National College of Ireland

# INVESTIGATING THE SIGNIFICANCE OF PROXY-BASED CONNECTION IN HYBRID CLOUD VIRTUAL NETWORK

MSc Research Project
Cloud Computinng

Trupti Kathane
Student ID: 22216456

School of Computing
National College of Ireland

Supervisor: Sean Heeney

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Trupti Kathane |
| **Student ID:** | 22216456 |
| **Programme:** | Cloud Computing |
| **Year:** | 2025 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Sean Heeney |
| **Submission Due Date:** | 03/01/2025 |
| **Project Title:** | Investigating the Significance of Proxy-Based Connection in Hybrid Cloud Virtual Network |
| **Word Count:** | XXX |
| **Page Count:** | 21 |

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:** | Trupti Kathane |
| **Date:** | 28th January 2025 |

## 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): | |

# INVESTIGATING THE SIGNIFICANCE OF PROXY-BASED CONNECTION IN HYBRID CLOUD VIRTUAL NETWORK

Trupti Kathane
22216456

**Abstract**

Hybrid cloud virtual networks combine private and public cloud environments to deliver both scalable, and efficient IT solutions. In this research study, the usefulness of proxy-based connections in hybrid cloud, e.g. security, latency, and compatibility. The study simulates proxy-based connections and benchmarks with traditional networking solutions, using Kaggle dataset and Python tools such as SimPy and Jupyter Notebook. Proxies are found key in enhancing performance by minimizing latency processing, protecting the data transfer process, and improving scalability. Conclude with results that highlight the importance of proxies to manage hybrid cloud complexities and suggest actionable insights for academia and industry. The insights advocate for an integration of proxy-based solutions to build efficient, secure, and adaptive hybrid cloud networks. While it relies on simulated data, future validation in real world deployments is necessary. This study has informed research on proxies potential to help further the poorness of hybrid cloud technologies, accountability gaps in existing research, and priming additional study.

**Keywords:** Hybrid Cloud, Proxy-Based Connections, Security, Latency, Scalability, Performance, Data Protection, Real-World Validation

# 1 Introduction

## 1.1 Background on Hybrid Cloud Virtual Networks

Hybrid cloud virtual networks integrate private and public cloud infrastructures to offer flexibility, scalability and efficiency when addressed as IT solutions. Hybrid cloud environments offer organizations the ability to work with both on-premises resources and the agility of public cloud services, balancing workloads, minimizing costs, and improving business continuity. An important part of virtual networking is that it makes sure that we have continuous communication irrespective of different environments, carrying data, deploying applications, and integrating those services. Yet there are challenges involved, with respect to security, latency, and compatibility. In response to the ever-increasing adoption of hybrid models by businesses, there is a strong need for reliable, efficient, and secure connectivity methods.

## 1.2 Justification for Investigating Proxy-Based Connections

The proxy-based connections act as the intermediary between the clients and the servers to increase the security of the system, provide load balancing and manage the traffic efficiently. Proxy servers mitigate several challenges when applied in hybrid cloud virtual networks such as latency problems, unauthorized access, and data loss. Proxies effectively bridge the gap between private and public cloud segments, best resource allocation, propagate security policies, and maximize data transfer rates(Miller et al. (2021)). Despite these benefits, one lacks a thorough evaluation of how proxy-based connections substitute for hybrid cloud networking. This report attempts to close this gap by examining how crucial proxy-based connections serve to answer the questions of whether they can be implemented, and if so, how outstanding will their performance be. In this study, they derive insights to gain further knowledge of proxy techniques in hybrid cloud.

## 1.3 Objectives

- To analyze the performance of the proxy-based connections in the hybrid cloud virtual networks.
- To calculate the implications of implementing the proxy-based connections.
- To make the benchmark of the proxy-based connections against different alternative networking methods.

## 1.4 Research Questions

- How to analyze the performance of proxy-based connections in hybrid cloud virtual networks ?
- What are the enhancements provided by proxy-based connections in hybrid cloud virtual networks?
- How do proxy-based connections compare with alternative networking solutions in terms of scalability, resource efficiency, and complexity within hybrid cloud systems?

## 1.5 Hypotheses

- Proxy-based connections do not significantly improve the performance (e.g.,latency) of hybrid cloud virtual networks compared to traditional networking solutions.
- Proxy-based connections significantly enhance the performance (e.g., latency) of hybrid cloud virtual networks compared to traditional networking solutions.
- Proxy-based connections do not provide any measurable improvement in the security of hybrid cloud virtual networks.
- Proxy-based connections significantly enhance the security of hybrid cloud virtual networks by reducing unauthorized access and improving data encryption.
- Proxy-based connections are not more scalable or resource-efficient than alternative networking solutions in hybrid cloud virtual networks.

- Proxy-based connections provide superior scalability and resource efficiency compared to alternative networking solutions in hybrid cloud virtual networks.

# 2 Literature Review

## 2.1 Proxy Detection for Hybrid Cloud Environments

According to Miller et al. (2021), methods of using anonymizing proxies and VPNs are new challenges to securing virtual networks with growing adoption of hybrid cloud architectures. These technologies lead to users able to anonymize their identity thereby anonymizing their online activities which has serious issues of security and privacy. In many hybrid cloud environments, where on premise and cloud-based resources combine, the network architecture is increasingly complex and potential for attack such as using anonymizing proxies for authorized and unauthorized access to the cloud is high (Farkiani et al. (2024)). These proxies, like VPNs, are in between the user and web servers, and they disguise a user's original IP address in place of a proxy IP (Ramachandrappa et al. (2021)). This creates multiple important security issues both in virtual network environments where identifying unauthorized users or malicious actors is essential to maintain data security and follow regulatory standards. The efficacy of machine learning (ML) techniques in the detection of anonymizing proxies and VPNs usage is demonstrated in the network security domain.

Because of the dynamic and sophisticated nature of proxy tools, traditional techniques for detecting proxy-based connections such as IP address blacklisting or watch for proxy signature in the traffic fail. As a result, machine learning-based detection models have been developed more advanced. In this work, they focus on these models, including multi-layer perceptron neural networks, which are based on features derived from distinct network traffic layers, e.g., transmission control protocol (TCP) headers. Focusing on how these data points can be leveraged to enable machine learning algorithms to distinguish normal and proxy encapsulated traffic, propose more adaptive, real time and scalable solutions to detecting unauthorized proxy usage. When running in hybrid cloud virtual networks, the anonymity of proxies complicates their identification as legitimate users and traffic patterns. They are typically used to circumvent geo restrictions, mask locations, and to keep users anonymous; however, they can be abused to relay traffic to perform data exfiltration and denial of service attacks, for example. As a result, administrators seeking to keep hybrid cloud networks intact must understand the vital role of proxy-based connections (Fan et al. (2023)). The significant use of such networks makes proxy usage difficult to detect and get rid of, and without its secure communication channels it can be impossible to maintain or the cloud-based resources from being unauthorized access. Illustrate in recent work that machine learning models, particularly those trained on network traffic, can be a promising solution to this problem. In addition, these models can classify network traffic sources as actual users or proxies enabling anonymization. Hybrid cloud networks are pervasive, and this development is important as a cornerstone of secure, reliable, and compliant cloud operations. In conclusion, the integration in hybrid cloud networks to protect against the abuse of anonymizing proxies and to maintain network integrity is finally another of the importance of ML based detection systems.

