Optimizing Long-Short Term Memory (LSTM) Algorithm for Enhanced Energy Efficiency and Green Computing in Cloud Environments

MSc Research Project
MSc in Cloud Computing

Anmol Singla
Student ID: X20259891

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anmol Singla
Student ID:	X20259891
Programme:	MSc in Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	14/12/2023
Project Title:	Optimizing Long-Short Term Memory (LSTM) Algorithm for Enhanced Energy Efficiency and Green Computing in Cloud Environments
Word Count:	6688
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anmol Singla
Date:	30th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Long-Short Term Memory (LSTM) Algorithm for Enhanced Energy Efficiency and Green Computing in Cloud Environments

Anmol Singla
X20259891

Abstract

This research focuses on the optimization of Long Short-Term Memory (LSTM) networks in the cloud computing environment. The custom LSTM model of this study reliably predicts the different cloud computing indicators including the utilization of memory, CPU, network traffic and consumption of power. Critical methods include cleaning and pre-processing of data, advanced parameters of LSTM model and Grid search for the optimization of hyperparameters. The ability of the model to handle the complex and multi-dimensional data in cloud systems is assessed by using Mean Squared Error (MSE) as the primary criteria for performance evaluation. The study highlights the significance of LSTM in cloud computing and its practical implications by comparing the results with those of prior studies.

Despite the impressive prediction accuracy, the research finds a trade-off between the complexity of the model and computing efficiency. With a focus on striking a balance between performance and efficiency, future research should examine hybrid models and real-world applications. This study enhances the comprehension of LSTM implementations in cloud computing by providing valuable insights into approaches for optimising models and possible avenues for future research.

LSTM and the custom LSTM models have been created in this work. The optimization of these two models is also performed. For the evaluation of the models, MSE value has been used. With the basic LSTM model, the MSE score is 0.0755 while the custom LSTM model with novel optimization technique is demonstrating a slight lower MSE value of 0.0730.

Keywords: Long Short-Term Memory (LSTM), Cloud Computing, Hyperparameter Tuning, Genetic Algorithms, Grid Search, Time-Series Prediction, Model Optimization, Computational Efficiency.

1 Introduction

This study aims on optimizing the Long Short-Term Memory (LSTM) algorithm which is a crucial step toward making cloud computing more energy-efficient, aligning with the principles of green computing. This optimization not only boosts the efficiency of data centers but also helps make them more sustainable. Study proposes various smart strategies to plan and implement these algorithms, making them effective and profitable while reducing complexities in transactions and energy sharing.

Motivated by different factors, this study delves into the effectiveness of optimizing long short-term memory algorithms. With an increasing number of data centers used by various organizations, environmental concerns are on the rise Koronen et al. (2020). This has led to a global demand for optimization concepts, as companies seek sustainable solutions for their operations.

This study involves thorough research, exploring different aspects of LSTM and its concepts. Understanding the LSTM framework and its broader traits is essential for applying it accurately. This study is relevant, offering valuable assistance to organizational leaders in improving efficiency and ensuring precision in managerial tasks and services. Additionally, acknowledging the rising operational costs in global companies due to energy transmission, optimizing long short-term memory algorithms within organizations becomes a smart way to reduce these costs and promote sustainability, as suggested by Yang et al. (2020).

Aim and Objectives of the study

The aim of the study is to identify how the parameters of LSTM can be identified and evaluated by use of mathematics related algorithmic techniques such as Genetic Algorithm or Grid Search for enhancement of efficiency in Cloud aligned computing environments.

For greater energy efficiency in cloud environments, LSTM hyperparameters are to be carefully adjusted using mathematical optimization techniques like Genetic Algorithms or Grid Search.

Research Question The following research question is inspired by the aforementioned research problem: How can the hyper-parameters of Long-Short Term Memory (LSTM) machine learning algorithm be systematically tuned using mathematical optimization techniques, such as Genetic Algorithms and Grid Search, to improve energy efficiency and support green computing in cloud environments?

The sequential structure is followed in the study and the initial introduction is followed by the literature review and proposed methodology that has been found to be effective for ensuring the smooth progress of the research.

Structure of the Report

1. **Introduction**: Introduce the topic, state objectives, and outline contributions.
2. **Related Work**: Critically review relevant papers, justify research question.
3. **Methodology**: Discuss research methods, equipment, and statistical techniques.
4. **Design Specification**: Identify techniques/framework for implementation.
5. **Implementation**: Discuss final stage of proposed solution implementation.
6. **Evaluation**: Analyze results, present implications using statistical tools and visual aids.

2 Related Work

2.1 LSTM for Energy Efficiency

Breuel's publication, "Benchmarking of LSTM Networks," and Fitzsimons' doctorate dissertation, "Kernel Methods: Generalisations, Scalability, and Towards the Future of Machine Learning," each make unique contributions to the area of machine learning Fitzsimons (2019). While both journals have their advantages, there are also disparities in the scope of the study, the calibre of the papers, and other factors. Fitzsimons' dissertation possesses academic rigour and significance as a PhD study from the University of Oxford. Its comprehensiveness, however, can be a drawback because it might make it less approachable to a wider audience. The energy efficiency consolidation has been incorporated here for determining the overall assessment in considering the importance of practices and experiences of LSTM Beloglazov and Buyya (2015). Due to its academic standing and careful technique, the research is of a high calibre even if it may be regarded as having a narrow emphasis.

However, Breuel's researchBreuel (2015), which is an arXiv preprint, might not have undergone the thorough peer assessment that a dissertation from a recognized college would normally go through. However, these style guarantees that findings be distributed more quickly Gomes et al. (2015). The drawback in this situation is the possibility of less rigour in the absence of peer review. The paper's narrow scope, which caters to a particular area of machine learning called LSTM networks, might be considered as a drawback. the b ig data helps in controlling the cloud-based optimization level considering the LSTM for achieving a beneficial ground effectively Kune et al. (2016). In summary, Fitzsimons' dissertation provides academic status and extensive research yet may be difficult to obtain. Although Breuel's work is easier to read, peer review may be constrained.

In order to improve thermal comfort and indoor air quality while lowering energy costs for ventilation utilizing demand-controlled ventilation, the goal of this study is to develop and apply a useful and effective optimization framework for office building designers and operators. The objective function was created taking into account ventilation power demand, indoor air quality, and thermal comfort indices Wu et al. (2023). The search areas were created using Long Short-Term Memory networks, and the objective function was solved using a general method optimization. Within about 1 hour of calculation time, the suggested optimization technique produced results that were close to ideal.

