UMEÅ UNIVERSITY

Accurate and Low-Overhead Workload Prediction for Cloud Management

Lidia Kidane

Doctoral Thesis, May 2025
Department of Computing Science
Umeå University
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

lkidane@cs.umu.se

ISBN 978-91-8070-712-1 (print)
978-91-8070-713-8 (digital)
ISSN 0348-0542
UMINF 25.09

Printed by Scandinavian Print Group
Umeå, 2025

Abstract

Cloud computing has transformed the IT landscape by offering users and organizations on-demand access to computing power, storage, data processing, and machine learning resources. Despite the benefits, cloud resource management faces challenges due to the heterogeneous and dynamic nature of workloads. Inefficient provisioning manifests in two critical forms: underprovisioning leads to degraded Quality of Service (QoS) and unmet Service-Level Agreements (SLAs), while overprovisioning results in unnecessary energy consumption and high operational costs. With the current rise of AI and machine learning innovations, machine learning-based workload prediction for resource provision plays a vital role in predicting future scenarios and identifying new occurrences, enabling service providers to prepare ahead of time. However, various challenges are associated with machine learning-based workload prediction.

This thesis addresses the challenges of machine learning-based workload prediction in cloud environments, including data drift due to dynamic workloads, high computational overhead, and storage overhead. Firstly, cloud workloads are dynamic, and models trained with old historical data can become obsolete over time. We addressed the challenge of accurate prediction and data drift by incorporating machine learning and streaming data processing algorithms to assist adaptive prediction. Secondly, constantly training and updating deep learning models adds significant computational overhead to the cloud infrastructure. We addressed this problem by proposing a solution that incorporates a knowledge base repository with transfer learning-based adaptation. Moreover, we explored the tradeoff between model accuracy and computational overhead. Finally, we propose a data compression mechanism that leverages an autoencoder to reduce storage overhead resulting from the continuous generation of monitoring data in cloud management systems.

Our findings reveal that the proposed methods have significantly improved the machine learning-based cloud management system. Extensive evaluation using real-world datasets reveals that the proposed methods facilitate the creation of accurate predictions, even in the face of ever-changing patterns in cloud workloads. Moreover, the methods reduced computation overhead by leveraging existing knowledge and highlighting the tradeoff required to achieve a balance between prediction accuracy and computation overhead.

Sammanfattning

Molntjänster har förändrat IT-landskapet genom att erbjuda användare och organisationer tillgång till datorkraft, lagring, databehandling och maskininlärningsresurser på begäran. Trots fördelarna står molnresurshantering inför utmaningar på grund av heterogena och dynamiska arbetsbelastningar. Ineffektiv resurstilldelning visar sig i två kritiska former: underallokering leder till försämrad tjänstekvalitet (QoS) och ouppfyllda servicenivåavtal (SLA), medan överallokering resulterar i onödig energiförbrukning och höga driftskostnader. I och med den ökade innovationen inom AI och maskininlärning spelar maskininlärningsbaserad arbetsbelastningsprognos för resursallokering en viktig roll i att förutsäga framtida scenarier och identifiera nya händelser för att notifiera tjänsteleverantörer i förväg. Dock finns diverse olika utmaningar kopplade till maskininlärningsbaserad arbetsbelastningsprognos.

Denna avhandling behandlar utmaningarna med maskininlärningsbaserad arbetsbelastningsprognos i molnmiljöer, såsom datadrift av dynamiska arbetsbelastningar, hög beräkningsoverhead samt lagringsoverhead. För det första är molnarbetsbelastningar dynamiska, och modeller som tränats med gamla historiska data kan bli föråldrade över tid. Vi antog utmaningen med noggrann prognos och datadrift genom att integrera maskininlärning och strömmande databehandlingsalgoritmer för att hjälpa i en adaptiv prognos. För det andra, att träna och uppdatera djupinlärningsmodeller konstant lägger betydande beräkningsoverhead på molninfrastrukturen. Vi tacklade detta problem genom att föreslå en lösning som integrerar en kunskapsbas med en transferinlärningsbaserad basanpassning. Dessutom utforskade vi avvägningen mellan modellnoggrannhet och beräkningsoverhead. Slutligen föreslår vi en datakomprimeringsmekanism som utnyttjar autoencoder för att minska den lagringsoverhead som uppstår från kontinuerlig generering av övervakningsdata i molnhanteringssystem.

Våra resultat visar att de föreslagna metoderna avsevärt förbättrade det maskininlärningsbaserade molnhanteringssystemet. Omfattande utvärdering med verkliga dataset visar att de föreslagna metoderna hjälper till att skapa noggranna prognoser trots de ständigt föränderliga mönstren i molnarbetsbelastningar. Dessutom minskade metoderna beräkningsoverhead genom att utnyttja gammal kunskap och belysa avvägningen för att uppnå en balans mellan prognosens noggrannhet och beräkningsoverhead.

Preface

The thesis comprises five scientific publications preceded by a short introduction. The introduction provides background information, motivation, objectives, methodology, and elaborates on the research problems, along with short summaries of the papers. The five publications are the following.

Paper I **Lidia Kidane**, Paul Townend, Thijs Metsch, and Erik Elmroth. When and How to Retrain Machine Learning-based Cloud Management Systems, *In Proceeding of* IEEE 36th International Parallel & Distributed Processing Symposium Workshops (IPDPSW), pp. 688-698, IEEE, 2022.

Paper II **Lidia Kidane**, Paul Townend, Thijs Metsch, and Erik Elmroth. Efficient Retraining of Machine Learning Algorithms in Cloud Management Systems, *Submitted for journal publication*, 2023, 18 pages.

Paper III **Lidia Kidane**, Paul Townend, Thijs Metsch, and Erik Elmroth. Automated Hyperparameter Tuning for Adaptive Cloud Workload Prediction, *In Proceedings of* the IEEE/ACM 16th International Conference on Utility and Cloud Computing (UCC), Article No.: 45, Pages 1 - 8, ACM, 2024.

Paper IV **Lidia Kidane**, Paul Townend, Thijs Metsch, and Erik Elmroth. A Hybrid Autoencoder-LSTM Framework for Efficient Workload Prediction, *Submitted for publication*, 2024, 10 pages.

Paper V **Lidia Kidane**, Paul Townend, Thijs Metsch, and Erik Elmroth. A Data-driven Framework for Efficient and Automated Workload Prediction in Cloud Computing, *Submitted for publication*, 2025, 9 pages.

This thesis project has been funded by the Knut and Alice Wallenberg Foundation under grant KAW 2019.0352 and by the eSSENCE Programme under the Swedish Government's Strategic Research Initiative.

Acknowledgements

I would like to extend my heartfelt appreciation to all those who have accompanied me throughout my doctoral studies. The achievement of this PhD journey has been challenging yet immensely fulfilling, and I would like to express my deepest appreciation to those who have supported me. This dissertation would not have been possible without their contributions, guidance, and encouragement.

Most of all, I would like to offer my sincerest thanks to my principal supervisor, **Erik Elmroth**, for providing me with the enriching opportunity to pursue my PhD in such an inspiring and stimulating research environment. His expertise and evident enthusiasm for the topic have profoundly shaped my academic growth and understanding. His patience and experience have ensured that I was able to overcome numerous research challenges along the way. I am especially thankful for his kindness and support during personally difficult times—his encouragement and understanding were instrumental in helping me persevere and complete this work.

I am also grateful to my co-supervisor, **Paul Townend**, whose technical expertise in the field has proven to be greatly valuable to my research. His ability to think problems over from multiple directions has provided necessary insight, providing depth and richness to this work. His thoughtful guidance and keen critiques have also been extremely beneficial, always challenging me to think more critically and rationally. I am truly grateful for his influence and contribution during the course of my doctoral studies.

I would also like to thank our co-author, **Thijs Metsch**, for his professional advice and constructive contributions to this research. His contribution provided the project with a solid foundation of hands-on experience that integrated industrial knowledge, significantly enhancing the research. I particularly appreciate his careful reading and critical comments on the research work and related publications, which significantly improved and tightened some crucial points in this thesis.

It has been an incredible privilege to be affiliated with the WASP graduate school. The rigorous academic environment, inspiring workshops, and exposure to outstanding researchers have profoundly impacted my academic perspective and growth. The experiences and collaborations fostered here have had a profoundly positive impact on my academic life.