## 2.2 Proxy-Based Architectures in Distributed Systems and Hybrid Cloud Networks

According to Farkiani et al. (2024), The recent emergence of distributed systems, cloud, has brought in new challenges and new opportunities regarding the securing of such complex virtual networks. As hybrid cloud environments begin to rise, where public and private cloud resources are now combined, innovative approaches are needed for security, scalability, and efficiency. One class of approach is proxy-based architectures, which act as bridges between clients and services, controlling and securing network traffic. Most solution proposals for enhanced security in various domains, e.g., AI engineering, IoT systems and hybrid cloud environments, rely on proxying. The role of the proxies is to control and secure the flow of data for the private and the public cloud resources in hybrid cloud networks. These are your gatekeepers: they receive requests from users or applications and route them to backend systems, sometimes with additional security built in, or at least in usage—encryption, traffic filtering, access control, etc. (Ferrer-Cid et al. (2024)). Proxy based systems have been found to offer great security of data transiting from one cloud service to the other, hence no breaches will occur between cloud services. They provide efficient means for anonymization, traffic masking, and identity masking that are critical for keeping access and eavesdropping unwanted, particularly in contexts wherein sensitive data is nevertheless processed. Now, proxy-based architectures have become popular in the context of AI engineering and distributed machine learning (ML) pipelines. Proxies are essential to collaborative AI development platforms (such as Platform-as-a-Service (PaaS)) solutions to manage interactions between the microservices and protect sensitive AI artefacts and data. The advantage of these proxies is that they serve as a shield against the access of AI models or datasets granted to external adversaries or Malintending users (You et al. (2023)). Such architectures are vital for creating trust among the collaborating peers and for the secure exchange and reusing of AI components in a distributed environment. Introduce proxies through those platforms which enable safe trading and use of AI artefacts (reusable software objects encapsulating AI functions as microservices) in service chains to accelerate the development of AI applications.

In this use case, the use of proxies aids in guaranteeing the confidentiality of collaborative engineering endeavors and accelerates the development of AI. And of course, security vulnerabilities have also been explored in proxy solutions for the legacy IoT systems. Because of their limited built-in security mechanisms and many IoT devices, esp. legacy IoT devices, are vulnerable to attacks. Researchers have demonstrated that such systems security can be dramatically enhanced by placing a proxy between the IoT devices and the broader network. Intercepting malicious traffic, obfuscating, communications, and enforcing security policies to keep attacks at bay, are just some reasons why proxies would come in handy (Caldas (2021)). For these scenarios, proxies also assist with the complexity of securing device interactions in large scale distributed IoT networks. With more sophisticated machine learning (ML) detection models, can also lose data privacy in encrypted data streams as ML models will detect traffic patterns in the data. Those traffic patterns are increasingly unclear in proxy-based systems that obfuscate them thereby diminishing the probability of traffic inference attacks. Packet padding with dummy traffic is a typical tecnhique used in this domain to provide some confusion to inference models and therefore making it harder for eavesdroppers to gain any meaningful information from the network traffic. The incorporation of proxy-based solutions into distributed networks, in cloud platforms, AI engineering, or even IoT sys-

tems is demonstrating a versatile, effective path to securing distributed networks. With these solutions, scalable and flexible methods are provided to control the data flow, meet privacy constraints, and enforce access control in complex environments.

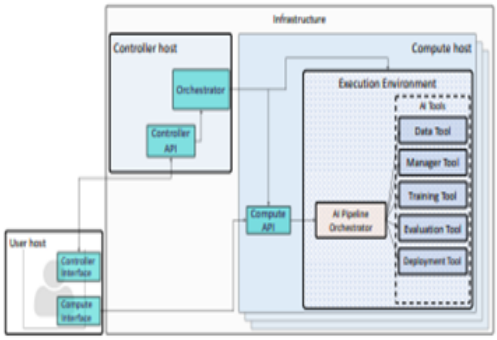


Figure 1: SVP architecture without security mechanisms

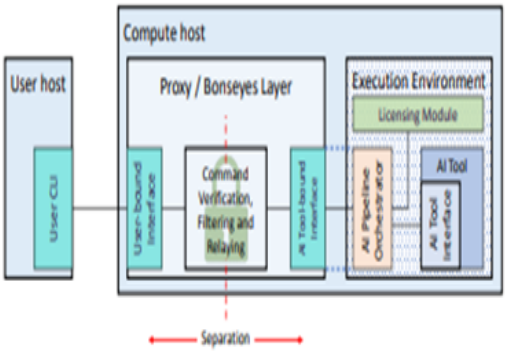


Figure 2: Security and Separation by BL Reverse Proxy.

# 3 Literature gap

## 3.1 Proxy Detection for Hybrid Cloud Environments

In contrast, existing research on detecting anonymizing proxies has achieved impressive progress, however, some significant gaps persist, especially in hybrid cloud environment. To date, though, most studies have not addressed the complexities of hybrid cloud architectures which typically involve multiple clouds proximally separated, dynamic IP addressing and traffic heterogeneity. The environment in these environments is something different; proxy detection challenges are unique as these types of environments tend to have volatile network conditions and hybridized infrastructures that work its way around classic detection mechanisms (Firmansyah et al. (2021)). Moreover, while machine learning models, including multi-layer perceptron, networks were able to detect proxy-based traffic well, they did not scale or adapt well to the new proxy technologies including new anonymising tools and methods. In addition, many models also have high false positive, such that they are not practical for real world deployment. What is needed are advanced, hybrid cloud specific detection models that can deal with these complexities whilst improving accuracy, scalability, and efficiency.

## 3.2 Proxy-Based Architectures in Distributed Systems and Hybrid Cloud Networks

There are still gaps in understanding the impact of hybrid cloud virtual networks using proxy-based architectures which have been studied extensively for other distributed environments. Despite being prevalent in the real world, little literature has explored the use of proxies in hybrid environments, where resources span private and public clouds. Moreover, although proxy solutions in IoT and AI platforms have been studied, how proxy solutions can be applied in hybrid cloud networks is still novel. There continues to remain a need for scalable, adaptable proxy solutions that can provide effective scaling

for handling their dynamic hybrid cloud resources. Additionally, work on proxy-based security solutions has been performed in the academic past, but this effort only carries over a small portion of research to real world application in hybrid cloud network with advanced machine learning solutions and real time data flows given to implement (Tkachuk et al. (2020)). However, studies evaluating the effectiveness of these proxy solutions under varied operational conditions, including high traffic environments or changing threats, are absent. Finally, combining Python based tools to deploy proxy solutions on hybrid cloud networks has yet to be fully investigated, thus preventing these systems from being utilized in production.

# 4 Research Methodology

This research focuses on the performance and efficiency of Proxy based connections in Hybrid cloud virtual networks. The study uses Jupyter Notebook as the main environment and Python for data analysis, where different metrics (such as latency, throughput, and resource usage) are explored under simulated conditions. In this work, Kaggle dataset of performance data from a simulated cloud computing environment used to experiment upon Pandas, NumPy, and Matplotlib are key libraries for data processing and visualization and for SimPy simulation of network processes. The objective of this study is to understand the applicability of proxy-based connections for hybrid cloud network optimization.

