Machine Learning-driven Resource Management for Enhanced System Performance

Project Team: Chaitanya Lakshmi Chava (280687527), Chih-Chia Chen (425269124), Rami-Lionel Kuttab (232798551), Shravan Sailada (528301335), Yaswanth Krishna Vayalapalli (376926618)

Abstract - **Machine learning (ML) is gaining popularity as a tool to optimize computing performance through better resource management. This research paper explores the use of ML algorithms to analyze computer resource usage data and system performance metrics to identify opportunities to optimize resource allocation and utilization, resulting in improvements in overall system performance, reduced energy consumption, and increased longevity of system components. The success of ML techniques can be used to optimize the utilization of available components such as RAM, GPU, and memory in computer system architecture. This involves analyzing past data to allocate resources according to the priorities assigned to different applications, ensuring that critical situations are handled in a prioritized manner. This research paper proposes an idea presenting the power of machine learning to optimize resource allocation based on system demands and historical data. The framework involves collecting system data, training machine learning models on this data, and using these models to make resource allocation decisions dynamically. One of the major advantages of ML-driven resource management over traditional methods is its ability to adapt to changing system demands in real-time, resulting in more efficient resource utilization and better system performance. ML-driven resource management has the potential to revolutionize the way computer systems are managed and optimized. With the increasing complexity of modern computer systems, ML-driven resource management is becoming an essential tool for achieving peak performance and efficiency. This research paper highlights the potential of machine learning as a powerful tool for optimizing computer system performance through resource management. In conclusion, the proposed ML-driven framework for resource management can adapt to changing system demands in real-time, making it an essential tool for achieving peak performance and efficiency in a wide range of applications. By leveraging the power of ML algorithms to predict system demands and adjust resource allocation dynamically, it is possible to achieve better system performance and resource utilization in various computer systems.**

1. INTRODUCTION

Resource management is a critical process in enhancing system performance in various domains such as cloud computing, edge computing, and wireless networks. With the emergence of big data and the Internet of Things (IoT), traditional resource management techniques have become insufficient to handle the exponentially growing available data and devices that need to be serviced. Machine learning algorithms have proven to be an effective approach for solving complex resource management problems. In this paper, The significance of machine learning-driven resource management and its applications in various domains will be explored, as well as how machine learning techniques can overcome the challenges faced by traditional systems and provide an overview of the various machine learning algorithms that can be used for resource management. Furthermore, The importance of data preprocessing, feature selection, and model selection in achieving optimal system performance. Finally, some case studies and real-world applications are presented where machine learning-driven resource management has been successfully implemented to improve system performance. In particular, the focus will be placed on the potential of machine learning algorithms to optimize resource allocation and utilization within existing computer systems, leading to improvements in overall system performance, reduced energy consumption, and increased longevity of system components.

Machine learning (ML) is a subset of artificial intelligence (AI) that focuses on creating algorithms that can learn from data and make predictions or decisions based on that learning. One of the key advantages of ML is its ability to analyze vast amounts of data and extract patterns that can be used to make accurate predictions. This makes ML a valuable tool for a wide range of applications, including computer system performance optimization. By using ML algorithms to analyze computer resource usage data and system performance metrics, it is possible to identify opportunities to optimize resource allocation and utilization within existing computer systems. For example, a machine learning model could be trained to predict when certain resources in the system may become unoccupied or underutilized, and then automatically adjust resource allocation accordingly to take advantage of those opportunities. This could lead to improvements in overall system performance, reduced energy consumption, and increased longevity of system components.

The incorporation of machine learning in computer system architecture aims to enhance system efficiency by optimizing the utilization of available components such as RAM, GPU, and memory. By using machine learning algorithms to determine the priorities of instructions registered in the CPU when running multiple applications simultaneously, the system can allocate resources according to the priorities assigned to different applications, ensuring that critical situations are handled in a prioritized manner. Additionally, users can customize application priorities to better align with their requirements. These features help minimize recurring issues and improve system efficiency by tailoring resource allocation to meet user needs.