I would like to acknowledge all past and present members of the **Autonomous Distributed Systems (ADS) lab** for their support, friendship, and immense contributions throughout my PhD. Having the privilege to work with such capable and committed members has been energizing and motivational. I would like to extend my sincere appreciation to my colleagues for their friendship and numerous fruitful discussions. Their support and cooperation created a lively atmosphere of scholarship. I am deeply grateful for the opportunity to be involved with such a research community. Thanks to **Ali** and **Sourasekhar** for providing me with documents and guidelines for writing the Ph.D. thesis. Special thanks to **Oliver**, who assisted in translating the Swedish segment of the kappa.

Finally, none of this would have existed without my family's unwavering love and encouragement. My parents, Tesfahiwet and Tsehainesh, thank you for your limitless faith in me. Most of all, I owe an unparalleled debt of gratitude to my husband, Simon, for being with me through every challenge, offering immeasurable emotional support, and pushing me forward even when the path forward was uncertain.

Lidia Kidane
Umeå, May 2025

Contents

Chapter 1

Introduction

Cloud computing is the main driving force behind the modern era of rapidly evolving technology and AI systems. It has revolutionized technology usage by providing developers, private users, and companies with big data processing, large model training, applications, and data services on demand while letting users rely on it regarding hardware, installation, and maintenance. In this chapter, we provide a detailed description of the motivation behind this work, followed by its leading research objectives, methodology, and contributions.

1.1 Research Motivation

The widespread adoption of cloud computing has enabled cost-effective, scalable access to computing resources and data storage management. It is crucial for the scalable training of deep learning models [Aba+16], real-time big data analytics for IoT [Bot+16; BB+12], bio-informatics [Kop+21], and autonomous systems [DLN23]. Cloud computing infrastructures are rapidly expanding to provide extensive services as demand grows; however, this brings about additional challenges on effective resource management and auto-scaling. The dynamic nature of cloud workloads and shared resources adds significant complexity to efficient resource management, underscoring the need for adaptive, intelligent solutions to handle real-time auto-scaling and resource management.

This work addresses the complexity of maintaining machine learning (ML) models used for cloud workload prediction, enabling accurate predictions over time. The core motivation of this research is to provide solutions that improve resource management in cloud computing environments through accurate, adaptive, and efficient workload predictions using machine learning. Accurate workload prediction is crucial in cloud platforms due to their dynamic and elastic nature, where resources are allocated on demand. This adds a layer of complexity in maintaining performance and ensuring safety from under- or over-provisioning. Therefore, developing effective workload prediction is crucial

for optimizing resource utilization, reducing operational costs, and ensuring optimal performance in cloud environments. Although machine learning and statistical models are commonly used for workload prediction [Kha+22], typical solutions fail to adapt well to temporal or seasonal fluctuations of workloads in cloud environments and do not incorporate mechanisms that utilize past knowledge. This often results in redundant calculations and suboptimal performance. Hence, there is a critical demand for workload prediction mechanisms that are not only computationally efficient but also adaptive and capable of reusing knowledge in the dynamic realm of cloud infrastructures. Therefore, this research is crucial to improve contemporary cloud environments.

1.2 Research Objectives

As highlighted in Section 1.1, the primary focus of this research is improving current cloud workload prediction methods to incorporate adaptability, resource efficiency, knowledge reusability, and accuracy. Cloud workloads are dynamic and complex to model, whilst static approaches fail to capture dependencies. Moreover, training deep learning models periodically is computationally intensive. The primary objective of this research is to investigate and develop effective strategies for training and sustaining accurate machine learning models specifically tailored for cloud workload prediction. This includes ensuring that the models are lightweight in terms of computational and memory overhead, while also being capable of dynamically adapting to fluctuations and evolving patterns in workload behavior over time. The study emphasizes the balance between predictive accuracy, resource efficiency, and adaptability in real-world cloud environments. The high-level research objectives are outlined as follows.

RO1: To propose, implement, and evaluate workload change detection mechanisms and efficient model update strategies for adaptive cloud workload prediction, aimed at enhancing accuracy in dynamic cloud environments

RO2: To design, implement, and assess methods for reusing knowledge to improve the computational efficiency of model updates.

RO3: To develop integrated methods for accurate cloud workload prediction that minimize computational overhead and optimize storage, with the objective of achieving a balanced trade-off between predictive accuracy, computational efficiency, and resource optimization for scalable and adaptive workload forecasting in dynamic cloud environments.

1.3 Methodology

The methodology followed in this thesis is based on the objectives of the Design Science Research (DSR) methodology [VHM20], as provided in Figure 1.1. To achieve these objectives, the following methodology is adopted: the first step involves identifying primary challenges in cloud resource prediction and problem identification, followed by finding the research gap. Next, following the definition of research objectives, a rigorous review of existing literature is conducted to achieve a thorough understanding of current knowledge and technological developments. A solution is then designed, followed by the implementation of the algorithm. The proposed solution is evaluated using selected metrics and real-world datasets, alongside state-of-the-art solutions from the literature. The iterative process facilitates the development of efficient and accurate workload prediction algorithms that align with the objectives. Finally, the results and findings are compiled into a research article for dissemination and communication.

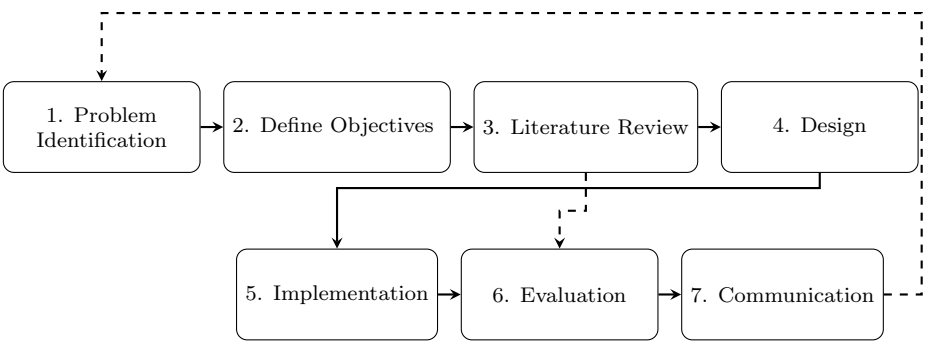

Figure 1.1: DSR methodology

1.4 Research Contributions

The main contribution of this study is designing, improving, implementing, and evaluating accurate and efficient machine learning based workload prediction methods for cloud environments for research objectives (**ROs**)defined in Section 1.2. Paper I addresses **RO1** by proposing and implementing a concept drift detection-based identification of model retraining steps to adapt to unforeseen changes in cloud workloads. Moreover, Paper II addresses both objectives **RO1** and **RO3** by introducing an efficient way of retraining machine learning models for dynamic workloads. It proposes a hybrid approach that combines concept drift based along prediction performance monitor to detect changes in workload and evaluate various baselines. In Paper III, we cover **RO2** by introducing a method that assists in prior knowledge reusability. We implement a knowledge base to save past knowledge along with trained models to assist in the adaptation to dynamic cloud workloads. Paper IV presents **RO3** by introducing and implementing an autoencoder-based workload prediction method to assist in storage efficiency and workload prediction accuracy. Finally, Paper V addresses **RO1**, **RO2**, and **RO3** by introducing and implementing a comprehensive solution for accurate, efficient, and adaptive workload prediction. It emphasizes the identification of various settings and the outcome in the solution's overall resource efficiency and accuracy.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 explores the fundamentals of cloud computing and resource management. Next, in Chapter 3, we identify the main challenges with machine learning-based cloud resource prediction and provide a comprehensive overview of machine learning in cloud workload prediction. Chapter 4 focuses on evaluation strategies, evaluation metrics, and datasets. An in-depth analysis of the detailed contribution of each paper is described in Chapter 5. Finally, Chapter 6 discusses potential future directions in the area of machine learning based cloud resource prediction and presents possible research tracks for further investigation.

Chapter 2

Foundation of Cloud Computing

This chapter provides a general overview of cloud computing environments with a special focus on cloud resource management. This chapter begins with a general background and definition of cloud computing, discusses its standard characteristics, service models, and classifications, and concludes with a section focusing on cloud resource management in general, as well as the challenges of cloud resource management and the characteristics of workloads.

