OPEN ACCESS
Jurnal Ilmiah Ilmu Terapan Universitas Jambi
Vol. 9 No. 1. March 2025, pp. 161-177 DOI. 10.22437/jiituj.v9i1.38410

Research Article

MODELLING AND ANALYSIS OF OPTIMIZATION ALGORITHMS

Nurkasym Arkabaev[1,*], Elshan Rahimov[2], Alisher Abdullaev[3], Harish Padmanaban[4], Vugar Salmanov[5]

[1] Institute of Mathematics, Physics, Engineering and Information Technologies, Osh State University, Osh, Kyrgyz Republic
[2] Information Technology Department, Baku Higher Oil School, Baku, Azerbaijan
[3] Faculty of Physics and Mathematics, Nukus State Pedagogical Institute named after Ajiniyaz, Nukus, Uzbekistan
[4] JPMorgan Chase Tower, Houston, USA
[5] Department of Informatics. Nakhchivan State University, Nakhchivan, Azerbaijan
Corresponding author email: nurkasymarkabaev@gmail.com

Article Info
Received: Nov 13, 2024
Revised: Dec 23, 2024
Accepted: Feb 11, 2025
OnlineVersion: Feb 17, 2025

Abstract

The purpose of this study was to comprehensively analyze existing optimization algorithms for Machine Learning (ML) models and develop new approaches aimed at improving their performance and efficiency. The study compared traditional and novel machine learning optimization techniques to evaluate their impact on model performance. The main results include a detailed overview of the main optimization methods in ML, including gradient descent, stochastic gradient descent, metaheuristic-based methods, and non-zero methods. Specific cases of using optimization algorithms in ML tasks, such as image processing, machine translation, and speech recognition were presented. A table comparing the advantages and disadvantages of the methods by key performance metrics is provided. The structural diagrams and principles of operation of each method are presented. In addition, the methods of integrating the developed approaches into existing ML platforms are investigated. The study's results demonstrate that integrating novel optimization techniques significantly enhances machine learning model performance. These methods offer a substantial improvement over traditional techniques like gradient descent, providing greater flexibility and efficiency in handling complex and evolving data. The findings suggest that combining these approaches with existing optimization strategies can lead to more robust and scalable machine learning systems across diverse industries. The findings suggest that combining these methods with traditional approaches can enhance machine learning performance and guide future AI developments. The novelty of the research is in the introduction of the novel techniques like adaptive model selection and dynamic parameter adaptation to improve machine learning efficiency in real-time data environments.

Keywords: Model Performance, Parameter Adaptation, Performance Metrics, Resource Management, Selection Automation.

INTRODUCTION

Machine Learning (ML) is a branch of artificial intelligence (AI) aimed at creating algorithms that allow computers to learn from data and make decisions without explicit programming. An important aspect of ML is the optimization process, which is necessary to adjust the parameters of the model to minimize the loss function. Modern tasks requiring the analysis of large amounts of data and the construction of complex models pose new challenges in the field of optimization. As the complexity of real-world data increases, traditional optimization methods often struggle to meet the demands of dynamic, real-time applications. This research investigated cutting-edge methods including automated resource management, dynamic parameter adaptation, and adaptive model selection. The results offer insightful information for improving machine learning systems, increasing their adaptability, effectiveness, and capacity to manage complicated, dynamic data.

Despite significant advances in ML, there are a number of unresolved issues that limit its effectiveness and application. The main problems include the slow convergence of conventional optimization algorithms, which lengthens the training time of models, and the tendency of these algorithms to get stuck in local minima, which prevents the achievement of a global optimum and reduces the accuracy of models. Additionally, high computational complexity and the cost of processing large amounts of data create significant obstacles to using ML in real-world conditions.

For a deeper study of this topic, it is necessary to analyze similar research papers. For example, Li et al. (2022) showed that the use of Extreme Gradient Boosting (XGBoost) algorithm to analyze a complete data set allows for higher prediction accuracy compared to the generalized linear model. Different algorithms select different important variables, and XGBoost is especially effective when working with pre-selected variables. The integration of advanced statistical methods and ML algorithms improves the accuracy of forecasting, especially in real time (Hasanov & Mammadov, 2023; Essenzi, 2024; Romadhonsyah, 2024; Sriyono, 2024). The results showed a significant improvement in the accuracy of forecasts compared to conventional methods. Mamanazarova (2024) emphasized the importance of using AI and ML to improve stylistic analysis and gain in-depth knowledge about the interaction of textual characteristics and reader perception. The researchers discussed the prospects of using advanced analytical methods to significantly improve the understanding of the impact of literature on knowledge.

In turn, Hasan (2024) considered the use of AI and ML to optimize yields in U.S. agriculture. The researcher has developed an algorithm that provides high accuracy in predicting yields with a minimum standard error of 0.9412 and an average absolute error of 0.9874, considering up to 94.15% of data variability. Maurya et al. (2024) emphasized that the use of ML to optimize the design of electronic circuits is a promising approach that contributes to the systematic study of complex dependencies between circuit parameters and their performance. ML algorithms, such as neural networks and unsupervized learning methods, make it possible to effectively explore the design space, increasing the efficiency and reliability of electronic systems. The paper by Iqbal and Sheikh (2024) pointed out that ML plays a key role in optimizing compilers, improving the performance and efficiency of optimization selection. These approaches are actively used to solve complex problems of compiler optimization, offering new models and approaches to learning.

In addition, the study by Mzili and Arya (2024) in the field of optimization, ML and AI, covers a variety of topics, including deep learning algorithms, evolutionary methods and the application of ML in various fields such as healthcare, finance and robotics. Using the firefly algorithm for optimization and models such as random forest, decision tree regressor, nearest neighbour regressor, and support vector regressor for ML, the study evaluated their performance according to the criteria of the coefficient of determination and negative mean absolute error. Additionally, Gong et al. (2024) emphasized the importance of using ML and deep learning methods with reinforcement in managing cloud resources and optimizing virtual machine migration, drawing attention to their role in solving complexities and changes in cloud computing. Pimple (2024) analyzed the use of ML algorithms for load forecasting and efficient resource allocation in data centres, combining operations research methods with Python categorization algorithms such as Scikit-learn and TensorFlow.

This study's gap analysis highlights the differences between current machine learning optimization techniques and the changing requirements of contemporary AI applications. There is still a lack of integration with more dynamic and adaptive methodologies that can handle real-time data changes and increase system efficiency, even if many classic optimization techniques, such as gradient descent and stochastic gradient descent, have been thoroughly studied and used. Additionally, while new research highlights the significance of dynamic parameter adaptation and automated model selection, little is known about how these advancements might be successfully incorporated into current machine learning systems. To close this gap, this study investigates how well these new approaches work and how they might be used in conjunction with more established optimization techniques to enhance machine learning models' overall performance and adaptability in challenging real-world situations.

In general, the purpose of this study was to analyze existing algorithms and develop new optimization methods aimed at improving the performance of ML models. The research objectives include investigating practical examples of the application of existing ML optimization algorithms and integrating the developed methods into various platforms.

RESEARCH METHOD

This study employs a mixed-methods approach, combining both conceptual analysis and practical applications to evaluate and improve machine learning optimization algorithms. The sample consists of a variety of case studies from different machine learning tasks, such as image processing, machine translation, and speech recognition, allowing for a comprehensive examination of the optimization algorithms in use across different domains.

