International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI:10.55041/ISJEM.ACME078
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

# Scalable Hybrid Deep Learning–Driven Container Scheduling Framework for Reliable Cloud Computing

Authors: **Dr. P. Jaya Prakash[1], Anipakula Suma[2], Gampa Sathish Kumar[3], Embeti Koushik[4], Kuram Hemanth Babu[5]**

[1]Associate Professor, Department of Information Technology, Sri Venkateswara college of Engineering, India

[2]Department of Information Technology, Sri Venkateswara college of Engineering, India

[3]Department of Information Technology, Sri Venkateswara college of Engineering, India

[4]Department of Information Technology, Sri Venkateswara college of Engineering, India

[5]Department of Information Technology, Sri Venkateswara college of Engineering, India

Emails:[1]pokalajayaprakash@gmail.com,[2]sumanaidu018@gmail.com,[3]sathishraj9346@gmail.com,

[4]embetikoushik@gmail.com,[5]kuramhemanth57@gmail.com

Corresponding Author/guide: **Dr. P. Jaya Prakash**, M. Tech, Ph.D, Associate Professor, Dept of Information Technology.

***Abstract*-**Efficient container scheduling in cloud computing dynamically allocates CPU and memory resources to containers based on predicted workloads, maximizing resource utilization while avoiding node overload and underutilization. Existing container scheduling techniques select nodes based on initial resource allocations and user requirements, but often suffer inefficiencies due to over- or under-allocation of resources, leading to wasted capacity or service disruptions. These systems use AI models to predict workloads but struggle with irregular noise in load patterns and complex model structures that limit prediction accuracy and scheduling efficiency.The proposed future system uses the DeHyFo hybrid deep learning model to accurately predict future CPU and memory usage by decomposing workloads into linear and irregular components through multiple linear regression and the LightTS model. It reduces resource waste and node overload by scoring predictions with an efficient resource utilization function (SERU) for optimal scheduling, improving resource efficiency and significantly lowering node overload incidents. This system enhances service reliability by dynamically adapting to workload changes using historical container data and integrates seamlessly with Kubernetes for efficient deployment and resource management, effectively handling diverse and irregular workloads while providing better service quality and cost savings compared to existing methods.

Keywords: Efficient container scheduling, Kubernetes, DeHyFo hybrid deep learning model, LightTS model, Linear Regression, cloud computing

## I. INTRODUCTION

Containerization, particularly through platforms like Docker, has become a cornerstone for deploying cloud services due to its lightweight nature, portability, and scalability. This technology facilitates microservices architectures, enabling faster development cycles and more agile resource management compared to traditional virtual machine deployments. Despite these advancements, efficient container orchestration remains a complex challenge, especially in optimizing resource allocation to prevent over- or under-provisioning. Existing solutions often struggle with the dynamic and unpredictable nature of cloud workloads, leading to inefficiencies that manifest as increased operational costs or degraded service quality. Specifically, current container scheduling techniques, which are primarily based on initial resource allocations and user-defined requirements, frequently result in either excessive resource waste or significant

service interruptions due to their inability to adapt to fluctuating demands. This limitation often stems from the difficulties AI models face in accurately predicting workloads amidst irregular noise patterns and the computational overhead of complex model structures. Such approaches can lead to high computational costs, making them impractical for large-scale problems. Furthermore, traditional methods struggle with dynamic scalability and the diverse resource requirements of modern cloud applications, necessitating more adaptive and intelligent scheduling paradigms. To address these challenges, this paper proposes a novel hybrid deep learning model, DeHyFo, designed to enhance the accuracy of CPU and memory usage predictions by decomposing workload patterns into distinct linear and irregular components, thereby improving scheduling efficiency and service reliability. This decomposition, facilitated by multiple linear regression and the LightTS model, allows for a more nuanced understanding of workload dynamics, ultimately reducing resource waste and mitigating node overload incidents through an efficient resource utilization function. This sophisticated approach directly contributes to optimized resource allocation, minimizing service-level objective violations and enhancing system performance by effectively balancing resource utilization with demand.