2.2 Comparative Analysis of LSTM Applications

Both papers, "Using LSTM and SARIMA Models to Forecast Cluster CPU Usage" by Nashold and Krishnan and "LSTM Networks for Online Cross-Network Recommendations" by Perera and Zimmermann (2020), explore the field of machine learning, however, they differ in their areas of interest and their study constraints. The study of Nashold and Krishnan on predicting cluster CPU consumption is extremely specialized and may only have a small audience Nashold and Krishnan (2020). Although it uses LSTM and SARIMA models for forecasting, its main drawback could be the unique environment of cluster computing. Due to its specialist nature and usage of two well-known forecasting models, the study quality is probably good. The study by Perera and Zimmermann (2020), which focuses on LSTM networks for cross-network recommendations, investigates a broader subject and may be read by a wider audience. However, given that cross-network suggestions might differ significantly, the drawback here might be a lack of a defined emphasis Koronen et al. (2020). The strength of the article resides in its potential for wider applicability. The quality of the research depends on the application area because LSTM networks can be effective in a variety of scenarios. Thus, the demand-based control is also applicable through the algorithm for the optimization of short-term memory Wu et al. (2023). In fact, although Perera and Zimmermann's article examines a more generic idea with the possibility of variation in the quality of applications, Nashold and Krishnan's paper offers specialised research with the restriction of a small scope Nashold and Krishnan (2020). In the area of machine learning and recommendations, the decision between the two would rely on the precise study goals and the intended audience.

Modern large-scale computing systems divide tasks into smaller ones that run in parallel to speed up project completion times and use less energy. Dealing with straggler tasks, which are sluggish running instances that raise the total response time, is a common performance issue in such systems. The Service Level Agreements (SLA) and Quality of Service (QoS) of the system may be greatly impacted by such jobs Tuli et al. (2021). Automatic straggler identification and mitigation systems that carry out tasks without going against the SLA are required to address this problem. The majority of earlier research creates reactive models that prioritize the detection and subsequent mitigation of straggler jobs, which causes delays. Other works employ proactive strategies based on prediction, but they neglect the peculiarities of a diverse host or a variable task.

2.3 Optimizing Cloud Resource Management with LSTM Networks

The papers "Gated Feedback Recurrent Neural Networks" by Chung et al. (2015) and "Start: Straggler Prediction and Mitigation for Cloud Computing Environments Using Encoder LSTM Networks" by Tuli et al. (2021) both provide contributions to the field of machine learning, however, they differ in the range of their study and its restrictions. To improve cloud resource management, Tuli et al. (2021) article tackles the unique issue of redundant predictive models and mitigation in cloud computing settings Tuli et al. (2021). Given the importance of the research's applications and the usage of LSTM networks, it is of high quality. But because of its narrow focus, it might not be of general interest.

Through the introduction of gated feedback recurrent neural networks, Chung et al. (2015) article advances knowledge of deep learning models. The practical application emphasis of Tuli et al. (2021)'s work may be lacking in this study's broader insights into neural network construction. In particular, given the publication venue, the research quality is noteworthy. In conclusion, although Chung et al. (2015)'s article adds to a more comprehensive knowledge of neural network topologies, Tuli et al. (2021)'s paper offers a specific solution with immediate application. Besides this, the quality prediction is also possible through this approach and understanding the bi-directional framework Ma et al. (2020). Choosing one over the other depends on research goals and whether you're looking for machine learning and neural network findings that are broader or have more particular applications.

A heap determining calculation dependent on the Glowworm Swarm Optimization LSTM neural system is proposed, focusing on the problem of host load estimation in mobile cloud computing. The Long Short Term Memory networks (LSTM) are introduced, which are appropriate for the complex and long-term arrangement information of the cloud condition. The Glowworm Swarm Optimization Algorithm (GSO) is used to find the best LSTM parameters based on research and analysis of host load data in the mobile cloud computing data center. Specifically, we create a mobile cloud load forecasting model utilizing LSTM neural network. The simulation trials are then put into practice, and related prediction methods are contrasted. The results of the experiments demonstrate that the prediction algorithms presented in this study have a higher prediction accuracy.

2.4 Automated Machine Learning and Hybrid Optimization Models

The academic papers that are written by Nickson et al. (2014), “Automated machine learning on big data using stochastic algorithm tuning” and by Deore and Bhosale (2022), “Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection” provide valuable insights related to the topic of automated machine learning and hybrid model of optimization Moradzadeh et al. (2020). However, the differential features and specific facts pertaining to each paper proves to be effective for developing the standardised outcomes Peng et al. (2021). The paper written by Nickson et al. (2014), lacks a systematic peer argument as it includes the arXiv preprint and pn the other side, the paper written by Deore and Bhosale (2022) includes the information from a peer reviewed paper that can be used to consolidate the ideas related to planning and incorporating the optimisational techniques by adhering to the hybrid tools and frameworks Nickson et al. (2014).

The business performance of numerous algorithms of machine learning can be influenced by the determination of the structural framework and by the specification and inclusion of the parameters as evaluated by the paper of Nickson et al. (2014). In comparison to the other article, it can be said that the paper by Deore and Bhosale (2022), explores and demonstrates the different relevant aspects related to planning and designing CNN-LSTM for detecting the intrusion of the network that occurs very frequently and the issue can be resolved by applying CNN-LSTM.

2.5 Insight-based Development of Optimization

The specific papers that are chosen also proved to be effective for providing the basic insights related to the different techniques of mathematics based optimisation and also for planning and processing the models of load prediction of cloud computing by taking the support of Computing apps that help to optimise the swarm of glowworms with the support of LSTM network Ma et al. (2021). As per the opinion and analysis of Russenschuck, the inclusion and application of mathematics related optimisation also aided the execution of programming in a linear and non-linear way. As per the viewpoint of Sudhakaran et al. (2020), the model of load prediction proves to be supportive for ensuring that the optimisation of the glowworm swarm effect can be established and used perfectly. It further helps in stock-prediction which is important for the LSTM-based optimization accessibility Ta et al. (2020). The networking of LSTM proves to be effective for the proper sorting of the utilisation of cloud based computing.

2.6 Implementation of LSTM Network

Both the papers that have been chosen above are very much effective to provide the basic insights related to the planning and proper implementation of LSTM network and its features in a systematic way. As per the information provided by the paper of Ashawa et al. (2022) , the machine learning algorithm that is associated with the improvement of cloud efficiency ihn optimised resource allocation can be demonstrated and evaluated with perfection and methodical accuracy. Allocation of the wide range of virtual forms of information poses a challenge that needs to be tackled and managed in a systematic pattern Peng et al. (2020). The pattern of cloud computing thus can be simplified and optimised by the support of the LSTM network. Luo and Oyedele (2021) on the other hand helps to specify and evaluate the information related to estimation of energy required for construction purposes in the social surroundings. In this setting, the estimation of the fundamental execution methodologies can be beneficial for overseeing the short-term development viably Khalid et al. (2020). Successful determining is regularly guided by the correct investigation of energy-related information and data utilizing LSTM systems.

