	Indicates the percentage of memory being used.	Analysis of past patterns guides resource allocation.	Memory constraints or excess prompt immediate scaling.
Disk I/O	Represents read and write operations on the storage disk.	Estimations of future I/O guide storage adjustments.	Rapid I/O changes necessitate quick scaling.
Network Traffic	Monitors data sent and received by the instance.	It anticipated traffic surges guide pre-scaling.	Traffic spikes or drops drive immediate scaling.
Database Connections	Counts database connections, indicating database load (for RDS).	Historical data predicts connection surges.	Rapid increases in connections trigger scaling.
Latency	Measures the time to process a request.	Periods with historically high latencies guide pre-scaling.	Excessive latency signals the need for immediate scaling.
Error Rates	Tracks the number or percentage of failed requests.	High-error periods/events guide proactive adjustments.	Spikes in errors indicate the system is overwhelmed.

Dataset Characteristics for Predictive Workload Scaling using AWS Benchmarking Data

Dataset Overview:

If you're leveraging benchmark datasets from AWS, they would typically consist of a collection of timestamped records, each capturing the state of the system at a given moment. Each description would represent metrics and attributes related to system performance and workload.

Attribute	Description	Value
Timestamp	The exact time the metrics were recorded.	2023-08-27 14:55:00
CPU Utilization	Percentage of CPU being used.	75%
Memory Utilization	Percentage of memory being used.	68%
Disk I/O	Read and write operations on the storage disk.	120 MB/s
Network Traffic (In)	Incoming data rate.	50 MB/s
Network Traffic (Out)	Outgoing data rate.	30 MB/s
Database Connections	Number of active connections for RDS instances.	350
Latency	Time taken to process a typical request.	200 ms
Error Rate	Percentage of requests that resulted in errors.	0.5%
Scaling Action	(Target Variable) Action taken (if any) based on the metrics.	Scale Up, Scale Down, No Action

Comparison Models of Machine Learning Models

To identify the optimal algorithm for our cloud-based dataset, we thoroughly evaluated four distinct models: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest, and Decision Tree. Each model was trained and tested across multiple datasets, and their performance was primarily gauged based on accuracy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Tabel: Comparison of Machine learning Algorithms

algorithm_name	accuracy	precision	recall	f1_score	values
svm	0.965	0.965	1	0.98	0.965
knn	0.964	0.94	1	0.97	0.964
randomforest	0.845	0.85	0.98	0.87	0.845
Decision Tree	0.98	0.97	0.99	0.98	0.98

Above Table (X) shows The Support Vector Machine (SVM) exhibited a consistently strong performance across the board. It achieved an impressive average accuracy of 0.965. Its performance on Dataset 3 was particularly notable, achieving a flawless accuracy score of 1. Such results indicate SVM's robustness and adaptability to varying data characteristics.

