Anomaly Detection of Low Energy Efficiency in VMCloud Data

1. Introduction:

In this report, we looked at identifying virtual machines (VMs) with anomalously low energy efficiency using Isolation Forest for anomaly detection. Subsequently, these anomalies are analyzed to find the underlying causes and future action plan. Python code can be found on Github (link).

2. Methodology:

2.1 Data Description:

The VMCloud dataset [1] includes performance metrics related to virtual machines:

- cpu_usage: Percentage of CPU utilization.
- memory_usage: Percentage of memory utilization.
- network_traffic: Amount of network data transmitted and received.
- power_consumption: Power consumed by the VM.
- execution_time: The time taken to complete specific tasks.
- num_executed_instructions
- task_type: network, io, compute
- task_priority: low, medium, high
- task_status: waiting, running, completed

2.2 Data Preprocessing:

The dataset was cleaned to remove missing values and handle any inconsistencies. Additionally, to ensure that all features contribute equally to the anomaly detection process, the data was scaled using the StandardScaler.

2.3 Anomaly Detection: Isolation Forest

We used the Isolation Forest, an unsupervised machine learning algorithm, to identify anomalous VMs. Isolation Forest works by isolating anomalies, which have shorter path lengths in the isolation trees. Each VM was then assigned an anomaly score (1: normal, -1: abnormal).

3. Analysis of Anomalous VMs:

The focus is on four contradicting cases where energy efficiency is low (below 0.4) despite low power consumption with variations in CPU, memory, and network usage:

- Case 1: **Low Power, Low CPU, Low Memory, High Network (1 VM)**: [6]
  - Possible Causes: Inefficient network protocols, excessive small packet transmissions, unnecessary network broadcasts.
  - Action: Network protocol analysis, packet capture, application logging analysis.
- Case 2: **Low Power, Low CPU, Low Network, High Memory (3 VMs)**: [4], [5]
  - Idle inefficiency: high memory usage without active processing or data transfer.

  - Possible Causes: Memory leaks, inefficient memory allocation, large unused caches, virtualization overhead.
  - Action: Memory profiling tools, application code reviews, virtualization memory checks.
- Case 3: **Low Power, Low Network, Low Memory, High CPU (4 VMs)**: [7]
  - High CPU usage but minimal productive output.
  - Possible Causes: Inefficient algorithms, CPU-bound tasks with minimal output, software bugs causing loops.
  - Action: Application profiling, code reviews, debugging.
- Case 4: **Low Power, Low CPU, Low Network, Low Memory (4 VMs)**: [2]
  - Passive inefficiency, consuming power with little activity.
  - Possible Causes: Slow I/O operations, software bugs causing delays, hardware issues.
  - Action: I/O monitoring, application debugging, hardware checks.

This project helps diagnose energy inefficiency in VMCloud systems by classifying anomalies into distinct inefficiency patterns.

4. Further research:

- Develop a real-time anomaly detection and alerting system to enable proactive intervention and prevent energy waste.
- Explore other anomaly detection algorithms and compare their performance, such as One-Class SVM or Autoencoders, and compare their performance to identify the most effective approach (include feature engineer if needed).
- Incorporate additional metrics, such as hardware-level power consumption, to improve accuracy.
- Explore other energy usage patterns (include task categories)

5. Reference:

[1] Entony. (2023). Cloud Computing Performance Metrics [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/6165137
[2] Energy Efficiency versus Power Consumption. (2020, April 7). Www.ibm.com. https://www.ibm.com/support/pages/energy-efficiency-versus-power-consumption
[3] What is Energy Efficiency & Why is it Important? | NVIDIA. (n.d.). Www.nvidia.com. https://www.nvidia.com/en-us/glossary/energy-efficiency/
[4] Stocker, M. (2024, June 8). Software Efficiency and Energy Consumption - Growing Green Software - Medium. Medium; Growing Green Software. https://medium.com/growing-green-software/software-efficiency-and-energy-consumption-916b390593ec
[5] Zhang, K., Ou, D., Jiang, C., Qiu, Y., & Yan, L. (2021). Power and Performance Evaluation of Memory-Intensive Applications. Energies, 14(14), 4089. https://doi.org/10.3390/en14144089

[6] International Energy Agency. (2023, July 11). Data Centres and Data Transmission Networks. IEA.
https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks
[7] Tradeoff Between Server Utilization and Energy Efficiency. (2022, February 16). Green Software Foundation.
https://greensoftware.foundation/articles/sustainable-systems-mastering-the-tradeoff-between-high-server-utilization-and-ha