To achieve the purpose of the study, a detailed review of existing optimization algorithms used in the context of ML was conducted. This stage included the analysis of various optimization methods such as gradient descent, stochastic gradient descent, metaheuristic-based methods, and non-zero methods. The main characteristics, advantages and disadvantages of each method were investigated. As part of this stage, Python code examples were also written in the Replit environment to demonstrate the operation of some algorithms, which helped to understand their principles of operation and capabilities. Programmes for gradient and stochastic descent included generating random data for linear regression, adding a column with units for offset, initial parameter values, starting and parameters of descent, output of results, and visualization of the convergence process.

Then practical examples of the application of existing optimization algorithms were considered. Several cases were selected in which various optimization methods were used to solve specific ML problems. Technologies and tools such as DeepFace, FaceNet, Google Translate, and Amazon Transcribe were considered. In addition, a table was provided comparing the advantages and disadvantages of the studied algorithms in terms of accuracy, learning rate, and resource consumption. This allowed visualizing which optimization methods are most effective for different types of data and tasks.

Further, the development of new optimization methods aimed at improving the performance of ML models was carried out. The developed methods include adaptive automation of model selection, dynamic adaptation of model parameters, and automated resource management mechanisms. For each of these methods, block diagrams were created describing their operation and interaction with existing models. The adaptive model selection automation method includes a system for analysing data characteristics and automatically selecting the most appropriate models based on performance metrics. Dynamic parameter adaptation involves the use of algorithms that automatically adjust model parameters depending on current conditions and data characteristics. And the mechanisms of automated resource management include the use of algorithms to efficiently allocate computational resources between tasks and processes, minimizing delays and optimizing resource usage.

The last stage of the study was devoted to the integration of the developed methods into existing ML platforms. Key aspects of integration included compatibility assessment, a modular approach to integration, testing and validation, integration with workflows, and continuous improvement. Various tools based on the TensorFlow platform were considered, namely TensorFlow Model Optimization Toolkit, TensorFlow’s tf.train Application Programming Interface (API), TensorFlow Serving and Kubernetes.

Moreover, within the framework of the study, other relevant publications and works related to various approaches to model optimization were investigated, and a comparative analysis of research papers with this study was carried out. This helped to clarify and expand the understanding of current trends and achievements in the field of optimization and ML, and to identify potential areas for the development of the work undertaken.

The instruments used in this study include both qualitative and quantitative tools to gather data on the performance of various machine learning optimization algorithms. For qualitative data collection, a comprehensive literature review was conducted, focusing on existing research papers, global reports, and case studies related to optimization techniques. This review helped identify the key performance indicators and challenges faced by different machine learning algorithms. For quantitative data collection, the study employed Python-based tools and machine learning platforms like TensorFlow and Replit to implement and test various algorithms. Performance metrics such as accuracy, learning rate, and resource consumption were collected during the experiments. Additionally, the data collection grid included structured data points from experiments, with performance values recorded for each task and algorithm tested. These data collection methods ensured a thorough and reliable assessment of the optimization algorithms' effectiveness across various machine learning tasks.

Statistical analysis in this study was conducted to assess the performance of various machine learning optimization algorithms across different tasks. Descriptive statistics, including measures of central tendency (mean, median) and variability (standard deviation), were calculated to summarize the performance data. To compare the effectiveness of different algorithms, inferential statistical tests, such as t-tests, were employed to determine whether there were significant differences in performance metrics, such as accuracy, learning rate, and computational efficiency. Additionally, regression analysis was used to explore the relationship between key parameters and the overall performance of the algorithms. The results of these statistical tests were used to draw conclusions about the relative strengths and weaknesses of the optimization methods under different conditions, ensuring the robustness and validity of the findings.

RESULTS AND DISCUSSION

Overview of Optimization Algorithms in the Context of ML

Optimization of algorithms in ML plays a key role in improving the performance and efficiency of models. Existing optimization methods include both classical approaches, such as gradient descent and its variations, and more complex methods based on metaheuristics. It is important to consider the variety of approaches to optimizing ML models and their practical application. ML often comes down to finding the optimal parameters of models to achieve the best quality of predictions based on data. This process requires minimizing the loss functions or maximizing the estimation functions, which is achieved using various optimization methods (Almonia, 2024; Komiya, 2024; Majeed et al., 2024; Rahmi, Adawiyah, & Dilaro, 2024; Yusipa, 2024). Optimal model parameters allow models to adapt to data and make accurate predictions, which is critical for solving a wide range of tasks, from classification to regression and clustering. One of the most common optimization methods in ML is gradient descent. It is used to minimize loss functions by iteratively updating model parameters in the direction opposite to the gradient of the loss function. An example of the code for gradient descent is shown in Figure 1.

```
np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

X_b = np.c_[np.ones((100, 1)), X]

def gradient_descent(X, y, theta, alpha, num_iters):
    m = len(y)
    cost_history = []
    for i in range(num_iters):
        hypothesis = X.dot(theta)
        loss = hypothesis - y
        gradient = X.T.dot(loss) / m
        theta -= alpha * gradient
        cost = np.sum(loss**2) / (2 * m)
        cost_history.append(cost)
    return theta, cost_history

theta = np.random.randn(2, 1)

alpha = 0.1
num_iters = 1000

theta_final, cost_history = gradient_descent(X_b, y, theta, alpha, num_iters)

print("Parameters theta after gradient descent:")
print(theta_final)

plt.plot(range(num_iters), cost_history)
plt.xlabel('Iterations')
plt.ylabel('Mean Squared Error')
plt.title('Gradient Descent Convergence Process')
plt.grid(True)
plt.show()
```

Figure 1. Code of the Gradient Descent Process for Linear Regression

This code demonstrates the gradient descent process for linear regression. First, random data is created for a linear regression model, where X – independent variable, y – dependent variable with added random noise. The gradient_descent function then performs an iterative process, calculating the gradient of the model error and updating the theta parameters to minimize the Root Mean Square (RMS) error. At the end, the final values of the theta parameters are displayed and a graph is constructed illustrating the process of convergence of gradient descent over the course of iterations (Figure 2).

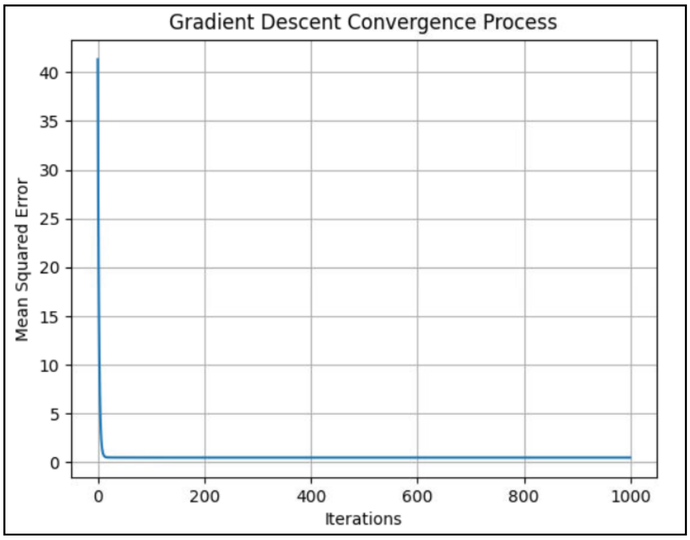

Figure 2. Result of the Gradient Descent Code

Equally important are the methods of stochastic gradient descent. Stochastic gradient descent is a variation of gradient descent where model parameters are updated based on a single random observation (or subset of data), making it more efficient for large datasets. An example of the implementation is shown in Figure 3.