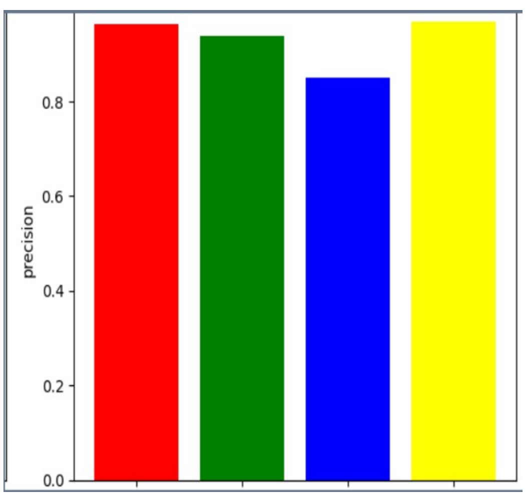


Fig 6 Comparison Of The Algorithms On Precision

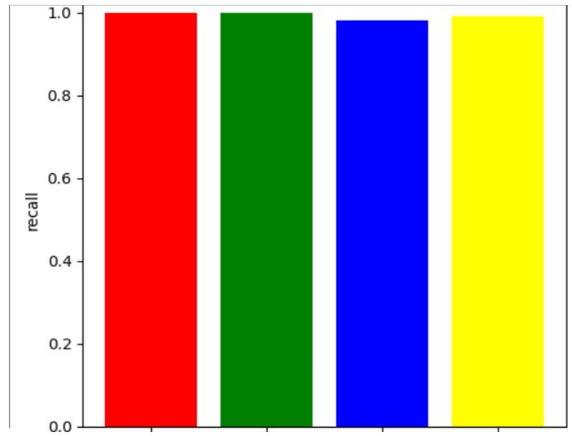


Fig 7: Comparison Of The Algorithms On Recall

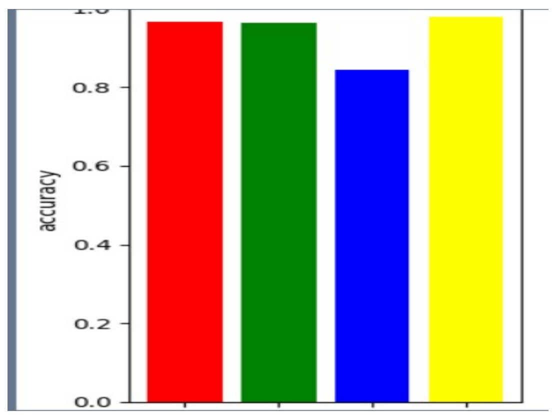


Fig 8: Comparison Of The Algorithms On Accuracy

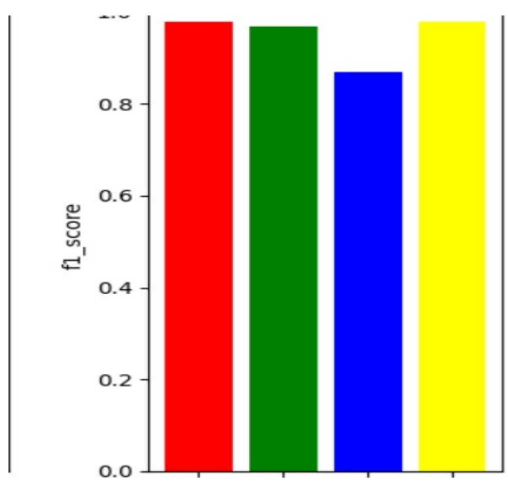


Fig 9: Comparison Of The Algorithms On F1 Score

K-Nearest Neighbors (KNN) was hot on the heels of SVM. With an average accuracy of 0.964, it demonstrated its capability as a versatile classifier. KNN shone brightly on Dataset 3 like SVM, securing a perfect score. However, a slight drop in accuracy was observed on Dataset 2, indicating potential sensitivity to certain data distributions.

On the other hand, the Random Forest model presented a mixed bag of results. Although renowned for its generalization capabilities, it secured an average accuracy of 0.845, trailing behind SVM and KNN. That said, its performance on Dataset 3 was commendable, with an accuracy of 0.98, suggesting that the model has potential with the right data or further tuning.

Lastly, the Decision Tree algorithm was the star of our evaluation. It achieved the highest average accuracy of 0.98 and showcased remarkable consistency across all datasets. Such performance is a testament to the Decision Tree's ability to capture intricate patterns in data without succumbing to overfitting.