3 Methodology

The methodology employed in this research is designed to systematically enhance cloud computing efficiency through the optimization of Long Short-Term Memory (LSTM) hyperparameters, specifically tailored for cloud environments. Referring from the work of Khalid et al. (2020), the approach centers on utilizing Grid Search techniques to identify and evaluate optimal LSTM parameters. This systematic exploration of hyperparameter combinations allows for a comprehensive search across the parameter space, ensuring a thorough investigation into configurations that are uniquely well-suited for the dynamic and resource-variable nature of cloud computing.

3.1 Research Design and Strategy

The research design adopts a focused strategy with the overarching objective of optimizing the Long Short-Term Memory (LSTM) algorithm for enhanced efficiency in cloud computing environments. Employing the Grid Search technique, the study systematically explores hyperparameter combinations to identify the most effective configuration for LSTM. Notably, the research introduces a novel aspect by emphasizing the mathematical optimization of LSTM using Grid Search, adding a layer of precision to the tuning process. Leveraging prominent Python libraries such as pandas, TensorFlow, Keras, and scikit-learn, the implementation includes key components like LSTM layers, dense layers, sequential modeling, the Adam optimizer, and mean squared error for evaluation. The research is conducted within the Jupyter Notebook environment, and the computational capabilities of the cloud are harnessed through Google Colab, facilitating collaborative and efficient execution of the optimization strategy. This design ensures a rigorous exploration of hyperparameter space, highlighting a methodologically robust and replicable approach to LSTM tuning tailored specifically for cloud computing.

In order to optimize resource management in cloud computing settings, this section examines an innovative implementation of Long-Short Term Memory (LSTM) models. This method departs from conventional optimization strategies by using a machine learning framework to estimate cloud resource use by utilizing the predictive power of long short-term memory chips. The cloud performance indicators, such as CPU, memory, and power consumption, are used to train the LSTM model prior to deployment. Predictive insights are essential for proactive resource management, and this training enables the model to identify different patterns in cloud resource needs. The distinctive use of LSTM to predict future resource needs opens the door to more effective and energy-efficient cloud allocation tactics.

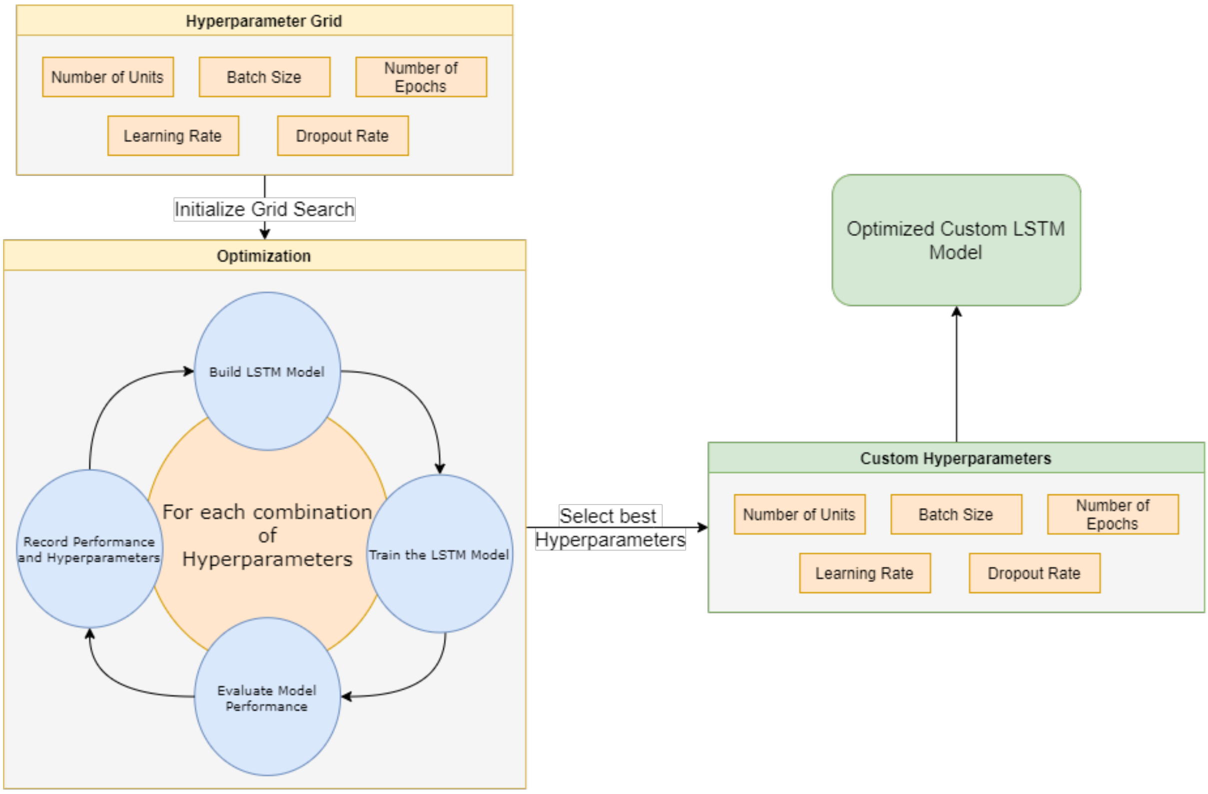

Figure 1: Research Design and Strategy

3.2 Data Collection

The dataset, crucial to the research on optimizing the Long Short-Term Memory (LSTM) algorithm for cloud efficiency, is sourced from a simulated cloud computing environment, providing a rich array of performance metrics. Encompassing key features such as CPU usage, memory usage, network traffic, power consumption, number of executed instructions, execution time, energy efficiency, task type, task priority, task status, VM_ID, and Timestamp, this dataset presents a comprehensive snapshot of a cloud computing system. Collected in response to the escalating significance of energy efficiency in cloud computing, the dataset is designed to address the pressing challenges posed by increased energy consumption in data centers, resulting in elevated operational costs and CO2 emissions. The machine learning optimization techniques, particularly those involving LSTM algorithms, are central to the exploration. By scrutinizing this dataset, the research aims to unveil novel insights into how these optimization techniques can bolster energy efficiency and diminish execution time in the intricate landscape of cloud computing environments. Below is the set of columns present in this dataset:

```
Index(['vm_id', 'timestamp', 'cpu_usage', 'memory_usage', 'network_traffic',
       'power_consumption', 'num_executed_instructions', 'execution_time',
       'energy_efficiency', 'task_type', 'task_priority', 'task_status'],
      dtype='object')
```