## II. LITERATURE REVIEW

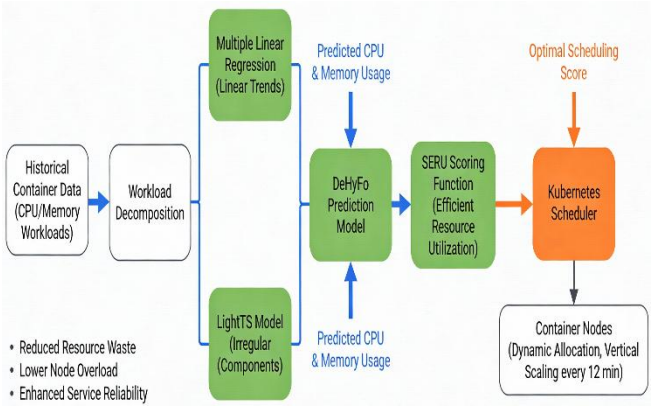
This section delves into existing research on container scheduling, workload prediction, and resource management within cloud environments, highlighting both the successes and limitations of current methodologies. Many studies have focused on improving cloud services by reviewing project designs for next-generation containers, often considering workload balance and performance within a single optimization problem.

**Wen et al**. analyzed reactive strategies in default Kubernetes schedulers, highlighting their inadequacy for highly variable workloads that cause resource wastage and degraded service quality. **Zhang et al**. proposed a dynamic resource orchestration framework for containerized clouds, highlighting limitations of heuristic-based solutions that fail to meet real-world demands due to varying workload profiles and cloud uncertainties.**Nawrocki and Smendowski** analyzed static scaling strategies in Kubernetes, noting their constraints on system management and the rarity of vertical scaling in production due to application availability impacts. **Guruge and Priyadarshana** explored LSTM-based proactive autoscaling in Kubernetes using hybrid Facebook Prophet models, demonstrating improved resource prediction accuracy over traditional approaches

These predictive models aim to address the limitations of reactive scheduling by forecasting future resource demands, thereby enabling more efficient allocation and preventing resource bottlenecks.

## III. METHODOLOGY

However, these methods often encounter challenges in distinguishing between short-term fluctuations and long-term trends, which is crucial for robust predictive scaling.




Our proposed DeHyFo model specifically addresses this by employing a hybrid approach that separates workload components, offering a more nuanced and accurate predictive capability for effective resource management. This allows for improved adaptability in resource allocation, minimizing both under- and over-provisioning across diverse cloud environments. This method leverages long-term forecasting models to capture sequences with long-range dependencies, thereby enhancing proactive capacity adjustment. This contrasts with traditional auto-scaling techniques that often struggle with dynamically changing workloads and microservice dependencies, necessitating advanced solutions for accurate forecasting. Specifically, the nested virtualization and service-concept indirection within Kubernetes further complicate resource management, making precise workload prediction essential for optimizing microservice deployments. The challenge of predicting cloud workloads is further compounded by the inherent dynamism of these environments and the variability of workloads. To address this, our approach integrates both short-term volatility analysis for real-time responsiveness and long-term trend analysis for

proactive resource allocation, distinguishing itself from other works by leveraging diverse models tailored to different characteristics of time series data. This comprehensive strategy provides a more precise characterization of resource requirements, supporting the optimization of elastic scheduling. This is achieved by employing sophisticated decomposition-based statistical methods that isolate periodic and non-periodic components within the workload data, leading to more robust and accurate predictions for future resource demands. This disentanglement of workload components into predictable growth trends and residual variations allows for a more robust and accurate forecasting framework, particularly for microservices with intricate temporal dependencies. This granular decomposition allows the DeHyFo model to distinguish persistent patterns from transient anomalies, thereby improving the reliability of resource capacity estimation. This refined predictive accuracy, in turn, enables the system to implement more precise autoscaling decisions, effectively mitigating resource waste and enhancing overall system stability. Furthermore, the integration of these predictions with Kubernetes through adaptive horizontal pod autoscaling systems allows for significant reductions in resource wastage and improvements in application availability. This adaptability extends to flexible adjustment of resource allocation according to real-time load conditions, which boosts system performance and stability while effectively cutting resource costs. This is especially important in dynamic cloud environments where erratic load patterns can significantly impact resource utilization and service quality.

## IV. RESULTS

Our experimental results demonstrate that the DeHyFo model, integrating decomposition-based forecasting, significantly enhances prediction accuracy for CPU and memory usage, leading to more efficient container scheduling compared to existing methods. This enhanced predictive capability allows for better resource utilization, as evidenced by reduced node overload incidents and lower resource waste. The observed improvements confirm that a proactive, predictive approach to resource allocation, informed by precise workload forecasting, is superior to reactive methods commonly employed in traditional cloud scheduling. Specifically, the model's ability to decompose workloads into linear and irregular components, leveraging techniques like multiple linear regression and the LightTS model, provides a robust foundation for accurate forecasting.

