17

Neuromorphic IoT Architecture for Efficient Water Management

Mugdim Bublin[1], Heimo Hirner[1], Antoine-Martin Lanners[1], and Radu Grosu[2]

[1]University of Applied Science FH Campus Wien, Austria
[2]Technische Universität Wien, Austria

Abstract

The exponential growth of IoT networks necessitates a paradigm shift towards architectures that offer high flexibility and learning capabilities while maintaining low energy consumption, minimal communication overhead, and low latency. Traditional IoT systems, particularly when integrated with machine learning approaches, often suffer from high communication overhead and significant energy consumption.

This work addresses these challenges by proposing a neuromorphic architecture inspired by biological systems. To illustrate the practical application of our proposed architecture, we present a case study focusing on water management in the Carinthian community of Neuhaus. Preliminary results regarding water consumption prediction and anomaly detection in this community are presented. We also introduce a novel neuromorphic IoT architecture that integrates biological principles into the design of IoT systems. This architecture is specifically tailored for edge computing scenarios, where low power and high efficiency are crucial. Our approach leverages the inherent advantages of neuromorphic computing, such as asynchronous processing and event-driven communication, to create an IoT framework that is both energy-efficient and responsive. Moreover, we show that not only is the architecture neuromorphically inspired, but it can also be partially realized, especially at the edge, by neuromorphic hardware. This case study demonstrates how the neuromorphic IoT architecture can be deployed in a real-world scenario, highlighting its benefits in terms of

energy savings, reduced communication overhead, and improved system responsiveness.

Keywords: IoT, edge computing, neuromorphic computing, deep learning, machine learning.

17.1 Introduction and Background

The exponential growth of the Internet of Things (IoT) networks necessitates a paradigm shift in how these systems are designed and deployed. Traditional IoT systems, especially those integrated with machine learning techniques, often face significant challenges, including high communication overhead, increased energy consumption, and latency issues [1]. As IoT devices become more prevalent, particularly in edge computing scenarios, the need for architectures that balance flexibility, learning capability, and energy efficiency becomes critical. These issues are exacerbated when machine learning models are employed, as they typically require substantial computational resources. The challenge lies in creating an IoT architecture that can efficiently process data at the edge, with minimal energy consumption and latency, while still providing the flexibility and learning capabilities necessary for complex tasks such as anomaly detection and prediction.

This paper addresses these challenges by proposing a neuromorphic IoT architecture inspired by biological systems, specifically tailored for scenarios where low power consumption and high efficiency are essential. Neuromorphic computing is an innovative approach in computer engineering that designs computational systems inspired by the architecture and functionality of the human brain and nervous system [2–4]. This brain-inspired computing method offers significant advantages, including [5], [6]:

- **Energy Efficiency:** Traditional deep learning models may require up to 20 MW of power, while the human brain operates on just about 20 W, highlighting the potential for substantial energy savings.
- **Latency:** Neuromorphic systems excel in parallel processing, allowing for faster computations and reduced latency.
- **Safety & Security:** These systems enhance reliability by using redundant and analog components, mimicking the robustness of biological neural networks.
- **Reduced Costs and Waste:** By leveraging materials that mimic biological processes, neuromorphic computing can lower production costs and minimize environmental impact.

These benefits point towards a new computing paradigm that could revolutionize how we approach computational tasks. The primary objective of this study is to propose a neuromorphic IoT architecture that integrates principles from biological systems into the design of IoT systems. This architecture is intended to offer significant improvements in energy efficiency, communication overhead, and system responsiveness. Additionally, we aim to validate the effectiveness of this architecture through a case study on water management in the small city Neuhaus, Austria, demonstrating its practical application and potential benefits.

17.2 Neuromorphic IoT Architecture

17.2.1 Design principles

The proposed neuromorphic IoT architecture is inspired by the efficiency of biological systems, particularly human nervous system, which have evolved to process information with minimal energy consumption and high responsiveness [6]. The key features of this architecture include: distributed control and learning, prediction instead of commands and reservoir computing.

17.2.2 Hierarchical distributed control and learning

Control and learning in an IoT network are distributed across various nodes, allowing the system to leverage locally available information for decision-making where it is most effective. This architecture is tailored for edge computing environments, prioritizing low power consumption and high efficiency. By processing data at the edge, the system can make real-time decisions without the need for constant communication with centralized servers.