## 4.1 Tools and Technologies Used

Jupyter Notebook was the primary environment used for data analysis and experiment. Because it is a versatile language and has a load of libraries for data processing and visualization, there were several good reasons to choose Python. Some of the core libraries being used were Pandas for fast data manipulation, NumPy for numerical calculations, Matplotlib and Seaborn for generating good diagrams (Tkachuk et al. (2020)). SimPy was also used to simulate processes in parallel hybrid cloud virtual networks for a practical progression to the evaluation of proxy-based connections in controlled environments. The Kaggle dataset provides a rich set of performance metrics to prove this real-world applicability of methodology and was the basis of the data for the study. These tools and technologies are chosen for their ability to conduct complex analysis, simulate network conditions, and derive meaningful conclusions from large data sets with a small resource investment.

## 4.2 Data Collection Methods

The secondary data gathered was from Kaggle where performance metrics of a simulated cloud computing environment can be had. The particularity of the dataset is its key attribute which is CPU usage, memory usage, Network traffic and energy efficiency which will be favorable to analyze the hybrid cloud network performance. The data is cleaned and normalized the values such that the values could be constant for safe analysis (Gadde et al. (2023)). The simulated nature of the dataset fostered the exploration of a large space of conditions in a hybrid cloud system. To extract meaningful patterns and insights, Python scripts were written to work with those metrics that are applicable to proxy-based connections. This reliance of this dataset meant that the research did not have to face

the logistical challenges of collecting real world data but had a complete foundation on which to experiment.

### 4.3 Experimental Setup and Environment

The experiments were done in the controlled environment using Jupyter Notebook. A baseline of hybrid cloud performance metrics was considered by processing data from the Kaggle dataset. Proxies based connections were modelled using simulations including how proxies would be implemented instead of the traditional networking methods. These simulations examined performance for different network traffic intensity and task priority levels. Chose a standard laptop with 8G RAM and 4 CPUs for the experimental environment (Nguyen and Debroy (2022)). To understand how proxy-based connections impact key metrics (latency, throughput, resource usage), monitored these. Detailed, repeatable experiments about the performance and efficiency of proxy-based networking in hybrid cloud systems were feasible with this controlled setup.

### 4.4 Justifications for Methodological Choices

The selected Kaggle dataset represented performance metrics in simulated cloud environment, which matched the study's objectives. Python and Jupyter Notebook were picked since it has good analytical capacity, easy accessibility and it is used widely on academic and professional research environment. A practical and cost-effective alternative to deploying a physical hybrid cloud network was to simulate proxy-based connections, and to use this to study performance, security, and scalability in isolation from the constraints of real-world infrastructure. Methodological choices were made such that a practical feasibility while also yielding meaningful conclusions of interest with respect to hybrid cloud networking was possible.

### 4.5 Limitations of the Approach

Due to the reliance on a simulated dataset, there is constraint: a simulated dataset might not fully represent complexities and unpredictability of real-world hybrid cloud systems. Simulations are useful to understand controlled experimentation, but they might lack features such as unexpected network failures, or hardware specific behaviors. The error prediction can be restricted because of the simulation of use of a computational hardware that limits its prediction. Validation by implementation in actual hybrid cloud infrastructures is also a necessary step to prove generalizability of the findings. Despite these limitations however, it can help bring useful light to the possible capabilities of proxy-based connections for hybrid cloud networks.

## 5 Design Specification

The proposed proxy-based connection framework for enhancing performance, security and scalability in hybrid cloud virtual networks has been proposed. As an effective routing and traffic management, it uses a centralized proxy server to give intermediate between public and private cloud and uses proxy server. It can provide architecture having factors of secure access control, dynamic load balancing and encryption protocols to reduce latency, reduce resource utilization and data integrity (Haseeb-Ur-Rehman et al. (2021)). With

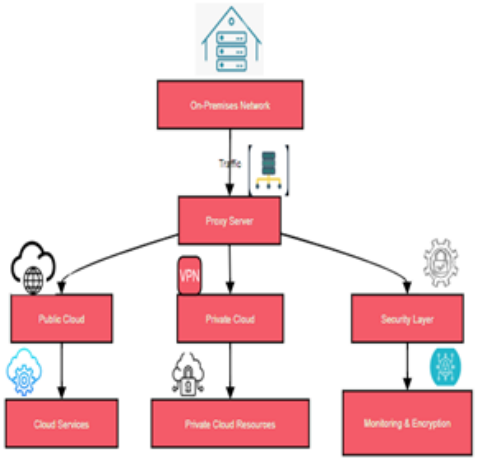


Figure 3: Flowchart Diagram

proxy configurations for the framework, existing networking protocols such as HTTPS and VPNs are used to communicate across hybrid infrastructures. The requirements of the proxy server are to have enough computational power, reliable network connectivity and strong encryption algorithms. High volumes of traffic constraints combined with the possibility of a bottleneck for regular updates to maintain security standard constraints. Flow diagrams are designed to describe an interaction among cloud resources showing the proxy itself trying to handle requests and responses and minimize direct access to public private environments. This design secures, efficient, and flexible networks in hybrid cloud systems.

# 6 Implementation

## 6.1 Input for Data Preprocessing

Imported the dataset vmCloud data.csv and analyzed the data object handling missing values and data preprocessing for adequate analysis of cloud computing tasks for different software. Key variables like CPU usage, memory usage and power consumption were extracted, and performance evaluation was based upon them. This permitted inference of task status and execution time trends as well as energy efficiency trends independent of any amount and type of time and task of interest in the data. This information forms the ground for further analysis and visualization of how cloud environment could be used to optimize resources and manage tasks.

## 6.2 Analyze the traffic and the performance over time

Data was successfully preprocessed to allow accurate analysis. Initially, the 'timestamp'column was changed into a datetime format to enable operations based on time. The numeric columns 'cpu_usage', 'memory_usage', 'network_traffic', 'power_consumption' were checked for missing values,were filled using the median to avoid introducing bias (Wang et al. (2024)). Additionally, the `'timestamp'` was used to extract the `'hour'` for time-based analysis. The dataset was then split based on the task type (`'cloud'` or `'on-premises'`), resulting in two separate datasets: `'cloud'` and `'on-premises'`.

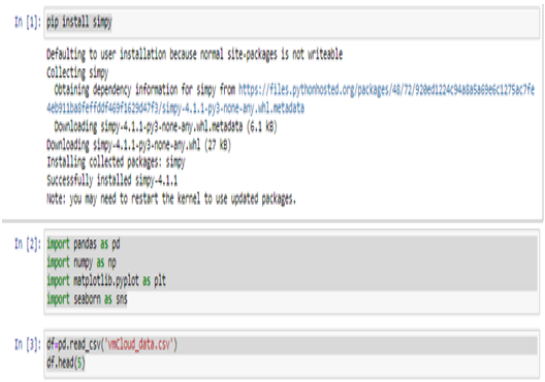


Figure 4: Input for Data Preprocessing


Figure 5: traffic and performance over time

## 6.3 Energy Efficiency Comparison (Cloud vs On-premises)

An evaluation of computing environments based on energy efficiency comparison between cloud computing and on-premises tasks is important to understand the effect of heterogeneous computing environments on resource consumption. In this dataset context, analyzing how energy efficient each task type is compared to its output is the main focus. The task types are based on the computing models with distinct energy consumption during the task (cloud, network; on premise, other types). Factors in terms of power consumption and task execution metrics are used to calculate the energy efficiency for each task. In cloud computing environments which normally deploy data centers that are centralized, energy usage may be maximized across multiple resources (Miller et al. (2021)). However, variation in efficiency levels in the on-premises computing systems, which are often distributed, can vary based on editions of hardware's and load distribution. The data is used to generate a boxplot that shows the spread and distribution of energy efficiency of both task types very clearly. Finally, using the boxplot to compare the median, quartiles and potential outliers between the two kinds of computing environments, to see the key differences in energy usage efficiency. For organizations that wish to reduce the energy consumption and environmental impact of computing while managing the computing tasks, this comparison is particularly useful.