2.1 Cloud Computing Environment

Cloud computing enables the delivery of services, including computing, networking, storage, databases, and software, to users without the need to own and manage physical infrastructure. Based on the definition from the National Institute of Standards and Technology (NIST), cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [MG11]. The NIST definition points out five essential characteristics as illustrated in Figure 2.1. It allows on-demand service where users can automatically provision their resources. Secondly, cloud services are available over a broad network and accessible by a set of heterogeneous devices. Third, rapid elasticity allows services to be scaled up and down to match workloads automatically, quickly, and seamlessly. Finally, it can be measured with automatic monitoring and control, allowing for pay-per-use billing. The characteristics enable the model to provide a cost-effective, flexible, and scalable service, in contrast to traditional IT infrastructure [Arm+10a; BVS18]. Services are offered through Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) models as shown in Figure 2.2.

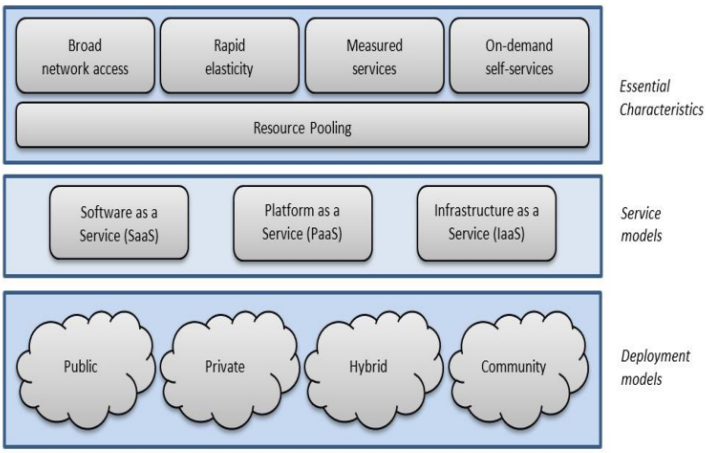

Figure 2.1: NIST visual model for cloud computing definition [Nav+16]

The key technology that enables cloud computing is virtualization. It abstracts physical resources to multiple virtual machines that share physical systems. It improves resource and power utilization through consolidation. Moreover, it provides a high degree of isolation due to isolated memory, processes, and operating environments [Sin18]. Other benefits of virtualization include load balancing, scalability, and security. There are several types of virtualization today as illustrated in Figure 2.3:

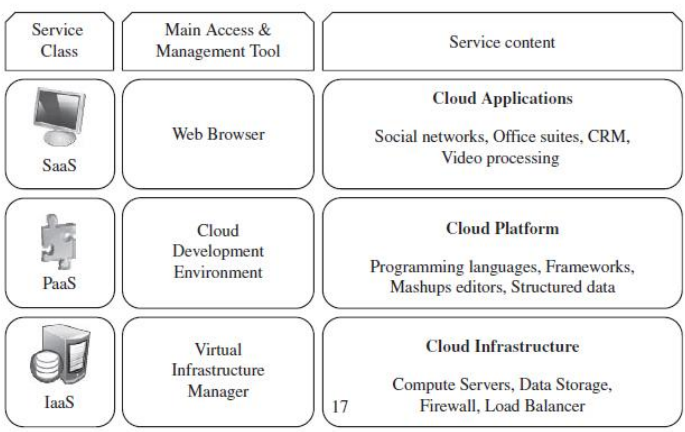

Figure 2.2: Cloud computing service models [BBG11]

Hypervisor-Based Virtualization: The hypervisor is a type of operating system that emulates the server hardware and enables multiple instances of virtual machines (VMs). The instances share the same hardware but are completely isolated. KVM [Djo+21], VMware ESXi [Mis10], and Hyper-V [Mic24] are widely used hypervisor-based virtualization technologies.

Container-Based Virtualization: Unlike hypervisors, container-based virtualization emulates the operating system (OS) level. They allow virtualization at the operating system level by packaging applications with their dependencies as lightweight instances. Docker [Smi17] is a widely used container-based virtualization technology that isolates containers with a shared host OS kernel. Compared to a hypervisor, containers have shorter startup time and lower performance overhead [BK21].

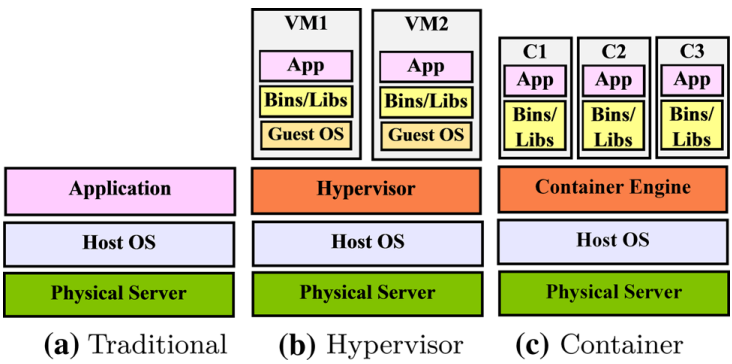

Figure 2.3: Overview of traditional vs. virtualized infrastructure models [BK21]

In order to allow modern cloud applications to be scalable, easy to develop, and maintain, Microservices come into the picture. They allow breaking down applications into small parts and are responsible for one task [Thö15]. They can be deployed independently across containers efficiently.

Several software systems are responsible for the management of data centers and workload orchestration. Kubernetes is an open-source software responsible for container orchestration by removing the burden of automating the scaling, management, and deployment process [Bur+16]. On the other hand, another open source software support managing cloud infrastructure by providing tools to manage compute, storage and network resources [SAE12]. Other technologies have also become important in cloud-native ecosystems:

Edge Computing: Edge computing transformed the real-time decision-making process by bringing the compute closer to the data sources, such as Internet of Things (IoT) devices [Shi+16]. It allows low-latency processing.

Serverless Computing: Serverless computing model allows deployment of code as functions that respond to specific events [Bal+17]. It removes

the burden of infrastructure provisioning, scaling, and maintenance from developers. As users are charged only for compute time utilized, it lowers cost and improves efficiency.

Infrastructure as Code (IaC): This enables automation of configuration and management of cloud infrastructure through code and configuration files [RMW19]. IT allows consistent deployments.

Cloud computing has become the primary backbone of contemporary information technology, enabling cost-effective solutions and eliminating the initial expenditure of building an IT infrastructure [Isl+23]. According to [GS21b], by 2025, the majority of new digital workloads are expected to be hosted on cloud-native platforms, with projections reaching over 95%, a significant increase from 30% in 2021. Moreover, Cloud has accelerated the innovation and usability of Artificial Intelligence (AI) and Machine Learning(ML). The convergence of AI and cloud computing services has led to AI-as-a-Service (AIaaS), enabling organizations to advance their data analytics, machine learning, and smart decision-making capabilities [Ram23].

Although cloud computing is widely popular and adaptable, it encounters multiple challenges that affect its efficiency. Firstly, security and privacy risks associated with storing private data in shared environments present significant issues [Par+22]. Additionally, resource retention in virtualized environments leads to performance fluctuations and potential declines in service quality [Ios+11]. Resource management is a complicated task due to dynamic workload patterns and heterogeneous resources [Arm+10b]. Additionally, the carbon emissions from data centers and concerns related to energy efficiency raise sustainability concerns [BBA10]. These obstacles underscore the need for intelligent resource management solutions, particularly those that are both cost-effective and resource-efficient.

2.2 Resource Management in Cloud Computing

Data centers' energy consumption is predicted to account for nearly 8% of global electricity demand by 2030 [Jon+18], thus making efficient resource management a necessary step. However, there are undeniable challenges to achieving this due to the dynamic nature of cloud environments [ZCB10]. Resource management involves efficiently allocating computational resources such as CPU, memory, storage, and network bandwidth to meet dynamic user demands based on real-time needs [BAB12]. Dynamic resource allocation is done through methods such as virtualization, load balancing, quality of service (QoS), and auto-scaling. Virtualization allows multiple virtual machines to run on a single physical server, thus enhancing efficient hardware utilization and elasticity[Arm+10a]. VM consolidation enhances virtualization by enabling the migration of underutilized virtual machines across servers, thereby reducing energy consumption [BAB12]. To ensure QoS, more complex solutions for scheduling are also used, such as

auto-scaling, load balancing, and VM consolidation [Li+14]. The load-balancing technique evenly distributes workloads across servers to prevent bottlenecks, while autoscaling dynamically adjusts resources in response to real-time demand [Guo+17]. While the sophistication is in place, resource management faces further challenges such as over-provisioning and interference between workloads [Li+14]. These challenges heightened the need to find more effective solutions through machine learning techniques [Guo+17].