In practice, different machine learning models are deployed across various layers of the IoT architecture: the device, or edge layer, fog layer, and cloud layer. At the edge layer, rapid decisions are made, such as automatically shutting off a water pipe in case of damage. The fog layer allows for more sophisticated decisions by integrating data from nearby devices to improve local water management. At the cloud layer, data from IoT devices and the internet is aggregated to manage overall water resources and address disaster scenarios.

We propose that higher architectural levels use predictions instead of commands, similar to the human visual [7] and motor systems [8]. In fog and cloud, models of water predictions on regional and global scales are built.

The model predictions are sent to lower levels, while error corrections are sent from lower to upper layers. Unlike federated learning [9], which relies on a single global model for all devices, this approach utilizes local machine learning models to make decisions based on locally available information. This strategy reduces communication overhead, latency, and energy consumption.

Neuromorphic Reservoir Computing is particularly well-suited for edge implementation due to its low computational overhead, which results in reduced energy consumption, and its ability to harness underlying physical processes for computation. Neuromorphic reservoir computing is a computational framework inspired by the brain's neural networks, particularly suited for processing time-series data and performing complex pattern recognition tasks [10–12]. This approach is based on the concept of a "reservoir," which is a recurrent neural network with fixed, randomly initialized connections. The key idea is that the reservoir can transform temporal input data into a higher-dimensional space, where the information is easier to analyse and predict. The advantage of this method lies in its simplicity: only the output layer is trained, while the reservoir itself remains unchanged. This reduces the computational burden associated with training deep neural networks.

In Figure 17.1 the analogy between human nervous system and the proposed IoT architecture is depicted.

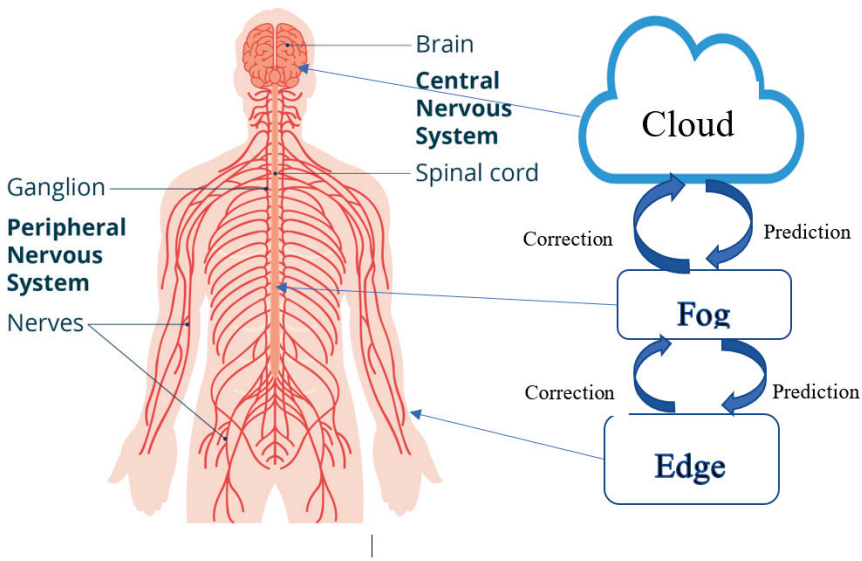

Figure 17.1 Analogies between human nervous system and the proposed IoT architecture.

Figure 17.2 depicts the black-box view of the overall system. The IoT network receives inputs from the environment (such as sensor signals like water consumption measurements, weather data, and other internet-sourced data) and executes commands on the environment (e.g., opening/closing pipes, issuing alarms, etc.).

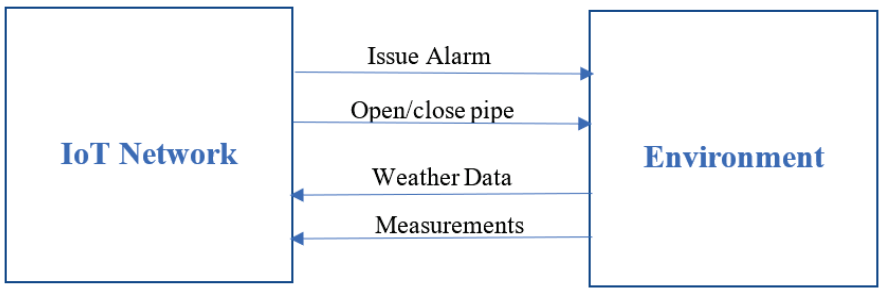

Figure 17.2 Black-Box view of the IoT network and the environment.

