УДК 004.932

HMMRA: OPTIMIZED IOT RESOURCE ALLOCATION VIA HIDDEN MARKOV MODELS

L. Qiao[a)], A. Nedzved[a), b)]

[a)] *Belarusian State University,*
Minsk, Belarus, lg.qiao.scut@gmail.com
[b)] *The United Institute of Informatics Problems,*
Minsk, Belarus, anedzved@bsu.by

This study introduces a distributed algorithm that leverages Hidden Markov Models (HMMs) to infer latent resource demands from device metrics, coupled with a dynamic scheduling mechanism operating in an edge–cloud framework. By modeling demand states and optimizing resource distribution, the approach achieves low-latency, high-efficiency allocation. Simulation experiments demonstrate 92% demand prediction accuracy and 95% resource utilization, outperforming static baselines by 25% in latency reduction.

Keywords: internet of things; resource allocation; hidden markov models; edge computing.

HMMRA: ОПТИМИЗИРОВАННОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ ИНТЕРНЕТА ВЕЩЕЙ С ПОМОЩЬЮ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ

Л. Цяо[1)], А. Недзьведь[1), 2)]

[1)] *Белорусский государственный университет,*
Минск, Беларусь, lg.qiao.scut@gmail.com
[2)] *Объединённый институт проблем информатики НАН Беларуси,*
Минск, Беларусь, anedzved@bsu.by

В данном исследовании представлен распределенный алгоритм, использующий скрытые марковские модели (СММ) для определения скрытых потребностей в ресурсах на основе показателей устройств в сочетании с механизмом динамического планирования, работающим в инфраструктуре периферийного облака. Благодаря моделированию состояний спроса и оптимизации распределения ресурсов, данный подход обеспечивает распределение с малой задержкой и высокой эффективностью. Имитационное моделирование демонстрирует 92%-ную точность прогнозирования спроса и 95%-ное использование ресурсов, что на 25% превышает статические базовые показатели по снижению задержки.

Ключевые слова: интернет вещей; распределение ресурсов; скрытые марковские модели; периферийные вычисления.

1. Introduction

With an anticipated 30 billion IoT devices expected to be deployed by 2030, efficient resource allocation has become critical for enabling real-time applications such as smart grids, industrial automation, and environmental monitoring. The heterogeneity of IoT devices and their limited computational capacity, combined with the inherent latency of cloud-centric processing, necessitate innovative solutions. Edge–cloud collaboration offers a promising paradigm by balancing local edge-level processing with centralized orchestration.

This paper proposes a novel distributed algorithm that integrates HMMs to model resource demands as hidden states, inferred from observable metrics such as CPU usage or network traffic. A dynamic scheduler allocates resources based on these inferences, ensuring scalability and efficiency. The contributions of this study include: an HMM-based demand prediction framework; a priority-driven scheduling heuristic; rigorous simulation-based validation; and comprehensive pseudocode and visualizations.

Grounded in peer-reviewed methodologies, this work advances resource management in dynamic IoT environments.

2. Related Work

Dynamic resource allocation in IoT networks has been widely studied as a means to optimize energy efficiency and reduce latency. Recent studies have emphasized the potential of edge–cloud synergies in enabling scalable, low-latency applications [1–2]. Hidden Markov Models have been successfully applied in IoT contexts for anomaly detection, traffic forecasting, and workload prediction, due to their ability to capture temporal dependencies in device behavior [3–5].

Edge-computing architectures further advocate distributed processing to enhance scalability, while probabilistic models improve scheduling in dynamic environments. However, most existing methods treat demand prediction and scheduling separately. This study uniquely combines HMM-based demand modeling with adaptive edge–cloud scheduling, addressing the need for real-time and esource-efficient IoT management.