2.3 Challenges of Cloud Resource Management and Characteristics of Workloads

Resource management in cloud computing requires handling dynamic resource allocation and heterogeneous workloads [ZCB10]. Due to this, various challenges arise, such as co-located workloads with shared resources competing for resources and resulting in performance degradation [Xu+13]. Moreover, rapid scaling to meet the quality of service requirements for users results in cost-ineffective provisioning and over-provisioning [Li+14]. Finally, energy-aware resource management techniques are also in great need due to high energy consumption by data centers [BAB12].

Workloads in cloud environments exhibit heterogeneous patterns that complicate resource management. Moreover, various workloads in web applications exhibit sudden, short-lived spikes [Guo+17]. Workloads can be categorized as data-intensive or time-critical. Data-intensive workloads for big data processing applications often prioritize I/O and network bandwidth to gain fast access and high throughput. On the other hand, time-sensitive workloads emphasize low latency [Wan+21]. Moreover, cloud workloads fluctuate due to user and application request fluctuations [TBG20]. These characteristics underscore the need for adaptive resource management strategies, where machine learning models facilitate proactive resource provisioning by predicting future workloads and implementing workload classification approaches [Che+19] or hybrid approaches.

Chapter 3

Machine Learning Based Resource Management in Cloud Computing

This chapter discusses machine learning (ML) techniques for managing cloud resources, including workload prediction models such as regression, neural networks, and reinforcement learning, as well as their associated challenges, including data variability, scalability, and real-time adaptability in dynamic cloud environments.

3.1 Workload Prediction in Cloud Management: Importance and Challenges

Machine learning (ML) models enable proactive scaling in the cloud by identifying and learning complex patterns within workloads. Predicting workload accurately enables the proactive optimization of resources at minimal cost, thereby maintaining service-level agreements (SLAs) [Isl+12] and considerably minimizing both under-provisioning and over-provisioning, thus enhancing Quality of Service (QoS). Cloud workload prediction is increasingly supporting autoscaling and dynamic scheduling in cloud environments [MK20].

3.2 Components of Machine Learning based Workload Prediction

The general architecture of the machine learning method comprises various components, including data collection, data preprocessing, feature engineering

and selection, model training, evaluation, deployment, and monitoring and feedback loop. The full process is illustrated in Figure 3.1.

Data collection: The data collection module collects data from various sources, including system logs, application performance metrics, VM resource utilization, and monitoring systems like Prometheus [Tur18]. Metrics collected include CPU usage, memory usage, and network throughput.

Data Preprocessing: The preprocessing module encompasses normalization, missing value handling, and time series transformation. The raw data collected from monitoring systems is noisy and unstructured, and often consists of missing values [Cor+20].

Feature engineering: The Feature engineering module consists of methods for feature extraction and selection steps. This stage performs statistical analysis and correlation assessment to determine the correlation level among features, and dimensionality reduction to transform time series data from a high-dimensional space to a lower-dimensional space. This module is a crucial step in improving prediction performance and reducing computational overhead through efficient training [Wan+22].

Model selection and training: The Model selection and training module is the central component that comprises the selection of feasible models ranging from traditional ML models, including linear regression and decision trees, to advanced deep learning approaches. Model training involves feeding data, optimizing hyperparameters, and validating results using selected metrics.

Evaluation and Deployment: The evaluation and deployment module evaluates the trained model using selected hyperparameters such as RMSE, MAE, or MAPE on test data. Once predictions have an acceptable level of accuracy according to system requirements, the model is deployed in the production environment.

Monitoring and feedback loop: The monitoring and feedback component assists in detecting model degradation and concept drifts [**lu2018learnin**] and ensures model usability in production.

3.3 Machine Learning based Workload Prediction Methods

Machine learning-based prediction of cloud workloads has transformed and enhanced resource allocation strategies in cloud environments, significantly improving resource utilization and efficiency. This step is possible due to the availability of training data generated from system logs or monitoring systems.

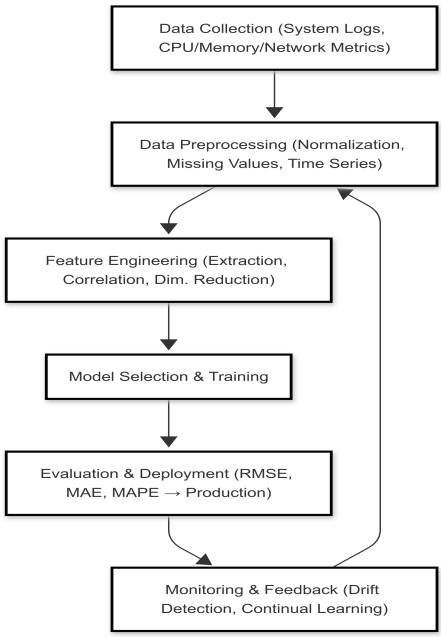

Figure 3.1: ML-based workload prediction

The prediction techniques range from simple time-series forecasting models to complex deep learning approaches, depending on the system's requirements.

3.3.1 Traditional Machine Learning

Traditional regression-based machine learning models have initially been used in cloud workloads. A Linear Regression [AA13] has been proposed by [Yan+14] to predict a number of requests for cloud service with the assumption that workloads are linear in short time intervals. Moreover, Support Vector Machines (SVM) are utilized to predict Service Level Agreement (SLA) violations based on workloads and resource allocation [BA13].

3.3.2 Deep Learning

Deep learning approaches have been widely applied to cloud workload prediction scenarios. After preprocessing, the data is passed through a deep learning model, and hyperparameters are tuned based on the learning process. Deep neural network-based models have successfully addressed the high-dimensional and highly variable features of cloud workloads [Xu+22] and workload prediction from multiple virtual machines to predict peak demands [BRC21]. Recurrent Neural Networks (RNN) have successfully worked with sequential data, includ-

ing time series data. However, Long Short-Term Memory (LSTM) models are introduced to address the limitations of RNNs in capturing long-term dependencies in time series data [Yu+19]. This significantly transformed machine learning-based cloud workload prediction approaches, which train on long-term historical workloads to produce accurate results [KGS18].

3.4 Challenges of Machine Learning-Based Workload Prediction

While machine learning techniques have significantly improved the accuracy of workload prediction in cloud environments, they introduce several unique challenges that impact their practical deployment [Che+19]. Cloud environments are dynamic, with ever-changing resource demands and user and application requests [GS21a]. One fundamental issue is the **data quality and availability** problem. ML models require large volumes of high-quality training data. Still, real-world cloud workloads often exhibit missing values, outliers, and noise due to multi-tenant interference and system failures [Xu+13]. Furthermore, the dynamic nature of cloud applications leads to **concept drift**, where statistical properties of workloads change over time, requiring continuous model retraining [Lu+21].

The **computational overhead** of complex ML models presents another significant challenge. Deep learning approaches, such as LSTMs and Transformers, while accurate, require substantial computational resources for both training and inference, potentially negating the energy savings they aim to achieve [Liu+20]. This is particularly problematic for real-time prediction scenarios where low latency is critical. Additionally, most state-of-the-art models suffer from **poor interpretability**, making it difficult for cloud operators to understand and trust the predictions [Aru+22].

The **diversity of workload patterns** across different cloud services exacerbates these challenges. A model trained on web application traces may perform poorly when applied to scientific computing workloads or serverless functions [Wan+21]. This necessitates either service-specific models, which increase maintenance complexity, or extremely generalized architectures, which reduce accuracy. Recent attempts to address this issue through meta-learning and transfer learning have shown promise, but they introduce new challenges in terms of model stability and training efficiency [Mah+22].

3.5 Efficient Workload Prediction in Cloud Environments

Despite the promising potential of machine learning-based cloud workload prediction solutions, various challenges arise that negatively affect their performance and applicability [Mir+19]. An efficient workload prediction is defined as a

method that can generate an accurate prediction based on system requirements, with minimal or acceptable computational overhead. Thus, computation and energy-aware prediction models are a necessary step to follow [Pre+14]. A list of possible challenges for an efficient workload prediction approach is stated below:

Concept Drift: Workload patterns in cloud environments are dynamic and ever-evolving due to changing user behavior and seasonal trends [Mir+19]. Static models become obsolete as new workload patterns arrive in the system. Solving this requires concept drift handling mechanisms, including online learning or retraining approaches [Gam+14].