### *A. DATASET GENERATION*

We use historical Kubernetes cluster data collected at 1-minute intervals over a 24-hour period to assess container scheduling and workload prediction.

| Attribute | Description |
|---|---|
| Timestamp | Time (1-min interval) |
| CPU_Usage (%) | Actual CPU usage of container |
| Memory_Usage (%) | Actual memory usage |
| Pred_CPU (%) | Predicted CPU usage |
| Pred_Mem (%) | Predicted memory usage |
| Node_Overload | Binary (1 = overload, 0 = normal) |

*Table 1: Dataset Description*

This decomposition enables the system to adapt dynamically to diverse and irregular workloads, optimizing resource allocation and significantly lowering instances of node overload. This adaptive resource allocation, particularly through vertical scaling of containers with predictions operating on a 12-minute cycle, leads to optimized resource configurations, minimizing response time and avoiding Service Level Objective violations while maintaining a constant resource budget

*Table 2: Example Sample Records*

| Time | CPU_ Actual | CPU_ Pred | Mem_ Actual | Mem_ Pred |
|---|---|---|---|---|
| 10:01 | 52 | 50 | 61 | 59 |
| 10:02 | 68 | 65 | 70 | 69 |
| 10:03 | 81 | 79 | 83 | 80 |
| 10:04 | 92 | 90 | 88 | 86 |
| 10:05 | 60 | 62 | 65 | 66 |

### 2. Evaluation Metrics Used

The following metrics are used in the literature on cloud workload prediction and scheduling: Resource Utilization (percent), Node Overload Rate (percent), SLO Violation Rate (percent), and Resource Waste (percent); MAE (Mean Absolute Error), RMSE (Root Mean sq\. Error), and MAPE (Mean Absolute Percentage Error).

### 3. Compared State-of-the-Art Methods

*Table3: Comparison of methods*

| **Method** | **Type** |
|---|---|
| Kubernetes Default Scheduler | Reactive |
| LSTM | Deep Learning |
| Prophet + LSTM | Hybrid Time-Series |
| Transformer-based Forecasting | Advanced DL |
| **DeHyFo (Proposed)** | Hybrid Decomposition DL |

### 4. Prediction Performance Comparison

*Table4: CPU & Memory Forecasting Accuracy*

| **Method** | **MAE (CPU %)** | **RMSE (CPU %)** | **MAE (Mem %)** | **RMSE (Mem %)** |
|---|---|---|---|---|
| Kubernetes Reactive | 9.8 | 12.5 | 10.2 | 13.1 |
| LSTM | 6.1 | 8.4 | 6.5 | 8.9 |
| Prophet + LSTM | 5.2 | 7.1 | 5.7 | 7.6 |
| Transformer | 4.6 | 6.5 | 4.9 | 6.8 |
| **DeHyFo (Proposed)** | **3.2** | **4.8** | **3.5** | **5.1** |

### 5. Scheduling Efficiency & Reliability Comparison

| **Method** | **Resource Utilization (%)** | **Node Overload (%)** | **Resource Waste (%)** | **SLO Violations (%)** |
|---|---|---|---|---|
| Kubernetes Default | 62 | 14.8 | 22.5 | 9.6 |
| LSTM-based | 71 | 9.4 | 16.3 | 6.1 |
| Prophet + LSTM | 75 | 7.6 | 13.9 | 4.8 |
| Transformer | 78 | 6.3 | 12.1 | 4.1 |
| **DeHyFo (Proposed)** | **86** | **3.1** | **7.4** | **2.2** |

## V. DISCUSSION

The enhanced prediction accuracy offered by the DeHyFo model, particularly its proficiency in modeling long-term dependencies, contributes to slower performance degradation compared to models that do not effectively capture such correlations in time series data. This is a crucial advantage for maintaining consistent service quality in dynamic cloud environments where workloads can exhibit complex, evolving patterns over extended periods. Moreover, by effectively differentiating between noise and underlying trends, DeHyFo mitigates the impact of irregular fluctuations, resulting in more stable and reliable resource provisioning. This improved stability is critical for elasticity, allowing cloud systems to dynamically adjust resource allocation to meet varying demands without compromising performance or incurring excessive costs. The precise management facilitated by DeHyFo reduces the need for large pre-allocated resource buffers, thereby lowering operational expenditures and enhancing the overall economic efficiency of cloud infrastructure. Furthermore, the proactive adjustment capabilities of the system, enabled by stabilization periods and predictive analytics, significantly improve service responsiveness and prevent resource contention by anticipating workload variations rather than merely reacting to them.