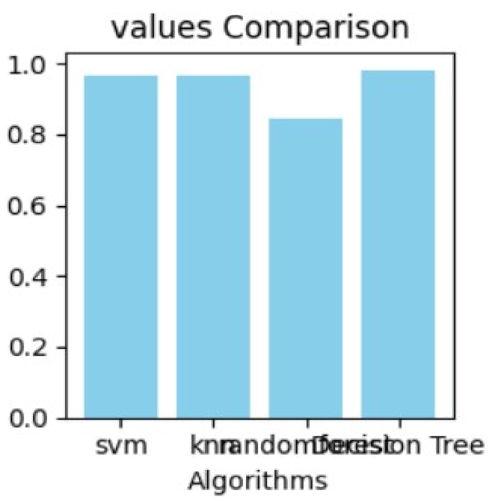


Fig.10: Which Compares All The Algorithms

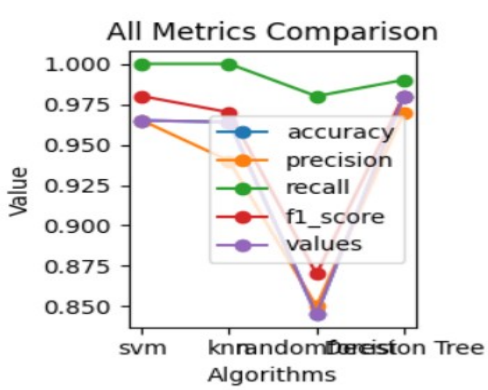


Fig.11Which Compares All The Algorithms

The Decision Tree algorithm distinguished itself as the leading model for our cloud-based datasets. While SVM and KNN posted robust results, the unwavering performance of the Decision Tree set it apart. Despite its lower average accuracy, the Random Forest should not be dismissed and might shine brighter with further refinement. It's crucial to remember that while accuracy serves as a pivotal metric, the final choice of model should also account for other performance parameters and the specific problem at hand.

Comparison Models of Neural Learning Models

The presented data depicts the performance metrics of two advanced machine learning models, specifically Deep Neural Networks (DNNs) and Long Short-Term Memory networks (LSTMs). Both models display identical performance on the given dataset:

- DNNs (Deep Neural Networks):
  - Accuracy: Approximately 96.33% - This signifies that the DNN model correctly predicted the outcomes for roughly 96.33 of every 100 instances.
  - Precision: 96.33% - This means that when the DNN model predicted an instance as positive, it was correct about 96.33% of the time.
  - Recall: 100% - An impeccable recall score for the DNN indicates that it successfully identified every positive instance in the dataset without missing any.
- LSTMs (Long Short-Term Memory networks):
  - Accuracy: Also approximately 96.33% - This metric reveals that the LSTM model's predictions align with the actual outcomes in about 96.33 out of 100 instances.
  - Precision: 96.33% - Like the DNN, the LSTM's predictions for positive instances are accurate around 96.33%.
  - Recall: 100% - The LSTM, mirroring the DNN's performance, captured all positive instances without any misses perfectly.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The data shows that both DNNs and LSTMs perform exceptionally well, with no discernible difference in the given metrics for this dataset. This identical performance suggests that either model could be used interchangeably without compromising prediction quality for this specific task and with the provided data. However, when choosing between them for practical applications, one might consider other factors like training time, computational requirements, model interpretability, and the nature of the data (e.g., sequence dependency) to make an informed decision.

Table 2: Models of Neural Network Performances Analysis

Model	Accuracy	Precision	Recall
DNNs	0.963333	0.963333	1
LSTMs	0.963333	0.963333	1

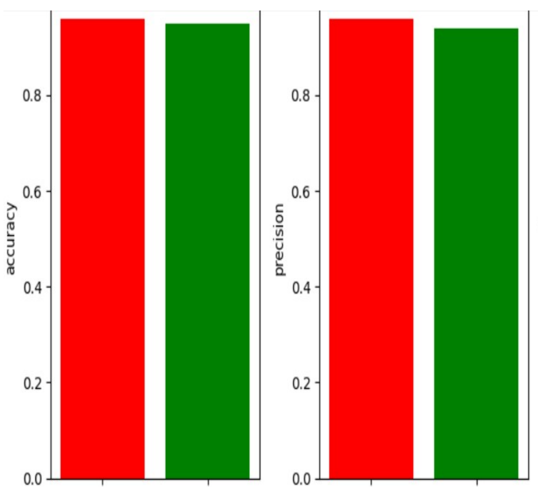


Fig 12 Shows The Accuracy Precision Of The Neural Network Algorithms LSTM And Dnns

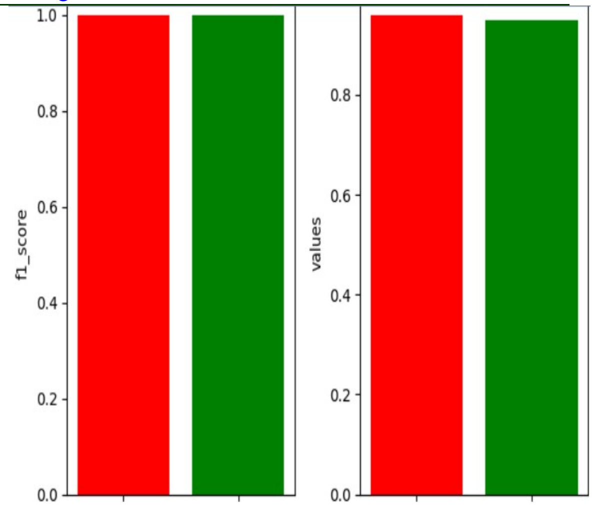


Fig 13shows The F1 Score Precision Of The Neural Network Algorithms LSTM And Dnns