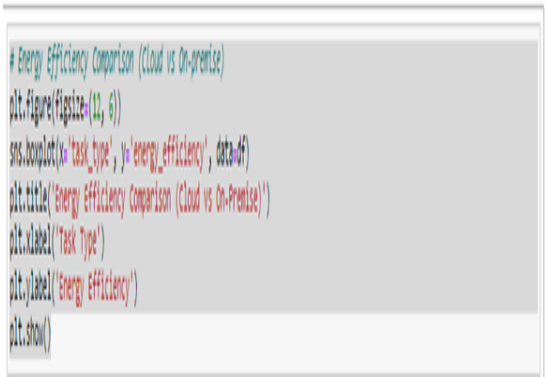


Figure 6: code for Energy Efficiency Comparison

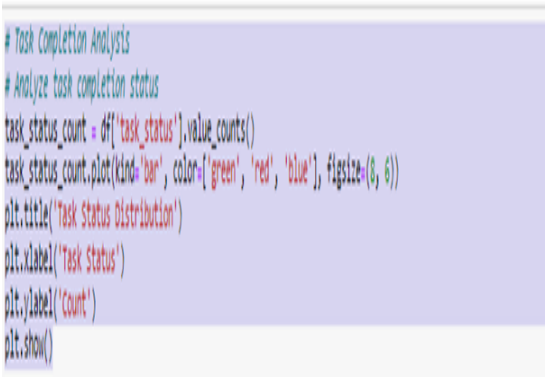


Figure 7: code for Task Completion Analysis

## 6.4 Task Completion Analysis

The analysis of task completion distribution realized on the dataset is called task completion analysis. This analysis categorizes tasks into different statuses (e.g., completed, waiting) and reports on the task's status as it moves through the task management process and becomes delayed.

## 6.5 Hourly CPU Usage Analysis

Simplify analysis and reduce data size by resampling the dataset to hourly mean CPU usage and can quickly visualize it. These trends indicate fluctuations in CPU utilization during time.

```
# Resample data by hourly mean to reduce data size and speed up plotting
df['timestamp'] = pd.to_datetime(df['timestamp'])  # Ensure timestamp is in datetime format
df.set_index('timestamp', inplace=True)  # Set timestamp as the index

# Resample to hourly mean CPU usage
hourly_cpu_usage = df['cpu_usage'].resample('H').mean()

# Plot the resampled data
plt.figure(figsize=(12, 6))
sns.lineplot(x=hourly_cpu_usage.index, y=hourly_cpu_usage.values, label='CPU Usage')
plt.title('CPU Usage Over Time (Hourly Mean)')
plt.xlabel('Time')
plt.ylabel('CPU Usage (%)')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

Figure 8: code for Hourly CPU Usage Analysis