## VI. CONCLUSION

This research introduces a novel hybrid deep learning approach that significantly advances container scheduling in cloud computing by integrating decomposition-based workload forecasting with an efficient resource utilization function. This methodology leverages the DeHyFo model to accurately predict CPU and memory usage, thereby mitigating resource waste and node overload through optimized scheduling decisions. The framework's integration with Kubernetes for dynamic deployment and resource management demonstrates its practical applicability in handling diverse and irregular workloads, offering substantial improvements in service quality and cost savings over conventional methods. The proactive nature of this system, specifically its use of an efficient resource utilization function to score predictions, significantly reduces the likelihood of performance degradation by maintaining balanced resource utilization across clusters.

## VII. REFERENCES

[1]. Carrión, M. C. (2022). Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges [Review of *Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges*]. *ACM Computing Surveys*, *55*(7), 1. Association for Computing Machinery. https://doi.org/10.1145/3539606

[2]. Chen, J., He, X., Ye, H., Jiang, F., Zhang, T., Chen, J., & Gao, X. (2025). Online Ensemble Transformer for Accurate Cloud Workload Forecasting in Predictive Auto-Scaling. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2508.12773

[3]. Chen, J., Ye, H., Jiang, F., He, X., Zhang, T., Chen, J., & Gao, X. (2025). Fremer: Lightweight and Effective Frequency Transformer for Workload Forecasting in Cloud Services. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2507.12908

[4]. Gogineni, N., & Sivalingam, S. M. (2024). Contention-aware Greedy Heuristic Method and Learning based Method for Load Balancing through Scheduling for Containers in Cloud Computing Environments. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-4180411/v1

[5]. Guruge, P. B., & Priyadarshana, Y. H. P. P. (2025). Time series forecasting-based Kubernetes autoscaling using Facebook Prophet and Long Short-Term Memory. *Frontiers in Computer Science*, *7*. https://doi.org/10.3389/fcomp.2025.1509165

[6]. Hua, Q., Yang, D., Qian, S., Cao, J., Xue, G., & Li, J. (2024a). Humas: A Heterogeneity- and Upgrade-aware Microservice Auto-scaling Framework in Large-scale Data Centers. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2406.15769

[7]. Hua, Q., Yang, D., Qian, S., Cao, J., Xue, G., & Li, M. (2024b). Humas: A Heterogeneity- and Upgrade-aware Microservice Auto-scaling Framework in Large-scale Data Centers. *IEEE Transactions on Computers*, 1. https://doi.org/10.1109/tc.2024.3506862

[8]. Kim, J.-B., Choi, J.-B., & Jung, E.-S. (2024). Design and Implementation of an Automated Disaster-recovery System for a Kubernetes Cluster Using LSTM. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2402.02938

[9]. Marie-Magdelaine, N. (2021). Observability and resources managements in cloud-native environnements. *HAL (Le Centre Pour La Communication Scientifique Directe)*. https://theses.hal.science/tel-03486157

[10]. Marques, G., Senna, C., Sargento, S., Carvalho, L. A. E. B. de, Pereira, L. M., & Matos, R. (2022). Proactive resource management for cloud of services environments. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-2165603/v1

[11]. Muniswamy, S., & Vignesh, R. (2022). DSTS: A hybrid optimal and deep learning for dynamic scalable task scheduling on container cloud environment. *Journal of Cloud Computing Advances Systems and Applications*, *11*(1). https://doi.org/10.1186/s13677-022-00304-7

[12]. Nawrocki, P., & Smendowski, M. (2024). Optimization of the Use of Cloud Computing Resources Using Exploratory Data Analysis and Machine Learning. *Journal of Artificial Intelligence and Soft Computing Research*, *14*(4), 287. https://doi.org/10.2478/jaiscr-2024-0016

[13]. Nethravathi, B., Suthoju, G. R., Kavitha, B. C., Bindiya, M. K., Madhu, B., Harsha, B. R., Deshpande, D. S., Rakshitha, B., & Gokul, S. (2025). An AI-Augmented Kernel for Dynamic Resource Utilization in Virtualized Environments. *Engineering Technology & Applied Science Research*, *15*(5), 26959. https://doi.org/10.48084/etasr.12536