Using hierarchical distributed control and learning with prediction/correction feedback loops, we can minimize communication overhead and energy consumption. Information is primarily processed where it is available; only predictions and corrections are communicated, not commands. A mathematical framework for this approach is provided by Friston's active inference and free energy principle [13], [14].

17.3 Free Energy Principle

Friston's free energy principle [13], [14] is a theoretical framework from neuroscience, which posits that the brain minimizes a quantity called free energy to maintain a stable internal state and make sense of the world. This principle is derived from thermodynamics and statistical mechanics, and it explains how biological systems (like the brain) resist disorder (entropy) by maintaining an internal model of the environment. In our case, each level of the IoT hierarchy has its own world model that is updated based on inputs from the next lower level. Active inference has also been applied to IoT in [15].

The free energy principle can also be broken down into terms involving surprise and approximation errors.

$$F = DKL(q(s) \parallel p(s \mid o)) - log\, p(o) \tag{17.1}$$

Where:

$DKL(q(s)||p(s|o))$ is the Kullback-Leibler (KL) divergence between the recognition density $q(s)$ and the posterior distribution $p(s|o)$, which measures how different the brain's internal model is from the real posterior probability of hidden states s given observations o. The hidden state, s, is a vector representing the state of the environment in each IoT layer. The observations o are vectors of measurements (over time) from the environment and signals from the lower layers.

$log\ p(o)$ is the log-evidence or surprise associated with the sensory input o. The brain aims to minimize surprise (unexpected sensory states).

To reduce free energy, the brain does two things:

Perception: Adjusting its internal model (recognition density $q(s)$) to match sensory input better. This helps reduce the Kullback-Leibler divergence (KLD), which improves the accuracy of its beliefs about hidden states.

Action: Taking actions to influence the environment in a way that makes sensory input more predictable, thus reducing surprise $log\ p(o)$.

In our setup, each hierarchy level of an IoT network creates its own internal model of the environment, updates the model according to sensor inputs from lower layers, and performs actions on the lower layers to reduce discrepancies between the model and the environment.

The free energy principle enables **more autonomy** for each layer, as it places fewer constraints on the internal model of each layer compared to, for example, reinforcement learning (RL), which assumes that agents maximize expected rewards. Furthermore, it also requires **less communication overhead**, since information is signaled to other layers only when the differences between predictions and actual measurements exceed a certain threshold.

17.4 Asynchronous Processing and Event-driven Communication

Unlike traditional computing systems, which rely on a global clock, the proposed architecture processes information asynchronously. This allows for more efficient use of resources, as computation only occurs when necessary.

In biological systems, communication between neurons occurs only when a certain threshold is reached, triggering a spike of activity. This principle is applied in the proposed architecture to reduce unnecessary communication, thereby lowering energy consumption and latency.

17.5 The Role of Thresholds in Hierarchical IoT Model

In hierarchical models, each layer generates predictions based on the lower layer's output, with higher layers handling more abstract representations. Thresholds, in this context, can refer to parameters that decide:

- **Uncertainty bounds:** Determining when prediction errors (discrepancies between predicted and observed states) should trigger adjustments in the model or action.
- **Signal passing thresholds:** Deciding when sufficient confidence is achieved in a layer to propagate the information upward or downward.

In active inference framework agents (IoT layers) continuously minimize their **variational free energy** (a measure of surprise or uncertainty about sensory inputs) by updating beliefs or performing actions. Key to this process is minimizing **prediction errors** through updating the generative model (beliefs about the world) or by acting to bring sensory inputs in line with predictions.

- **Perception:** Update internal states to minimize prediction error.
- **Action:** Perform actions to reduce the discrepancy between expected and actual sensory inputs.

17.5.1 Setting adaptive thresholds using prediction errors

In active inference, **prediction error** (the difference between the predicted input and actual sensory input) drives updates to beliefs or actions. To adaptively set thresholds at different hierarchical layers, you can allow thresholds to be modulated by the **magnitude of prediction error** and the **precision (inverse uncertainty)** associated with each layer's predictions.