Furthermore, ML models have proven to be invaluable in terms of tackling existing problems as well as future challenges, resulting in both more accurate results that reflect real-world weather patterns and shorter processing time of data compared to traditional weather forecasting models. ML has also been applied to hardware acceleration for graphics processing units (GPUs) to develop precise predictive models for optimizing GPU performance.

In conclusion, the proposed ML-driven framework for resource management has the potential to revolutionize the way computer systems are managed and optimized. With the increasing complexity of modern computer systems, ML-driven resource management is becoming an essential tool for achieving peak performance and efficiency. By leveraging the power of ML algorithms to predict system demands and adjust resource allocation dynamically, it is possible to achieve better system performance and resource utilization in various computer systems.

2. LITERATURE REVIEW

2.1 Overview

Based on the information provided by the sources that will be cited in the reference section, there are several trend that can be observed. First, it is possible to implement a dynamic system resource allocation scheme trained via ML on various applications and in this paper specifically, the following will be explored: cloud computing resource allocation, local memory management, Internet resource management, and computing efficiency.

2.2 ML-driven Cloud Computing Resource Allocation Scheme

Historically speaking, the upfront cost to purchasing state-of-the-art or otherwise comparable computation hardwares are tremendous. Such cost presented a difficult barrier-to-entry for newer organizations as well as businesses since computing by this point is a necessary resource in order to achieve success. However, as cloud technology becomes more mature, additional tasks and applications can now be delegated from the consumer or user side to the cloud. However, that shifted the problem of needing computational resource instead to the IaaS (Infrastructure-as-a-Service) providers such as Amazon, Google and Microsoft. Namely as new and upcoming technology and application becomes more integrated with the cloud and by extension the internet, there will be more demand for computational resources. However, increasing available resources physically takes time and capital and considering the reality that computational demand are variable, it is not necessarily always the best solution to increase those resources physically. For instance, during daytime when most people are either working or in school and so on, an online streaming business like netflix is unlikely to se the huge spike in demand vs when in the evening people are now relaxing at home and watching shows. In order to service millions of users at once, Clouds heavily rely on virtualization technologies which partitions an existing physical system into multiple independent VMs (Virtual Machines). However, that introduced a new problem since for many reasons such as security, these VMs can not have access to information to other VMs on the same system. As such, this do not ensure performance isolation and that since each VM has its own resource scheduler, which manages a physical shared resources without visibility of others, this could lead to inconsistent performance across VMs as it is difficult to determine appropriate scaling policies in a dynamic non-stationary environment [1].

Traditionally speaking, threshold based policies have been the most widely used mechanism for auto scaling applications in the cloud, but they have limitations in terms of their ability to support higher business functions and objectives. However, threshold based policy do not facilitate higher business functions or objectives, namely adapting system resource allocation based on dynamic event. Some proposed solutions aim to address this issue, such as Claudia, which operates at a higher level of abstraction but still requires domain knowledge and ad hoc rules. Elastic storage solutions also use threshold based approaches, but they lack predictive power and may not respond optimally to fast-changing events [1]. Reinforcement learning methods are introduced as an alternative, which can reason under uncertainty based on environmental observations and adapt to suit the environment based on its own experience [1]. A key benefit of reinforcement learning is its ability to handle changes in the workload request model without requiring a model change for the threshold based approach.

One way to create a ML driven resource allocation scheme was to utilize Q-learning as it does not require a complete model of the environment which is very useful for conducting experiments because the access to environment may not always be available. The following figure 4 from Barrett et al.'s (2013) study presents a high-level abstraction for a parallel Q-learning architecture in a typical cloud environment:

