International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

AI-Driven Dynamic Scaling Frameworks for Resilient Microservices in Cloud-Based E-Commerce Platforms

Dr Somasundaram Krishnan

Professor, Department of Computer Science and Engineering, Sri Muthukumaran Institute of Technology, Chennai, India

ABSTRACT: Microservices play a crucial role in delivering fast, reliable and scalable digital commerce experiences. But the rules-based, manual/automatic resource scaling can lead to over-provisioning resources, failing to meet all combinations of resources demand or simply being too slow to meet demand needs. The proposed research paper proposes an idea of how to create Resilient Microservices for Cloud Based ecommerce Application with AI based Dynamic Scaling Framework. Ensuring the availability and performance of the system is addressed via the framework by means of real-time monitoring, workload prediction, anomaly detection, intelligent allocation of resources and feedback based optimization. Through a machine learning model, the framework understands these characteristics in these historical workloads and uses these to forecast future trends in workload, CPU/memory usage, transaction query latency, traffic volume etc., depending on which services are in use. Based on these predictions, an automatic scaling process which is based on a container orchestration system like Kubernetes can scale up or down by horizontally or vertically scaling the microservices. Also part of the proposed framework are various resilience aspects like service health monitoring, fault detection, load balancing, circuit breaking and self-healing to minimize service downtimes and keep services up and running. The paper highlights the advantages of AI for scaling, including faster response times, optimized cloud resource usage, increased fault tolerance, and reduced costs. When not only is it capturing data over time but also leveraging insights from data to dynamically adjust the scaling, it's an appealing solution for complex and highly trafficked microservices that are considered important for meeting enterprise objectives. This research also contributes to the advocacy of intelligence cloud administration to build up scalable and resilient e-commerce structure which are cost efficient.

KEYWORDS: AI-driven autoscaling, dynamic scaling, microservices, cloud computing, e-commerce platforms, Kubernetes, workload prediction, resilience, resource optimization, fault tolerance.

I. INTRODUCTION

Cloud-based ecommerce solutions provide the crucial foundation for e-commerce in today's digital age, allowing businesses to offer products and services to their customers via highly responsive, scalable and available online systems. Today's ecommerce systems include a number of systems which change their loads as part of the product search, login, shopping cart, ecommerce payment processor, customer support, product recommendation and systems, order management system and product promotions are but some of the countless systems that need to be managed. These changes in users within a short span of time might be because of events such as a “flash sale” or festive sales, launching of a new product or because of an ad hoc surge in users. Without the efficient scalability it can cause latency concerns, performing transactions, service interruptions, and lose the confidence from customers. Hence those are critical when it comes to cloud based ecommerce systems, scalability and resilience.

A microservices architecture has become very popular over the past couple of years while building any large e-commerce applications, it's dividing and conquering a monolithic architecture into smaller deployable services. The microservices are each associated with a specific business process such as User authentication, Product catalog, Order fulfilment, Inventory management etc. The design is Modular, allowing for flexibility in the system, maintainability and deployment as continuous services. With microservices, though, comes some new operational hurdles. Different services are interdependent of each other through apis and failure / overload in any of the service would impact the performance of the platform. As an example, in a sale scheme, when the payment service is not functioning well due to the high numbers of customers, customers cannot do any sales even if the product catalog and its shopping cart

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

application works well. So resilience for microservices isn't only about scaling the whole application, but scaling out the services while taking into account their load and criticality.

Cloud Computing offers elastic infrastructure resources – virtual machines, containers, serverless functions, container orchestration – via Kubernetes. These technologies allow to add or remove the necessary resources as needed. One traditional method to scaling was to follow a set of rules, e.g. if the CPU utilisation is above 80%, scale out. This measure can be useful, but it is a reactive measure, pro-active is better. Taking a rule based approach to scaling is 'reacting' to the resource pressure, in the general. This latency can sometimes cause a temporary delay in service in fast-changing eCommerce industries, longer response times and a poor user experience. Further, a single mathematical threshold is not enough to represent the various characteristics of complicated workloads when it comes to memory footprints, networking, request queues, transaction rates, relationships between services/applications, and/or time of year.