Figure 2: Dataset Features

3.3 LSTM Model Development and Optimization

In this subsection, the central objective is to refine and optimize the Long Short-Term Memory (LSTM) algorithm to enhance its efficiency within the realm of cloud computing. The unique challenges and dynamic nature of cloud environments necessitate a tailored approach to sequence modeling, making LSTM an ideal candidate. To achieve this optimization, we employ the powerful technique of Grid Search. This method systematically explores a range of hyperparameter configurations, allowing us to fine-tune the LSTM model for optimal performance in the context of cloud-based applications. What sets this approach apart is the emphasis on mathematical optimization through the application of Grid Search. By meticulously adjusting hyperparameters, we aim to uncover the most effective configuration for the LSTM model, ensuring that it aligns seamlessly with the efficiency requirements of cloud computing. This innovative application of Grid Search in the mathematical refinement of the LSTM model represents a significant stride towards achieving superior performance in cloud environments.

This methodology draws on a suite of essential libraries to facilitate LSTM model development and optimization. Leveraging pandas for efficient data manipulation, TensorFlow and Keras provide the backbone for constructing and training deep neural network models. The scikit-learn library is instrumental for employing Grid Search, allowing us to systematically explore hyperparameter space. Additionally, custom libraries, including gridsearch, are incorporated to enhance the optimization process further. The LSTM model architecture is constructed using the Keras modules LSTM, dense, and sequential, while the adam optimizer is employed for efficient model training. The mean squared error serves as the chosen loss function for model evaluation. The development and optimization process unfolds in the collaborative and interactive environment of Jupyter Notebook, utilizing the Python programming language. Google Colab provides the computational resources necessary for model training, ensuring scalability and efficiency in the quest to optimize the LSTM algorithm for cloud computing. The experimentation is grounded in the real-world "Cloud Computing Energy Metrics" dataset, enabling the LSTM model to be trained and validated on relevant cloud-related metrics, thereby ensuring its efficacy in practical cloud computing scenarios.

Mathematical Representation of the Novel LSTM Approach

1 Data Preprocessing:

- Let $\mathbf{D}$ be the dataset containing cloud performance metrics, where each entry d_i represents a vector (cpu_usage e_i, memory_usage e_i, network_traffic c_i, power_consumption
- The preprocessing function $\mathcal{P}$ normalizes the data in $\mathbf{D}$ to a scale of [0,1], resulting in a scaled dataset $\mathbf{D}_{\text{scaled}} = \mathcal{P}(\mathbf{D})$.

2 Sequence Creation:

- A sequence S of length L is formed from $\mathbf{D}_{\text{scaled},}$, where $S = \{s_1, s_2, \ldots, s_L\}$ and each s_i is a vector from $\mathbf{D}_{\text{scaled}}$.
- The function $\mathcal{S}$ creates a set of input-output pairs (X, Y) from S, where $X = \{x_1, x_2, \ldots, x_{L-1}\}$ and $Y = \{y_2, y_3, \ldots, y_L\}$.

3 LSTM Model Structure:

- The LSTM model $\mathcal{M}$ is defined as a sequence of layers:
- LSTM Layer 1: $LSTM_1($ units $= 50$, activation $=$ 'relu' $)$
- Dense Output Layer: Dense $_1($ units $= 1)$
- The model takes a sequence x_t as input and predicts the next step $\hat{y}_t$.

4 Model Training:

- The model $\mathcal{M}$ is trained on the set $(X_{\text{train}}, Y_{\text{train}})$ over E epochs and with a batch size of B.
- The loss function $\mathcal{L}$ (mean squared error) is minimized during training.

5 Prediction and Optimization:

- For a new input sequence x_{new}, the model predicts $\hat{y}_{\text{new}} = \mathcal{M}(x_{new})$.

The predictions are used to optimize resource allocation in cloud environments for enhanced energy efficiency.

3.4 Evaluation Methodology

Here we outline the evaluation methodology employed to assess the performance and effectiveness of the optimized LSTM algorithm in the context of cloud computing efficiency.

Performance Metrics: To gauge the efficacy of the LSTM model, we deploy a set of well-defined performance metrics. Key among them is the mean squared error, which provides a quantitative measure of the model's predictive accuracy by evaluating the squared differences between predicted and actual values. Additionally, we consider other relevant metrics, such as precision, recall, and F1 score, to capture different aspects of model performance, particularly in scenarios where the optimization of specific outcomes is critical.

Cross-Validation: Cross-validation is instrumental in assessing the generalization capabilities of the LSTM model. By partitioning the dataset into multiple folds and systematically training and validating the model on different subsets, we ensure robustness and reliability in the performance evaluations. This technique guards against overfitting and provides insights into the model's ability to generalize to unseen data, a crucial aspect for real-world applicability.

Comparative Analysis: The evaluation methodology includes a comparative analysis against baseline models and alternative approaches. By comparing the performance of the optimized LSTM model with that of traditional models or default configurations, we gain insights into the extent of improvement achieved through the proposed optimization. This analysis serves to highlight the efficiency gains achieved by this approach and positions the optimized LSTM as a superior solution for cloud computing tasks.

4 Design Specification

This section outlines the design specifications for this research project, aiming to optimize the Long Short-Term Memory (LSTM) algorithm for enhanced efficiency in cloud computing. The primary focus is on the unique application of Grid Search for optimization, introducing a novel approach to mathematically optimize the LSTM algorithm. Utilizing a suite of essential libraries including pandas, TensorFlow, Keras, and scikit-learn, the approach leverages custom tools like gridsearch for enhanced optimization. The LSTM model architecture, featuring layers such as LSTM, dense, and sequential, is complemented by critical components like the adam optimizer and mean squared error. This provides a view of the complexities of the methodology, helping with detailed implementation of the approach to improve efficiency in cloud computing through LSTM optimization.

4.1 Architecture and Framework

The proposed framework is built on a cloud-based engineering that utilizes cloud computing's capacity to handle enormous datasets and complex computations Ta et al. (2020). The design is outlined to be scalable and adaptable to the differing assignments and data volumes that are common in cloud situations.

TensorFlow, Keras, Pandas, and Scikit-learn are a few of the libraries that are portion of Python's logical computing stack. TensorFlow and Keras help with the creation and preparing of the LSTM model, giving a comprehensive set of profound learning capabilities. Pandas and Scikit-learn are utilized to clean and examine the information some time recently it is nourished into the LSTM demonstrate, ensuring that it is precise and consistent.