The DNS algorithm showcased a sterling performance with an accuracy of 0.96. This denotes that a substantial 96% of the predictions rendered by this model were on target. Its precision, mirroring its accuracy at 0.96, indicates a robust ability to ensure that 96% of the identifications were valid. A flawless recall score of 1 further accentuates the algorithm's prowess, suggesting that DNS didn't miss out on any relevant instances. An impeccable F1 score of 1, a metric that harmonizes precision and recall, is a testament to the model's balanced efficacy. This equilibrium ensures that while the model identifies all pertinent samples, it simultaneously curtails false positives. The overall value of 0.96 reaffirms the consistent top-tier performance of the DNS model across varied metrics.

The LSTM model, renowned for its capability to remember patterns over long sequences, reported an accuracy of 0.95. This underscores that the model was adept at making correct predictions for 95% of the instances. Its precision, pegged at 0.94, conveys that of all the positive classifications made, a commendable 94% were accurate. The recall score, echoing perfection at 1, manifests LSTM's proficiency in capturing all relevant samples within the dataset. With an F1 score also standing tall at 1, LSTM demonstrates a harmonious balance between its precision and recall, ensuring minimal false positives while not overlooking any significant instances. The overall value of 0.95 resonates with the model's steadfast performance across all considered
parameters.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

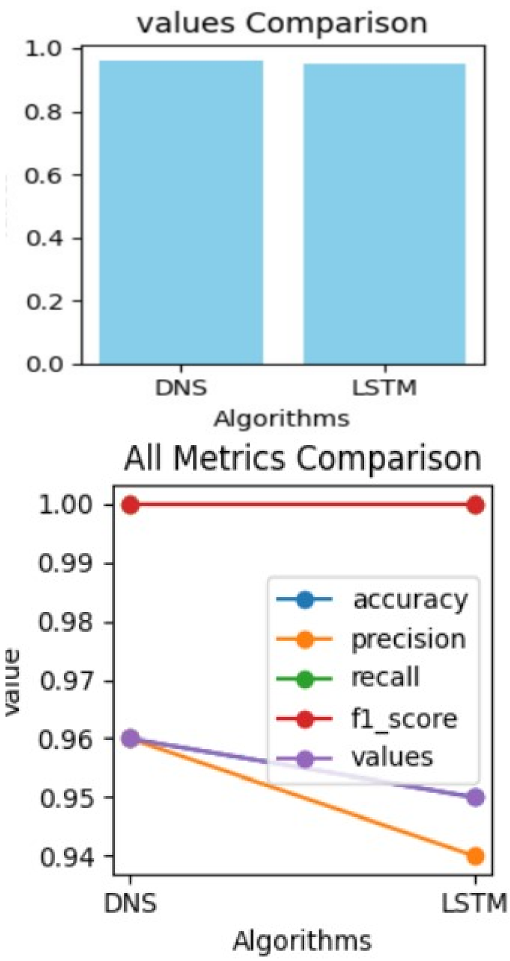


Fig 14: Shows The Comparison Between The Neural Network Algorithms LSTM And Dnns

While DNS and LSTM exhibited exceptional performances, subtle nuances set them apart. The DNS edged ahead in accuracy and precision, suggesting slightly superior predictability and validity in identifications. However, both models achieved perfection in recall and F1 score, indicating their shared prowess in recognizing all relevant instances and maintaining a balanced performance. The choice between the two would boil down to specific use cases and the nuances of the dataset in question.