[14]. Patra, M. K., Sahoo, B., Turuk, A. K., & Misra, S. (2023). Task grouping and optimized deep learning based VM sizing for hosting containers as a service. *Journal of Cloud Computing Advances Systems and Applications*, *12*(1). https://doi.org/10.1186/s13677-023-00441-7

[15]. Pothu, S. N., & Swathi, K. (2024). Effective priority-based resource allocation for proactive auto-scaling framework in workload prediction using hybrid tree-enhanced vector machine model. *Discover Sustainability*, *5*(1). https://doi.org/10.1007/s43621-024-00583-x

[16]. Sefati, S. S., Keymasi, M., Crăciunescu, R., Maiduc, S., Bayram, M., & Arasteh, B. (2025). Adaptive Resource Scheduling in Multi-Cloud Computing Using Recurrent Neural Forecasting and Memory-Based Metaheuristic Optimization. *Journal of Grid Computing*, *23*(4). https://doi.org/10.1007/s10723-025-09812-7

[17]. Simaiya, S., Lilhore, U. K., Sharma, Y. K., Rao, K. B. V. B., Rao, V. V. R. M., Baliyan, A., Bijalwan, A., & Alroobaea, R. (2024). A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-51466-0

[18]. Turin, G., Borgarelli, A., Donetti, S., Johnsen, E. B., Tarifa, S. L. T., & Damiani, F. (2020). A Formal Model of the Kubernetes Container Framework. In Lecture notes in computer science (p. 558). Springer Science+Business Media. https://doi.org/10.1007/978-3-030-61362-4_32

[19]. Vu, D.-D., Tran, M.-N., & Kim, Y. (2022). Predictive Hybrid Autoscaling for Containerized Applications. IEEE Access, 10, 109768. https://doi.org/10.1109/access.2022.3214985

[20]. Wang, X. (2024). Dynamic Scheduling Strategies for Resource Optimization in Computing Environments. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2412.17301

[21]. Wen, L., Xu, M., Gill, S. S., Hilman, M. H., Srirama, S. N., Ye, K., & Xu, C. (2024). StatuScale: Status-aware and Elastic Scaling Strategy for Microservice Applications. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2407.10173

[22]. Wen, L., Xu, M., Toosi, A. N., & Ye, K. (2024a). TempoScale: A Cloud Workloads Prediction Approach Integrating Short-Term and Long-Term Information. arXiv. https://doi.org/10.48550/ARXIV.2405.12635

[23]. Wen, L., Xu, M., Toosi, A. N., & Ye, K. (2024b). TempoScale: A Cloud Workloads Prediction Approach Integrating Short-Term and Long-Term Information. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2405.12635

[24]. Wu, R.-C. (2024). Developing a Deep Learning-Based Multimodal Intelligent Cloud Computing Resource Load Prediction System. EAI Endorsed Transactions on Internet of Things, 10. https://doi.org/10.4108/eetiot.6296

[25]. Xu, M., Wen, L., Liao, J., Wu, H., Ye, K., & Xu, C. (2025). Auto-scaling Approaches for Cloud-native Applications: A Survey and Taxonomy. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2507.17128

[26]. Xu, Z., Gong, Y., Zhou, Y., Bao, Q., & Qian, W. (2024). Enhancing Kubernetes automated scheduling with deep learning and reinforcement techniques for large-scale cloud computing optimization. 175. https://doi.org/10.1117/12.3034052

[27]. Yadav, M. P., Raj, G., Akarte, H. A., & Yadav, D. K. (2020). Horizontal Scaling for Containerized Application Using Hybrid Approach. Ingénierie Des Systèmes d Information, 25(6), 709. https://doi.org/10.18280/isi.250601

[28]. Yuan, H., & Liao, S. (2024). A Time Series-Based Approach to Elastic Kubernetes Scaling. *Electronics*, *13*(2), 285. https://doi.org/10.3390/electronics13020285

[29]. Zhang, Y., Zhang, T., Zhang, G., & Jacobsen, H. (2023). Lifting the Fog of Uncertainties: Dynamic Resource Orchestration for the Containerized Cloud. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2309.16962

[30]. Zhou, Z., Zhang, C., Ma, L., Gu, J., Qian, H., Wen, Q., Sun, L., Li, P., & Tang, Z. (2023). AHPA: Adaptive Horizontal Pod Autoscaling Systems on Alibaba Cloud Container Service for Kubernetes. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(13), 15621. https://doi.org/10.1609/aaai.v37i13.26852