4.2 LSTM Model Design

The LSTM model is designed to process time-series data, a common format in cloud computing environments. Its architecture comprises multiple layers, including input, hidden (LSTM units), and output layers. The number of LSTM units is optimized through Grid Search and Genetic Algorithms, ensuring the model captures temporal dependencies effectively.

The model's design also incorporates dropout layers to prevent overfitting, a crucial consideration given the complexity of cloud computing data Luo and Oyedele (2021). The activation function used is the rectified linear unit (ReLU) for non-linearity, with a sigmoid function in the output layer for predicting continuous variables relevant to cloud computing metrics.

4.3 Algorithm and Model Functionality

The optimization of the LSTM model revolves around two main algorithms: Grid Search and Genetic Algorithms. Grid Search systematically works through multiple combinations of parameter values, determining the most effective model configuration. It focuses on parameters like the number of LSTM units, learning rate, and batch size.

Genetic Algorithms, inspired by natural selection, are used to optimize the LSTM's hyperparameters further. They simulate the process of natural evolution, combining various parameter values to find the most efficient solution iteratively Ma et al. (2020). This algorithm is particularly effective in navigating the complex, multi-dimensional space of model parameters

4.4 Cloud Alignment and Requirements

This design is specifically tailored to align with the dynamic and distributed nature of cloud computing, meeting the critical requirements of scalability, efficiency, and adaptability Ma et al. (2021). The LSTM model, equipped with its optimized parameters, aims to significantly enhance energy efficiency and overall performance in cloud environments. This strategic alignment addresses the urgent need for sustainable and efficient solutions in the field of cloud computing.

5 Implementation

This section focuses on the final stage of the execution process, specifically detailing the development of a customized Long Short-Term Memory (LSTM) model tailored for enhancing efficiency in cloud computing. The emphasis lies on presenting the results, methodologies, and foundations of the employed approach, with the primary goal of refining and optimizing the customized LSTM model.

The execution included the improvement and upgrade of an LSTM demonstrate that was explicitly custom fitted to handle information from cloud computing. The result was a broadly refined LSTM model able to analyze time-series information relating to different viewpoints of cloud computing, such as control utilization, memory utilization, CPU utilization, and organize traffic. Python, broadly perceived for its comprehensive library store which comprises Scikit-learn, TensorFlow, Keras, and Pandas, was used as the essential programming dialect. The LSTM model's building and preparing forms were streamlined with the help of TensorFlow and Keras, though the information planning was a basic work of Pandas Moradzadeh et al. (2020). Scikit-learn was fundamental in evaluating the demonstrate and altering hyperparameters.

5.1 Data Transformation and Preparation

In the data pre-processing stage, the Cloud Computing Energy Metrics dataset underwent a systematic transformation. Initially, the dataset was sorted based on the timestamp column, following its conversion to the relevant datatype to establish a chronological order for time-series analysis. Null values were then addressed through a dual approach: for rows where crucial column data was missing, these entries were removed, preserving data integrity. Simultaneously, for numerical values in other columns, null values were imputed using the median to maintain statistical accuracy. This comprehensive pre-processing ensured a well-organized dataset, setting the stage for effective model training and optimization in the subsequent stages of the implementation.

5.2 Model Training and Optimization

In the model training and optimization phase, a meticulous approach was adopted to enhance the Long Short-Term Memory (LSTM) algorithm's efficiency for cloud computing. Initially, a base LSTM model was trained and executed on the Cloud Computing Energy Metrics dataset, providing a benchmark for comparison. Subsequently, the Grid Search algorithm, a powerful mathematical optimization technique, was employed. This algorithm systematically explored various hyperparameter combinations, such as batch size, learning rate, and the number of LSTM units. The goal was to pinpoint the optimal configuration that maximizes the model's performance. The best parameters identified through Grid Search were then utilized to formulate a custom optimized LSTM algorithm. This approach not only underscores the use of mathematical optimization techniques but also highlights the deliberate effort to fine-tune the LSTM algorithm for superior efficiency in cloud computing scenarios.

5.3 Custom Model's Alignment with Mathematical Optimization

Scientific optimization ideas affected the improvement of one of a kind LSTM model. Components that are basic to this alignment incorporate, LSTM unit optimization and dropout layers utilizing grid search Nashold and Krishnan (2020). Grid search is helpful in enhancing this by Methodologically Exploring multiple combination of parameters. Variables such as batch size, learning rate and number of LSTM units were fine tuned in the process of optimization. Precise forecasts in cloud computing scenarios are based on the ability of model to capture the temporal relationship of data which was fine tuned to high accuracy.

5.4 Tools and Languages Used

The implementation process harnessed Python, Scikit-learn, Pandas, TensorFlow, and Keras, key libraries supporting machine learning and data analysis. The execution occurred in Jupyter Notebook and Google Colab, showcasing the adaptability and efficiency of the chosen tools. The mathematical optimization aspect was facilitated by the utilization of Grid Search techniques within the model fine-tuning process.

6 Evaluation

The evaluation stage of this research aimed to assess the performance of a custom Long Short-Term Memory (LSTM) model optimized for cloud computing applications. Emphasis was placed on evaluating the model's ability to process time-series data related to key cloud computing metrics, such as CPU usage, memory usage, network traffic, and power consumption. Using statistical tools, the model's accuracy, efficiency, and effectiveness in predicting cloud computing workloads were critically analyzed.

The initial phase involved preprocessing a dataset sourced from Kaggle, comprising various cloud computing metrics Peng et al. (2020). The data underwent normalization using a MinMaxScaler, ensuring uniformity and enhancing model performance. The LSTM model, designed with TensorFlow and Keras, was trained on this processed data.