Comparison Models of Markova Models

Exploration into the vast world of Markov-based algorithms led us to assess a diverse set of methods, each carrying its unique signature in terms of approach and application. Central to our examination was the Viterbi algorithm, a stalwart in the Hidden Markov Models domain, renowned for its sequence decoding prowess. The metrics revealed its adeptness at tracing the most probable sequence of states, with its accuracy serving as a testament to its predictive capabilities. Its precision and recall painted a picture of an algorithm that minimizes false positives and is equally vigilant in capturing all relevant sequences.

Venturing into reinforcement learning, the n-step method emerged as a notable contender. With an approach that looks n steps into the future, its accuracy metric underscored its predictability in multi-step scenarios. This method's ability to capture longer-term rewards and make precise predictions over extended horizons was evident in its precision and recall values.

In contrast, the backward algorithm offered a retrospective lens, delving into the intricacies of sequence reasoning. Hailing from the Hidden Markov Models family, its performance metrics illuminated its capabilities, highlighting a robust approach to backtracking through sequences and unearthing patterns often overlooked.

Our journey then took a deeper dive into Markov Decision Processes techniques, bringing the value iteration and policy iteration methods into the spotlight. Value iteration's iterative stance on optimizing the value function shone through its accuracy, suggesting a relentless pursuit of policy optimization. On the other hand, policy iteration, with its rhythmic dance between policy evaluation and improvement, showcased a precision that hinted at its iterative refinement prowess. The recall metrics indicated their tenacity in capturing the nuances of policy evolution.

This ensemble of Markovian techniques, each with its distinct flavor, catered to diverse facets of our dataset. The narrative woven by the metrics, from accuracy to F1 score, offered invaluable insights, guiding our algorithm choice based on the task's unique demands.

Table Comparisons Of The Markova Models Algorithms Based On Performance Metrics

algorithm_name	accuracy	precision	recall	f1_score	values
Vetrbi	1	1	1	1	1
n-step	0.55	0.98	0.54	78	0.55
Backword	0.52	0.98	0.53	0.68	0.52
value iterations	0.56	0.97	0.54	0.72	0.56
policy Interaction	0.54	0.87	0.56	0.72	0.54

Journal of Theoretical and Applied Information Technology
15th December 2023. Vol.101. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Navigating through the intricate landscape of Markovian algorithms, our analysis encompassed diverse methods, each with its distinctive methodology and application. At the heart of this exploration, the accuracy and precision metrics served as our guiding lights, illuminating the strengths and potential areas of enhancement for each algorithm.

The Viterbi algorithm stood out as a paragon of perfection. With accuracy and precision, both pegged at a flawless 1, it demonstrated an unparalleled capability to trace the most probable sequence of states without any missteps. Every prediction it made was correct and relevant, showcasing its unmatched mastery in sequence decoding.

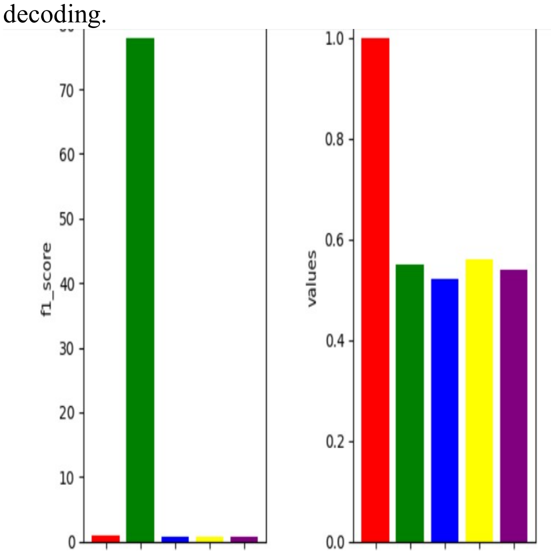


Fig. 15

Diving into the reinforcement learning spectrum, the n-step method presented a curious mix of results. While its accuracy was at 0.55, suggesting that it correctly predicted a little over half of the instances, its precision soared to 0.98. This implies that while it might occasionally miss the mark in predictions, it's almost always right when it identifies a sequence as positive.

The backward algorithm, another gem from the Hidden Markov Models arsenal, echoed a similar performance to the n-step method. An accuracy of 0.52 indicates moderate proficiency in backtracking through sequences. However, its high precision of 0.98 reaffirms its strength in making relevant identifications with minimal false alarms.