Computation Overhead: Deep learning models utilized in cloud workload prediction often result in significant computational overhead during training and prediction steps. While they assist in accurate predictions, the overhead resulting from utilizing those models decreases their overall usability and benefits [Che+20; HMM18].

3.5.1 Navigating Concept Drift Aware Workload Prediction

Concept drift refers to unforeseen changes in the statistical properties of data over time [Lu+18], as illustrated in Figure 3.2. This situation is particularly challenging in the case of workload prediction systems relying on historical data. Types of concept drifts include:

- Sudden drift: Abrupt changes in workload patterns, e.g. hardware update or popularity of deployed application [Seh+18].
- Gradual drift: Slow evolution of usage patterns over time.
- Recurring drift: Seasonal or periodic changes in workload patterns over time [Sun+19].
- Incremental drift: Continuous small changes either in ascending or descending order in workload patterns.

3.5.2 Concept Drift Detection Approaches

Concept drift detection algorithms are generally based on a variety of techniques spanning fields such as statistical, window-based, and model error-based.

Statistical methods: These approaches utilize statistical methods to evaluate ongoing predictions against both the existing data and the initial training data to determine whether concept drift is occurring. One such method is the Page-Hinckley Test (PH), a method based on the cumulative sum (CUMSUM) algorithm, which computes a running average across the data

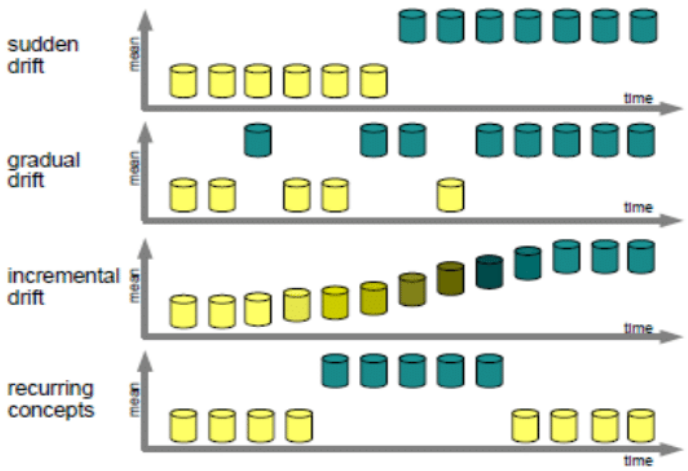

Figure 3.2: Types of concept drift [Lu+18]

points. Computes a running mean and adjusts it on the arrival of a new entry. A detected alarm is set when the variation between the minimum mean and the current mean surpasses a user-defined threshold [Gam+14].

Window based Window-based techniques are employed when it is not practical to store all streaming data. Adaptive Windowing (ADWIN), which utilizes a sliding window of variable size. New data points are added to the window as long as no concept drift has been identified. A change in distribution is recognized when the average distance between the two sub-windows surpasses a confidence value defined by the user, resulting in the removal of the older sub-window [BG07].

Model error rate based These algorithms are performed by identifying changes in systems through the monitoring of prediction errors from a forecasting model currently in use [Bai+20]. Drift Detection Method (DDM) tracks the error rate of the prediction model and detects drift when significant changes are seen. It uses a binomial distribution to detect drift and warning levels when the error rate exceeds a pre-set threshold level [Gam+14]. Moreover, the Early Drift Detection Method (EDDM) works by monitoring the distance between two consecutive errors, generating warning and alarm rates [Bae+06]. This method improves DDM by detecting gradual concept drift.

We address change detection in cloud workloads through concept drift detection methods in this work. Statistical and window-based algorithms perform a change in distribution test in the incoming workload data; they have been

used in workload change detection with various metrics. On the other hand, model error rate methods track changes in the workload prediction model. The techniques have been applied to workload change detection using various metrics. At later stages, a combination of the methods is also adapted to remove false positives and enable smooth adaptation.

3.5.3 Approaches for Concept Drift-Aware Workload Prediction

Various steps can be followed to adapt models to concept drift in workloads. The approaches vary in terms of usability and the methods employed as illustrated in Figure 3.3.

- Passive mode: This can be accomplished by periodically updating models with new incoming data through online learning techniques [MK15], or by employing ensembles of models trained on data from different time windows. Notable approaches include incremental learning [He+11], weighted ensemble methods [KM07], and the sliding window technique [Gam+14]. While online learning offers adaptability, it is often computationally intensive—particularly when applied to deep learning models in cloud computing environments. In contrast, ensemble-based methods mitigate this overhead by limiting the frequency of model retraining, instead retaining and reusing models trained on distinct temporal segments [MSB15].
- Active mode: This employs concept drift detection mechanisms to monitor prediction errors and identify shifts in the distribution of incoming workload data. The model is retrained whenever a change in data distribution is detected or a significant decline in prediction accuracy occurs [Lu+18].

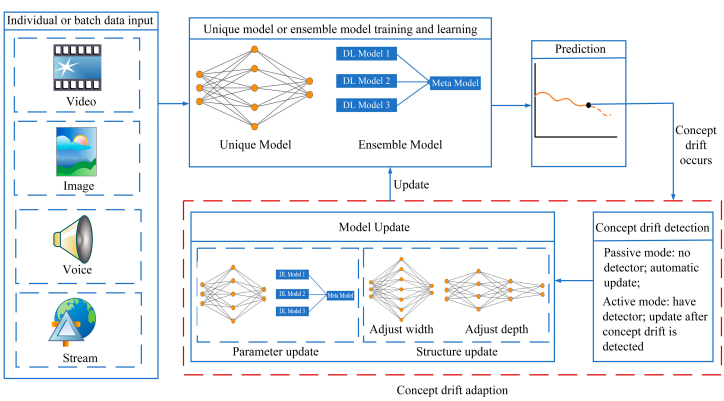

Figure 3.3: Concept drift adaptation techniques [Xia+23]

3.5.4 Challenges in Production Systems

- Latency requirements: Real-time adaptation for workload prediction models must meet Service Level Agreement (SLA) requirements.
- Resource Constraints: Adaptive models require more computational resources to train models and adapt to new patterns in workload, either periodically or on change detection [Che+20; HMM18].
- Cold start problems: The period after drift detection and until enough data is collected to update the working model needs a handling mechanism.

Chapter 4

Evaluation Strategies

This chapter presents an extensive assessment of the proposed workload prediction model for application in cloud management. The primary objectives are to assess its accuracy, computational efficiency, and feasibility in real-world cloud environments. Comparison with state-of-the-art methods is made based on synthetic and real-world datasets and metrics.

4.1 Experimental Setup

4.1.1 Datasets

The evaluation uses both real-world and synthetic workload traces:

- **Synthetic Dataset**: We implement a load generator with specifications described in the cloud-native microservices demo application [1]. The implemented load generator has been modified to incorporate abrupt concept drift in the dataset and runs on two Kubernetes clusters, with samples recorded at a frequency of 15 minutes.
- **Wikipedia Traces**: Data consists of 10% of all HTTP requests directed to all wiki projects from Sept 19, 2007, to Dec 31, 2013. Front-end proxy caches generate requests [Urd+09], and contain HTTP requests for all languages supported by Wikipedia. Each request is aggregated over a one-hour period and contains a unique ID, a timestamp, and the number of requests at that specific time. We select the Arabic wiki trace because it exhibits both gradual and sudden concept drifts over the period during which the traces were collected.
- **Ericsson Dataset**: A real-world dataset collected from Ericsson Sweden data center. We select two performance metrics from the traces for prediction: CPU utilization and memory utilization.

[1] `https://github.com/GoogleCloudPlatform/microservices-demo/`

4.1.2 Baseline Models

We compare our proposed approach with the widely adopted prediction baseline, the Long Short-Term Memory Network (LSTM) model. All models are optimized via cross-validation and trained on identical data splits.

4.1.3 Evaluation Metrics

To assess prediction performance, we use the following standard metrics:

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4.3}$$

Training time: Time taken to train a model across the selected workload dataset.

Prediction time: Time taken to compute a single prediction, averaged over all instances.

Resource Overhead: Measured by CPU and memory usage during model inference.