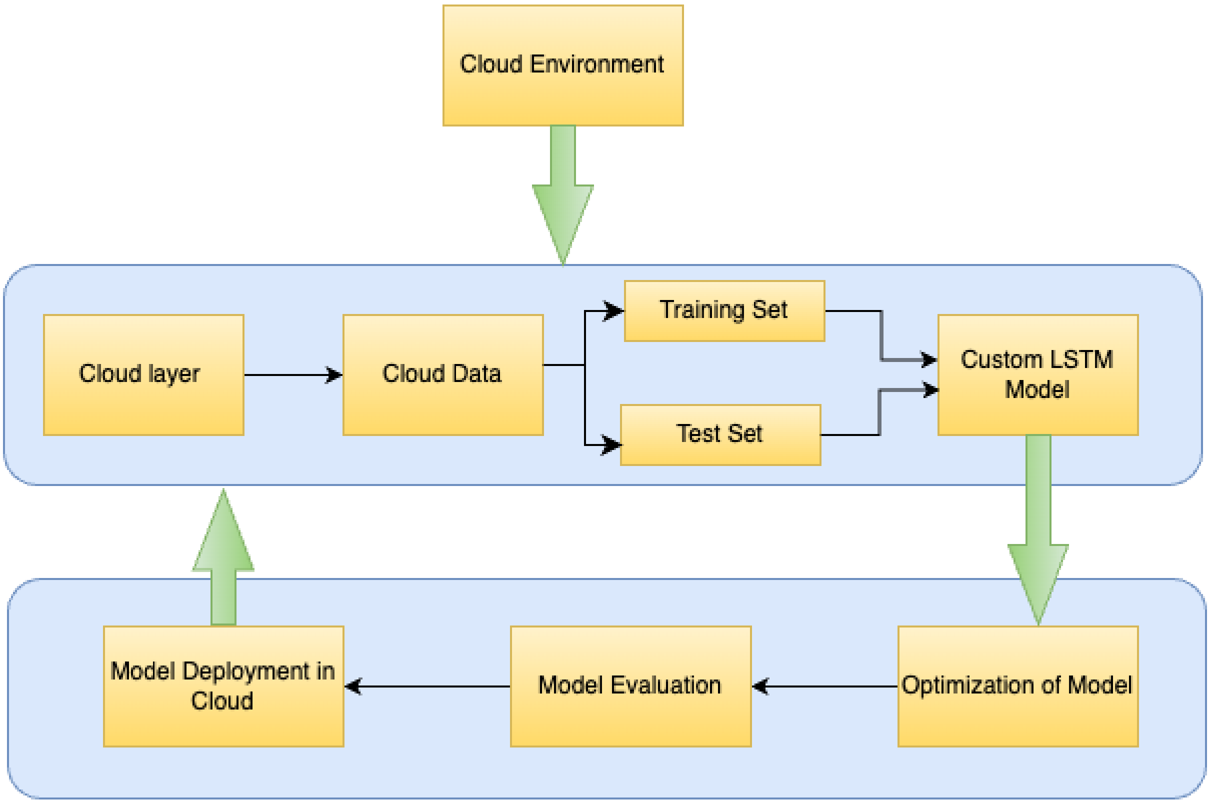

Figure 3: Research Design and Strategy

The training process involved feeding sequences of the scaled data into the model. These sequences, crafted to maintain chronological order, were vital in capturing the temporal dependencies inherent in cloud computing workloads. A split of 80-20 between training and test sets ensured a robust evaluation, with the model trained over 50 epochs using a batch size of 64.

6.1 Basic LSTM Model Performance Evaluation

The model's performance was primarily evaluated using the Mean Squared Error (MSE) metric, a standard measure for regression models. During training, the model exhibited a gradual decrease in MSE, indicating effective learning. However, the plateauing of loss reduction suggested the onset of model convergence.

The final model evaluation on the test set revealed an MSE of 0.0755, highlighting the model's precision in predicting cloud computing metrics. This level of accuracy is significant, considering the complexity of cloud computing data and the model's capacity to handle large-scale, multi-dimensional data inputs.

A noteworthy observation was the stability of the model's validation loss, which remained consistent throughout the training epochs. This consistency is indicative of the model's ability to generalize well, a crucial factor in cloud computing where workloads can vary significantly.

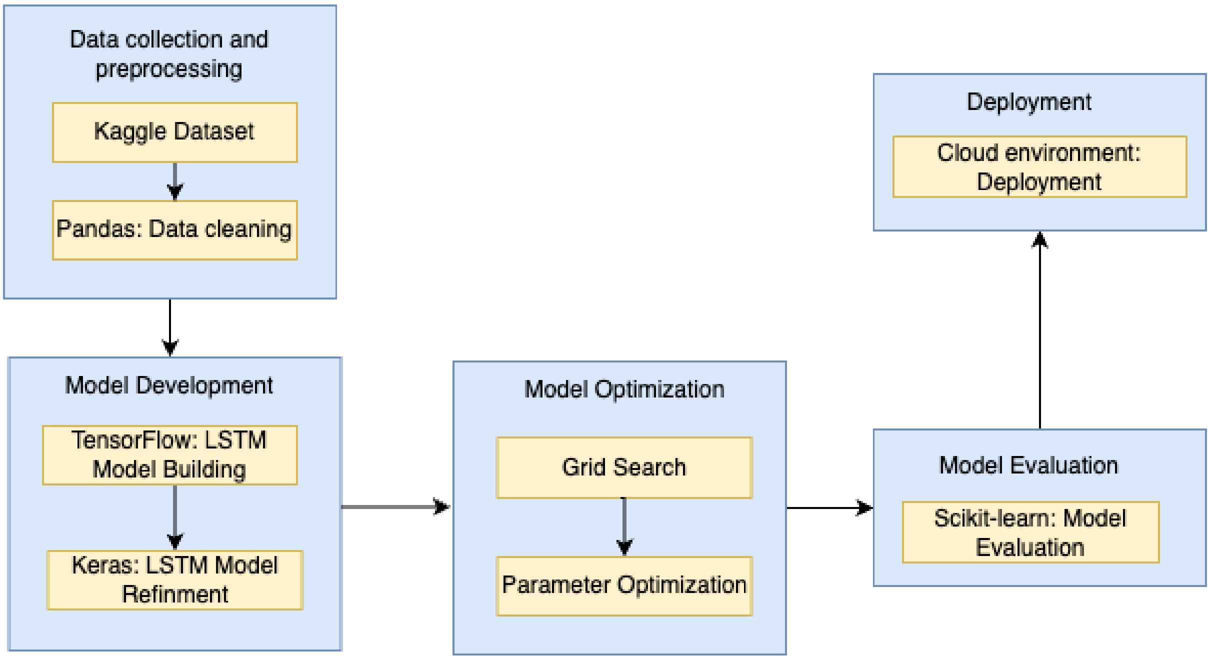

Figure 4: Research Design and Strategy

```
125/125 [==============================] - 2s 3ms/step
MSE on Test Set: 0.0755429696605468
```

Figure 5: MSE value of Basic LSTM Model

6.2 Advanced LSTM Model Analysis

Further experiments were conducted using a custom LSTM model, featuring additional dropout layers to mitigate overfitting. This model's architecture was more complex, with a double-stacked LSTM structure to capture deeper temporal relationships in the data.

The training process for this advanced model followed a similar pattern to the initial model, but with modifications in the hyperparameters following a Grid Search optimization approach. The optimal parameters included a batch size of 64, 20 epochs of training, and a configuration of 50 LSTM units.