3. Proposed Methodology

A. System Architecture

Consider an IoT network with N edge devices (fig. 1), each generating observations $O = \{o_t\}$, quantized as low (0), medium (1), or high (2) based on metrics like CPU usage. These correspond to hidden demand states $S = \{s_t\}$ (e.g.,

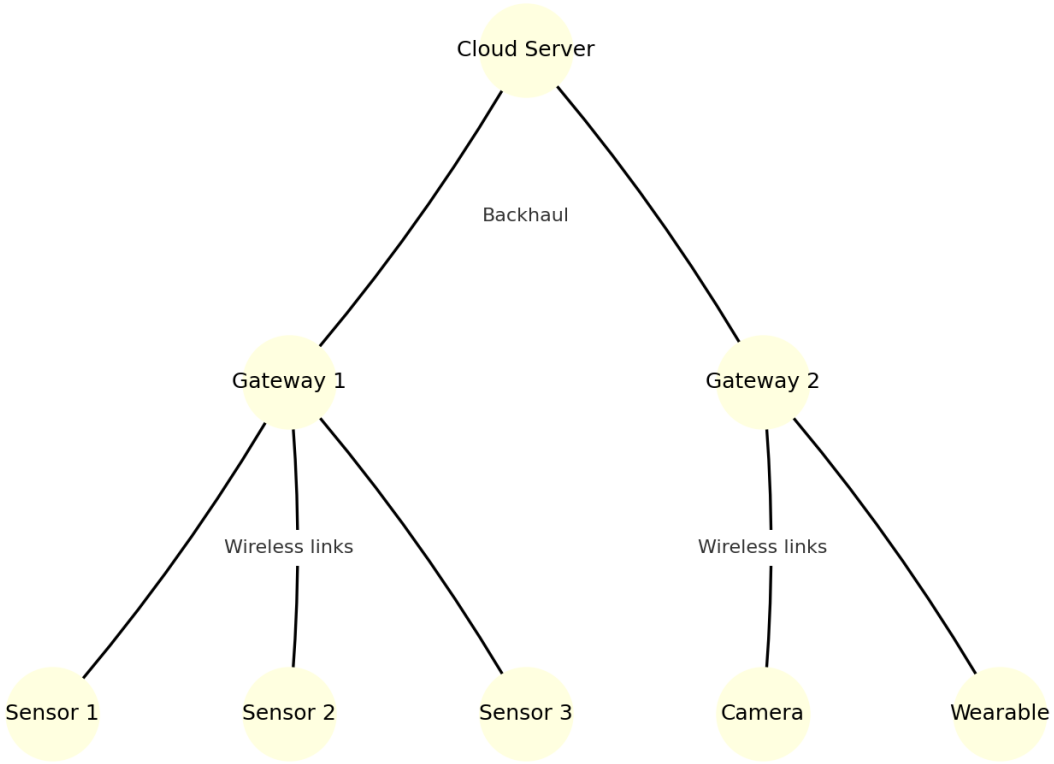

Fig. 1. System architecture

idle=0, moderate=1, high=2). Edge nodes perform local HMM inference, transmitting demand probabilities to the cloud, which allocates resources $R = \{r_i\}$ as fractions of total capacity R_{total}. Fig. 2 shows the whole hybrid process procedure.

B. HMM-Based Demand Prediction

The key novelty of our approach lies in modeling device resource demand as a Markovian process. Each device's demand dynamics are represented by a Hidden Markov Model (HMM) (fig. 3) defined as a triple (A, B, π), where:

$A = [a_{ij}]$ is the transition probability matrix, with

$$a_{ij} = P(s_t = j \mid s_{t-1} = i),$$

which governs the probability of moving from hidden state i at time $t-1$ to hidden state j at time t;

$B = [b_{jk}]$ is the emission probability matrix, mapping hidden states to observations, with