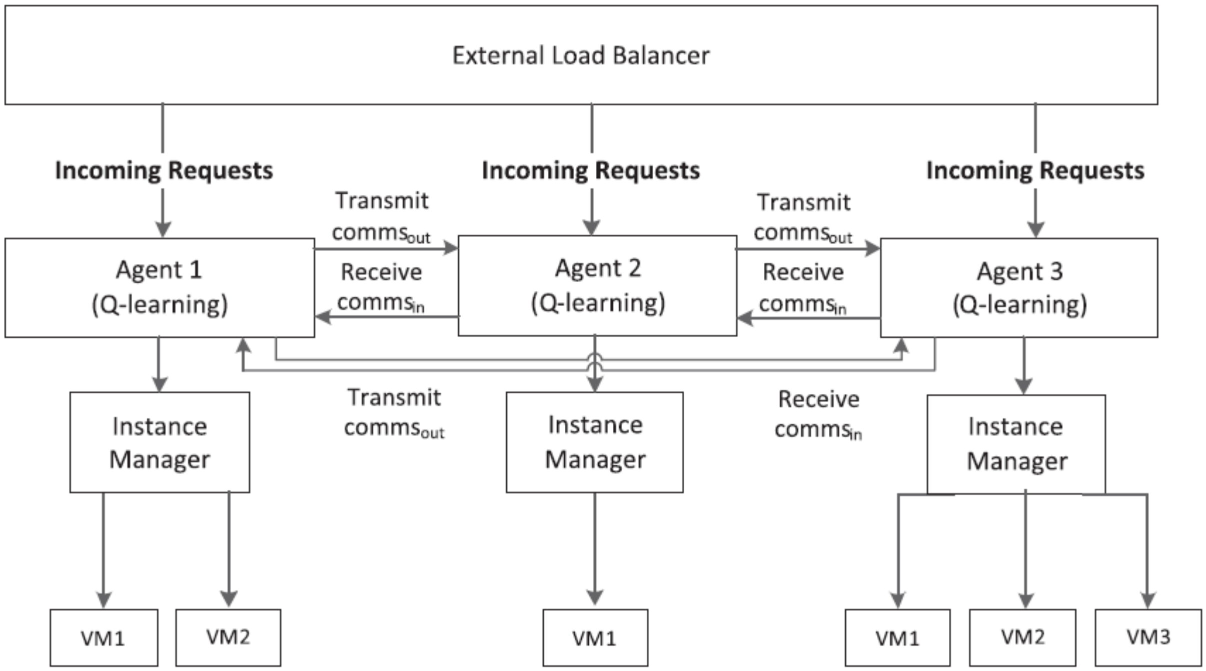

Figure-2.1: Parallel Q-Learning Architecture

Essentially, the figure describes the use of a multiple independent agents system which operate independently. These agents all experiences rewards and penalties independently based on its own decision about incoming user requests. Then the agents share information with each other about their observations and communicate directly with all other agents in the system. The agents attempt to learn an individually optimal policy for allocating resources, and the instance manager executes actions based on the agents' instructions, such as adding, removing, or maintaining the existing number of allocated virtual machines. This approach ensures that each agent's environment is insular, and learning in parallel does not introduce non-stationarity, which in the context of cloud based environment describe the irregularity and unpredictable nature of the arrival of computational demand from users, into other agent’s environments [1].

This ML-driven resource allocation scheme proves to be better than the previously mentioned traditional threshold-based policy because the ML-driven scheme is not limited by the lack of domain and application knowledge, and that reinforcement learning algorithms can adapt to changes over time in terms of workloads, application updates, resource variability and other variables. Also, since each agents can communicate with each other, this helps with reducing learning time needed to reach a stable allocation strategy because there are more data aggregates for policy adjustment [1]. Overall, ML-driven resource allocation seems to be a promising alternative for the IaaS providers as opposed to the traditional threshold based policy approach.

2.3 Local Resource Management

Jose Martinez article discusses the use of machine learning techniques for dynamic multicore resource management in on-chip hardware agents. Multicore architectures have become the primary mechanism to reap the benefits of Moore's Law in the billion-transistor era, but not all multicore resources scale easily. Effective resource management in future multicore architectures will be important for delivering cost-effective, high-performance products. The article proposes using machine learning to tackle resource allocation, scheduling, load balancing, and design space exploration problems in dynamic multicore resource management. The expert's time is best invested in determining which objective function to optimize, which automatic learning technique best suits the problem, and which variables should drive the resulting self-optimizing mechanism at runtime. This approach contrasts with the traditional approach of directly specifying at design time how the hardware should accomplish the desired goal. Overall, the article highlights the potential of machine learning techniques in producing self-optimizing on-chip hardware agents capable of learning, planning, and continuously adapting to changing workload demands. This can result in more efficient and flexible management of critical hardware resources at runtime, which is essential for delivering cost-effective, high-performance products in future multicore architectures.