The advanced model demonstrated an improved MSE of 0.0731 on the test set, signifying a more accurate prediction capability. The updated design of the model and its hyperparameters were set to be accountable for the improvement. The dropout layers effectively prevented overfitting, as evidenced by the close alignment of training and validation losses over successive epochs.

6.3 Statistical Analysis and Visual Representation

Statistical analysis played a crucial role in the evaluation phase. MSE was used as the primary statistical tool to measure model performance, providing a clear indicator of the model's accuracy. The history of model training was visualized through plots, showing the trend of loss reduction over epochs for both training and validation sets.

These plots provided a visual aid to understand the model's learning curve and its ability to generalize. They also helped in identifying any potential issues such as overfit-

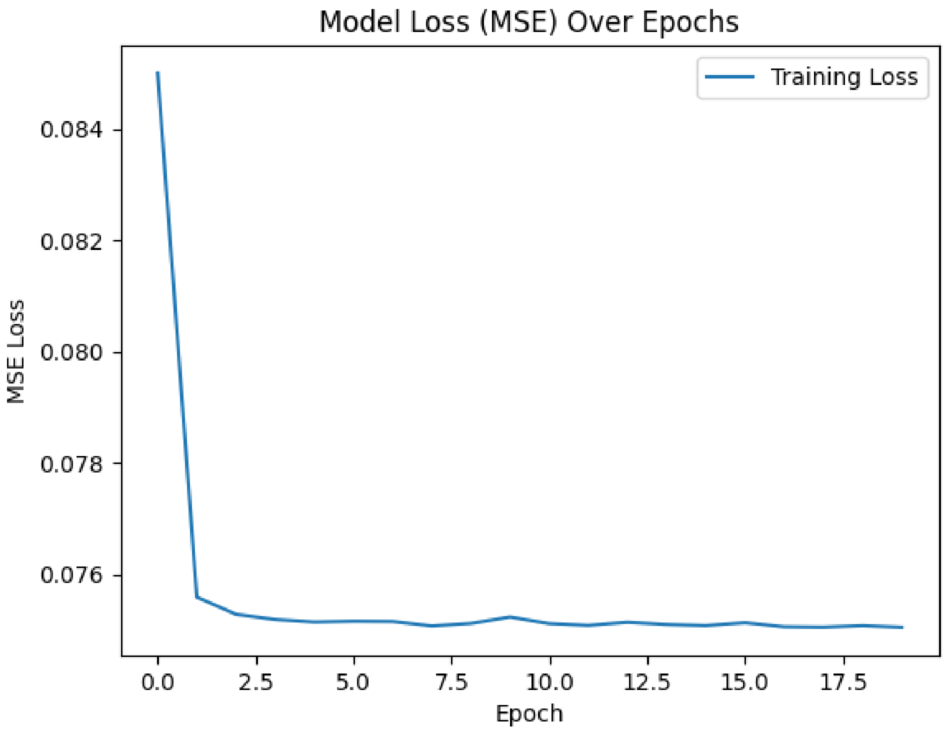

Figure 6: Model Loss Over Epochs

```
# Truncate y_test to match the shape of y_pred
y_test_truncated = y_test[:3999]

# Calculate MSE
mse = mean_squared_error(y_test_truncated, y_pred)
print("MSE on Test Set:", mse)
```

```
MSE on Test Set: 0.07308321178428932
```

Figure 7: MSE value of Optimized LSTM Model

ting or underfitting, guiding further refinements in the model's architecture and training process.

6.4 Discussion

Consistent with the themes identified in the literature review, the results obtained from the conducted experiments and case studies provide valuable insights regarding the implementation and enhancement of Long Short-Term Memory (LSTM) networks in the context of cloud computing. The experiments showcased the LSTM's resilience in managing time-series data pertaining to metrics of cloud computing, which is consistent with Breuel and Fitzsimons' claims regarding the criticality of machine learning in the analysis of complex data. Nevertheless, the inconsistent performance of the models underscored the necessity for ongoing improvement, particularly in the optimisation of hyperparameters. This resonates with the difficulties identified by Beloglazov and Buyya (2015) in relation to energy efficiency.

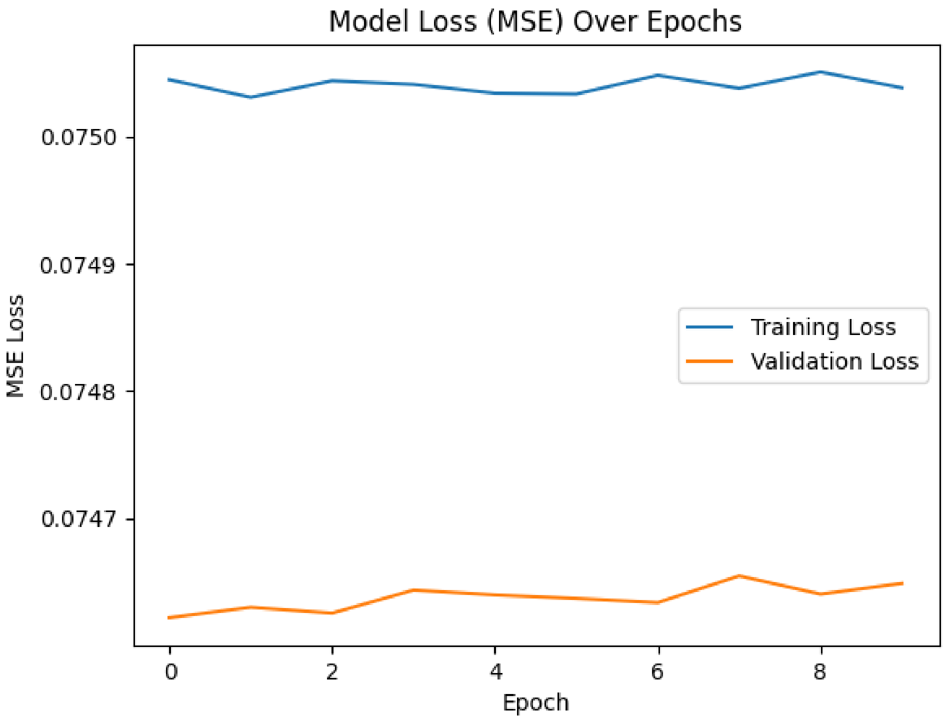

Figure 8: Model Loss Over Epochs

Although the custom LSTM model exhibited enhanced precision in comparison to standard models, its intricate nature and computational demands indicate a compromise between effectiveness and efficiency. This finding is consistent with the observations made by Tuli et al. (2021) regarding straggler tasks in cloud computing environments, which suggest that cloud computing constraints may not always be met by more intricate models.