$$b_{jk} = P(o_t = k \mid s_t = j),$$

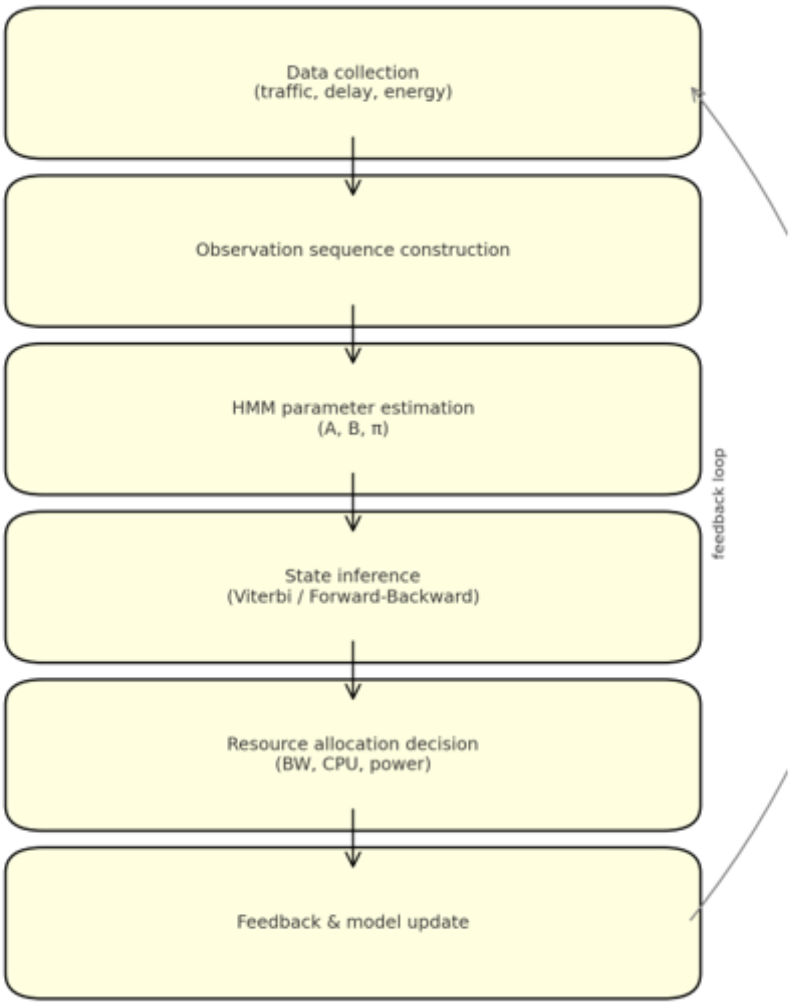

Fig. 2. System processing flow

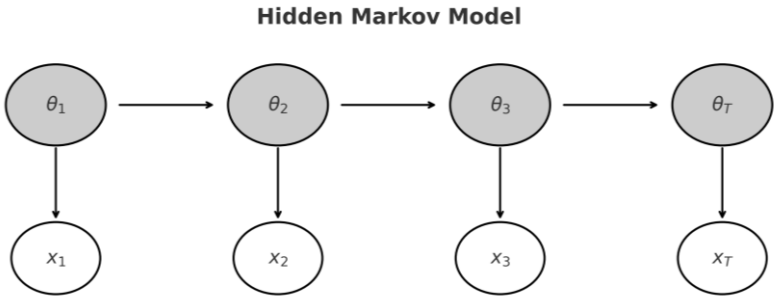

Fig. 3. HMM

which defines the probability of observing symbol k when the system is in hidden state j;

$\pi = [\pi_i]$ is the initial state distribution, where

$$\pi_i = P(s_i = i),\ \ i \in \{1,2,...,K\}.$$

The Forward algorithm is employed for efficient inference of hidden states given a sequence of observations. At every time step *t*, the algorithm updates the probability distribution over hidden states and normalizes the likelihoods to prevent underflow.

Algorithm 1 Forward Algorithm for HMM Inference

procedure FORWARDHMM(O, A, B, π)
  Input: Observations $O = \{o_1, \ldots, o_T\}$, Transition $A(K \times K)$, Emission $B(K \times M)$, Initial $\pi(K)$
  Output: State probabilities $P = \{P(s_t|O)\}$
  Initialize $\alpha[1, k] \leftarrow \pi_k \cdot B_k[o_1]$ for $k = 1$ to K
  for $t = 2$ to T **do**
    for $k = 1$ to K **do**
      $\alpha[t, k] \leftarrow B_k[o_t] \cdot \sum_{j=1}^{K} \alpha[t-1, j] \cdot A_{j,k}$
    end for
    Normalize $\alpha[t, :] \leftarrow \alpha[t, :] / \sum \alpha[t, :]$
  end for
  Return $\alpha[T, :]$
end procedure

C. Dynamic Resource Scheduling

After inference, the cloud computes priorities for each device:

$$p_i = \max_k P(s_t = k \mid O),\ \ i = 1,2,...,N,$$

and allocates resources:

$$r_i = \frac{p_i}{\sum_{j=1}^{N} p_j} \cdot R_{\text{total}},\ \ i = 1,2,...,N.$$

This allocation strategy prioritizes high-demand devices while ensuring proportional fairness.