Federated learning has emerged as a promising solution for training machine learning models on decentralized data held by mobile devices. While this technique has garnered significant attention, the use of resource-constrained mobile devices raises concerns about energy efficiency, which can significantly impact learning time and model accuracy. Unfortunately, existing work on federated learning has primarily focused on developing fast learning algorithms with provable convergence speed, while largely ignoring the critical issue of energy efficiency. To address this challenge, the authors of the paper propose a deep reinforcement learning-based approach to optimize the CPU-cycle frequency of mobile devices in federated learning. The proposed approach involves formulating an optimization problem that minimizes the total system cost, which is a weighted sum of training time and energy consumption. To solve the optimization problem, a DRL agent is designed, which allocates computational resources to mobile devices based on observed learning speeds in previous iterations. The agent is trained offline using the PPO algorithm and can converge to near-optimal solutions without knowledge of network quality. The proposed approach has been evaluated through experiments and simulations and has been shown to outperform existing solutions. Additionally, the authors highlight the importance of designing the state, action, and reward for the DRL method and demonstrate the superiority of their proposed design. The use of DRL and PPO for efficient training and adaptation to the dynamic environment of federated learning are also key features of the proposed algorithm. Overall, the paper makes a valuable contribution to the literature on federated learning by addressing the critical issue of energy efficiency for mobile devices. The proposed approach offers a novel solution to the optimization problem of federated learning and demonstrates the effectiveness of deep reinforcement learning in this context. The results of the experiments and simulations suggest that the proposed algorithm can significantly improve system performance compared to existing approaches. By doing so, the authors have provided a new direction for research in federated learning, which can lead to more sustainable and efficient use of mobile devices for machine learning applications.

2.4 Internet Resource Management

Resource management in computer systems and networks is a complex problem due to the heterogeneity and dynamic nature of the resources involved. Traditional approaches to resource allocation and scheduling rely on static heuristics or mathematical optimization models, which may not be able to cope with the increasing scale and complexity of modern systems. Machine learning, and in particular reinforcement learning (RL), has shown promise in addressing these challenges by enabling systems to learn from experience and adapt to changing conditions. Mao et al. propose the use of deep reinforcement learning (DRL) for resource management in clusters, where multiple tasks with different resource demands compete for shared resources. The authors introduce a new framework called DeepRM, which translates the problem of packing tasks with multiple resource demands into a learning problem. The goal is to learn a policy that maximizes the overall system utility while meeting the constraints of the available resources.

The authors argue that RL approaches are well-suited to resource management systems due to the repetitive nature of decisions and the ability to model complex systems using deep neural networks. They compare DeepRM to state-of-the-art heuristics for resource allocation and show that it achieves comparable performance while also being adaptable to different conditions and learning sensible strategies from experience. The authors also discuss some of the challenges and limitations of using DRL for resource management, such as the need for large amounts of training data, the difficulty of handling continuous state and action spaces, and the potential for overfitting. They suggest some possible directions for future research, such as incorporating domain knowledge into the learning process, exploring alternative RL algorithms, and developing methods for handling uncertainty and partial observability.

Overall, the article presents a compelling case for the use of DRL in resource management systems and provides a promising example of how it can be applied in practice. The results suggest that DRL can be a viable alternative to traditional approaches and can potentially lead to improvements in system performance and efficiency.

2.5 Computational Efficiency

The paper "Managing Power Consumption and Performance of Computing Systems Using Reinforcement Learning" presents a novel approach to managing the power consumption and performance of large-scale IT systems such as data centers. The proposed approach uses reinforcement learning (RL) to train policies for a CPU frequency controller embedded in the Blade servers' firmware, with a multi-criteria reward signal based on application performance and CPU power consumption.The study addresses the challenges of limited decision sampling rates and multiple reward functions by using innovative practical solutions. The results show performance improvements compared to hand-designed policies and "cookbook" RL implementations. This is an important development, as energy consumption is a growing concern in the IT industry, and researchers are exploring intelligent power control of processors, memory chips, and whole systems to reduce power consumption and heat densities. The experimental testbed consists of a Workload Generator that produces HTTP-based workloads of varying intensity, which are routed to a collection of blade servers contained in a single chassis. The RL-based power manager and commercial performance manager aim to optimize a joint power-performance objective cooperatively by adjusting control parameters while sharing information. The state is characterized by observable performance, power, and load intensity metrics collected in a data collection module, while the action is a throttling of CPU frequency achieved by setting a powercap on each blade.