The answer is a more flexible application that is driven by AI. AI-based scaling can leverage significant amounts of data collected from physical infrastructure and applications to anticipate infrastructure trends and the potential scenarios that could impact workload performance, and make changes accordingly. With machine learning models, trends in user traffic, the frequency of transactions, server usage, response time and error rates can be understood. Given these predictions, the system can be scaled up or down depending on whether a service that's getting a great deal of attention is upping or downing the number of containers involved in the system, or add compute resources in anticipation of a foreseen performance cap. Such a predicting ability can prove pretty useful to eCommerce enterprises that could see their need rising quickly while promoting deals or even during the holidays.

Self-healing of the system with the help of AI-based dynamic scaling framework is achieved to handle the anomaly detection and enhance the system resilience. Data analysis and comparing the existing metrics with the expected behaviour of the system, can enable AI models to identify the abnormal behavior of the system – microservices crash, traffic suddenly surges, network is congested, Database is slowing down, etc. When this anomaly is identified, the framework can react to it and redeploy the failed containers, move the traffic to the healthy instances, increase or decrease the services affected and even notify the system administrators. This will decrease the instances of people handling it and increase continuity of services.

The proposed study is aimed at developing an AI-Supported Dynamic Scaling Scheme for the Resilient Microservices in Cloud-Based E-commerce Applications. They comprise of real-time monitoring, workload prediction, intelligent decision making, container orchestration, self-healing and Operations. First, it gathers all the metrics of the system within services, such as CPU usage, memory, requests, latency of the requests, error-rate and load of transactions. These parameters are then used by machine learning algorithms to make predictions upon request, and to alert of possible performance issues that might arise in the near future. The scaling decision module, based on the prediction results, decides if horizontal or vertical scaling is needed or load balancing and/or service recovery are needed. Then these decisions are acted upon using the orchestration layer, using cloud-native infrastructure—Kubernetes.

This study is important as the e-commerce should be cost effective with a high level of performance. Cloud resources, if provisioned, end up becoming labor costs and if provisioned to be under provisioned, will degrade the service quality and business end up losing money. AI tools can help with resource allocation based on current and future needs, as well as the cost of the resources, performance and costs. It enables it to continue to have adequate resource for the delivery of essential resources at times when there is high demand and it prevents wasting resources for the other parts of the time.

Moreover, the analysis will also contribute to intelligent cloud operation (also referred to as AIOPs) market. The combination of AI and Cloud native Microservices is the building block on which automatic infrastructure are managed in the proposed framework. In this way it reduces manual configuration to a minimum and systems can continually self adapt to a changing business/technical situation. Every second counts with ecommerce customers' satisfaction and selling conversion, and this type of smart automation makes it a game-changer for ecommerce businesses.

Lastly, dynamic scaling is not simply a technical demand, but it's a tactical requirement in these days, especially in e-commerce organizations, and especially in cloud configurations. The more unpredictable and complex workloads proved to be a problem that could only be coped with using the traditional methods for scalability. By forecasting demand and proactive detection and adjustment in response to abnormal situations, an AI-driven system could be instrumental in enhancing scalability, resilience, resource utilisation and the CX. Given the context of this situation,

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

this research aims to discuss how to effectively apply AI to a microservice-architecture e-commerce system to achieve, as an underlying cloud infrastructure, adaptability, reliability and cost-effectiveness in any systems.

II. RELATED WORK

Even in recent times, the scaling in and out of cloud systems driven by application services is not anymore just a threshold but is done through intelligent scaling taking into account service-level-objectives (SLOs) and workload adaption. Applications are broken up into several loosely coupled microservices, where some microservices have a high workload, some have a low workload and some are latency sensitive. In a world of static provisioning, it can be inefficient especially in cloud ecommerce applications which generally suffer from a very rapid growth of its ecommerce traffic, e.g. during sales or during flash-demand days in particular seasons.

The work of Qiu et al. was also novel in delivering intelligent fine-grained management of services and resources to the SLO level of microservices using the virtualization layer–based FIRM (1). FIRM would like to identify performance-bottlenecks in a micro-scale and make allocations decisions for microservices to meet the SLOs. Much of it is the control of resources/per-service – fighting coarse application-level scaling. This is very important when it comes to resilient e-commerce systems as user-facing services like product search, adding items to the cart, processing the payment and providing recommendations may not all behave the same when under pressure. This would be a nice use-case for fine-grained SLO driven scaling to prevent over provisioning and mandating latency guarantees.