Chapter 5

Summary of Contributions

5.1 Paper I

Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. MicroSplit: When and How to Retrain Machine Learning-based Cloud Management Systems *In Proceeding of 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).*

5.1.1 Paper Contributions

Paper I addresses **RO1** by examining the detection of changes in cloud workloads, the degradation of workload prediction models, and exploring methods for retraining the models used in cloud workload prediction.

This paper introduces and explores methods for retraining machine learning models in cloud management systems in the presence of concept drift. By leveraging concept drift detection approaches with Long Short-Term Memory, the work explores various methods for retraining a model in the presence of concept drift. The analysis of when to retrain examines approaches for detecting when retraining is necessary, utilizing concept drift detection and prediction error thresholds, as well as determining the optimal point at which retraining should occur. Additionally, the analysis of how to retrain focuses on the data required for retraining, and the proportion should be taken from before and after the need for retraining is detected. The primary objective of this model is to determine when and how to retrain these models effectively in the presence of concept drift. Through a comprehensive evaluation using synthetic and real-world workload data, the approach improved performance on all models after concept drift. Furthermore, it showed that we provided recommendations for cloud management systems with support for the automatic retraining of ML-based models.

5.2 Paper II

Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. Efficient Retraining of Machine Learning Algorithms in Cloud Management Systems, 2023, 18 pages.

5.2.1 Paper Contributions

Paper II targets **RO1** and **RO3** and focuses on detecting changes that manifest in cloud workloads and efficient retraining of workload prediction models.

In this paper, we examine an efficient method for retraining machine learning models used in cloud workload prediction models. While state-of-the-art data mining techniques have been developed to address this problem, the extreme characteristics of data produced in cloud environments raise concerns about their applicability. In this work, we perform an experimental evaluation of methods to detect when retraining is needed, how retraining should be done, and what data should be used. However, current concept drift detection methods in the literature are prone to false positives. To mitigate this, we propose a hybrid concept drift detection approach that combines data monitoring with model performance metrics. Our evaluation of synthetic and real-world cloud workload datasets highlights challenges in applying concept drift-aware prediction methods in cloud environments. The results show that our proposed approach and implemented methods improve drift detection and lead to more than 60% improvements over the baseline prediction accuracy. Furthermore, the proposed method significantly reduces computation by eliminating false detections, which in turn eliminates the need for necessary retraining.

5.3 Paper III

Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth . Automated Hyperparameter Tuning for Adaptive Cloud Workload Prediction, *In Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing (UCC)*, Article No.: 45, Pages 1 - 8, ACM, 2024.

5.3.1 Paper Contributions

Paper III addresses **RO2** by presenting a knowledge reusability framework built upon a pre-existing knowledge base for a cloud workload prediction system. It investigates key concepts such as workload change detection, adaptive mechanisms, and computational efficiency.

This paper proposes an innovative approach to accurately and efficiently adapting cloud workloads by introducing a knowledge base. It proposes an automated framework for cloud workload prediction that addresses critical challenges of accuracy, adaptability, and computational efficiency. The key innovation is a self-tuning approach that dynamically optimizes model hyperparameters and detects concept drift using a pre-built knowledge base derived from historical workload statistics, thereby eliminating the need for manual intervention. The framework adapts pre-trained models to evolving workload patterns by integrating transfer learning, while embedded concept drift detection ensures robustness to non-stationary data streams. Evaluated on synthetic and real-world datasets, the method achieves 50% higher prediction accuracy compared to static approaches, significantly reducing computational overhead. The solution bridges theory and practice, providing cloud operators with a scalable and deployable tool for resource provisioning that strikes a balance between accuracy, automation, and efficiency.

5.4 Paper IV

Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. A Hybrid Autoencoder-LSTM Framework for Efficient Workload Prediction, 2024, 10 pages.

5.4.1 Paper Contributions

Paper IV contributes to **RO3** by proposing a method for efficient workload prediction and data compression.

This paper presents a deep learning framework for efficient cloud workload prediction, addressing the challenges of high-dimensional monitoring data and computational complexity. The proposed solution combines autoencoders for data compression and feature extraction with LSTM networks for accurate workload forecasting, significantly reducing processing demands while maintaining prediction quality. By training the LSTM on compressed data representations, the framework achieves a balance between dimensionality reduction and model performance, demonstrating up to 60% data compression with minimal reconstruction loss. Experimental results on both synthetic and real-world datasets show improvements in both prediction accuracy and computational efficiency compared to baseline methods while systematically analyzing the trade-offs between compression ratios and forecasting performance. This approach provides cloud systems with a solution for resource provisioning that addresses the critical challenges of high-volume workload processing and workload prediction compression.

5.5 Paper V

Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. A Data-driven Framework for Efficient and Automated Workload Prediction in Cloud Computing, 2025, 9 pages.

5.5.1 Paper Contributions

Paper V contributes to **RO3** by presenting a combined framework for accurate and efficient workload prediction. It incorporates change detection and adaptation, knowledge reusability, and workload compression, while also analyzing the trade-offs between different techniques in terms of prediction accuracy and efficiency.

This study introduces a comprehensive data-driven framework for automated workload prediction in cloud computing, specifically designed to minimize operational overhead for resource providers. The framework addresses five critical challenges: workload variability, predictive accuracy, computational efficiency, storage optimization, and knowledge reusability. Key innovations include (1) a knowledge base system that stores and reuses prediction models to enable rapid adaptation while eliminating redundant computations and (2) an autoencoder-based compression method that reduces long-term historical data storage requirements without sacrificing prediction quality. Through the systematic evaluation of various configurations, the framework demonstrates significant improvements in both operational efficiency, which reduces computational and storage costs, and prediction accuracy. Experimental results show that this integrated approach maintains high service quality while enhancing cost-effectiveness for cloud providers, offering a practical solution for dynamic cloud environments where prediction overhead directly impacts operational economics. The framework's balanced optimization of accuracy and efficiency represents a meaningful advance toward sustainable, scalable workload prediction systems.

5.6 Contributions by the Author of This Thesis

The contributions of the thesis author to these papers are as follows: Starting from a high-level description of the problem domain, the thesis author refined and formulated specific research problems for each paper, developed corresponding solutions, designed and conducted experiments, and produced the papers. Throughout the process, the work was carried out with continuous feedback, suggestions, and guidance from the co-authors.

Chapter 6

Future Research Directions

As more users and organizations utilize cloud assets, various problems and research issues are likely to arise when managing resources. This chapter outlines a number of specific problems and provides information on potential research directions and innovations. Moreover, emerging cloud paradigms, such as edge computing and serverless architectures, introduce additional complexity. The **geographical distribution** of edge resources creates prediction scenarios that differ markedly from traditional cloud workloads [Sha+21].

6.1 Cross-Domain Transfer Learning and Meta-Learning

A major challenge associated with predicting cloud workload behaviors using machine learning methods is the computational overhead incurred during the model training and inference phases. It is necessary to develop computationally lightweight models. The knowledge base and transfer learning approach developed in this thesis serve as a baseline for further research on cross-domain adaptation and meta-learning in cloud settings. Future work can investigate federated meta-learning [FAL17] for allowing decentralized model adaptation across various cloud infrastructures without compromising data privacy. Additionally, methods such as neural architecture search (NAS) [ZL16] should be investigated to facilitate the automatic creation of lightweight yet accurate prediction models.

6.2 Better Compression of Long-Term Cloud Monitoring Data

The steep rise in cloud monitoring data underscores the urgent need for sophisticated compression methods that optimize storage efficiency while ensuring

efficient retrieval. Future research would examine hybrid compression schemes that combine both lossy and lossless compression algorithms [CLL22], as well as learned compression algorithms [The+17], to adapt to temporal changes in resource metrics dynamically. Research should also focus on trade-offs involving compression ratios, computational overheads, and their effect on subsequent analytics processes.

6.3 Interpreting Explainability for Dynamic Cloud Contexts

As cloud infrastructures become more dynamic and complex, it is important to ensure the transparency of resource management decisions. Future research should explore explainable artificial intelligence (XAI) techniques [Arr+20] tailored to cloud environments, including real-time feature attribution methods and visualization tools for temporal models. Additionally, hybrid approaches [Zha+21] that combine rule-based systems and deep learning techniques can enhance trust and simplify debugging in autonomous cloud orchestration.

Bibliography

[AA13] Ratnadip Adhikari and Ramesh K Agrawal. "An introductory study on time series modeling and forecasting". In: *arXiv preprint arXiv:1302.6613* (2013).