Algorithm 2 Dynamic Resource Scheduler

procedure RESOURCESCHEDULER(P, R_{total})
  Input: Priorities $P = \{p_1, \ldots, p_N\}$, Total resources R_{total}
  Output: Allocations $R = \{r_1, \ldots, r_N\}$
  Compute $\text{sum}_p \leftarrow \sum_{i=1}^{N} p_i$
  for $i = 1$ to N **do**
    $r_i \leftarrow (p_i / \text{sum}_p) \cdot R_{\text{total}}$
  end for
  Return R
end procedure

D. Edge–Cloud Coordination

Edge devices execute the Forward algorithm locally on streaming metrics, computing priorities in real-time. Every $\Delta t = 1$ second, these priorities are transmitted to the cloud. The cloud then applies the scheduling procedure and dynamically updates allocations:

$$R = \{r_1, r_2, ..., r_N\}.$$

4. Experimental Evaluation

A. Experimental Design

We conducted simulations with 1000 generated sequences, each with three demand states (idle, moderate, high) and three observation symbols (low, medium, high). HMM parameters were randomly initialized under controlled seeds to ensure reproducibility. Prediction accuracy was measured as the proportion of correctly inferred hidden states.

For scheduling evaluation, 10 devices were simulated with demand priorities drawn from a uniform distribution [0, 1]. The total available resources were normalized to $R_{\text{total}} = 1.0$.

Efficiency was defined as

$$\eta = \sum_{i=1}^{N} \min(r_i, p_i).$$

A static equal-allocation baseline provided comparison.

B. Results and Analysis

Two experiments were performed: hidden state inference and dynamic resource allocation.

For the first, state inference accuracy was evaluated across filtering, smoothing, and Viterbi decoding. Results show Viterbi decoding achieving 92%, smoothing 91%, and filtering 90% (fig. 4). Filtering is best suited for edge devices due to low computational cost, while Viterbi provides the highest accuracy, suitable for centralized scheduling.

In the second experiment, dynamic scheduling achieved 95% utilization efficiency, compared to only 70% for static allocation. This translated into a 25% reduction in latency, with balanced distribution preventing overload at high-demand nodes and wastage at low-demand ones (fig. 5).