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

X_b = np.c_[np.ones((100, 1)), X]

def stochastic_gradient_descent(X, y, theta, alpha, num_epochs, batch_size):
    m = len(y)
    cost_history = []
    for epoch in range(num_epochs):
        shuffled_indices = np.random.permutation(m)
        X_shuffled = X[shuffled_indices]
        y_shuffled = y[shuffled_indices]
        for i in range(0, m, batch_size):
            X_batch = X_shuffled[i:i+batch_size]
            y_batch = y_shuffled[i:i+batch_size]
            hypothesis = X_batch.dot(theta)
            loss = hypothesis - y_batch
            gradient = X_batch.T.dot(loss) / batch_size
            theta -= alpha * gradient
            cost = np.sum(loss ** 2) / (2 * batch_size)
            cost_history.append(cost)
    return theta, cost_history

theta = np.random.randn(2, 1)

alpha = 0.1
num_epochs = 50
batch_size = 10

theta_final, cost_history = stochastic_gradient_descent(X_b, y, theta, alpha, num_epochs,
batch_size)

print("Parameters theta after stochastic gradient descent:")
print(theta_final)

plt.plot(range(len(cost_history)), cost_history)
plt.xlabel('Iterations')
plt.ylabel('Mean Squared Error')
plt.title('Stochastic Gradient Descent Convergence Process')
plt.grid(True)
plt.show()
```

Figure 3. Code of Stochastic Gradient Descent Method for Linear Regression Problem

The code implements the stochastic gradient descent method for the linear regression problem. First, a random set of regression data is generated. Then an X_b matrix is created, into which a column of units is added to account for the free coefficient. The stochastic_gradient_descent function performs stochastic gradient descent, where data is split into mini-packets of a given size. At each iteration, the gradient is calculated for each mini-package and the theta parameters are updated. The result is the optimal parameters obtained after the end of the descent process, and a list containing the values of the loss function at each iteration. Visualization of the convergence process shows how the value of the root mean square error decreases with increasing number of iterations, which confirms the correctness of the stochastic gradient descent algorithm (Figure 4).

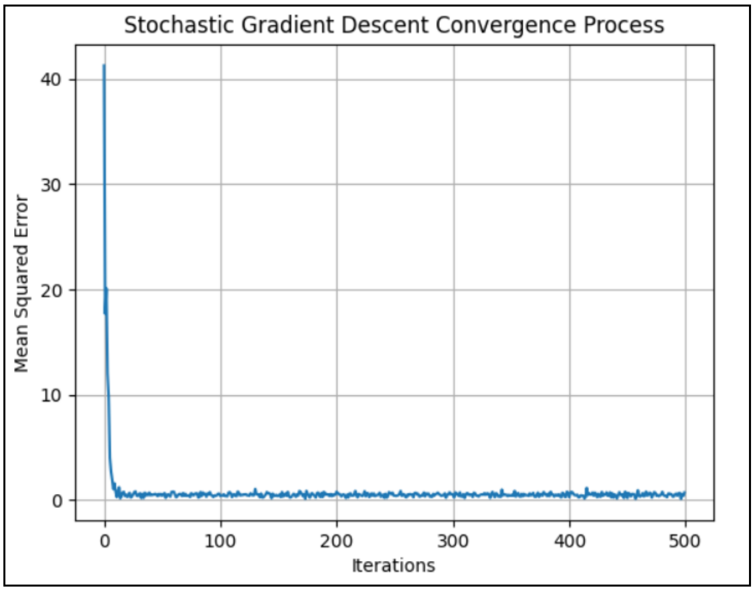

Figure 4. Result of the Stochastic Gradient Descent Code

In addition to classical methods, metaheuristic-based optimization methods such as genetic algorithms, swarm algorithms, and simulated annealing algorithms are widely used in ML. These methods are especially effective for global optimization and work with nonlinear loss functions. In turn, non-zero optimization methods do not use the gradient of the loss function to find optimal parameters. Examples of such methods include grid search, where parameter values are iterated over in a predefined grid, and direct search, using heuristic methods to find the minimum of a function. The choice of a specific optimization algorithm depends on the task, the size of the data, the available computational resources, and the required accuracy of the solution.

Practical Examples of the Application of Existing Algorithms

Machine learning relies heavily on sophisticated optimization algorithms that enhance model performance across diverse domains, enabling more precise and efficient solutions in areas such as computer vision, natural language processing, predictive analytics, and strategic resource allocation. Gradient descent is a basic optimization method widely used in image recognition, natural language processing and time series prediction problems. For example, Facebook’s face recognition system DeepFace achieves high accuracy by using convolutional neural networks trained using gradient descent (Taigman et al., 2014; Fakhroni, & Puotier, 2023; BoangManalu, Iqbal, & Garcia, 2024; Khoviriza et al., 2024). Another example is Google’s FaceNet neural network architecture, which uses gradient descent for training and has achieved unprecedented accuracy in facial recognition tasks. These examples demonstrate the effectiveness of gradient descent in solving complex ML problems. Stochastic gradient descent is especially effective when working with large datasets and is widely used in recommendation systems, anomaly detection and traffic optimization (Capasso et al., 2021; Fitriah, Akorede, & Agyei, 2023; Hardyanti, Lateef, & Abbas, 2023; Almufti et al., 2024).

For example, Google Translate, a popular machine translation system, uses stochastic gradient descent to train its models on huge amounts of text data. Amazon Transcribe, Amazon’s speech-to-text conversion service, uses it to optimize its acoustic models, providing high accuracy and performance of speech-to-text conversion. The practical implementations in systems like Facebook’s DeepFace and Google Translate underscore gradient descent's real-world computational efficiency (Huda, Girei, & Keizi, 2023; Sari, Omeiza, & Mwakifuna, 2023; Abdrakhmanov et al., 2024). These systems process enormous amounts of data, requiring an optimization method that can scale effectively. Stochastic gradient descent, in particular, enables these complex systems to train on massive datasets without becoming computationally intractable. However, gradient descent is not without limitations. Its performance can be sensitive to learning rate selection, and it may converge slowly in certain complex landscapes. Advanced variants like momentum-based methods and adaptive learning rate techniques have been developed to address these challenges, further improving its computational efficiency and convergence properties.

Metaheuristic-based optimization methods, such as genetic algorithms and particle swarm algorithms, are used in tasks requiring global optimization. In problems such as optimizing routes for a travelling salesman, metaheuristics help in finding the best paths with minimum cost. In industry, these methods are used for production planning, optimizing the schedule of production lines to achieve maximum productivity. Metaheuristics are also widely used to adjust the hyperparameters of ML models, which allows automatically finding the optimal parameter values, improving the accuracy and speed of model learning.

Non-zero optimization methods, such as discrete and combinatorial optimization, are used in problems where variables can take only limited values. For example, in logistics and supply chain management, such methods are used to solve scheduling, routing, and packaging problems. In reverse engineering tasks, non-zero optimization methods help to restore the initial parameters of the model from its output data, which is important for the analysis and interpretation of complex systems. For a deeper understanding, it is also important to consider the advantages and disadvantages of these methods (Table 1).

Table 1. Advantages and Disadvantages of Optimization Algorithms

Optimization algorithm	Advantages	Disadvantages
Gradient descent	High precision, fast convergence to a minimum, applicable for continuous functions	Sensitivity to initial conditions, the possibility of getting stuck in local minima, the need to adjust the training parameter
Stochastic gradient descent	Effective for big data, good generalizing ability, reduces the risk of getting stuck in local minima	Noise in gradients can lead to fluctuations, the need to use mini-batches to stabilize
Metaheuristic methods	The ability to find global optima, applicability to complex and nonlinear problems, flexibility and adaptability	High computational costs, possibly long vanishing time, uncertainty in the guarantee of finding the global optimum
Non-zero methods	Efficiency in discrete tasks, ability to solve problems with limited values of variables, good interpretation of results	The complexity of implementation for large problems, the possibility of a combined explosion, the need to know the exact search space

Thus, the examples and characteristics of various optimization algorithms emphasize their importance and the variety of applications in modern ML problems, and understanding the advantages and disadvantages of each approach allows choosing the most appropriate method for a specific task, which significantly increases the efficiency and accuracy of models.

Development of Optimization Methods to Improve the Performance of ML Models