The paper also discusses the use of RL to learn effective control policies for complex systems such as a population of human users interacting with a commercial web application. The researchers have developed innovative tricks to improve RL performance in a particular application domain, such as training two separate function approximators to estimate future discounted reward components, designing a new type of neuronal output unit to learn performance-related values, and dividing the entire training set into subsets to address a specific rate limitation in the system.

In summary, the paper presents a successful application of batch RL with nonlinear function approximation to the autonomic management of power and performance in web application servers. The approach achieved high-quality management policies that outperformed the best hand-crafted policy, resulting in power savings while maintaining performance. This research is a significant contribution to the field and has important implications for energy-efficient management of large-scale IT systems.

3. POTENTIAL PROBLEMS AND FUTURE DIRECTIONS

3.1 Potential Problems with Machine Learning for Resource Management

The potential problems and future directions in machine learning-driven resource management have significant implications for the development of intelligent systems. Although machine learning algorithms have shown great promise in enhancing system performance, there are several challenges that need to be addressed to ensure that these systems can be effectively implemented in real-world scenarios.

3.1.1 Data Quality

One of the main challenges in implementing machine learning algorithms is data quality. Poor data quality can lead to inaccurate predictions and poor system performance. Therefore, techniques such as data cleansing, normalization, and augmentation can be employed to improve data quality. Data cleansing involves removing or correcting inaccurate or incomplete data, normalization involves scaling the data to a standard range, and augmentation involves generating new data to increase the size of the dataset.

3.1.2 Overfitting

Another challenge in machine learning is overfitting, which occurs when the model is too complex and fits the training data too closely. This can lead to poor performance on new data. Regularization techniques, cross-validation, and hyperparameter tuning can be used to address this issue. Regularization techniques involve adding a penalty term to the model to discourage overfitting, to perform cross-validation, it is necessary to divide the data into sets for training and validation purposes to evaluate model performance, and hyperparameter tuning involves adjusting the model parameters to optimize performance.

3.1.3 Expensive

Machine learning algorithms can also be computationally expensive and require significant resources such as memory and processing power. This can be addressed by using techniques such as model compression, distributed training, and hardware acceleration. Model compression involves reducing the size of the model without significantly impacting performance, distributed training involves training the model across multiple devices to reduce the training time, and hardware acceleration involves using specialized hardware such as GPUs to speed up model training and inference.

3.1.4 Privacy and Security

Privacy and security are also important considerations in machine learning-driven resource management. Techniques such as secure federated learning, differential privacy, and robustness to adversarial attacks can be employed to address these concerns. Secure federated learning involves training the model on data that is distributed across multiple devices without sharing the data, differential privacy involves adding noise to the data to protect the privacy of individual records, and robustness to adversarial attacks involves designing the model to be resistant to attacks that aim to manipulate the model's output.

3.1.5 Expandability

As machine learning algorithms become more complex, understanding how they make decisions becomes increasingly important. XAI techniques such as model visualization, feature importance analysis, and decision rule extraction can be employed to make machine learning algorithms more transparent and interpretable. Model visualization involves generating visual representations of the model to help users understand its internal workings, feature importance analysis involves identifying which features of the data are most important in the model's decisions, and decision rule extraction involves extracting rules from the model that can be easily understood by humans.

3.1.6 Lack of Diversity

Another potential problem with machine learning in resource management is the lack of diversity in the training data. If the training data is biased or does not represent the full range of possible scenarios, the resulting model may not perform well in real-world situations. To address this issue, techniques such as data augmentation and diverse training data can be employed. Data augmentation involves generating new data from existing data to increase the diversity of the training data, and diverse training data involves collecting data from a wide range of sources and scenarios to ensure that the model is robust to different situations.