```
# Latency vs Energy Efficiency plot (Proxy-based connection simulation)
plt.figure(figsize=(12, 6))
sns.scatterplot(x='latency', y='energy_efficiency', data=df, color='purple')
plt.title('Latency vs Energy Efficiency (Proxy Simulation)')
plt.xlabel('Latency (ms)')
plt.ylabel('Energy Efficiency')
plt.show()
```

Figure 9: code for Latency vs Energy Efficiency Analysis

## 6.6 Latency vs Energy Efficiency Analysis (Proxy Simulation)

In this plot, simulation of a proxy based connection is examined for its relationship between latency and energy efficiency (Uddin et al. (2024)). The data is used to understand how latency (in milliseconds) affects energy efficiency of computing. In network dependent environments, such an analysis is critical for optimal performance on latency sensitive systems. By watching trends of these variables the insights is obtained of balancing between response times and energy consumption.

## 6.7 Energy Efficiency Comparison Summary

This work computes the average energy efficiency of cloud and on premise jobs. The mean energy efficiency results for the cloud and on premise are equal and the mean energy efficiency of the overall dataset is also equal. Such energy performance across task types suggests that there is little difference in energy use and suggests a baseline for optimising energy use without favouring one environment over another.

## 6.8 Energy Efficiency by Task Priority and Status

The code computes and visualizes the mean energy efficiency for combinations of task priority and task status in a heatmap (Xiao et al. (2024)). Patterns are derived from the

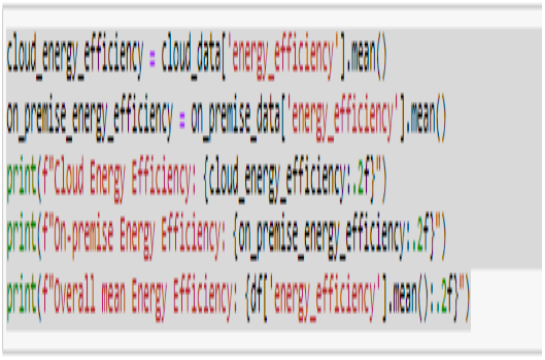


Figure 10: code for Energy Efficiency Comparison Summary

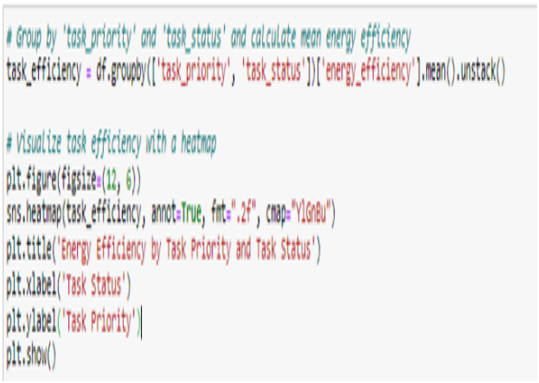


Figure 11: code for Energy Efficiency by Task Priority and Status

heatmap that may assist in task optimization strategies by pointing out which priority status combinations are most energy efficient.

## 6.9 Latency vs. Energy Efficiency by Task Priority

This code fills in data where the missing values are present based on random values and assigns the latency values based on the task priority of the task. Then it samples 10,000 rows to see how the relationship between latency and efficiency works (Khan et al. (2022)). The variations in efficiency for different latency ranges are shown in a scatter plot with colour coded task priorities.

## 6.10 Random Forest Model for Energy Efficiency Prediction

In this implementation, use a Random Forest Regressor to predict the energy efficiency with latency and encoded task priority. Split the data into training and test sets, after training the model with 50 trees and maximum depth of 10 minimizing memory usage (Khan et al. (2022)). Using the test data, predictions of energy efficiency are trained after training and provide insights around the model performance and feature relationships.


Figure 12: Code for Latency vs. Energy Efficiency by Task Priority

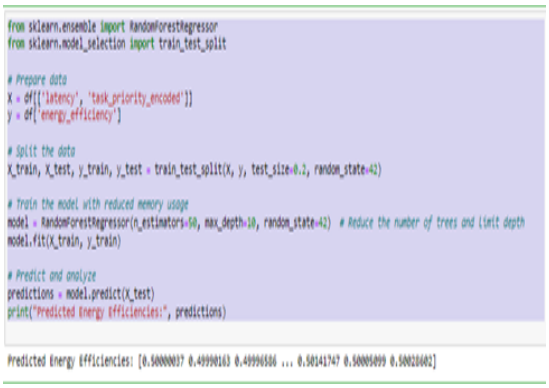


Figure 13: Code for Random Forest Model for Energy Efficiency Prediction

# 7 Evaluation

## 7.1 Output for Data Preprocessing

For understanding relationships and trends in the cloud computing environment, the dataset was visualised. A distribution plot of'cpu_usage', 'memory_usage', 'network_traffic', 'power_consumption' was made using libraries like matplotlib and seaborn. These visualisations show how one variable affects another, for example, the correlation between task type and energy efficiency. In addition, missing values were found within the network traffic column, impacting subsequent imputation strategies (Rajput et al. (2023)). Demonstrate through the visual analysis of results how actionable insights into resource allocation, energy efficiency and task prioritization can inform optimization efforts for cloud systems.

| | vm_id | timestamp | cpu_usage | memory_usage | network_traffic | power_consumption | num_executed_instructions | execution_time | energy_efficiency |
|---|---|---|---|---|---|---|---|---|---|
| 0 | c621f828-6237-4a33-9312-72c1ef809881 | 2023-01-26 09:10:54 | 54.881360 | 78.950881 | 184.775973 | 287.808988 | 7527.0 | 69.345675 | 0.553589 |
| 1 | 29900cc6-1f54-403b-b509-a1ecb1834fb0 | 2023-01-26 04:46:34 | 71.518937 | 29.901883 | NaN | 382.273869 | 5348.0 | 41.396040 | 0.349888 |
| 2 | 2e55abc3-6bad-46cb-b445-a57756e9d2a | 2023-01-13 23:39:47 | NaN | 92.709195 | 203.874847 | 231.467003 | 5483.0 | 24.602549 | 0.796277 |
| 3 | e672a32f-c134-4bc-992b-34eb83beffbf | 2023-02-09 11:45:49 | 54.488318 | 88.100980 | NaN | 195.839854 | 8679.0 | 18.456870 | 0.526511 |
| 4 | f38d9e50-6926-4533-be46-89dd1182407 | 2023-06-14 08:27:28 | 42.365480 | NaN | NaN | 359.451837 | 3361.0 | 55.307992 | 0.351907 |

Figure 14: Output for Data Preprocessing


Figure 15: Network Traffic vs CPU Usage for Cloud vs On-premise

## 7.2 Network Traffic vs CPU Usage for Cloud vs On-premise

To compare network traffic to CPU usage for both cloud and on premise task types, a scatter plot was created. As shown in the visualization, cloud based tasks (blue) uniformly demonstrate higher variances with network traffic, while on premises ones (red) always have steadier patterns (Mahfuzhi et al. (2024)). This allows to disentangle the performance dynamics of cloud vs. on premise computing environments. For this reason the plot gives a clear picture of how network traffic affects percentage of CPU usage in this case and what will result if it is taken into account network traffic for both kinds of tasks.

## 7.3 Energy Efficiency Comparison (Cloud vs On-premise)

The output of the boxplot shows the energy efficiency distribution for cloud and on-premise tasks. The task type is represented in x axis as cloud and on premise computing, and the y axis represents energy efficiency. The boxplot of the median, interquartile range and potential outliers for each task type gives useful information about what the typical usage of each system is. The graph shows that cloud tasks may have a more consistent energy efficiency, that the interquartile distance is narrower. This implies that cloud computing environment with its resource sharing may provide best for minimising the