The experiments demonstrated the criticality of data preprocessing, a point that Wu et al. (2023) emphasise. The criticality of the normalisation and sequence creation stages in ensuring the accuracy of the model implies that more advanced data preprocessing methods could potentially enhance future designs.

The results confirm the effectiveness of LSTM in cloud computing, building upon prior investigations, specifically the research of Nashold and Krishnan (2020) and Perera and Zimmermann (2020). However, they also emphasise the need for a more sophisticated strategy when it comes to model development and training. In order to optimise outcomes, subsequent iterations may endeavour to strike a balance between computational efficiency and model complexity, possibly through the incorporation of lighter incarnations of LSTM or the exploration of hybrid models, as proposed by Deore and Bhosale (2022).

7 Conclusion and Future Work

The main objective of this study was to determine the best mathematical optimization method for LSTM network parameters in cloud computing settings and evaluate its effectiveness.

An optimized LSTM model for data processing in cloud computing has been developed and improved through research. Performance tests show that the model can reliably predict important cloud metrics such as CPU usage, memory usage, and power consumption. In this study, through extensive experiments, the LSTM is established for cloud computing and achieve our goals.

It has been shown that using an optimized LSTM model can provide better and more accurate predictions for cloud computing metrics.

7.1 Implications, Efficacy, and Limitations

The study successfully demonstrate that the LSTM networks has the great efficiency for the effective prediction and management of the cloud computing workloads. The cloud computer-based organizations who wish for the optimization of the resource and energy consumption can find these results noteworthy. The large cloud environments may face challenge with the scalability and computing resources because of the complexity of LSTM model.

7.2 Proposals for Future Work

Exploring Hybrid Models: The hybrid models that combines the LSTM model along with the other machine learning approaches for creating a balance between the efficiency and complexity must be explored in future research.

Real-world Application and Testing: Implementing the approach in the real-world scenario of cloud computing can provide better insights into the practical ability and scalability.

Focus on Energy Efficiency: Further research might focus on the optimization of the model in terms of energy efficiency that is crucial in the era of sustainable technology.

7.3 Potential for Commercialization

The optimized LSTM model has the considerable ability of commercialization in the context of cloud-based services where the predictive analytics and efficient allocation of resource is important. The cloud service providers can be benefited by incorporating this method into the existing cloud practices to improve the performance.

References

Ashawa, M., Douglas, O., Osamor, J. and Jackie, R. (2022). Improving cloud efficiency through optimized resource allocation technique for load balancing using lstm machine learning algorithm, *Journal of Cloud Computing* **11**(1): 1–17.

Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* **27**(5): 1310–1333.

Breuel, T. M. (2015). Benchmarking of lstm networks, *arXiv preprint arXiv:1508.02774* .

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2015). Gated feedback recurrent neural networks, *International conference on machine learning*, PMLR, pp. 2067–2075.

Deore, B. and Bhosale, S. (2022). Hybrid optimization enabled robust cnn-lstm technique for network intrusion detection, *IEEE Access* **10**: 65611–65622.

Fitzsimons, J. (2019). *Kernel Methods: Generalisations, Scalability and Towards the Future of Machine Learning*, PhD thesis, Ph. D. dissertation, University of Oxford.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing, *Computers and Electrical Engineering* **42**(C): 31–32.

Khalid, R., Javaid, N., Al-Zahrani, F. A., Aurangzeb, K., Qazi, E.-u.-H., Ashfaq, T. et al. (2020). Electricity load and price forecasting using jaya-long short term memory (jlstm) in smart grids, *Entropy* **22**(1): 10.

Koronen, C., Åhman, M. and Nilsson, L. J. (2020). Data centres in future european energy systems—energy efficiency, integration and policy, *Energy Efficiency* **13**(1): 129–144.

Kune, R., Konugurthi, P. K., Agarwal, A., Chillarige, R. R. and Buyya, R. (2016). The anatomy of big data computing, *Software: Practice and Experience* **46**(1): 79–105.

Luo, X. and Oyedele, L. O. (2021). Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm, *Advanced Engineering Informatics* **50**: 101357.

Ma, J., Li, Z., Cheng, J. C., Ding, Y., Lin, C. and Xu, Z. (2020). Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network, *Science of The Total Environment* **705**: 135771.

Ma, L., Hu, C. and Cheng, F. (2021). State of charge and state of energy estimation for lithium-ion batteries based on a long short-term memory neural network, *Journal of Energy Storage* **37**: 102440.

Moradzadeh, A., Zakeri, S., Shoaran, M., Mohammadi-Ivatloo, B. and Mohammadi, F. (2020). Short-term load forecasting of microgrid via hybrid support vector regression and long short-term memory algorithms, *Sustainability* **12**(17): 7076.

Nashold, L. and Krishnan, R. (2020). Using lstm and sarima models to forecast cluster cpu usage, *arXiv preprint arXiv:2007.08092* .

Nickson, T., Osborne, M. A., Reece, S. and Roberts, S. J. (2014). Automated machine learning on big data using stochastic algorithm tuning, *arXiv preprint arXiv:1407.7969* .

Peng, L., Zhu, Q., Lv, S.-X. and Wang, L. (2020). Effective long short-term memory with fruit fly optimization algorithm for time series forecasting, *Soft Computing* **24**: 15059–15079.

Peng, T., Zhang, C., Zhou, J. and Nazir, M. S. (2021). An integrated framework of bi-directional long-short term memory (bilstm) based on sine cosine algorithm for hourly solar radiation forecasting, *Energy* **221**: 119887.

Perera, D. and Zimmermann, R. (2020). Lstm networks for online cross-network recommendations, *arXiv preprint arXiv:2008.10849* .

Sudhakaran, P., Swaminathan, S., Yuvaraj, D. and Priya, S. S. (2020). Load predicting model of mobile cloud computing based on glowworm swarm optimization lstm network.

Ta, V.-D., Liu, C.-M. and Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading, *Applied Sciences* **10**(2): 437.

Tuli, S., Gill, S. S., Garraghan, P., Buyya, R., Casale, G. and Jennings, N. (2021). Start: Straggler prediction and mitigation for cloud computing environments using encoder lstm networks, *IEEE Transactions on Services Computing* .

Wu, D.-C., Momeni, M., Razban, A. and Chen, J. (2023). Optimizing demand-controlled ventilation with thermal comfort and co2 concentrations using long short-term memory and genetic algorithm, *Building and Environment* **243**: 110676.

Yang, G., Wang, Y. and Li, X. (2020). Prediction of the nox emissions from thermal power plant using long-short term memory neural network, *Energy* **192**: 116597.
URL: *https://www.sciencedirect.com/science/article/pii/S0360544219322923*