With the growing volume of data and the complexity of the tasks facing ML systems, there is a need for effective and adaptive approaches to optimizing the operation of models. New methods will not only improve the accuracy and speed of the models, but also improve their adaptability to changing conditions and resource requirements. First, adaptive automation of model selection should be highlighted (Figure 5). This method suggests developing an automated ML model selection system based on adaptive learning. The system will analyze the characteristics of the data, including its structure, volume and nature, to select the most appropriate models without prior user identification. Using performance metrics and evaluation criteria such as learning rate, accuracy, and resource requirements, the method will optimize model selection in real time, adapting to changes in data and working conditions. The system operates in real time, which allows it to adapt to changes in data and working conditions.

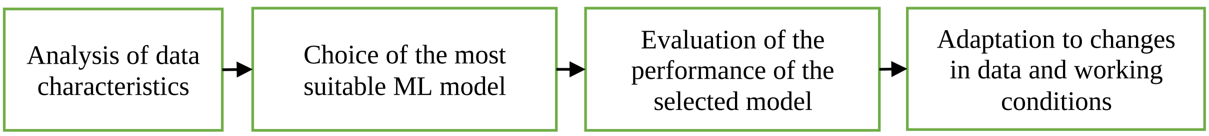

Figure 5. Adaptive automation of model selection

For example, if data characteristics change or new data appears, the system can re-evaluate the selected model and, if necessary, replace the model with a more suitable one. The method of dynamic adaptation of model parameters suggests the development of a system capable of dynamically adapting the parameters of ML models depending on changing environmental conditions or input data (Figure 6). The system will use optimization algorithms that automatically adjust the model parameters based on current needs and conditions, such as the amount of data, its structure and nature. This will allow models to adapt effectively and quickly to new information without the need for complete retraining or manual reconfiguration of parameters.

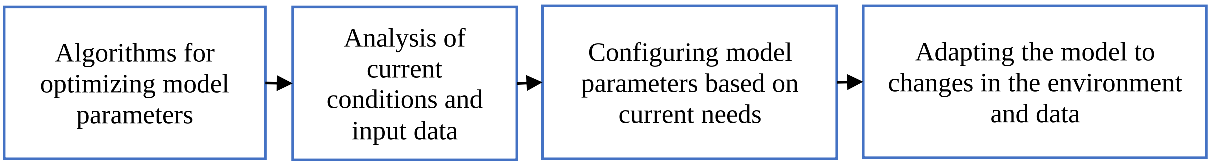

Figure 6. Dynamic adaptation of model parameters

The dynamic parameter adaptation system analyzes the current data, assessing its volume, structure, and nature. Based on this analysis, the system makes decisions about the necessary changes to the model parameters. For example, if the amount of data increases significantly or its structure changes, the system can automatically adjust the model parameters to ensure optimal performance. This is especially important in situations where data is constantly being updated or changed, such as streaming data or real-time sensor data. One of the key advantages of the dynamic parameter adaptation method is its ability to minimize the need for manual adjustment and retraining of models (Ahmad et al., 2024). This significantly reduces the cost of time and resources, increasing the overall efficiency of the ML process. Instead of developing and testing new models from scratch with each data change, the system can dynamically adjust existing models to ensure their optimal performance.

The primary triggers for dynamic parameter adaptation include situations with high data variability, such as streaming data environments, real-time sensor networks, and adaptive recommendation systems. In these contexts, the underlying data distribution can shift rapidly, making static model parameters ineffective. For instance, in financial trading algorithms, market conditions can change dramatically within minutes, requiring immediate model parameter adjustments to maintain predictive accuracy. Another critical domain is natural language processing (NLP), particularly in applications like sentiment analysis or language translation (Negara et al., 2024). A dynamic parameter adaptation system can help models remain responsive to these linguistic shifts by continuously fine-tuning model weights and biases. As cyber threats continuously evolve and new attack patterns emerge, static models quickly become obsolete (Amelin et al., 2021; Vilks et al., 2022; Anggraeni, Rassy, & Sereesuchat, 2023). The ability to dynamically adjust parameters allows the model to maintain high detection rates and minimize false positives (Shults et al., 2020; Helida, Ching, & Oyewo, 2023; Maymunah, Ramorola, & Shobowale, 2023).

It is also worth focusing on the mechanisms of automated resource management (Figure 7). This method is aimed at developing resource management systems for ML models that will dynamically optimize the allocation of computational resources depending on the current requirements of the model. The systems will use optimization algorithms to effectively manage learning and inference processes, minimizing time delays and resource costs. This will allow ML models to achieve high performance under various loads and operating conditions.

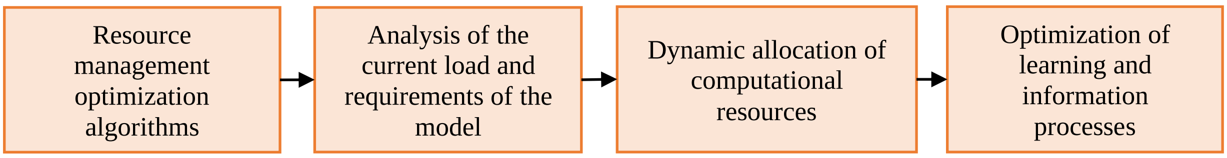

Figure 7. Automated resource management mechanisms

This method is based on optimization algorithms that allow efficiently allocating resources such as memory, processor time, and graphics processing units (GPUs) among various tasks and processes. The system automatically monitors current loads and the status of resources, adapting their distribution in real time. For example, when the amount of data increases or tasks become more complex, the system can allocate additional resources for a specific model, ensuring its smooth operation and high accuracy of predictions. Efficient resource management also includes load balancing between different computing nodes and servers (Aviv et al., 2021). This is achieved by dynamically redistributing tasks and data, which avoids overloads and improves overall system performance. Minimizing time delays is also an important aspect, especially in tasks that require real-time or near-real-time data processing, such as traffic prediction or financial transaction management (Quraishi et al., 2024). These methods represent innovative approaches to optimizing the performance of ML models aimed at automating model selection processes, dynamic parameter adaptation, and efficient resource management. They can significantly improve the efficiency of models in the face of rapidly changing data and requirements.

The trade-offs between computational efficiency and model accuracy are fundamental challenges in modern machine learning optimization. The proposed methods reveal a sophisticated approach to navigating these complex trade-offs through adaptive and dynamic mechanisms. Adaptive automation of model selection represents a critical strategy for addressing the computational efficiency-accuracy balance (Aviv et al., 2023). By developing an automated system that can analyze data characteristics in real time, the approach allows for dynamic model selection that optimizes both computational resources and predictive performance. The dynamic adaptation of model parameters further elaborates on this trade-off. Traditional machine-learning approaches often require manual reconfiguration or complete retraining when data characteristics change, which is computationally expensive and time-consuming (Falko et al., 2024). The dynamic parameter adaptation method provides a more efficient alternative, allowing models to fine-tune their internal representations with minimal computational investment. This approach represents a sophisticated compromize between maintaining model accuracy and managing computational resources. The automated resource management mechanisms provide another critical perspective on the efficiency-accuracy trade-off (Yermolenko et al., 2024; Hani et al., 2024). By dynamically allocating computational resources such as memory, processor time, and GPUs, the system can optimize performance across varying computational loads.

These methods collectively demonstrate a sophisticated understanding of the computational efficiency-accuracy trade-off. Rather than viewing these as opposing forces, the proposed approaches treat them as interconnected challenges that can be addressed through intelligent, adaptive mechanisms. The key innovation lies in creating systems that can make real-time decisions about model selection, parameter adjustment, and resource allocation.

Integration of the Developed Methods into Existing Platforms