3.1.7 Ethical Concerns

Machine learning algorithms can also raise ethical concerns, particularly in resource management applications. For example, biased algorithms can lead to unfair resource allocation or discrimination. To address these concerns, techniques such as fairness metrics and ethical guidelines can be employed. Fairness metrics involve evaluating the performance of the model across different subgroups to ensure that it is not biased towards any particular group, and ethical guidelines involve developing guidelines for the use and deployment of machine learning algorithms to ensure that they are used in a responsible and ethical manner. Also, in one of the referenced sources, the paper does not provide a detailed discussion of the potential ethical implications of self-optimizing hardware agents [2].

3.1.8 Lack of Human Involvement

Finally, a potential problem with machine learning-driven resource management is the lack of human involvement. While machine learning algorithms can automate many tasks and improve system performance, they may not always take into account the broader context or human factors. Therefore, it is important to ensure that there is appropriate human oversight and input in the design and deployment of these systems. This can be achieved through techniques such as human-in-the-loop systems, where humans are involved in the decision-making process, or transparent decision-making processes that allow for greater understanding and input from stakeholders.

3.2 Future Directions for Enhancing Resource Management with Machine Learning

As the demand for computing resources continues to grow, there is a need for more efficient and effective resource management techniques. Machine learning has emerged as a powerful tool for resource management, enabling systems to optimize resource allocation based on real-time performance feedback. In this context, this article explores future directions for enhancing resource management with machine learning, including self-learning algorithms, edge computing, reinforcement learning, and combining machine learning with rule-based and expert systems. These approaches have the potential to revolutionize resource management, creating more efficient and effective systems that can adapt to changing conditions in real-time.

3.2.1 Self-Learning Algorithms:

Self-learning algorithms can adapt to changing environments without human intervention, enabling more efficient resource management. These algorithms can continuously monitor system performance and adjust resource allocation to optimize performance. In the context of resource management, self-learning algorithms could be used to monitor and analyze patterns of resource usage, detect anomalies and take appropriate actions to optimize the usage of resources. Self-learning algorithms can also be used to predict resource requirements based on historical data and expected workload, and allocate resources accordingly. This can help in reducing resource waste, optimizing resource utilization, and enhancing overall system efficiency.

3.2.2 Edge Computing:

Edge computing is a computing model that involves the distribution of computational processes and data storage in close proximity to the location where data is generated, rather than relying on centralized cloud-based computing. By leveraging edge computing, resource management systems can improve response times, reduce network latency, and improve overall system performance. Machine learning can be used to optimize resource allocation at the edge by monitoring system performance, detecting anomalies, and making real-time decisions based on performance feedback. Edge computing can be especially useful in resource-constrained environments, where network bandwidth and latency are major concerns.

3.2.3 Reinforcement Learning:

Reinforcement learning is a machine learning technique that enables systems to learn how to allocate resources based on performance feedback. In the context of resource management, reinforcement learning can serve to optimize resource allocation based upon the system performance. For example, a reinforcement learning agent could learn to allocate more resources to a particular application if it is experiencing higher usage, while reducing resources allocated to less used applications. Reinforcement learning can help in optimizing resource usage in dynamic environments where resource requirements change frequently.

3.2.4 Combining Machine Learning with Rule-Based and Expert Systems:

Combining machine learning with other techniques such as rule-based systems or expert systems could enable more robust and efficient resource management. Rule-based systems use a set of predefined rules to make decisions, while expert systems use expert knowledge to make decisions. By combining these techniques with machine learning, resource management systems can leverage the strengths of each approach to create more effective and efficient systems. For example, a rule-based system could be used to set resource allocation policies, while machine learning could be used to optimize resource allocation within those policies based on performance feedback.

3.2.5 Federated Learning:

Federated learning is a type of machine learning technique that allows the training of machine learning models on the decentralized data. In the context of resource management, federated learning can also be used to train machine learning models on data that is already distributed across multiple devices, without any need for the data to be centralized. This can help to address privacy concerns, as the data remains on the device and is not shared with a centralized server. Federated learning can also enable more efficient training of machine learning models, as it allows for distributed processing of data.