Dynamic scaling of ecommerce was finally explored by Subramani and was focussed on microservices, latency, compliance and resilience [2]. This work is relevant, as well as directly related to the current work, in the sense that it is a technique of autoscaling that is integrated in the functioning of an eCommerce site. The secure transaction processing, together with the response time (regulatory requirements and high availability are related to that) are peculiarities to achieve e-commerce systems and are a challenge to design, build and operate if compared with the typical cloud app. Performance optimisation is not the only problem with dynamic scaling: it's an issue of resilience and compliance, too, according to the study. This view underscores the importance of AI-powered systems which can predict shifts in demand, optimise staffing, and keep deals alive in times of sudden demand surges.

For the problem of SLOs Park et al proposed a proactive resource scaling framework (GRAF for scalable microservices) which uses a graph neural network and assesses relations between nodes, with nodes representing microservices [3]. The interesting thing to look at the GRAF is that Microservices are tightly coupled, if one of the resources is heavily used then it can cause ripple effect with other microservices across the service chain. GRAF is different from standalone scale models, in that it makes it possible to model behaviours based on dependency aware microservice relationships. If you have certain other services that are directly connected with your ecommerce site, such as inventory, payment gateway, confirmation and authentication then it is really beneficial. Proactively identifying impacts related to these upstream and downstream conditions before the point of SLO violation can be accomplished by using the graph.

Yu et al. proposed an online learning automatic microscaler, called Microscaler [4]; it is a part of the literature, because you could make scaling-choices based on system behaviour, continuously. Such learning is appropriate for workload fluctuating over time occurring, like in cloud retail applications. When the customers are browsing at regular time of the year, for example, they're doing something else than when they're browsing at holiday times or atlimited time offers. This implies that it is possible to do more than achieve higher responsiveness with fixed rule scaling using the learning based auto-scaling

The authors of Choi et al. suggest using an active autoscaling mechanism for microservice chains named pHPA [5]. In the classic horizontal pod autoscaler, scaling action may only start once the scaling configurations hit thresholds–and it can take time for that to occur. The problem is that pHPA can be used to predict upcoming demand, and scale microservice chains before they start to impact on the performance. Latency is very important in ecommerce and can cause a loss of trust and greater cart abandonment if shoppers are waiting for checkout or to view the payment information.

Gias et al. have proposed an autoscaling of the microservices using the model-driven approach (ATOM) [6]. A key focus of ATOM is to gain insight into scaling decisions made from application performance models. It's useful when dealing with model driven autoscaling as it shows the context of how work load and resource allocation works and how

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

such changes interact with response time. These types of models can be applied for e-commerce applications to predict the number of times (with desirable latency) that a varying traffic workload might need to be executed.

To scale, authors of this paper, Kwan et al., have proposed the concept of Poisson Hybridisation of dockerized micro-services to scale in Cloud data-centres that are named HyScale [7]. This paper takes into account not just CPU/memory restrictions, but also networking concerns. Overheads of communication and latency in the network are major concerns, since an application can be a set of some communication services. In ecommerce scenarios, for example, 'network aware' scalabilty is also necessary, as services for product search, product recommendation, payments and order creation also need to communicate with each other to complete an order.

Tamanah (Baarzi and Kesidis) was an initiative to downsize and effectively plan microservices [8]. Their work is relevant because if the cloud isn't provisioned enough the latency would be violated, and if it's provisioned more, would costs the customer more for their cloud. The problem of SHOWAR can be stated as: How to satisfy the following contradictory objectives: Efficient utilization of resources, and: Guarantee of performances. Smooth resource scheduling is critical for eCommerce businesses since they can experience extremely varied structure habits and areas of profits could be impacted from legislative charges for over usage of resources, when utilizing cloud.

Balla et al. conducted study on adaptive scaling of the pods with an application deployed on Kubernetes [9]. AutoScalers are now an integral part of the cloud paradigm as Kubernetes is increasingly becoming the orchestration tool for deployment of microservices. Their work helps to understand the scaling behavior on the pod level, and reminds of adaptations that might be necessary when scaling behavior to the work load. So, there should be a smart scaling system which understands and is in sync with kubernetes.