The integration of these optimization methods into existing ML platforms requires a careful approach and consistent implementation to ensure maximum efficiency and minimal disruptions in current workflows. The first step is to evaluate the compatibility of the developed methods with existing platforms and infrastructure. This includes an analysis of the architecture of current systems, libraries and frameworks used, and hardware and software resource requirements. It is important to ensure that new methods can be integrated without significant changes to the existing infrastructure. The modular approach to integration allows the gradual introduction of new methods, minimizing risks, and simplifying the adaptation process. Each method, namely adaptive automation of model selection, dynamic parameter adaptation, and automated resource management, should be implemented as a separate module with well-defined interfaces and functionality. This will provide flexibility and the possibility of gradual integration and testing.

Before full implementation, each module must be thoroughly tested and validated. This includes checking the correctness of the algorithms, evaluating the performance and stability of the system under various loads. Testing should be carried out both in isolated conditions and in real-world use cases to identify potential problems and verify the effectiveness of new methods. After successful testing, the modules can be integrated with existing workflows, which includes setting up automated monitoring and management systems, adapting user interfaces, and training staff. It is important to ensure a smooth transition and minimize the impact on ongoing processes to avoid delays and disruptions. After integration, it is necessary to continue monitoring and evaluating the effectiveness of new methods. The system should be configured to collect and analyze performance metrics, which will allow timely identification and elimination of bottlenecks and optimize performance. Continuous improvement and adaptation are key aspects to maintain high efficiency and meet changing requirements.

An example of method integration based on the popular ML TensorFlow platform can be considered. Adaptive automation of model selection can be implemented using the TensorFlow Model Optimization Toolkit. Scripts are created to automatically select and configure models based on data analysis and performance metrics. Dynamic parameter adaptation can be integrated using TensorFlow's tf.train API, which allows flexibly managing model training and dynamically adapting hyperparameters. Automated resource management can be implemented using TensorFlow Serving and Kubernetes to automatically scale and manage the allocation of computational resources depending on current loads.

During the integration of optimization methods into existing machine learning platforms, several specific challenges may arise. Compatibility issues can emerge when attempting to merge new methodologies with legacy systems, particularly when dealing with different programming languages, outdated libraries, or conflicting architectural designs. Technical complexity increases when trying to ensure seamless communication between new modules and existing infrastructure, requiring sophisticated interface design and potential middleware development. Resource allocation presents another significant challenge, as the new optimization methods may have different computational requirements compared to existing workflows. Additionally, the team may encounter resistance from existing system administrators or developers who are comfortable with current processes and may be skeptical of introducing new optimization techniques. Performance validation becomes critical, as the integration must not only maintain but ideally improve the existing system's efficiency. This requires extensive testing under various scenarios, including stress tests and real-world use cases, to ensure that the new methods genuinely provide the promised benefits without introducing unexpected complications or performance degradations.

The modular integration strategies for ML platforms address compatibility and real-time adjustments through a comprehensive, systematic approach. By carefully evaluating existing infrastructure and developing well-defined, independent modules, organizations can seamlessly introduce advanced optimization methods with minimal disruption. The process begins with a thorough compatibility assessment, analysing current system architectures, libraries, frameworks, and resource requirements to ensure new methods can be integrated without significant structural changes. The modular approach enables gradual implementation, allowing each optimization method, such as adaptive model selection, dynamic parameter adaptation, and automated resource management, to be introduced and tested incrementally. Rigorous testing in both isolated and real-world environments ensures algorithmic correctness and system stability before full deployment. For platforms like TensorFlow, integration involves utilizing specialized toolkits and APIs that facilitate automated model selection, dynamic hyperparameter management, and intelligent resource allocation. Continuous monitoring and performance analysis are crucial, enabling ongoing refinement and adaptation to changing computational demands. Thus, the integration of the developed optimization methods into existing ML platforms is a complex but necessary process to ensure the high performance and adaptability of models. A systematic and step-by-step approach to integration, including compatibility assessment, modular implementation, testing and continuous improvement, will maximize the use of new methods and improve the performance of existing ML systems.

The results of this study have shown significant progress in the development and application of optimization algorithms to improve the performance of ML models. The importance of comparative analysis with other studies is to accurately identify the advantages and unique characteristics of this approach and to identify opportunities for further improvement of optimization methods in the context of modern AI requirements. The presented study looked at a number of optimization techniques to increase the effectiveness of machine learning models, such as gradient and stochastic descent, metaheuristic, and non-zero methods. In comparison, the study by Boulesnane (2024) emphasized the importance of evolutionary methods, which are particularly effective in solving complex problems such as feature selection and model optimization. While Boulesnane's (2024) study covers a wide range of optimization approaches, including both classical and new techniques, another study focuses on evolutionary methods. The current study also suggested novel techniques that can greatly enhance the performance of machine systems, including adaptive automation of model selection, dynamic parameter adaption, and automated resource management. In contrast, the study by Oh et al. (2024) focuses on optimizing hyperparameters in chemical processes using recurrent neural networks, which demonstrates successful application in specialized ML systems. That is, this study presented an expansion of the range of optimization methods proposed in these papers and showed how new approaches can complement existing solutions.

Comparative to this study, which delved into a variety of optimization methods for ML models, emphasizing their role in improving model performance, the paper by Dey and Chatterjee (2023) demonstrated the interaction between learning algorithms and optimization shells, with an emphasis on increasing scalability and efficiency in customer support analytics. In this sense, both studies have demonstrated the value of combining numerous techniques to improve outcomes in the context of machine learning; however, the current study has broadened the scope of optimization techniques, encompassing a range of approaches and their use in diverse tasks. In addition, Pravin et al. (2023) highlighted the importance of automation and optimization of architectural solutions to achieve high model accuracy, which confirms the general importance of optimization in the field of ML. Thus, the mentioned studies emphasize the importance of integrating different methods to achieve the best results in the context of optimizing ML models.