3.2.6 AutoML:

AutoML, or automated machine learning, refers to the use of machine learning algorithms to automate the process of model selection, hyperparameter tuning, and feature engineering. AutoML can enable more efficient development of machine learning models, as it automates the tedious and time-consuming aspects of the process. In the context of resource management, AutoML can be used to develop more accurate and efficient machine learning models for resource allocation and optimization.

3.2.7 Multi-Objective Optimization:

Multi-objective optimization involves optimizing multiple objectives simultaneously, rather than just a single objective. In the context of resource management, multi-objective optimization can be used to optimize multiple performance metrics simultaneously, such as response time, throughput, and energy consumption. This can enable more comprehensive and efficient resource management, as the system can optimize multiple metrics simultaneously.

3.2.8 Explainable AI:

Explainable AI, or XAI, refers to the use of machine learning techniques to create models that are transparent and explainable. In the context of resource management, XAI can help to increase user trust and understanding of the system, by enabling users to understand how the system is making decisions. This can be particularly important in resource management systems, where the consequences of incorrect decisions can be significant.

In this context, the future directions for enhancing resource management with machine learning involve using self-learning algorithms, edge computing, reinforcement learning, and combining machine learning with rule-based and expert systems. These approaches can help in creating more efficient, robust, and transparent resource management systems that can adapt to changing conditions in real time.

4. CONCLUSION

In this research paper, the potential of machine learning-driven resource management to optimize computer system performance through better resource allocation and utilization is explored widely. By analyzing past data and system performance metrics, machine learning algorithms can identify opportunities to optimize resource allocation, resulting in significant improvements in overall system performance, reduced energy consumption, and increased longevity of system components. The research work has also discussed the importance of data preprocessing, feature selection, and model selection in achieving optimal system performance. Furthermore, this paper has discussed research papers and real-world applications where machine learning-driven resource management has been successfully implemented.

Machine learning-driven resource management has the potential to revolutionize the way computer systems are managed and optimized. By leveraging the power of machine learning algorithms to predict system demands and adjust resource allocation dynamically, it is possible to achieve better system performance and resource utilization in various computer systems. The proposed framework can adapt to changing system demands in real-time, making it an essential tool for achieving peak performance and efficiency.

ACKNOWLEDGMENT

This paper is the result of the collaborative effort of our team members who have contributed their vast industry knowledge and expertise to make this work possible. Although not explicitly cited in the referenced sources, the extensive information presented in this paper pertaining to specific topics is derived largely from our team's practical experience in the field of machine learning-driven resource management. Their insights have been invaluable in shaping our understanding of the subject matter and providing valuable insights into the practical aspects of resource management.

We also want to extend our sincere appreciation to the authors of the original research paper for their significant contribution to the field of machine learning and computer systems. Their research has laid the foundation for our work and inspired us to explore new areas of research and development. The summary provided in this paper aims to provide a comprehensive and accessible overview of the original research. We hope that this work will serve as a catalyst for future research and advancements in the field of machine learning-driven resource management, leading to better performance, efficiency, and sustainability of computer systems.

REFERENCES

[1] E. Barrett, E. Howley, and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Computat.: Pract. Exper., vol. 25, pp. 1656-1674, 2013, doi: 10.1002/cpe.2864.

[2] G. Tesauro, R. Das, H. Chan, J. Kephart, D. Levine, F. Rawson, and C. Lefurgy, "Managing power consumption and performance of computing systems using reinforcement learning," in Proc. 21st Annual Conference on Neural Information Processing Systems (NIPS '07), 2007.

[3] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource Management with Deep Reinforcement Learning," in Proc. 15th ACM Workshop on Hot Topics in Networks (HotNets '16), New York, NY, USA, 2016, pp. 50-56, doi: 10.1145/3005745.3005750.

[4] J. F. Martinez and E. Ipek, "Dynamic Multicore Resource Management: A Machine Learning Approach," in IEEE Micro, vol. 29, no. 5, pp. 8-17, Sept.-Oct. 2009, doi: 10.1109/MM.2009.77.

[5] Y. Zhan, P. Li and S. Guo, "Experience-Driven Computational Resource Allocation of Federated Learning by Deep Reinforcement Learning," 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 2020, pp. 234-243, doi: 10.1109/IPDPS47924.2020.00033.