[Aba+16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: `1603.04467 [cs.DC]`. URL: `https://arxiv.org/abs/1603.04467`.

[Arm+10a] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. "A view of cloud computing". In: *Communications of the ACM* 53.4 (2010), pp. 50–58.

[Arm+10b] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. "A view of cloud computing". In: *Communications of the ACM* 53.4 (2010), pp. 50–58.

[Arr+20] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges". In: *Information Fusion* 58 (2020), pp. 82–115.

[Aru+22] Siva Arunachalam et al. "Explainable AI for cloud workload prediction". In: *ACM Transactions on Autonomous and Adaptive Systems* 17.2 (2022), pp. 1–28.

[BA13] Akindele A Bankole and Samuel A Ajila. “Predicting cloud resource provisioning using machine learning techniques”. In: *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2013, pp. 1–4.

[BAB12] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. “Energy-efficient resource management in virtualized cloud data centers”. In: *Future Generation Computer Systems* 28.5 (2012), pp. 755–768.

[Bae+06] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. “Early drift detection method”. In: *Fourth international workshop on knowledge discovery from data streams*. Vol. 6. Citeseer. 2006, pp. 77–86.

[Bai+20] Lucas Baier, Marcel Hofmann, Niklas Kühl, Marisa Mohr, and Gerhard Satzger. “Handling Concept Drifts in Regression Problems–the Error Intersection Approach”. In: *arXiv preprint arXiv:2004.00438* (2020).

[Bal+17] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. “Serverless computing: Current trends and open problems”. In: *Research advances in cloud computing* (2017), pp. 1–20.

[BB+12] Prahlada Rao B.B., P. Saluia, Nancy Sharma, A. Mittal, and S. Sharma. “Cloud computing for Internet of Things & sensing based applications”. In: *2012 Sixth International Conference on Sensing Technology (ICST)* (2012), pp. 374–380. URL: `https://api.semanticscholar.org/CorpusID:24568233`.

[BBA10] Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy. “Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges”. In: *arXiv preprint arXiv:1006.0308* (2010).

[BBG11] Rajkumar Buyya, James Broberg, and Andrzej Goscinski. *Cloud Computing: Principles and Paradigms*. Wiley, 2011. ISBN: 978-0-470-88799-8.

[BG07] Albert Bifet and Ricard Gavalda. “Learning from time-changing data with adaptive windowing”. In: *Proceedings of the 2007 SIAM international conference on data mining*. SIAM. 2007, pp. 443–448.

[BK21] Aditya Bhardwaj and C Rama Krishna. “Virtualization in cloud computing: Moving from hypervisor to containerization—a survey”. In: *Arabian Journal for Science and Engineering* 46.9 (2021), pp. 8585–8601.

[Bot+16] Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. "Integration of Cloud computing and Internet of Things". In: *Future Gener. Comput. Syst.* 56.C (Mar. 2016), pp. 684–700. ISSN: 0167-739X. DOI: 10.1016/j.future.2015.09.021. URL: https://doi.org/10.1016/j.future.2015.09.021.

[BRC21] Paras Bhagtya, S Raghavan, and K Chandraseakran. "Workload classification in multi-vm cloud environment using deep neural network model". In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 2021, pp. 79–82.

[Bur+16] Brendan Burns, Brian Grant, David Oppenheimer, Eric Brewer, and John Wilkes. "Borg, omega, and kubernetes". In: *Communications of the ACM* 59.5 (2016), pp. 50–57.

[BVS18] Rajkumar Buyya, Christian Vecchiola, and S Thamarai Selvi. *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2018.

[Che+19] Tianyi Chen et al. "Deep learning for workload prediction in cloud computing". In: *IEEE Transactions on Parallel and Distributed Systems* 30 (2019), pp. 987–1001.

[Che+20] Chunlei Chen, Peng Zhang, Huixiang Zhang, Jiangyan Dai, Yugen Yi, Huihui Zhang, and Yonghui Zhang. "Deep learning on computational-resource-limited platforms: A survey". In: *Mobile Information Systems* 2020.1 (2020), p. 8454327.

[CLL22] Tianqi Chen, Zhiyuan Liu, and Mu Li. "Deep compression for cloud monitoring data: A survey". In: *IEEE Transactions on Cloud Computing* 10.2 (2022), pp. 1450–1465.

[Cor+20] Juan Antonio Cortés-Ibáñez, Sergio González, José Javier Valle-Alonso, Julián Luengo, Salvador García, and Francisco Herrera. "Preprocessing methodology for time series: An industrial world application case study". In: *Information sciences* 514 (2020), pp. 385–401.

[Djo+21] Borislav Djordjevic, Valentina Timcenko, Nenad Kraljevic, and Nemanja Macek. "File System Performance Comparison in Full Hardware Virtualization with ESXi, KVM, Hyper-V and Xen Hypervisors." In: *Advances in Electrical & Computer Engineering* 21.1 (2021).

[DLN23] Gerasimos Damigos, Tore Lindgren, and George Nikolakopoulos. "Toward 5G edge computing for enabling autonomous aerial vehicles". In: *IEEE Access* 11 (2023), pp. 3926–3941.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning* 70 (2017), pp. 1126–1135.

[Gam+14] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. "A survey on concept drift adaptation". In: *ACM computing surveys (CSUR)* 46.4 (2014), pp. 1–37.

[GS21a] Mostafa Ghobaei-Arani and Ali Shahidinejad. "An efficient resource provisioning approach for analyzing cloud workloads: a metaheuristic-based clustering approach". In: *The Journal of Supercomputing* 77.1 (2021), pp. 711–750.

[GS21b] Laurence Goasduff and C Stamford. "Gartner Says Cloud Will Be the Centerpiece of New Digital Experiences". In: *Retrieved December* 9 (2021), p. 2022.

[Guo+17] Zhenyu Guo et al. "A survey on load balancing algorithms for virtual machines in cloud computing". In: *Concurrency and Computation: Practice and Experience* 29.12 (2017), e4123.

[He+11] Haibo He, Sheng Chen, Kang Li, and Xin Xu. "Incremental learning from stream data". In: *IEEE Transactions on Neural Networks* 22.12 (2011), pp. 1901–1914.

[HMM18] Ying-Feng Hsu, Kazuhiro Matsuda, and Morito Matsuoka. "Self-aware workload forecasting in data center power prediction". In: *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE. 2018, pp. 321–330.

[Ios+11] Alexandru Iosup, Simon Ostermann, M Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema. "Performance analysis of cloud computing services for many-tasks scientific computing". In: *IEEE Transactions on Parallel and Distributed systems* 22.6 (2011), pp. 931–945.

[Isl+12] Sadeka Islam et al. "Empirical prediction models for adaptive resource provisioning in the cloud". In: *Future Generation Computer Systems* 28 (2012), pp. 155–162.

[Isl+23] Rafia Islam, Vardhan Patamsetti, Aparna Gadhi, Ragha Madhavi Gondu, Chinna Manikanta Bandaru, Sai Chaitanya Kesani, and Olatunde Abiona. "The future of cloud computing: benefits and challenges". In: *International Journal of Communications, Network and System Sciences* 16.4 (2023), pp. 53–65.

[Jon+18] Nicola Jones et al. "How to stop data centres from gobbling up the world's electricity". In: *nature* 561.7722 (2018), pp. 163–166.

[KGS18] Jitendra Kumar, Rimsha Goomer, and Ashutosh Kumar Singh. "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters". In: *Procedia computer science* 125 (2018), pp. 676–682.

[Kha+22] Tahseen Khan, Wenhong Tian, Guangyao Zhou, Shashikant Ilager, Mingming Gong, and Rajkumar Buyya. "Machine learning (ML)-centric resource management in cloud computing: A review and future directions". In: *Journal of Network and Computer Applications* 204 (2022), p. 103405.

[Kid+22] Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. "When and How to Retrain Machine Learning-based Cloud Management Systems". In: *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2022, pp. 688–698. DOI: 10.1109/IPDPSW55747.2022.00120.

[Kid+23] Lidia Kidane, Paul Townend, Thijs Metsch, and Erik Elmroth. "Automated Hyperparameter Tuning for Adaptive Cloud Workload Prediction". In: *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*. 2023, pp. 1–8.

[KM07] J Zico Kolter and Marcus A Maloof. "Dynamic weighted majority: An ensemble method for drifting concepts". In: *The Journal of Machine Learning Research* 8 (2007), pp. 2755–2790.