Liu et al. have come up with the idea of fuzzy-based autoscaler for web applications in cloud environments [10]. If uncertainty arises and if thresholds are not crisp or there are multiple thresholds being actions when the scaling process occurs, perhaps fuzzy logic will be useful. It's crucial when autoscaling with AI since the workload of e-commerce applications are somewhat erratic, and static thresholds will not raise or lower based on real performance. Fuzzy based control provides a flexible control mechanism which balances between performance and the resources.

Finally, Al-Dhuraibi et al., discussed in detail the fundamentals concept of elasticity in cloud computing and the methods and challenges it faces [11]. Their research findings lay the groundwork to the general theory of elasticity (i.e. automatic provisioning, resource adaptation and quality of service maintainment). The literature analyzed shows that such a change has happened – from Reactive elasticity all the way to smart, proactive and dependency-aware SLO-driven scalability. However, one solution that can address all of the above, including microservice interdependency, latency, resilience, forecasts of workloads, compliance and cost, for ecommerce applications in the cloud is still needed.

III. AI-DRIVEN DYNAMIC SCALING FRAMEWORK FOR RESILIENT MICROSERVICES

Besides availability, scalability and cost of such platforms, cloud-based e-commerce platforms are also dealing with reliability and resiliency issues of microservices.In the context of e-commerce appropriate for this eService, AI-Driven Dynamic Scaling Framework is an attempt to make cloud based e-commerce platforms scalable, reliable, available and affordable, by dynamically scaling resources to make them more resilient and dependable. The architecture is designed to be based on AI, orchestration on the cloud, real-time monitoring, predictive analytics, and ability to self-heal for intelligent resources management. As compared to the existing threshold based auto scaling of microservices, the proposed solution is predictive & smart without affecting end user, while keeping the system from being overloaded.
The microservices application layer can be use to manage services and provide data collection, prediction (AI layer), decisionmaking layer, orchestration and scaling layer, and resilience and optimization layer. There is a distinct role for each layer and feedback between the layers.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

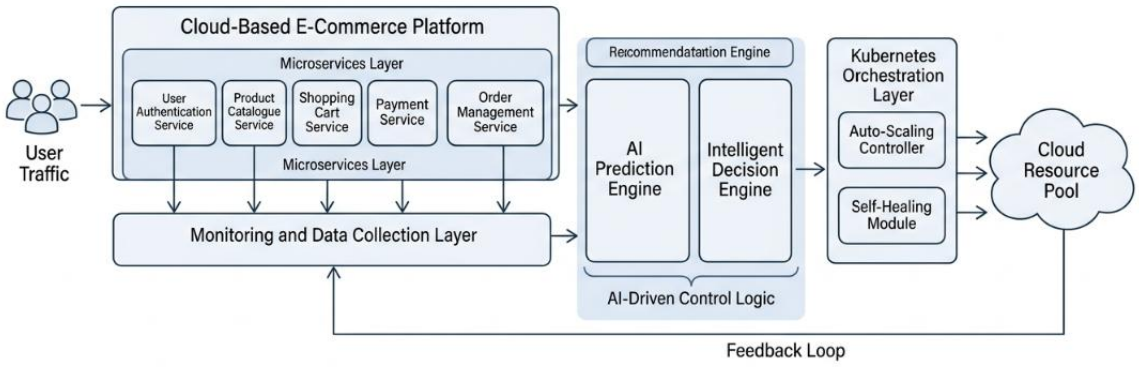

Figure 1: Overall Architecture of the AI-Driven Dynamic Scaling Framework

3.1 Microservices Application Layer

Base layer: all of the required e-commerce microservices. User authentication, product catalogue, search, shopping cart, payment gateway, order management, inventory control, recommendation system and notification service and customer support service. Each Microservice in a container. Allows to scale up/down of each services independently according to the workload.

For example, during a holiday promotion for certain products, there may be a greater number of requests for the product search and catalogue service during the "browse" phase than for the payment service. When a customer attempts to make a payment, then taps into a service for checkout, there could be a 'burst' on the cart and payment services. The drawback with this is that to scale up the application at all levels is not efficient. Proposed framework is service specific scaling and that is because only when a service is potentially about to become overloaded, or is overloaded, it should get more resources.