1. **Prediction Error Calculation:** At each layer L_i, prediction error is computed as:
$$\epsilon_i = input_i - prediction_i \tag{17.2}$$
where ϵ_i is the prediction error at layer *i*, and $input_i$ is the input to the layer (could be sensory data for the bottom layer or output from the previous layer).
2. **Precision-Weighted Prediction Error:** To adaptively set thresholds, weight the prediction error based on the layer's **precision** Πi which represents the inverse of uncertainty:
$$\widetilde{\epsilon}_i = \Pi_i \cdot \epsilon_i \tag{17.3}$$

Here, $\widetilde{\epsilon}_i$ is the precision-weighted prediction error.

3. **Adaptive Threshold Adjustment:** Let the threshold at layer L_i denoted τ_i be adjusted based on the running average or variance of prediction errors at that layer. For example, using an exponentially weighted moving average (EWMA):

$$\tau_i(t+1) = \alpha \cdot \tau_i(t) + (1-\alpha) \cdot |\widetilde{\epsilon}_i| \tag{17.4}$$

where α is a smoothing factor, controlling how quickly the threshold adapts to changes in prediction error magnitude.

4. **Hierarchical Precision Tuning:** Active inference frameworks often adjust precision at each layer based on uncertainty in the environment. If a lower-level layer exhibits high variability (uncertainty), precision at higher levels may be reduced (lower confidence in predictions), increasing tolerance for prediction errors.
We can compute adaptive precision Π_i at each layer based on past errors:

$$\Pi_i = \frac{1}{\sigma_i^2 + \beta} \tag{17.5}$$

where σ_i^2 is the variance of prediction errors at layer L_i, and β is a small constant to avoid division by zero.

5. **Threshold Propagation Through Layers:** In a hierarchical model, the adaptive thresholds τ_i can also be influenced by errors at neighboring layers. For instance:

$$\tau_i = f(\epsilon_{i-1}, \epsilon_{i+1}) \tag{17.6}$$

where τ_i is set adaptively based on errors in both the layer above and the layer below, helping each layer balance local and global model adjustments.

17.5.2 Incorporating actions into threshold setting

Active inference involves both **belief updates** and **action selection**. Thresholds can be adapted based on both the **perceptual** prediction errors (used to update internal beliefs) and the **action** outcomes (used to reduce discrepancy between predicted and actual sensory inputs).

For example:

- If actions reduce prediction error, thresholds may decrease, indicating higher precision in predictions.
- If actions increase prediction error, thresholds may increase to allow more flexibility in updating the generative model.

17.5.3 Optimizing thresholds using free energy minimization

In active inference, the agent seeks to minimize **free energy**, which combines **prediction error** and **model uncertainty**. Thus, thresholds τ_i can be adaptively set by minimizing the total free energy at each hierarchical level.

The free energy at each layer i can be expressed as:

$$F_i = \frac{1}{2}\widetilde{\epsilon}_i x^2 + H(\Pi_i) \tag{17.7}$$

where $H(\Pi_i)$ represents the entropy or uncertainty in the precision at that layer. Minimizing F_i can help set optimal thresholds at each layer by balancing prediction accuracy and uncertainty.

17.5.4 Threshold setting summary

To adaptively set thresholds in hierarchical layers using active inference:

1. **Monitor prediction errors** at each layer.
2. **Adjust thresholds** based on the precision-weighted errors.
3. **Tune precision** and thresholds across layers by minimizing free energy and propagating uncertainty estimates.
4. **Use feedback from action** to refine thresholds.

This approach aligns with the active inference principle of reducing prediction errors while accounting for uncertainty in a dynamic environment.

17.6 Implementation

To implement the proposed architecture, we developed a neuromorphic IoT framework that integrates neuromorphic devices for data processing. These devices are deployed on edge devices, which are responsible for collecting data, processing it locally, and making decisions in real-time. The framework is designed to be modular, allowing for easy integration with existing IoT systems and flexibility in terms of the types of sensors and devices used.