[Kop+21] Saraswati Koppad, Annappa B, Georgios V Gkoutos, and Animesh Acharjee. "Cloud Computing Enabled Big Multi-Omics Data Analytics". In: *Bioinformatics and Biology Insights* 15 (2021). PMID: 34376975, p. 11779322211035921. DOI: 10.1177/11779322211035921. eprint: https://doi.org/10.1177/11779322211035921. URL: https://doi.org/10.1177/11779322211035921.

[Li+14] Jing Li et al. "Efficient and fair resource allocation in cloud computing environments". In: *IEEE Transactions on Parallel and Distributed Systems* 25.12 (2014), pp. 3124–3135.

[Liu+20] Yang Liu et al. "Deep learning-based workload prediction for cloud resource management". In: *IEEE Transactions on Cloud Computing* 8 (2020), pp. 1233–1245.

[Lu+18] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. "Learning under concept drift: A review". In: *IEEE transactions on knowledge and data engineering* 31.12 (2018), pp. 2346–2363.

[Lu+21] Yang Lu et al. "Concept drift adaptation for cloud workload prediction". In: *IEEE Transactions on Cloud Computing* 9 (2021), pp. 1234–1248.

[Mah+22] Sami Mahmoud et al. "Predictive resource allocation for hybrid clouds: A meta-learning approach". In: *Journal of Cloud Computing* 11 (2022), pp. 1–22.

[MG11] Peter Mell and Tim Grance. "The NIST definition of cloud computing". In: *NIST Special Publication* 800 (2011), p. 145.

[Mic24] Microsoft Corporation. *Hyper-V Documentation*. Accessed: 2025-04-24. Microsoft. 2024. URL: https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/.

[Mir+19] Seyedehmehrnaz Mireslami, Logan Rakai, Mea Wang, and Behrouz Homayoun Far. "Dynamic cloud resource allocation considering demand uncertainty". In: *IEEE Transactions on Cloud Computing* 9.3 (2019), pp. 981–994.

[Mis10] Dave Mishchenko. *VMware ESXi: Planning, implementation, and security*. Course Technology Press, 2010.

[MK15] Veena Mittal and Indu Kashyap. "Online methods of learning in occurrence of concept drift". In: *International Journal of Computer Applications* 117.13 (2015).

[MK20] Mohammad Masdari and Afsane Khoshnevis. "A survey and classification of the workload forecasting methods in cloud computing". In: *Cluster Computing* 23.4 (2020), pp. 2399–2424.

[MSB15] Bruno Iran Ferreira Maciel, Silas Garrido Teixeira Carvalho Santos, and Roberto Souto Maior Barros. "A lightweight concept drift detection ensemble". In: *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2015, pp. 1061–1068.

[Nav+16] P Naveen, Wong Kiing Ing, Michael Kobina Danquah, Amandeep S Sidhu, and Ahmed Abu-Siada. "Cloud computing for energy management in smart grid-an application survey". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 121. 1. IOP Publishing. 2016, p. 012010.

[Par+22] Fatemeh Khoda Parast, Chandni Sindhav, Seema Nikam, Hadiseh Izadi Yekta, Kenneth B Kent, and Saqib Hakak. "Cloud computing security: A survey of service-based models". In: *Computers & Security* 114 (2022), p. 102580.

[Pre+14] John J Prevost, Kranthimanoj Nagothu, Mo Jamshidi, and Brian Kelley. "Optimal calculation overhead for energy efficient cloud workload prediction". In: *2014 World Automation Congress (WAC)*. IEEE. 2014, pp. 741–747.

[Ram23] Vijay Ramamoorthi. "Applications of AI in Cloud Computing: Transforming Industries and Future Opportunities". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 9.4 (2023), pp. 472–483.

[RMW19] Akond Rahman, Rezvan Mahdavi-Hezaveh, and Laurie Williams. "A systematic mapping study of infrastructure as code research". In: *Information and Software Technology* 108 (2019), pp. 65–77.

[SAE12] Omar Sefraoui, Mohammed Aissaoui, and Mohsine Eleuldj. "OpenStack: toward an open-source solution for cloud computing". In: *International Journal of Computer Applications* 55.3 (2012).

[Seh+18] Naresh Kumar Sehgal, Pramod Chandra P Bhatt, Naresh Kumar Sehgal, and Pramod Chandra P Bhatt. "Cloud workload characterization". In: *Cloud Computing: Concepts and Practices* (2018), pp. 61–83.

[Sha+21] Amir Shahidinejad et al. "Resource provisioning in edge-cloud federated environments". In: *IEEE Transactions on Services Computing* (2021).

[Shi+16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge computing: Vision and challenges". In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646.

[Sin18] Manjeet Singh. "Virtualization in cloud computing-a study". In: *2018 international conference on advances in computing, communication control and networking (ICACCCN)*. IEEE. 2018, pp. 64–67.

[Smi17] Randall Smith. *Docker orchestration*. Packt Publishing Ltd, 2017.

[Sun+19] Yange Sun, Zhihai Wang, Jidong Yuan, and Wei Zhang. "Tracking recurring concepts from evolving data streams using ensemble method." In: *Int. Arab J. Inf. Technol.* 16.6 (2019), pp. 1044–1052.

[TBG20] Hajer Toumi, Zaki Brahmi, and Mohammed Mohsen Gammoudi. "Extended hoeffding adaptive tree based-server load prediction in cloud computing environment". In: *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*. 2020, pp. 161–168.

[The+17] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. "Lossy image compression with compressive autoencoders". In: *arXiv preprint arXiv:1703.00395* (2017).

[Thö15] Johannes Thönes. "Microservices". In: *IEEE software* 32.1 (2015), pp. 116–116.

[Tur18] James Turnbull. *Monitoring with Prometheus*. Turnbull Press, 2018.

[Urd+09] Guido Urdaneta et al. "Wikipedia workload analysis for decentralized hosting". In: *Computer Networks* 53 (July 2009), pp. 1830–1845. DOI: 10.1016/j.comnet.2009.02.019.

[VHM20] Jan Vom Brocke, Alan Hevner, and Alexander Maedche. "Introduction to design science research". In: *Design science research. Cases* (2020), pp. 1–13.

[Wan+21] Liang Wang et al. “Workload characterization in modern cloud services”. In: *IEEE Transactions on Cloud Computing* 9 (2021), pp. 1243–1256.

[Wan+22] Can Wang, Mitra Baratchi, Thomas Bäck, Holger H Hoos, Steffen Limmer, and Markus Olhofer. “Towards time-series feature engineering in automated machine learning for multi-step-ahead forecasting”. In: *Engineering Proceedings* 18.1 (2022), p. 17.

[Xia+23] Qiuyan Xiang, Lingling Zi, Xin Cong, and Yan Wang. “Concept drift adaptation methods under the deep learning framework: A literature review”. In: *Applied Sciences* 13.11 (2023), p. 6515.

[Xu+13] Hong Xu et al. “Does cloud computing need a variance analysis?” In: *IEEE International Symposium on Workload Characterization*. 2013, pp. 51–60.

[Xu+22] Minxian Xu, Chenghao Song, Huaming Wu, Sukhpal Singh Gill, Kejiang Ye, and Chengzhong Xu. “esDNN: deep neural network based multivariate workload prediction in cloud computing environments”. In: *ACM Transactions on Internet Technology (TOIT)* 22.3 (2022), pp. 1–24.

[Yan+14] Jingqi Yang, Chuanchang Liu, Yanlei Shang, Bo Cheng, Zexiang Mao, Chunhong Liu, Lisha Niu, and Junliang Chen. “A cost-aware auto-scaling approach using the workload prediction in service clouds”. In: *Information Systems Frontiers* 16 (2014), pp. 7–18.

[Yu+19] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. “A review of recurrent neural networks: LSTM cells and network architectures”. In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[ZCB10] Qi Zhang, Lu Cheng, and Raouf Boutaba. “Cloud computing: state-of-the-art and research challenges”. In: *Journal of internet services and applications* 1.1 (2010), pp. 7–18.

[Zha+21] Yushan Zhang et al. “Interpretable deep learning for resource allocation in cloud computing”. In: *Proceedings of the ACM Symposium on Cloud Computing*. 2021, pp. 456–471.

[ZL16] Barret Zoph and Quoc V Le. “Neural architecture search with reinforcement learning”. In: *arXiv preprint arXiv:1611.01578* (2016).