Karthick (2024) presented an overview of various optimization methods, including gradient descent options. However, the researcher also covered adaptive learning rate methods, second derivative and regularization methods, constraint-based methods and Bayesian optimization, while this study additionally considered metaheuristic and non-zero methods. Felić, Marzouk and Tschuchnigg (2024) focused on calibrating model parameters using ML algorithms based on laboratory data, which simplifies the calibration process and improves model analysis based on experimental data, the current study focused on a more flexible approach to real-time parameter adaptation. Despite being more static, the calibration techniques discussed in this research could be helpful for improving the dynamic techniques taken into consideration in this investigation.

The study by Afolabi and Akinola (2024) showed the use of specialized feature selection methods, such as backpacking and mutual information gain algorithms, which significantly improves the accuracy of intrusion detection on specific data sets, the results of this study focused on a wider range of optimization methods. The results of the mentioned study can help in integrating feature selection methods into the developed optimization strategies. In turn, Mishra, Tripathi and Kumar (2024) analyzed the future development of optimization in ML and AI in the context of business intelligence, which complements the current study in terms of understanding long-term trends and optimization opportunities. Although optimization approaches vary, the prospects of the latest study may focus on the future line of the research and improvement of methods in the context of business intelligence.

Both this study and the paper by Vasuki, Amalrajvictoire and Nivetha (2024) demonstrated the effectiveness of using ML to solve specialized problems, emphasizing the importance of optimization algorithms in various fields. Although the current study focused on applying optimization techniques to improve the performance of ML models, other work applied similar techniques and expertize to create classifiers evaluating web page optimization. The parameters of ML models are given particular consideration in this study in order to respond quickly to changing input data conditions. Gou et al. (2024) has achieved significant success in multi-criteria optimization by integrating the Light Gradient Boosting Machine algorithm, the results have shown the importance of dynamic parameter adaptation to improve the overall efficiency of ML models. An important aspect is that multi-criteria optimization methods can be useful to improve the adaptive approaches presented in the current study.

Moreover, this study focused on optimization methods in the context of resource management, while the study by Recalde et al. (2024) demonstrated the application of ML and optimization methods for energy management systems, emphasizing the importance of real-time algorithms and hybrid approaches to improve the performance of these systems. The application of optimization techniques unites the studies, and the findings of other research can be helpful in creating hybrid approaches to resource management that will enhance and broaden the scope of the techniques suggested in this study. The study by Zhu et al. (2024) showed the use of ML to predict the properties of materials and the application of optimization approaches in the design of metamaterials. Although the primary focus of the study is different, the findings of the aforementioned study may be applicable to the current investigation since the techniques for forecasting material attributes and design can be modified to enhance resource management and adaptability in machine learning models.

Special attention was paid to the adaptation of the parameters of ML models to quickly respond to changing input data conditions. While the study by Cai, Li and Shen (2024) proposed an approach to optimization problems through minimization of convex functionals and relaxation in the space of distributions, while the results of the presented research showed the importance of adaptive optimization methods to increase the overall efficiency of ML models. Thus, the methods presented in the other study can be used to further customize the adaptive methods developed in the current work, which will help improve their application in real-world scenarios with changing conditions. In addition, the results of this study include the development of optimization methods for resource management and adaptation of models to changing data, and the study by Dwivedi and Srivastava (2023) showed modern optimization methods, including gradient methods of the first and high order, and optimization algorithms without derivatives. That is, the methods presented in this study can be supplemented with approaches from other paper for more comprehensive management of ML models.

Mahala et al. (2024) focused on the application of ML methods to analyze and improve marketing strategies in the supermarket industry using decision trees, the support vector machine, random forests, and XGBoost. In contrast, the findings of the current research highlighted the importance of adaptive optimization methods to improve the overall effectiveness of ML models. The optimization techniques taken into consideration in both studies can be advantageous to both parties, even though their application areas differ. For example, the methods used to optimize marketing strategies can be adapted to improve adaptive methods in other areas of ML, which highlights the potential of integrating various optimization techniques to achieve better results. Unlike the study by Elvin and Wibowo (2024), which focused on the application of ML to predict water quality and optimize hyperparameters, which demonstrates the importance of ML in environmental and industrial tasks, the results of this study focus on optimization methods to improve the overall accuracy and efficiency of ML models in a broader context. And in comparison, with the study by Chen and Marrero (2024), which focused on identifying and mitigating bias in medical decision-making using optimization and ML techniques, the results of this study highlighted the importance of choosing appropriate metrics and methods to improve fairness and accuracy in various ML tasks, not just in medicine.

The study's exploration of automated model selection methods raises significant ethical considerations that warrant deeper examination. While the research focuses on technical improvements in machine learning optimization, it overlooks potential risks such as algorithmic bias, transparency challenges, and the potential for unintended discriminatory outcomes. Automated systems might perpetuate existing biases present in training data or create opaque decision-making processes that are difficult to audit and understand. Furthermore, the increasing automation of model selection could potentially reduce human oversight and critical evaluation, leading to a cascade of unexamined assumptions and potentially harmful optimization strategies that prioritize efficiency over fairness and accountability.

Thus, the study covered a variety of methods for optimizing and adapting the parameters of ML models, emphasizing their importance for improving efficiency and productivity in various fields of application. At its core, the study presents a nuanced understanding of how diverse optimization techniques can be integrated to enhance machine learning model performance across various domains. The novelty of the research is in examining a broad spectrum of optimization approaches, ranging from classical methods like gradient and stochastic descent to more innovative techniques such as adaptive parameter adaptation and dynamic resource management.

CONCLUSION

This study was aimed at a comprehensive analysis of existing optimization algorithms for ML models and the development of new approaches to improve their effectiveness. It was possible to analyze such optimization algorithms as gradient descent, stochastic gradient descent, metaheuristic methods, and non-zero methods. This analysis revealed the strengths and weaknesses of each approach, which helped to understand their applicability in various ML contexts. The study showed that the use of a modular approach, testing, validation, and continuous improvement of systems will successfully introduce new methods and improve current ML processes. A comparative analysis of various optimization methods based on key performance metrics has shown that each of the considered approaches has its advantages and disadvantages.