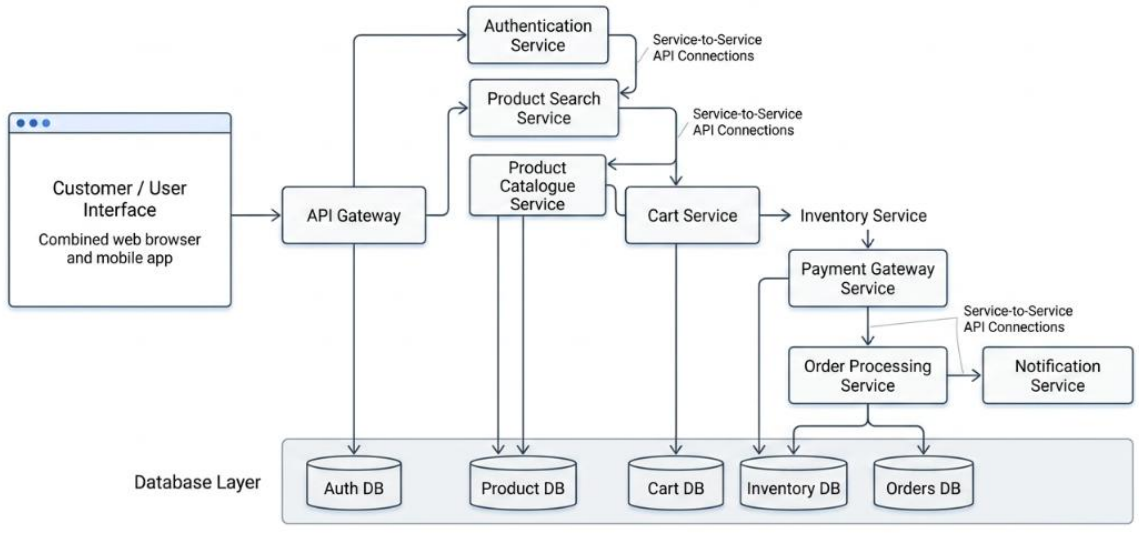

Figure 2: Microservices Communication Workflow in E-Commerce Platform

3.2 Monitoring and Data Collection Layer

All operational information of all the microservices is continuously fed in the second layer. CPU time, memory receive, disk in/out, network traffic, number of incoming requests, response time, error rates, queues, number of transactions, and container health & service dependency metrics. Active users, add-to-cart, checkout attempts, number payment failures, order completion rate, among other business metrics, are also tracked.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

This is as it plays a vital role due to the accuracy and timely data on which AI based scaling is dependent. Data collected is then used in a centralized monitoring system where it is data cleaned, data normalized and placed onto a data repository for near real-time data analysis. Can include of monitoring/observability tools like: Prometheus, Grafana, ElasticSearch, Fluentd, etc. and cloud-native observability platforms. The aim is not just to look at how the system is working, but also to establish a solid data basis for intelligent prediction and decision making.

3.3 AI-Based Workload Prediction Layer

The third layer is solution focused, dealing with the prediction of the future workload conditions with AI and ML models. Traditional models scale conventions are to just add more containers as soon as your application's CPU hits a threshold (e.g. 80%). However, e-commerce workloads are extremely dynamic, and, frequently, complex. There's a lot that can be done for season, marketing, social media whats hot and whats selling, etc that can be done to help get more visitors.

AI Prediction Layer, Sends receives and real time data, provides Prediction. Machine learning algorithms such as Long Short-Term Memory (LSTM) networks, Random Forest (RF), Gradient Boosting (GB) or reinforcement learning algorithms can be employed depending on the workload to be dealt with. For example, time-series forecasting models can be used to predict short-term traffic spikes and anomaly detection models can be used to detect any unusual traffic — for example, payment system delays, abrupt and severe service disruptions or bot traffic.

This layer obtains the predictions for each microservice's workload. For example, it can forecast that after 10 minutes, the product catalogue service will receive 40 per cent more calls than the payment service which is needed when checking out. The system dynamically scales proactively based on these forecasts, when it is not at overload.