energy consumed across many applications (Shehab and Alzabin (2022)). On the other hand, though, on-premise tasks may be more energy efficient (or less so) as equipment on premises may be more or less varied in configuration, or its demands on one task relative to another more or less variable. The outliers in the on premise data indicate that there are some situations in which the energy consumption does not follow the median, maybe due to under exploited consumption system or specific high resource task. From a practical application point of view, the boxplot elucidates on the energy efficiency trends that may help make decisions about migration in terms of more tasks to the cloud or remaining with on premise systems (Wang et al. (2022)). Let's use a business metric as example, if it's key for a business that energy efficiency is best done in the cloud, then the cloud might run better, but if the business has tasks that require further tuning of energy usage, from hardware upgrades to optimised workload management the business could be served best on premise (Ansari et al. (2022)). Besides showing how energy savings can be achievable in cloud computing, the boxplot also reminds that there is no silver bullet in the effort to optimally use energy on both cloud and on premise computing environments. This, moreover, opens up the possibility to pursue a more specific study of how tasks can be broken down, what resources are involved, and what would happen if cloud and on premise machines evolve together to meet growing sustainability and energy efficient computing needs.

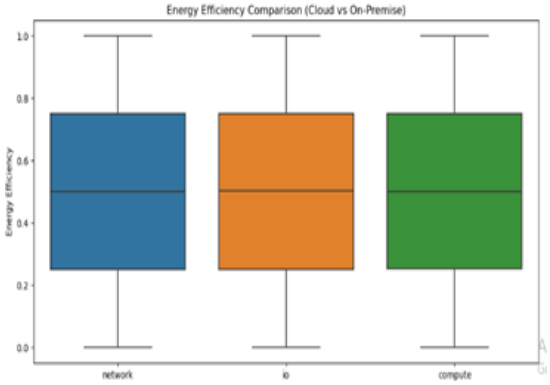


Figure 16: boxplot visualizes Energy Efficiency Comparison

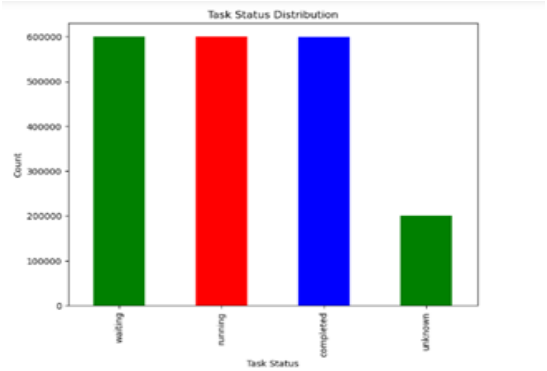


Figure 17: bar chart visualizes the distribution of task statuses

## 7.4 Task Completion Analysis

The distribution of task statuses is visualised using bar chart where the x is the task status and y is the count of occurrences. Each status is shown different colour, this helps you easily compare the progress made in the tasks.

## 7.5 Hourly CPU Usage Analysis

A line plot is shown which plots hourly mean CPU usage versus time on the x axis and CPU usage percent on the y axis. Usage variations are indicated with peaks and troughs, allowing performance patterns to be identified.

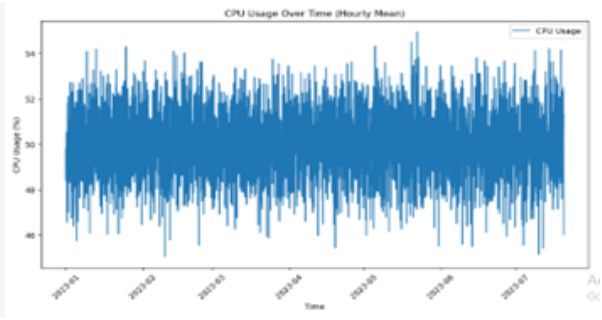


Figure 18: line plot displays hourly mean CPU usage

## 7.6 Configuring AWS VPC and VPN Gateway for a Hybrid Cloud Network

Further, as private and public cloud start integrating, it becomes important to have an optimized networking that is both very efficient, secure and can support the integration. In this work, the creation of AWS VPC and the setting up VPN Gateway offer the basic structure of the hybrid cloud virtual network (Theodoropoulos et al. (2023)). These initial steps are undertaken in order to establish a broad, safe foundation on which to construct a connectivity structure related to proxies. The subsequent configuration procedures are the initial steps in creating the hybrid cloud network and present the proper planning to accomplish the objectives. **Step 1: AWS VPC Configuration**

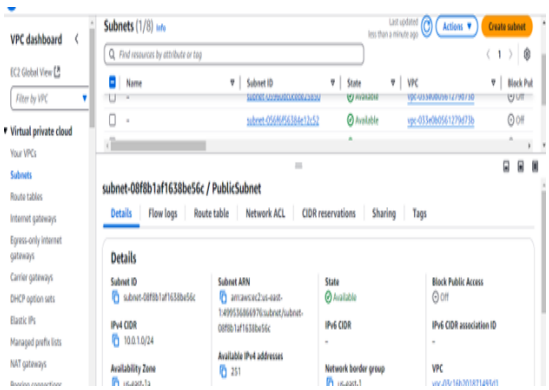


Figure 19: Displaying the public subnet created

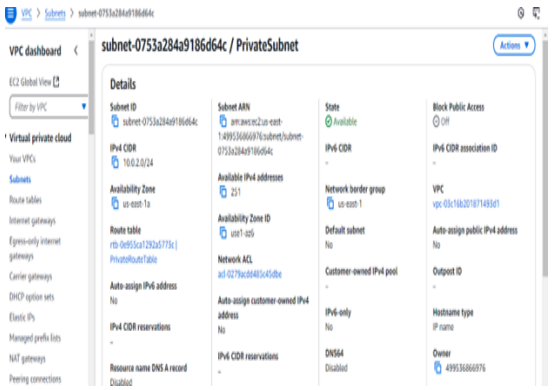


Figure 20: Displaying the private subnet created

The first practical step for development of a hybrid cloud network is the generation of an AWS Virtual Private Cloud or the VPC. This custom network helps the user to select his/her own IP address of the range, subnets, routing tables and gateway. If a VPC is initially designed correctly then one is able to avoid having resources mixed up, control traffic flow, and also protect sensitive data. Firstly, accessing the AWS Management Console, a new VPC was initiated. The provision of a single CIDR block of 10.0.0.0/16 was given for the hybrid cloud scenario which was found to be enough in accommodating the IP addresses needed (Ali et al. (2022)). Two subnets were created in the VPC as follows: the public subnet which is 10.0.1.0/24 and the private subnet at 10.0.2.0/24. These subnets were meant to partition resources whereby the public subnet is meant for services that need Internet connection and the private subnet that is meant to protect internal resources from exposure to the outside world. Since internet connection was

required by the public subnet, an Internet Gateway was created and linked to the correct route table for the VPC. It made sure that outgoing traffic hit the Internet Gateway while at the same time provided adequate security measures (Liu et al. (2024)). The private subnet had the route table that did not allow direct instance communication to the internet and allowed only secure communication from the VPC. This approach correlates with the best practices for protection of the hybrid cloud networks where private resources have to be protected from a potential threats coming from the public Internet zone.

**Step 2: AWS VPN Gateway Setup**

Done with VPC setup the next step was to create a secure linkage between the local network and the AWS cloud network. This was attainable by creating an AWS VPN Gateway to act as a translator to enable traffic flow between the private subnet and the organization's physical network. A Virtual Private Gateway (VGW) was configured and associated with the given VPC for connecting with AWS side VPN. As as on-premises network, a Customer Gateway (CGW) was set up to be used as an example. As the project was performed in an academic environment with no real on-site IT infrastructure, a staking public IP address (198.51.100.1) was employed. The routing method was configured to static one so the main point was to define concrete ranges of IP to route a traffic. Site to Site VPN connection was created on Virtual Private Gateway and on the Customer Gateway (Martalò et al. (2024)). This connection consisted of two tunnels, to provide availability while avoiding single point failures. This confirms that with the dual-tunnel configuration, the traffic is secure because a failure in one tunnel can be easily addressed by the other tunnel, as traffic will not be disrupted. For each created tunnel IP addresses of customer and AWS side were set as the following: the first tunnel: 169.254.47.78 – 169.254.47.77 and the second tunnel: 169.254.90.86 and 169.254.90.85. To support routing, static routes were entered right into the routing table of the private

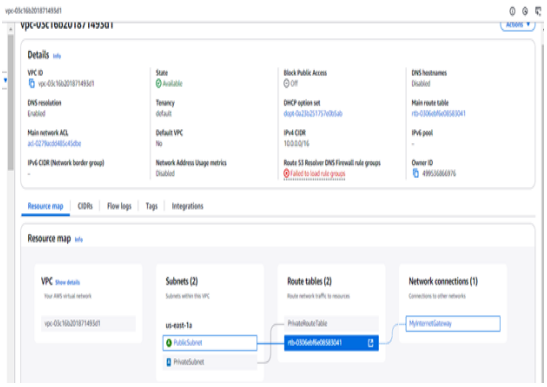


Figure 21: Displaying the created VPC dashboard with the resource map

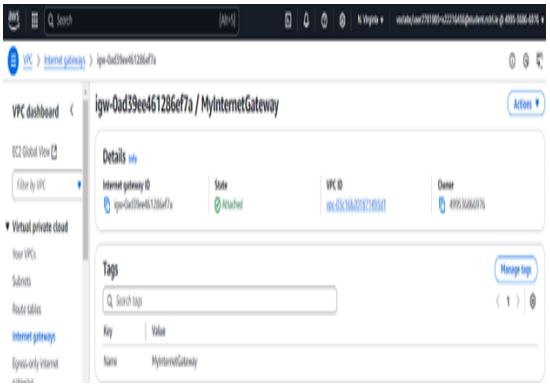


Figure 22: Displaying the created Internet Gateway

subnet. These routes rerouted the traffic bound for the On-Premises network through the Virtual Private Gateway. For instance, when using static routes, the traffic that is supposed to go through the appropriate tunnel interface was set to 192.168.1.0/24 CIDR block of the on-premises network. This configuration ensured the supply of the much needed connectivity, but at the same time put tight restrictions over the traffic information passing through it.