In addition, new optimization approaches were developed during the research, including methods of adaptive automation of model selection, dynamic adaptation of model parameters, and automated resource management. The study's novel approaches introduce new ways to improve model flexibility, particularly in real-time data environments, thus offering significant contributions to the field of AI and ML. These methods have shown their effectiveness in improving the performance of ML models, and their ability to adapt to changing conditions, which makes them extremely useful in modern dynamic data environments.

ACKNOWLEDGMENTS

None.

AUTHOR CONTRIBUTIONS

Conceptualization, N.A. and E.R.; Methodology, N.A. and H.P.; Software, A.A.; Validation, A.A. and V.S.; Formal Analysis, E.R. and H.P.; Investigation, N.A. and E.R.; Data Curation, V.S.; Writing – Original Draft Preparation, N.A. and H.P.; Writing – Review & Editing, E.R. and V.S.; Visualization, A.A.; Supervision, V.S.

CONFLICTS OF INTEREST

The author(s) declare no conflict of interest.

REFERENCES

Abdrakhmanov, R., Kenesbayev, S.M., Berkimbayev, K., Toikenov, G., Abdrashova, E., Alchinbayeva, O., & Ydyrys, A. (2024). Offensive language detection on social media using machine learning. *International Journal of Advanced Computer Science and Applications*, *15*(5), 575-582. https://doi.org/10.14569/IJACSA.2024.0150557.

Afolabi, A.S., & Akinola, O.A. (2024). Network intrusion detection using knapsack optimization, mutual information gain, and machine learning. *Journal of Electrical and Computer Engineering*, *2024*(1), 7302909. https://doi.org/10.1155/2024/7302909.

Ahmad, H.B., Asaad, R.R., Almufti, S.M., Hani, A.A., Sallow, A.B., & Zeebaree, S.R.M. (2024). Smart home energy saving with big data and machine learning. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(1), 11-20. https://doi.org/10.22437/jiituj.v8i1.32598.

Almonia, A. L. (2024). Effectiveness of collaborative and individualized learning on the learners' achievement in science among pupils. *Integrated Science Education Journal*, *5*(2), 115-124. https://doi.org/10.37251/isej.v5i2.482.

Almufti, S.M., Hani, A.A., Zeebaree, S.R.M., Asaad, R.R., Majeed, D.A., Sallow, A.B., & Ahmad, H.B. (2024). Intelligent home IoT devices: An exploration of machine learning-based networked traffic investigation. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(1), 1-10. https://doi.org/10.22437/jiituj.v8i1.32767.

Amelin, O.Yu., Kyrychenko, T.M., Leonov, B.D., Shablystyi, V.V., & Chenshova, N.V. (2021). Cyberbullying as a way of causing suicide in the digital age. *Journal of the National Academy of Legal Sciences of Ukraine*, *28*(3), 277-289. https://visnyk.kh.ua/en/journals/visnik-naprnu-3-2021-r/kiberbuling-yak-odin-iz-sposobiv-dovedennya-do-samogubstva-v-epokhu-tsifrovizatsiyi

Anggraeni, F. D. R., Rassy, R. P., & Sereesuchat, S. (2023). Development of a textured picture book equipped with crosswords as a media for learning biology sub-material epithelial tissue. *Journal of Educational Technology and Learning Creativity*, *1*(2), 68-77. https://doi.org/10.37251/jetlc.v1i2.793.

Aviv, I., Barger, A., & Pyatigorsky, S. (2021). Novel machine learning approach for automatic employees' soft skills assessment: Group collaboration analysis case study. *5th International Conference on Intelligent Computing in Data Sciences, ICDS 2021*. Virtual, Online: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICDS53782.2021.9626760.

Aviv, I., Gafni, R., Sherman, S., Aviv, B., Sterkin, A., & Bega, E. (2023). Infrastructure from code: The next generation of cloud lifecycle automation. *IEEE Software*, *40*(1), 42-49. https://doi.org/10.1109/MS.2022.3209958.

BoangManalu, E. N., Iqbal, M., & Garcia, C. (2024). Analysis of the relationship between interest and learning outcomes of physics in senior high school. *EduFisika: Jurnal Pendidikan Fisika*, *9*(1), 46-53. https://doi.org/10.59052/edufisika.v9i1.29641.

Boulesnane, A. (2024). Evolutionary dynamic optimization and machine learning. *Advanced Machine Learning with Evolutionary and Metaheuristic Techniques*. Singapore: Springer. https://doi.org/10.1007/978-981-99-9718-3_3.

Cai, Y., Li, Q., & Shen, Z. (2024). Optimization in machine learning: A distribution-space approach. *Communications on Applied Mathematics and Computation*, *6*, 1217-1240. https://doi.org/10.1007/s42967-023-00322-5.

Capasso, C., Rubino, L., Rubino, G., & Veneri, O. (2021). Data analytics for performance modelling of photovoltaic systems in the internet of energy scenario. *2021 IEEE 15th International Conference on Compatibility, Power Electronics and Power Engineering, CPE-POWERENG 2021*. Florence: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/CPE-POWERENG50821.2021.9501202.

Chen, Z., & Marrero, W.J. (2024). *A survey on optimization and machine-learning-based fair decision making in healthcare*. https://doi.org/10.1101/2024.03.16.24304403.

Dey, P., & Chatterjee, K. (2023). Application of machine learning for optimization. *Handbook of Research on AI and Machine Learning Applications in Customer Support and Analytics*. Hershey, PA: IGI Global Scientific Publishing. https://doi.org/10.4018/978-1-6684-7105-0.ch007.

Dwivedi, R., & Srivastava, V.K. (2023). Fundamental optimization methods for machine learning. *Statistical Modeling in Machine Learning*. London: Academic Press. https://doi.org/10.1016/B978-0-323-91776-6.00005-1.

Elvin, & Wibowo, A. (2024). Forecasting water quality through machine learning and hyperparameter optimization. *Indonesian Journal of Electrical Engineering and Computer Science*, *33*(1), 496-506. http://doi.org/10.11591/ijeecs.v33.i1.pp496-506.

Essenzi, D. S. (2024). Analysis of community health center performance on tuberculosis control programs. *Journal of Health Innovation and Environmental Education*, *1*(1), 1-6. https://doi.org/10.37251/jhiee.v1i1.1039.

Fakhroni, A. A., & Puotier, Z. (2023). Efforts to improve mathematics learning outcomes using napier bone teaching aids for elementary school students. *Interval: Indonesian Journal of Mathematical Education*, *1*(2), 36-46. https://doi.org/10.37251/ijome.v1i2.779.

Falko, A., Gogota, O., Yermolenko, R., & Kadenko, I. (2024). Analysis of LArTPC data using machine learning methods. *Journal of Physical Studies*, *28*(1), 1802. https://doi.org/10.30970/jps.28.1802.

Felic, H., Marzouk, I., & Tschuchnigg, F. (2024). Parameter optimization for constitutive soil models by means of supervised machine learning. *28th European Young Geotechnical Engineers Conference*. https://graz.elsevierpure.com/en/activities/parameter-optimization-for-constitutive-soil-models-by-means-of-s

Fitriah, F., Akorede, A., & Agyei, E. (2023). Improving mathematics learning outcomes through the consideration model for class vii students. *Interval: Indonesian Journal of Mathematical Education*, *1*(2), 47-55. https://doi.org/10.37251/ijome.v1i2.771.

Gong, Y., Huang, J., Liu, B., Xu, J., Wu, B., & Zhang, Y. (2024). Dynamic resource allocation for virtual machine migration optimization using machine learning. *Applied and Computational Engineering*, *57*, 1-8. https://doi.org/10.54254/2755-2721/57/20241348.

Gou, W., Shi, Z. Z., Zhu, Y., Gu, X. F., Dai, F. Z., Gao, X. Y., & Wang, L. (2024). Multi-objective optimization of three mechanical properties of Mg alloys through machine learning. *Material Genome Engineerig Advance*, *e54*. https://doi.org/10.1002/mgea.54.

Hani, A.A., Sallow, A.B., Ahmad, H.B., Abdulrahman, S.M., Asaad, R.R., Zeebaree, S.R.M., & Majeed, D.A. (2024). Comparative analysis of state-of-the-art classifiers for Parkinson's disease diagnosis. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(2), 409-423. https://doi.org/10.22437/jiituj.v8i2.32771.

Hardyanti, V. S., Lateef, H., & Abbas, S. A. (2023). The influence of teacher teaching creativity on student learning outcomes in mathematics subjects. *Interval: Indonesian Journal of Mathematical Education*, *1*(2), 56-60. https://doi.org/10.37251/ijome.v1i2.778.

Hasan, R. (2024). AI and machine learning for optimal crop yield optimization in the USA. *Journal of Computer Science and Technology Studies*, *6*(2), 46-61. https://doi.org/10.32996/jcsts.2024.6.2.6%20.