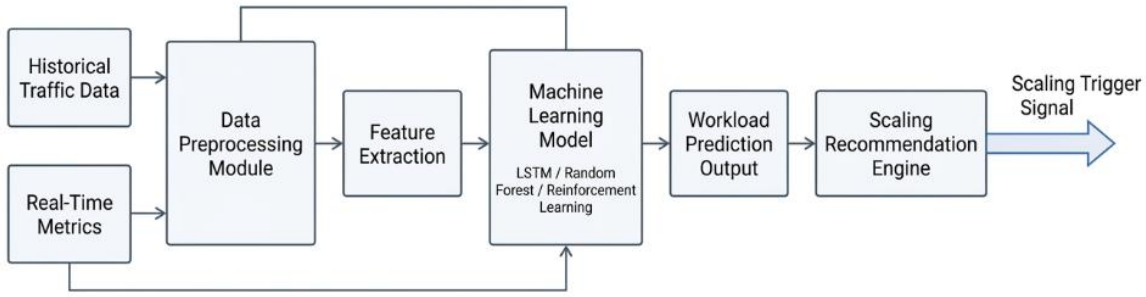

Figure 3: AI-Based Predictive Scaling Workflow

3.4 Intelligent Decision-Making Layer

The fourth layer provides interpretation of the AI's 'projections' to an expectation that is suitable for scaling. It examines all current system metrics, workload, services priority, cost constraint and performance requirements. The decision making module will decide whether or not to scale vertically, horizontally scale the traffic or to recover from a failure or to throttle traffic.

Adding/removing container instances is called horizontal scaling. This facility is beneficial in case there are more requests for it. Vertical Scaling is when a service is scaled with an increase in memory or CPU. Load redistribution occurs when it spread user's service demands from the overloaded service instance to healthy service instances. However, in the framework, rate limiting will also be used, to ensure a certain service is not overwhelmed.

The decision making paint is utilized as well, whereby the business importance is also taken into account. For example, a payment service or a checkout service, is one that, otherwise, would not result in a gain or loss of money. While that's important, a suggestion machine may run on a strength of zero in a high-volume time of the day. Thus priority aware scaling – such that business critical services will get resources first – is in effect.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

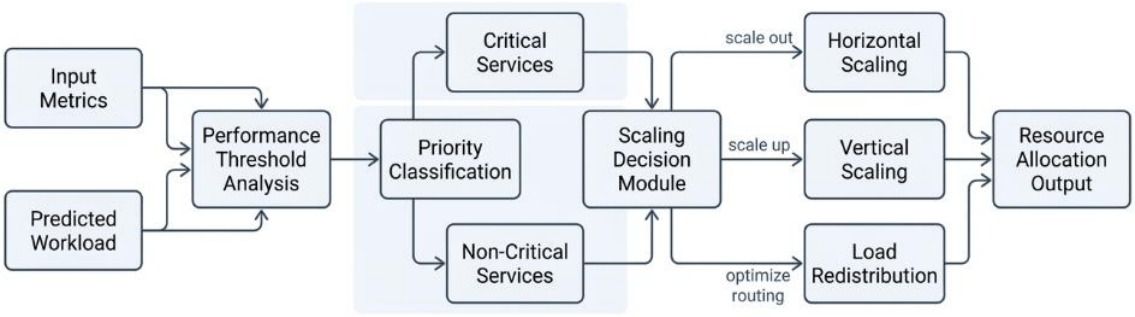

Figure 4: Intelligent Decision-Making and Auto-Scaling Process

3.5 Orchestration and Scaling Layer

Operation is done at the cloud-native operations level (5th layer) with cloud-native orchestration tools. The attributes such as containers, service discovery, load balancing, auto-scaling, rolling update and self healing make Kubernetes an ideal choice to achieve that. It can be used with any other custom Autoscaler controllers in AI services and integrates with the Horizontal Pod Autoscaler, Vertical Pod Autoscaler and Cluster Autoscaler.

If the decision making layer has determined that this new scaling is in need, the orchestration layer dynamically creates leftovers of services, eliminates unwanted containers, reallocates resources or even moves work loads from node to node. As an example, the AI model could predict a surge in cashier cashouts at a certain location and kickoff the orchestration layer to create an increase in the number of pods running in the area.

This layer also plays a role in making sure that there is no effect on the users' activities when scaling activities are performed. Fineness of sessions is a must for e-commerce Web sites, as the customer gets irritated and may cancel orders if the Web site flakes when they're going through the order process. So, scaling should be seamless, precise and be amenable to load balancers, API gateways, and service mesh technologies.