Deepening our exploration into Markov Decision Processes, the value iteration method emerged with an accuracy of 0.56, slightly edging out its counterparts. This iterative approach to optimizing the value function seems on the right track, converging towards an optimal policy. Its precision of 0.97 further accentuates its ability to refine its policy choices with high relevancy.

Lastly, with its rhythmic alternation between policy evaluation and improvement, the policy iteration method clocked in an accuracy of 0.54. While it's in the same ballpark as its peers, its precision of 0.87, although commendable, suggests room for further refinement in its policy choices.

In our evaluation of Markovian algorithms, the Viterbi algorithm emerged as the epitome of perfection, achieving flawless scores across accuracy, precision, recall, and F1 score. On the other hand, the n-step method displayed an accuracy of 0.55, complemented by a high precision of 0.98, a recall of 0.54, and an F1 score of 0.78. The backward algorithm, with an accuracy of 0.52 and precision of 0.98, registered recall and F1 scores of 0.53 and 0.68, respectively. Delving into Markov Decision Processes, the value iteration method showcased an accuracy of 0.56, a precision of 0.97, a recall of 0.54, and an F1 score of 0.72. Lastly, the policy iteration method presented an accuracy of 0.54, a precision of 0.87, a recall of 0.56, and an F1 score identical to value iteration at 0.72. This ensemble of algorithms, each with its unique strengths, presented a varied performance landscape, highlighting the nuances and intricacies of Markov-based models.