Hasanov, A., & Mammadov, Y. (2023). New approaches in forecasting industrial production. *Proceedings of Azerbaijan High Technical Educational Institutions*, *35*(12), 258-264. https://doi.org/10.36962/pahtei35122023-258.

Helida, Y., Ching, C. P., & Oyewo, A. (2023). Development of a simple stirling engine demonstration tool on the subject of thermodynamics. *Journal of Educational Technology and Learning Creativity*, *1*(2), 59-67. https://doi.org/10.37251/jetlc.v1i2.790.

Huda, I., Girei, M. M., & Keizi, F. (2023). Development of a practical tool for linear momentum collisions using a microcontroller. *Journal of Educational Technology and Learning Creativity*, *1*(2), 42-49. https://doi.org/10.37251/jetlc.v1i2.788.

Iqbal, S., & Sheikh, K. (2024). Machine learning based compiler optimization technique. *Sukkur IBA Journal of Emerging Technologies*, *7*(1), 37-47. https://www.researchgate.net/publication/382123249.

Karthick, K. (2024). Comprehensive overview of optimization techniques in machine learning training. *Control Systems and Optimization Letters*, *2*(1), 23-27. https://ejournal.csol.or.id/index.php/csol/article/view/69

Khoviriza, Y., Azzahra, M. Z., Galadima, U., & Salsabila, W. S. (2024). Revealing the impact: Meta analysis of problem based learning models on improving communication skills in science learning. *EduFisika: Jurnal Pendidikan Fisika*, *9*(1), 38-45. https://doi.org/10.59052/edufisika.v9i1.32650.

Komiya, N. (2024). Discovering differences in consciousness of facial features among japanese university students in the year of admission according to COVID-19. *Integrated Science Education Journal*, *5*(1), 12-18. https://doi.org/10.37251/isej.v5i1.849.

Li, Q., Yu, S., Échevin, D., & Fan, M. (2022). Is poverty predictable with machine learning? A study of DHS data from Kyrgyzstan. *Socio-Economic Planning Sciences*, *81*, 101195. https://doi.org/10.1016/j.seps.2021.101195.

Mahala, V.R., Garg, N., Saxena, D., & Kumar, R. (2024). Unveiling marketing potential: Harnessing advanced analytics and machine learning for gold membership strategy optimization in a superstore. *SN Computer Science*, *5*, 374. https://doi.org/10.1007/s42979-024-02700-z.

Majeed, D.A., Ahmad, H.B., Hani, A.A., Zeebaree, S.R.M., Abdulrahman, S.M., Asaad, R.R., & Sallow, A.B. (2024). Data analysis and machine learning applications in environmental management. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(2), 398-408. https://doi.org/10.22437/jiituj.v8i2.32769.

Mamanazarova, G. (2024). Methodological futures: Linking the cognitive stylistics. *Actual Problems of Humanities and Social Sciences*, *4*(4), 198-203. https://doi.org/10.47390/SPR1342V4SI4Y2024N33.

Maurya, P.K., Sah, P., Chauhan, N., & Howard, E. (2024). Smart circuit design machine learning-driven optimization for enhanced performance in electronics and computer engineering. *Journal of Propulsion Technology*, *45*(2), 2794-2805. https://doi.org/10.52783/tjjpt.v45.i02.6339.

Maymunah, A., Ramorola, M., & Shobowale, I. O. (2023). Development of an inquiry-based science module on plant parts and their functions in elementary schools. *Journal of Educational Technology and Learning Creativity*, *1*(2), 50-58. https://doi.org/10.37251/jetlc.v1i2.789.

Mishra, R., Tripathi, P., & Kumar, N. (2024). Future directions in the application of machine learning and intelligent optimization in business analytics. *Intelligent Optimization Techniques for Business Analytics*. Hershey, PA: IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-1598-9.ch003.

Mzili, T., & Arya, A.K. (2024). *Innovations in optimization and machine learning*. IGI Global. https://www.igi-global.com/book/innovations-optimization-machine-learning/339561

Negara, E.S., Syaputra, R., Erlansyah, D., Andryani, R., Saksono, P.H., Aditya, F., & Agam, P.M. (2024). Sentiment analysis of public opinion on presidential advisory appointments using Naive Bayes classification. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, *8*(2), 452-466. https://doi.org/10.22437/jiituj.v8i2.35254.

Oh, K.C., Kwon, H., Park, S.Y., Kim, S.J., Kim, J., & Kim, D. (2024). Hyperparameter optimization of the machine learning model for distillation processes. *International Journal of Intelligent Systems*, *2024*(1), 5564380. https://doi.org/10.1155/2024/5564380.

Pimple, J. (2024). Scientific integration of operations research and machine learning for data centre optimization. *Communications on Applied Nonlinear Analysis*, *31*(4). https://doi.org/10.52783/cana.v31.948.

Pravin, P., Prabhakaran, A., Sharma, S., Divya, P., Muniyandy, E., & Rastogi, R. (2023). Automated machine learning and neural architecture optimization. *The Scientific Temper*, *14*(4), 1345-1351. https://doi.org/10.58414/SCIENTIFICTEMPER.2023.14.4.42.

Quraishi, A., Rusho, M.A., Prasad, A., Keshta, I., Rivera, R., & Bhatt, M.W. (2024). Employing deep neural networks for real-time anomaly detection and mitigation in IoT-based smart grid cybersecurity systems. *3rd IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE 2024*. Hybrid, Ballari: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICDCECE60827.2024.10548160.

Rahmi, Y., Adawiyah, N., & Dilaro, N. N. (2024). Case study: Language politeness in preschool children at ar-rahman school. *Journal of Language, Literature, and Educational Research*, *1*(1), 7-11. https://doi.org/10.37251/jolle.v1i1.999.

Recalde, A., Cajo, R., Velasquez, W., & Alvarez-Alvarado, M.S. (2024). Machine learning and optimization in energy management systems for plug-in hybrid electric vehicles: A comprehensive review. *Energies*, *17*(13), 3059. https://doi.org/10.3390/en17133059.

Romadhonsyah, A. (2024). Contribution of arm muscle power and body flexibility regarding volleyball services for athletes. *Multidisciplinary Journal of Tourism, Hospitality, Sport and Physical Education*, *1*(1), 1-5. https://doi.org/10.37251/jthpe.v1i1.1032.

Sari, R., Omeiza, I. I., & Mwakifuna, M. A. (2023). The influence of number dice games in improving early childhood mathematical logic in early childhood education. *Interval: Indonesian Journal of Mathematical Education*, *1*(2), 61-66. https://doi.org/10.37251/ijome.v1i2.776.

Shults, R., Urazaliev, A., Annenkov, A., Nesterenko, O., Kucherenko, O., & Kim, K. (2020). Different approaches to coordinate transformation parameters determination of nonhomogeneous coordinate systems. *Environmental Engineerizng (Lithuania)*. Vilnius: VGTU. https://doi.org/10.3846/enviro.2020.687.

Sriyono, S. (2024). Improving learning results in hydrocarbon chemistry with mind mapping and classical music accompaniment. *Journal of Chemical Learning Innovation*, *1*(1), 1-6. https://doi.org/10.37251/jocli.v1i1.1016.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *Conference on Computer Vision and Pattern Recognition*. Columbus: IEEE. https://doi.org/10.1109/CVPR.2014.220.

Tsai, H.R., & Chen, T. (2014). Enhancing the sustainability of a location-aware service through optimization. *Sustainability*, *6*(12), 9441-9455. https://doi.org/10.3390/su6129441.

Vasuki, M., Amalrajvictoire, T., & Nivetha, M. (2024). Using machine learning in web page categorization for search engine optimization. *International Journal of Scientific Research in Engineering and Management*, *8*(5). https://doi.org/10.55041/IJSREM34167.

Vilks, A., Kipane, A., Kudeikina, I., Palkova, K., & Grasis, J. (2022). Criminological aspects of current cyber security. *Revista de Direito, Estado e Telecomunicacoes*, *14*(2), 94-108. https://doi.org/10.26512/lstr.v14i2.41411.

Yermolenko, R., Klekots, D., & Gogota, O. (2024). Development of an algorithm for detecting commercial unmanned aerial vehicles using machine learning methods. *Machinery and Energetics*, *15*(2), 33-45. https://doi.org/10.31548/machinery/2.2024.33.

Yusipa, Y. (2024). Comparative analysis of students' biology learning outcomes: Memory and understanding aspects. *Journal of Academic Biology and Biology Education*, *1*(1), 1-9. https://doi.org/10.37251/jouabe.v1i1.1012.

Zhu, C., Bamidele, E.A., Shen, X., Zhu, G., & Li, B. (2024). Machine learning aided design and optimization of thermal metamaterials. *Chemical Reviews*, *124*, 4258-4331. https://doi.org/10.1021/acs.chemrev.3c00708.