3.6 Resilience and Self-Healing Layer

6th layer is called resilience of the system, where the concept of fault detection and fault recovery processes is applied, completely automatically. Most failure cases of the microservices happen due to software failures, microservice container failures, microservice latency in networks, database overload, microservice memory leaks and even 3rd party payment gateway failure cases. There are periodic health checks among the framework, and also, issues in the services are identified by leveraging the use of AI.

When a microservice becomes "unhealthy," the self-healing mechanism can have a failed microservice container restarted or a fresh instance created in the same zone, re-route traffic, fence off and isolate the microservice or invoke fallback. On the other hand, if the recommendation service is not operating (at the site for some reason), he/she can select "best seller" (as a site of some sort) for some time. If the payment service experiences problems, this traffic can be rerouted to another payment gateway.

This layer is used to reduce the downtime and improve user experience. It also can be used to restrict propagation of failure, one of the main issues in microservices architectures. Various services are interdependent on each other and any service failure can result in service failures across the platform. Protection properties increase the availability of the critical services and/or decrease the exposure of the less stable elements by the resilience levels in order to avoid these types of failures.

3.7 Feedback and Continuous Optimization

The framework is an on-going circle of feedback. Measuring the impact of each scaling action is an automatic follow-up, which will measure the impact on performance, costs and reliability. New operational data is provided to the AI models, which can help it be more accurate in future predictions. By reinforcement learning, it is possible to improve scaling policies continuously, based on successes or failures in making the decision.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

As the system over-provisoned the containers, for example in a relatively small traffic "spike", the feedback module will log it as so. The degree of under provisioning will be established by the need to install the system at a late life with resize, and degraded response time in the system. Scaling (different workloads) is learnt over time with the best scalable solution being the most successful.

IV. PERFORMANCE EVALUATION AND VALIDATION OF THE AI-DRIVEN SCALING FRAMEWORK

4.1 Evaluation Setup

The proposed framework performance is tested over the cloud based ecommerce microservices deployed via container orchestration. The major services in the test environment are the product catalogue, user authentication, cart, payment, order processing and recommending. This system is subject to simulation where regular traffic, traffic peaks, and large surges of traffic and even partial failure of the service are dealt with. The AI solution is compared to a conventional threshold-based auto-scaling policy to validate its properties in the areas of scalability, responsiveness, resilience and resource usage.

4.2 Evaluation Metrics

A set of performance measures are used to ascertain the success of the framework. Among those are the average response time, request throughput, CPU utilization, memory usage, scaling delay and service available, request error rate and cost of the cloud resources. Response time – speed of responding from the platform, of the users' request. Please Note: Throughput is expressed as the number of requests per second. Scaling delay is the time required to scale up or down, when the change in workload is detected. Availability indicates which is the ability of your system to keep running in case of failure or increment of traffic.

4.3 Scalability Performance

While workloads may increase over the years, the planned AI solution can provide more sustainable solutions that predict workload is increasing instead of being a burden. The framework takes into account traffic patterns and enforces creating more copies of microservice at a pre-methodical level rather than waiting for an X% of CPU/memory utilization. This will reduce reaction delays in the event of busy times and flash sales. Service specific scaling is also helpful, so that only the needed services can be scaled up: for example, scaling up of the cart service and payment service can be correlated to a specific time at which customers are maximum on the checkout.

4.4 Resilience and Fault Tolerance

It is a framework that takes advantage of abnormal behavior detection and self-healing to enhance systems' resiliency. When one of the microservices slow down or no longer exists, the system restarts the failed containers, the reroute or provision new instances. This guarantees that the failures of other services won't affect this service. That means even if a part of the system can't operate, vital functions of ecommerce, such as log-in, check-out and payment processing can still be performed.

4.5 Resource Utilization and Cost Efficiency

Scaling powered by AI ensures optimal resource utilization, eliminating under provisioning and over provisioning situations. So when offset times are not used, the extra containers are removed, thereby reducing a cost for cloud resources. During peak demand, resources are only supplied to those services which require them. This is a balance allocation to help with cost effectiveness and are still able to maintain the quality of services.

4.6 Overall Evaluation Outcome

According to the results of the complete evaluation, the proposed framework for dynamic scaling with the use of AI is beneficial compared to the conventional rule-based scaling. It lowers latency, enhances availability, boosts throughput and maximises the cloud resource usage. Hence its usefulness may be well applied to statistical eCommerce systems using a microservices based platform residing on the cloud where an intelligent, resilient and cost efficient microservice management is essential.