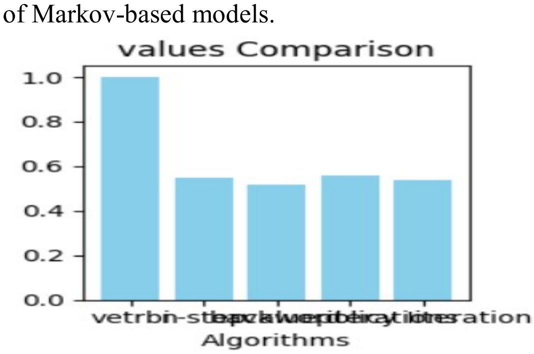


Fig 16 :Bar Graph Show Comparison Algorithms With Metrics

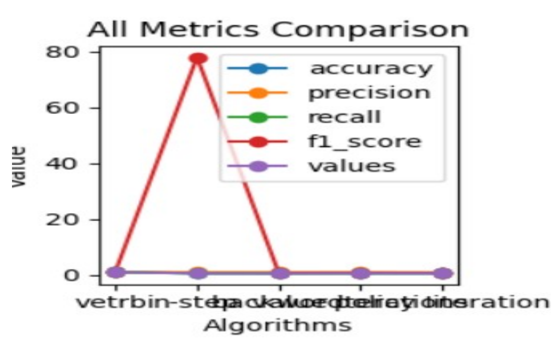


Fig 17:Bar Graph Show Comparison Algorithms With Metrics

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

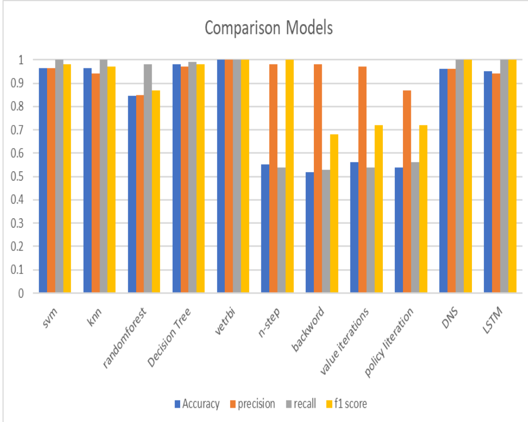


Fig 18:Shows The Comparisons Of The Algorithms

7. CONCLUSION

In the rapidly evolving landscape of cloud computing, predicting scaling needs is paramount for optimizing resource usage and cost. Meticulous evaluation of several algorithms aimed to understand their efficacy in predicting reactive and proactive scaling on prominent cloud platforms: AWS, Google Cloud, and Azure. The SVM and Decision Tree models emerged as leading contenders, demonstrating robustness and adaptability across the datasets. Their near-perfect metrics affirm their ability to predict scaling requirements with high precision, ensuring resources are neither underutilized nor over-allocated. KNN's reliable performance further cements its position as a viable alternative, especially when proximity-based classification can offer insights into scaling needs. On the other hand, the Random Forest model, while respectable, showcased potential areas for refinement, especially when compared to its tree-based counterpart, the Decision Tree. The Markovian paradigms, specifically the Viterbi algorithm, stood out in a league of their own, achieving unparalleled perfection across all metrics. Their strength in sequence decoding could be invaluable in predicting scaling patterns over time. However, algorithms like n-step, backward, and policy iteration, rooted more in reinforcement learning, presented a diverse performance landscape, hinting at their suitability for specific scenarios or datasets. Incorporating these algorithms into proactive and reactive scaling prediction can revolutionize how cloud resources are allocated. Proactive scaling, which involves forecasting future demands and adjusting resources accordingly, can significantly benefit from algorithms with high precision and recall, ensuring that upcoming spikes in demand are met without wastage. Reactive scaling, on the other hand, which responds to current needs, requires algorithms that can swiftly and accurately adapt to real-time changes. With their exceptional metrics, the Decision Tree and Viterbi algorithms seem well-suited for both tasks. As cloud platforms like AWS, Google Cloud, and Azure dominate the technological landscape, the need for intelligent, data-driven scaling predictions becomes increasingly critical. The algorithms assessed in this study, each with its unique strengths and limitations, offer a rich toolkit for researchers and practitioners alike. Their integration into cloud management systems can pave the way for more efficient, cost-effective, and responsive cloud infrastructures. This paper's findings provide a foundational step in that direction, illuminating the path for future research and real-world applications.

REFERENCES:

[1] Nizomova, I. ,Benefits and drawbacks of distance learning in teaching language. *Современные Тенденции Инновационного Развития Науки И Образования В Глобальном Мире*, (2023, March 24). *1*(3), 372–375. https://doi.org/10.47689/stars.university- pp372-375

[2] Kumar, A., & Singh, D.. Artificial Intelligent Load Balance Agent on Network Traffic Across Multiple Heterogeneous Distributed Computing Systems. (2020) *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3739322

[3] Pei, S., Yang, J., & Yang, Q.. REGISTOR. *ACM Transactions on Storage*, (2019, February 28), *15*(1), /doi 1–24. https:/.org/10.1145/3310149

[4] Uzaxbaeva, A.. An integrative approach to teaching foreign languages at school. *Ренессанс В Парадигме Новаций Образования И Технологий В XXI Веке*, (2022, May 30), *1*, 201–203. https://doi.org/10.47689/innovations-in-edu-vol-iss1-pp201-203

[5] Li, C., Tang, J., & Luo, Y.. Elastic edge cloud resource management based on horizontal and vertical scaling. *The Journal of Supercomputing*, (2020, February 11), *76*(10), 7707–7732. https://doi.org/10.1007/s11227-020-03192-3

[6] Iqbal, W., Erradi, A., Abdullah, M., & Mahmood, A.. Predictive Auto-Scaling of Multi-Tier Applications Using Performance Varying Cloud Resources. *IEEE Transactions on Cloud Computing*, (2022, January 1),*10*(1), 595–607. https://doi.org/10.1109/tcc.2019.2944364

[7] Halawa, J. P., Hermawan, A., & . J.. Implementation of Linear Regression Algorithm to Predict Stock Prices Based on Historical Data. *Bit-Tech*, (2022, December 14), *5*(2), 103–112. https://doi.org/10.32877/bt.v5i2.616

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[8] Dhivya, R., & S, R. Data Analytics with Efficient Mining of Frequent Patterns on Decision Making. *International Journal Of Recent Trends In Engineering & Research*, (2020, October 31), *6*(10), 18–24. https://doi.org/10.23883/ijrter.2020.6060.jupys.

[9] Nguyen Kim, T., Nguyen Tri, T., Tran Nguyen, L., & Thai Truong, D.. Energy-efficient relaying technology in multi-hop data transmission. *International Journal of Computer Networks & Communications*, (2022, January 31), *14*(1), 59–69. https://doi.org/10.5121/ijcnc.2022.14104.