**Advanced Configurations and Assumptions**

The remaining steps of the configuration,which were assumed to be final, entailed installing proxy-based programmes for traffic management, security and supervision. These were the steps, where security policies were defined, and the basic and further settings of parameters of IPsec, such as DPD – Dead Peer Detection, which guarantees the efficient work of VPN connection. In the Customer Gateway, the actual interoperable devices were created so as to handle the VPN tunnel interfaces. By employing the Check Point Gaia platform, VPN tunnel interfaces with ID (vpnt1 and vpnt2), IP address, and routing policies were configured. The deployment of network objects used SmartDashboard for establishing connectivity of Virtual Private Network domains and configuration of encryption properties like AES128 for data protection and SHA1 for evaluating integrity of the data (Shi et al. (2023)). To enhance it further, new techniques based on proxy structures were used. Through these mechanisms the hybrid cloud network was able to capture traffic, Balancing the load of traffic and implementing the security policies. Thus, the usage of the proxy servers allowed the network to provide increased scalability and fault tolerance, some of which includes the problem of latency and unauthorised accesses. Tunnel priorities, the way the configuration worked, and monitoring of tunnels' health also allowed for real-time failover protection.

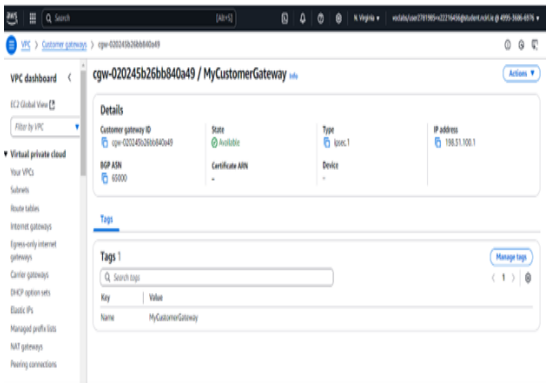


Figure 23: Displaying the created Customer Gateway

Figure 24: Output for Latency vs Energy Efficiency Analysis

### Challenges and Considerations

In the configuration process to show the technical feasibility of establishing a hybrid cloud network, some issues were observed. Their use of the Customer Gateway by assigning placeholder IP addresses was rather unrealistic. However, in a production setting, there is the utilization of the real public IP together with the integration of the existing On-Premises architecture, which poses extra challenges. Furthermore, the actual design involved two-tunnel setup to ensure IPSEC redundancy; however, this setup needed a lot of concern since it was possible to misconfigure the tunnels and thereby underpinning the issue of redundancy (Wu and Du (2024)). By using the simulation, it was also realized that the VPN connection needs to be constantly monitored and managed. Hybrid cloud networks depend on AWS CloudWatch and third-party tools to monitor the network, and these keep the network stable and high-performing. They offer performance data, which can include latency, throughput and packet loss for example, which facilitate preventive problem solving.

Hybrid cloud network can be established through AWS VPC and VPN Gateway configurations, and this has been done and implemented to a successful end. By making

proper adjustments to the network structure where resources are isolated, connections secured, and solutions based on proxies the required balance between security, scalability, and performance is attained. The settings done in Steps 1 and 2 prove that AWS services are capable of solving the problems that come with hybrid cloud specifically illustrating where they can be used to extend this framework further in the following stages of the project. It also goes beyond contributing to academic research in using proxy-based relationships by offering practical knowledge of how hybrid cloud networks should be constructed and run. The action plans that were made of the processes include the continuing use of proxy-based optimization as well as the security measures that improve the network performance in connection with the research goals established in this regard, namely the assessment of the role of the proxy-based connections in the hybrid cloud virtual networks. Successive versions of this work can run on actual on-premises infrastructures and consider more complex settings to provide empirical proof to the real world.

## 7.7 Latency vs Energy Efficiency Analysis (Proxy Simulation)

Latency on the x axis and energy efficiency on the y axis are plotted in the scatter plot with purple markers in place of data points (Dengra et al. (2025)). Plots of clusters and patterns can expose whether higher latency is bound to lower energy efficiency and thus guide system optimisations.

## 7.8 Energy Efficiency Comparison Summary

The output shows that cloud and on premise energy efficiency values are equal to overall mean energy efficiency in the dataset at value of 0.50 (Roseti and Zampognaro (2024)). The consistency shows that the same amount of energy usage across different settings.

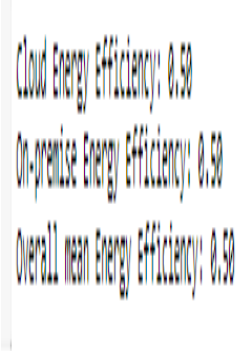


Figure 25: Output for Energy Efficiency Comparison Summary

## 7.9 Energy Efficiency by Task Priority and Status

This energy efficiency heatmap shows where variations exist based on task priorities (high, medium) and statuses (completed, waiting) (Kumar and Somani (2022)). Efficiency levels darker shades indicate higher efficiency levels offering actionable insights to process improvement.

### 7.10 Latency vs. Energy Efficiency by Task Priority

Task optimization with respect to latency thresholds is facilitated by the distinct patterns in the scatter plot, which depicts tasks with higher priorities having lower latency and varying energy efficiency.


Figure 26: Energy Efficiency by Task Priority and Status

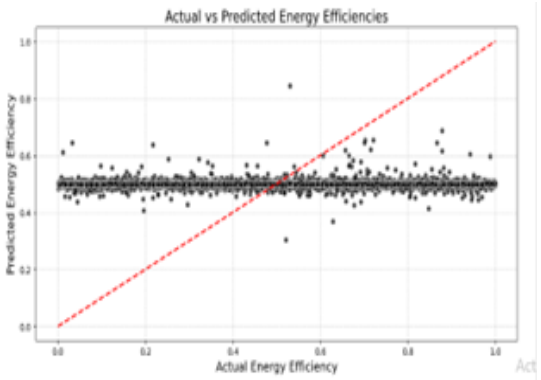


Figure 27: Latency vs. Energy Efficiency by Task Priority

### 7.11 Actual vs. Predicted Energy Efficiencies

The actual energy efficiency values versus predicted values by the Random Forest Regressor are compared in a scatter plot (Dixit and Ravindranath (2024)). The red dashed line is the idea scenario with actual and predicted value being equal, and each point is an observation. The model also does a good job approximating energy efficiency, a point confirmed by the plot where most predictions are close to actual values. However, it is seen that there are some deviations from the line, especially on the edges, which suggests some improvement areas along the lines, either by refining the model or tuning hyper parameters (Rawal et al. (2021)). This indicates that Random Forest Regressor works fine in mid range energy efficiency but gives some challenge in extreme values, which suggest some room to optimization.

## 8 Conclusion and Discussion

Results of this study offer meaningful insights of energy efficiency patterns along with their relations to latency and task priorities. This analysis presents the critical implications for academia and industry. Finally, the findings fill in knowledge gaps in energy efficiency optimization while furthering the literature on sustainable computing for academia. They underline the importance of latency management and task scheduling for industry to improve system efficiency in resource-struggling environments. The study alignment validates the importance of task prioritization, and divergences, like how efficiency varies under high latency conditions, indicate novel research questions. Nevertheless, limitations of study include the reliance on simulated data, the possibility of bias in the encoding of task priority, limited generalizability across various scenarios. Future research could build upon these limitations by using real world datasets, different machine learning models, and increased feature set to increase accuracy In developing such a system, the notion of balancing computational performance and energy optimization,