Auto Regressive Integrated Moving Average (ARIMA) is a classical model used for time-series forecasting, which combines autoregression, differencing, and moving average components. Both moving average with threshold comparison, as required in each layer of our IoT architecture, and ARIMA can be realized using neuromorphic computing. Moving average and threshold comparison can be implemented in following way [16], [17] (see Figure 17.3):

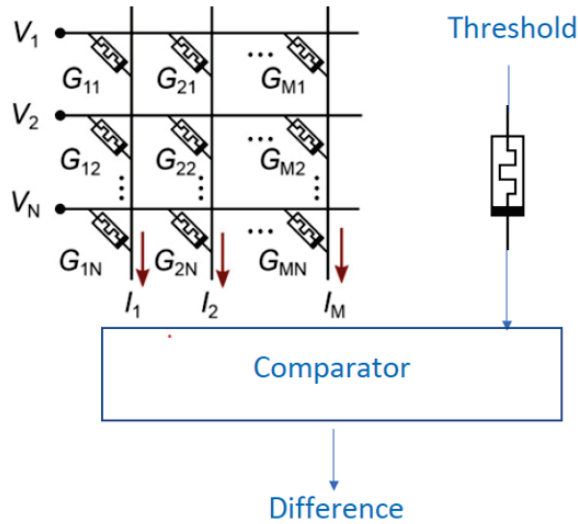

Figure 17.3 Memristor implementation of moving average and threshold comparison.

The implementation consists of the following components:

1. **Memristor Array for Moving Average:**
   - The inputs are weighted by the conductance of each memristor.
   - By applying voltages (input signals) across memristors with different conductances, a weighted sum of the inputs can be computed. Since the conductance (inverse of resistance) of the memristor changes based on the applied voltage, it represents the weight of each input in the moving average. The total output current represents the weighted sum (i.e., the moving average).
2. **Threshold Comparator:**
   - A reference memristor holds the threshold value.
   - The output current from the memristor array (moving average) is compared to the current through the threshold memristor.
3. **Output Decision:**
   The difference in the currents between the input memristor and the threshold memristor can be calculated using a current subtractor circuit or a differential amplifier. This will output a current (or voltage) proportional to the difference between the input signal and the threshold.

17.7 Case Study: Smart Village Water Management

17.7.1 Context and objectives

The municipality of Neuhaus in south-eastern Carinthia, Austria, on the Slovenian border with 1015 inhabitants (as of 1 January 2024, https://www.statistik.at) and an area of 36.34 kmš offers a unique opportunity to apply the proposed neuromorphic IoT architecture in a real-life scenario. Like other similarly structured micro-communities, Neuhaus is facing the typical problems of rural areas in Central Europe: declining population figures combined with shrinking financial and human resources and new challenges due to climate change. Water management is of crucial importance in this region, as efficient utilization of water resources is essential for both environmental sustainability and economic viability. In the last two years, Neuhaus was confronted with a period of drought in the summer of 2023 and enormous amounts of rain, flooding, and landslides in August 2024. Both have a negative impact on financial resources and the quality of life in the region.

In order to overcome these challenges and reduce the administrative expenses, the municipality decided back in 2020 to replace all 377 water meters in the municipality with smart meters. The water meters have to be replaced every 4 years anyway for calibration reasons, so the one-off additional costs were limited. A municipal LoRaWAN radio network was set up for data transmission. LoRaWAN fulfils the required criteria for long range in rural areas, low energy consumption of battery-operated smart meters, and low installation and operating costs. The first automatic meter reading took place in September 2021. In addition to the water meters, the reservoirs of the three separate water supply systems, in which the spring water is collected, were equipped with solar-powered level and flow meters. A total of around 560 LoRaWAN sensors are currently installed in the municipal area. In addition to the water meters, these include weather stations, road temperature sensors, snow depth gauges, and indoor CO_2 sensors, for. In July 2022, the research cooperation between the municipality of Neuhaus and the University of Applied Sciences FH Campus Wien was launched. The aim of this cooperation is to do research on IoT solutions for small municipalities. This gives us access to all measurement data of the LoRaWAN sensor network. Instead of simulated laboratory data, we can work with real data, coming from a lossy IoT network. The first project realized as part of this collaboration was a water management system for real-time water

balances tailored to small communities with their limited resources. It has been in operation since July 2023 and is used by the municipality.

In the present case study, we extend this system to include predictions and alerts. The problems we are facing are typical for IoT sensor networks. We have limited edge devices (our battery-powered water meters) and an unreliable network with very limited data rates. Typically, the transmission of water meter data occurs once daily to save battery power. An immediate message should only be sent in the event of an exceptional incident, e.g. a burst pipe after the meter. So, the decision must be made locally at the edge device. And, since these devices are battery-powered, they are also limited in their processing power and memory. The objectives of this case study are to:

1. Predict water consumption patterns in the community.
2. Detect anomalies in water usage that may indicate leaks or other issues.
3. Demonstrate the effectiveness of the proposed architecture in reducing energy consumption, communication overhead, and latency.