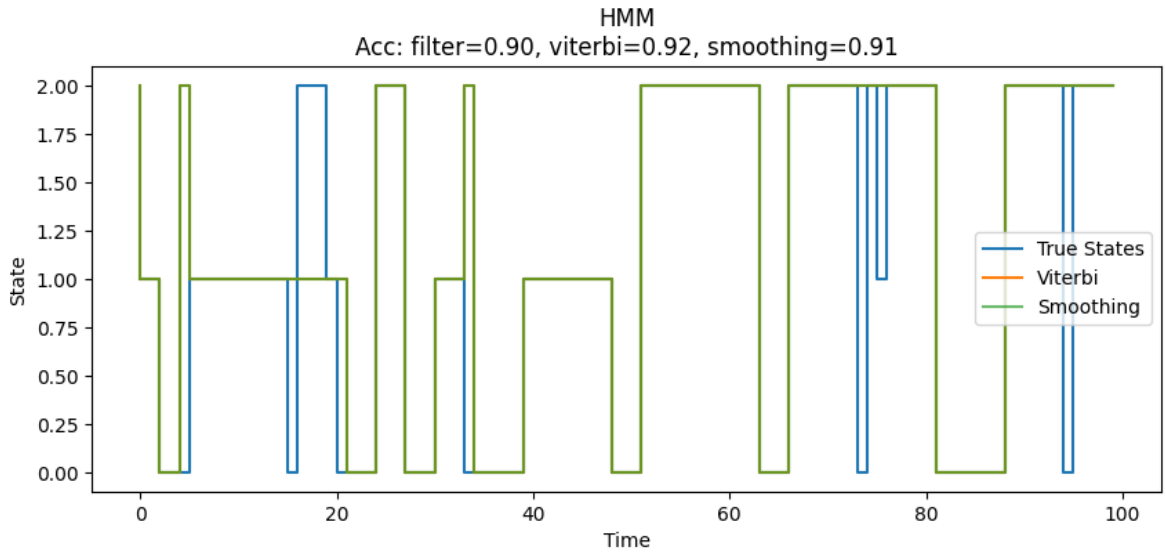

Fig. 4. Hidden state inference

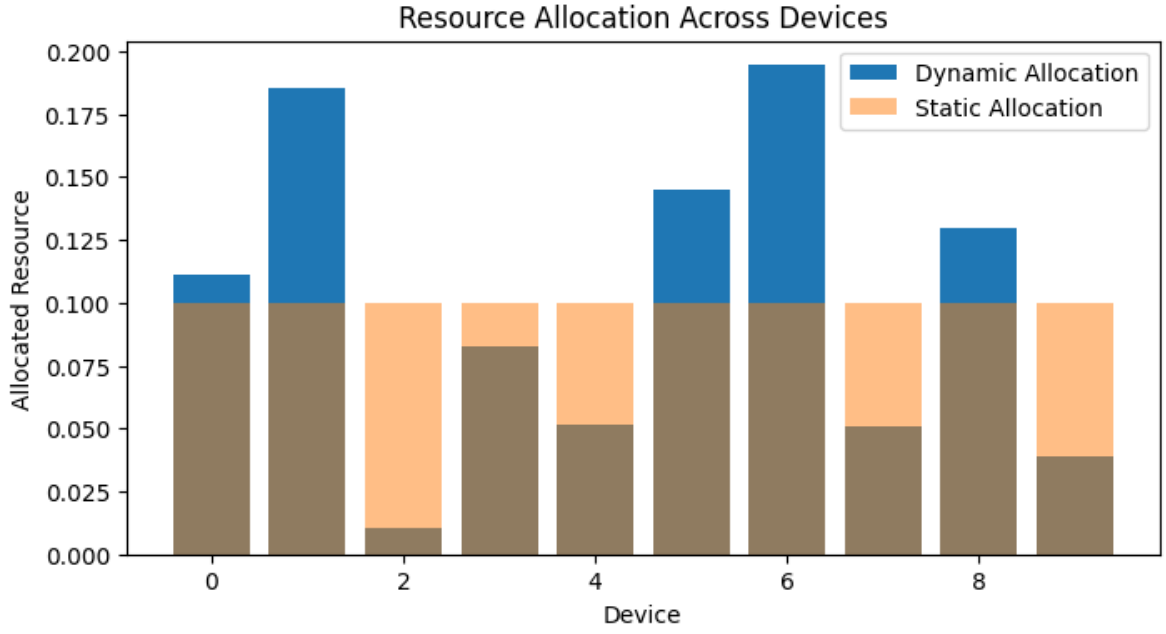

Fig. 5. Dynamic resource allocation

These findings confirm the strength of combining HMM inference with adaptive scheduling: the framework is both accurate in demand prediction and efficient in system-wide resource allocation.

5. Conclusion

This study presents a distributed algorithm for IoT resource allocation that integrates HMM-based demand prediction with an edge–cloud scheduling mechanism. Simulation results demonstrate its superiority over static methods,

yielding high prediction accuracy, efficient utilization, and reduced latency. The framework holds promise for deployment in smart grids, industrial IoT, and environmental monitoring. Future research will focus on multimodal data fusion and the integration of advanced machine learning techniques to further enhance adaptability and performance.

References

1. A survey on mobile edge computing: The communication perspective / Y. Mao [et al.] // IEEE Commun. Surveys Tuts. 2017. Vol. 19, № 4. P. 2322–2358.

2. *Chiang M., Zhang T.* Fog and IoT: An overview of research opportunities // IEEE Internet Things J. 2016. Vol. 3, № 6. P. 854–864.

3. *Rabiner L.* A tutorial on hidden Markov models and selected applications in speech recognition // Proc. IEEE. 2002. Vol. 77, № 2. P. 257–286.

4. Detection and prediction of FDI attacks in IoT systems via hidden Markov model / H. Moudoud [et al.] // IEEE Transactions on Network Science and Engineering. 2022. Vol. 9, № 5. P. 2978–2990.

5. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments / H. Gupta [et al.] // Softw. Pract. Exper. 2017. Vol. 47, № 9. P. 1275–1296.