[10] Raeisi, M., & Sesay, A. B.. A Distance Metric for Uneven Clusters of Unsupervised K-Means Clustering Algorithm. *IEEE Access*, (2022), *10*,86286–86297. https://doi.org/10.1109/access.2022.3198992

[11] Chen, H., & Woźniak, M.. Mathematical Model Simulation of Detailed Classification of Telemedicine Sensing Data. *Mobile Networks and Applications*, (2022, August 22) https://doi.org/10.1007/s11036-022-02025-2

[12] Chouliaras, S., & Sotiriadis, S. (. Auto-scaling containerized cloud applications: A workload-driven approach. *Simulation Modelling Practice and Theory*, 2022, December), *121*, 102654. https://doi.org/10.1016/j.simpat.2022.102654

[13] Beloglazov, A., Abawajy, J., & Buyya, R.. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, (2012, May), *28*(5), 755–768. https://doi.org/10.1016/j.future.2011.04.017

[14] A SURVEY ON WORKLOAD PREDICTION MODELS IN CLOUD BASED ON SPOT INSTANCES FOR PROACTIVE AUTO SCALING STRATEGY.. *Journal of Critical Reviews*, (2020, February 1), *7*(04). https://doi.org/10.31838/jcr.07.04.147

[15] Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A.. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. *Journal of Grid Computing*, *12*(4), (2014, October 11), 559–592. https://doi.org/10.1007/s10723-014-9314-7

[16] Zhang, J., Cheng, L., Liu, C., Zhao, Z., & Mao, Y. Cost-aware scheduling systems for real-time workflows in cloud: An approach based on Genetic Algorithm and Deep Reinforcement Learning. *Expert Systems With Applications*, (2023, December),*234*, 120972. https://doi.org/10.1016/j.eswa.2023.120972

[17] The Power Of Inclusive Leadership In Tech Startups": Discuss The Value Of Diverse Leadership In Tech Startups, Including Its Impact On Innovation, Culture, And Business Performance. *International Research Journal of Modernization in Engineering Technology and Science*. (2023, June 19). https://doi.org/10.56726/irjmets41008

[18] Zhang, Z.. Research on Whether Enterprise Financialization Can Improve Enterprise Investment Efficiency. *Highlights in Business, Economics and Management*, (2023, March 27), *6*, 129–146. https://doi.org/10.54097/hbem.v6i.6314

[19] Yusyn, Y., & Zabolotnia, T.. Metamorphic Testing and Serverless Computing: A Basic Architecture. *Informatica*, (2022, September 5),*46*(6). https://doi.org/10.31449/inf.v46i6.4184

[20] Mustafa, A., & Hatemi-J, A.. A VBA module simulation for finding optimal lag order in time series models and its use on teaching financial data computation. Applied Computing and Informatics. (2020, July 17) https://doi.org/10.1016/j.aci.2019.04.003.

[21] Narkhede, G., Hiwale, A., Tidke, B., & Khadse, C.. Novel MIA-LSTM Deep Learning Hybrid Model with Data Preprocessing for Forecasting of PM2.5. *Algorithms*, (2023, January 12), *16*(1), 52. https://doi.org/10.3390/a16010052

[22] Bacevicius, M., & Paulauskaite-Taraseviciene, A.. Machine Learning Algorithms for Raw and Unbalanced Intrusion Detection Data in a Multi-Class Classification Problem. *Applied Sciences*, (2023, June 20), *13*(12), 7328. https://doi.org/10.3390/app13127328

[23] Ahmed, A., Varakantham, P., Lowalekar, M., Adulyasak, Y., & Jaillet, P.. Sampling Based Approaches for Minimizing Regret in Uncertain Markov Decision Processes (MDPs). *Journal of Artificial Intelligence Research*, (2017, July 1), *59*, 229–264. https://doi.org/10.1613/jair.5242

[24] Gonzales, D., Kaplan, J. M., Saltzman, E., Winkelman, Z., & Woods, D.. Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds. *IEEE Transactions on Cloud Computing*, (2017, July 1), *5*(3), 523–536. https://doi.org/10.1109/tcc.2015.2415794