V. CONCLUSION AND FUTURE WORK

Analyzing the results obtained by conducting the research one can see that the dynamic scaling obtained by utilizing the AI will be a successful solution to the improvements of microservices system's resilience, scalability and cost effectiveness when being deployed on cloud environment used for the provisioning of e-commerce services. The traditional auto-scaling approaches that run off thresholds (and triggered by spikes in workloads) work okay to scale in

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 2, March – April 2025 ||

DOI: 10.15680/IJCTECE.2025.0802013

response to the growth, but don't have much skill in understanding how to respond when the workload drops—how much does the system need to scale down during a sudden dip in the number of flash sales offered, or during a seasonal sale, or some other unforeseen spike in traffic? At the same time, the proposed framework has the following properties when it comes to proactively and services scaling: real-time monitoring, workload prediction, intelligent decision making, container orchestration and self-healing.

The framework makes the e-commerce perform better since it can predict the e-commerce's demand even before it gets degraded. This allows to allocate resources to the important microservices like payment, order, cart and authentication based on their workload priority. This reduces latency, improves overall availability and prevents cascading failures and improves overall customer experience. For the case of performance vs cloud resource cost in kubernetes provides AI based orchestration to optimise the provisioning of the cloud resources, reducing the need for over or under providing resources.

Further works can be done to improve the proposed approach and use it on a real-time object deployment on e-commerce environment by using real traffic datasets. More advanced Deep Learning and Reinforcement Learning applications would be interesting to investigate for better predictions and analysis of adaptive scaling aspects. In addition to multi-cloud deployments and hybrid clouds can be explored to show the portability, fault tolerance and inter-cloud and inter-vendor scaling in future deployments. Additionally, security-aware scaling can take place as well, acknowledging any irregular traffic as a result of either bot (or DDoS) attacks or fraudulent transactions.

One of the future research directions is to describe the particular choices of scaling decisions so as to reach system admins with an understanding of the choice made by AI models. This will increase the transparency, trust and control of operations. Some research opportunities may be the design of such systems that can maximize the use of cloud infrastructure and minimize energy waste. Last but not least, the proposed system can serve as a good blue print for autonomous, resilient smart e-commerce system/service architecture in the cloud.

REFERENCES

[1] H. Qiu, S. S. Banerjee, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, "FIRM: An intelligent fine-grained resource management framework for SLO-oriented microservices," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 805–825.
[2] V. Subramani, "Dynamic scaling in e-commerce platforms: Microservices for latency, compliance, and resilience," *Computer Fraud and Security*, vol. 2024, no. 11, 2024. [Online]. Available: https://computerfraudsecurity.com/index.php/journal/article/view/879
[3] J. Park, B. Choi, C. Lee, and D. Han, "GRAF: A graph neural network based proactive resource allocation framework for SLO-oriented microservices," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, 2021, pp. 154–167.
[4] G. Yu, P. Chen, and Z. Zheng, "Microscaler: Automatic scaling for microservices with an online learning approach," in *2019 IEEE International Conference on Web Services (ICWS)*, 2019, pp. 68–75.
[5] B. Choi, J. Park, C. Lee, and D. Han, "pHPA: A proactive autoscaling framework for microservice chain," in *5th Asia-Pacific Workshop on Networking (APNet 2021)*, 2021, pp. 65–71.
[6] A. U. Gias, G. Casale, and M. Woodside, "ATOM: Model-driven autoscaling for microservices," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 1994–2004.
[7] A. Kwan, J. Wong, H.-A. Jacobsen, and V. Muthusamy, "HyScale: Hybrid and network scaling of dockerized microservices in cloud data centres," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 80–90.
[8] A. F. Baarzi and G. Kesidis, "SHOWAR: Right-sizing and efficient scheduling of microservices," in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 427–441.
[9] D. Balla, C. Simon, and M. Maliosz, "Adaptive scaling of Kubernetes pods," in *NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–5.
[10] B. Liu, R. Buyya, and A. N. Toosi, "A fuzzy-based auto-scaler for web applications in cloud computing environments," in *International Conference on Service-Oriented Computing*, Springer, 2018, pp. 797–811.
[11] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in cloud computing: State of the art and research challenges," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 430–447, 2017.