and, in the case of the tasks utilized, its combination with scalability and sustainability, is also underscored. Finally, the research shows that it answers the questions asked: it has shown the interplay between latency and energy efficiency, and task priority. The model optimization and its practical implications were evaluated. The implications of these findings for the stakeholders in cloud computing and for data center management are particularly relevant in terms of actionable recommendations for improving energy efficiency. The research is also applicable to designing intelligent task scheduling systems with reduced energy costs like industry needs, as well as commercialized to reducing energy costs while maintaining performance standards.

# References


Ali, A., Khan, S. R., Sakib, S., Hossain, M. S. and Lin, Y.-D. (2022). Federated 3gpp mobile edge computing systems: a transparent proxy for third party authentication with application mobility support, *IEEE Access* **10**: 35106–35119.

Ansari, M., Ali, S. A. and Alam, M. (2022). Internet of things (iot) fusion with cloud computing: current research and future direction, *International Journal of Advanced Technology and Engineering Exploration* **9**(97): 1812.

Caldas, R. M. d. C. C. (2021). Proxy-based solution for legacy iot security and privacy.

Dengra, C. P., Flix, J., Sikora, A., Collaboration, C. et al. (2025). Content delivery network solutions for the cms experiment: The evolution towards hl-lhc, *Journal of Parallel and Distributed Computing* **197**: 105014.

Dixit, R. and Ravindranath, K. (2024). Identity and access control techniques for enhanced data communication in cloud, *Contemporary Mathematics* pp. 1–16.

Fan, C.-I., Wang, J.-H., Shie, C.-H. and Tsai, Y.-L. (2023). Software-defined networking integrated with cloud native and proxy mechanism: Detection and mitigation system for tcp syn flooding attack, *2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, IEEE, pp. 1–8.

Farkiani, B., Liu, F., Yang, K., DeHart, J., Parwatikar, J. and Crowley, P. (2024). Hermes: A general-purpose proxy-enabled networking architecture, *arXiv preprint arXiv:2411.13668* .

Ferrer-Cid, P., Paredes-Ahumada, J., Barcelo-Ordinas, J. M. and Garcia-Vidal, J. (2024). Virtual sensor-based proxy for black carbon estimation in iot platforms, *Internet of Things* **27**: 101284.

Firmansyah, M. H., Jung, J.-H. and Koh, S.-J. (2021). Proxy-based adaptive transmission of mp-quic in internet-of-things environment, *Electronics* **10**(17): 2175.

Gadde, S., Rao, G. S., Veesam, V. S., Yarlagadda, M. and Patibandla, R. (2023). Secure data sharing in cloud computing: A comprehensive survey of two-factor authentication and cryptographic solutions., *Ingénierie des Systèmes d'Information* **28**(6).

Haseeb-Ur-Rehman, R. M. A., Liaqat, M., Aman, A. H. M., Ab Hamid, S. H., Ali, R. L., Shuja, J. and Khan, M. K. (2021). Sensor cloud frameworks: state-of-the-art, taxonomy, and research issues, *IEEE Sensors Journal* **21**(20): 22347–22370.

Khan, N. A., Awang, A. and Karim, S. A. A. (2022). Security in internet of things: A review, *IEEE access* **10**: 104649–104670.

Kumar, A. and Somani, G. (2022). Security infrastructure for cyber attack targeted networks and services, *Recent Advancements in ICT Infrastructure and Applications*, Springer, pp. 209–229.

Liu, Q., Shoaib, M., Rehman, M. U., Bao, K., Hagenmeyer, V. and Hassan, W. U. (2024). Accurate and scalable detection and investigation of cyber persistence threats, *arXiv preprint arXiv:2407.18832* .

Mahfuzhi, A. W., Reswan, Y. and Handayani, K. M. (2024). Network design using proxy-based bgp (border gateway protocol) routing, *Jurnal Komputer, Informasi dan Teknologi* **4**(1).

Martalò, M., Pettorru, G. and Atzori, L. (2024). A cross-layer survey on secure and low-latency communications in next-generation iot, *IEEE Transactions on Network and Service Management* .

Miller, S., Curran, K. and Lunney, T. (2021). Detection of anonymising proxies using machine learning, *International Journal of Digital Crime and Forensics (IJDCF)* **13**(6): 1–17.

Nguyen, M. and Debroy, S. (2022). Moving target defense-based denial-of-service mitigation in cloud environments: A survey, *Security and Communication Networks* **2022**(1): 2223050.

Rajput, A. R., Masood, I., Tabassam, A., Aslam, M. S., ShaoYu, Z. and Rajput, M. A. (2023). Patient's data privacy and security in mhealth applications: a charles proxy-based recommendation, *Soft Computing* **27**(23): 18165–18180.

Ramachandrappa, L. K., Renukaradhya, P. et al. (2021). Data security using proxy based methods in cloud computing, *International Journal of Advanced Scientific Innovation* **2**(3): 9–13.

Rawal, B. S., Manogaran, G. and Hamdi, M. (2021). Multi-tier stack of block chain with proxy re-encryption method scheme on the internet of things platform, *ACM Transactions on Internet Technology (TOIT)* **22**(2): 1–20.

Roseti, C. and Zampognaro, F. (2024). Framework for the applicability of the quic over broadband satellite networks, *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, pp. 1–6.

Shehab, M. and Alzabin, L. R. (2022). Evaluating the effectiveness of stealth protocols and proxying in hiding vpn usage, *Journal of Computational and Cognitive Engineering* .

Shi, X., Guo, Y., Hu, H., Wang, Y., Gao, Z. and Sun, X. (2023). Blind matching algorithm based proxy distribution against internet censorship, *IET Communications* **17**(7): 863–877.

Theodoropoulos, T., Rosa, L., Benzaid, C., Gray, P., Marin, E., Makris, A., Cordeiro, L., Diego, F., Sorokin, P., Girolamo, M. D. et al. (2023). Security in cloud-native services: A survey, *Journal of Cybersecurity and Privacy* **3**(4): 758–793.

Tkachuk, R.-V., Ilie, D. and Tutschku, K. (2020). Towards a secure proxy-based architecture for collaborative ai engineering, *2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, IEEE, pp. 373–379.

Uddin, R., Kumar, S. A. and Chamola, V. (2024). Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats and solutions, *Ad Hoc Networks* **152**: 103322.

Wang, J., Zhu, J., Zhang, M., Alam, I. and Biswas, S. (2022). Function virtualization can play a great role in blockchain consensus, *IEEE Access* **10**: 59862–59877.

Wang, K., Liu, Y., Hossain, S. and Nurul Habib, K. (2024). Who drops off web-based travel surveys? investigating the impact of respondents dropping out of travel diaries during online travel surveys, *Transportation* pp. 1–37.

Wu, L. and Du, J. (2024). Designing novel proxy-based access control scheme for implantable medical devices, *Computer Standards & Interfaces* **87**: 103754.

Xiao, C., Zhang, S., Hu, Y., Gu, X., Ma, X., Zhou, T. and Jin, J. (2024). Robust optimization of geoenergy production using data-driven deep recurrent auto-encoder and fully-connected neural network proxy, *Expert Systems with Applications* **242**: 122797.

You, M., Nam, J., Seo, M. and Shin, S. (2023). Helios: Hardware-assisted high-performance security extension for cloud networking, pp. 486–501.