17.7.2 Data collection and preprocessing

Data on water consumption were collected from various sensors deployed throughout the community. These data included hourly and daily water usage statistics, from the smart meters, as well as information on environmental

Table 17.1 Input and output information available at each IoT layer.

IoT Layer	Input from layer above	Other Inputs	Output to layer above	Time and space scale
Edge	Local water consumption threshold	Actual water consumption of each household	Difference between expected average local water consumption and the threshold	Seconds/Local (household)
Fog	Regional water consumption threshold	Regional water supply, Regional water pipe connection map	Difference between expected average regional water consumption and the threshold	Days/Regional
Cloud	-	Time, date and season, weather information from internet, water supply	Output to human: statistics, predictions and alarms over global water consumption	Months/Global

factors such as temperature and humidity, which may influence water consumption patterns. The data were preprocessed to remove noise and fill in missing values due to losses in the LoRaWAN network before being fed into the neuromorphic IoT framework. The following table shows the input and output information available at each IoT layer (see Table 17.1).

17.7.3 Prediction models and performance

To evaluate the effectiveness of the proposed architecture, we compared the performance of several machine learning models in predicting water consumption. The models used included a Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM) network, Auto Regressive Integrated Moving Average (ARIMA) model, and Random Forest. The mean absolute percentage error (MAPE) was calculated for each model to assess their predictive accuracy.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\overline{y}_i - y_i}{y_i}\right| \tag{17.8}$$

Where:
$\overline{y}_i$is predicted value for ith data point
y_i is actual value for ith data point
N is the number of observations

The results (see Table 17.2) indicate that the Random Forest model performed the best in terms of hourly prediction accuracy, with a MAPE of 26.05%. For daily predictions, the LSTM model achieved the lowest MAPE of 4.87%.

Overall, the ARIMA model achieves quite good performance in both hourly and daily water consumption prediction.

ARIMA (AutoRegressive Integrated Moving Average) is a classical model used for time-series forecasting, which combines autoregression, differencing, and moving average components.

Table 17.2 Hourly and daily prediction errors of different algorithms.

Algorithm	Hourly Prediction MAPE [%]	Daily Prediction MAPE [%]
MLP	41.10	5.05
LSTM	33.51	4.87
ARIMA	30.06	5.18
Random Forest	26.05	5.40

Reservoir computing can approximate ARIMA by leveraging its ability to model nonlinear relationships and memory of past inputs, which are essential components of ARIMA models. In particular, the recurrent nature of the reservoir allows it to capture the autoregressive and moving average aspects of the time series, while the nonlinear transformation within the reservoir can approximate the differencing and other complex relationships present in ARIMA models. The **equivalence of reservoir computing to nonlinear vector autoregression (NVAR)** is rooted in the way both models handle temporal data [16], [17]. NVAR is a statistical model that predicts future values of a time series based on past values, accounting for possible nonlinear relationships between variables. Reservoir computing, by transforming input sequences into a dynamic state within the reservoir, effectively performs a similar operation to NVAR. It creates a nonlinear mapping of past inputs, which can then be used to predict future values. This equivalence is particularly useful because it shows that reservoir computing can be viewed as a form of nonlinear autoregression, with the reservoir acting as the nonlinear transformation that enables the prediction of future states based on past data. This understanding opens up the possibility of using reservoir computing as a powerful tool for time-series prediction, where traditional methods like NVAR are used.

In summary, by exploiting the equivalence between reservoir computing and nonlinear vector autoregression, reservoir computing can be effectively used to approximate the behavior of ARIMA models, offering a powerful alternative for time-series forecasting that is particularly well-suited to handling nonlinear and complex patterns in the data. Neuromorphic reservoir computing is particularly interesting for IoT networks due to its low computational overhead and wide range of possibilities for physical implementations. In this way, not only is the IoT architecture neuromorphically inspired, but it can also be partially realized, especially at the edge, by neuromorphic hardware.

17.7.4 Anomaly detection

Anomaly detection is a critical aspect of water management, as it allows for the identification of unusual patterns in water usage that may indicate leaks or other issues. In this study, we applied both the mean + 3 sigma rule and an LSTM-based anomaly detection method to the water consumption data. The results showed that the mean + 3 sigma rule detected more anomalies than

the LSTM model, suggesting that simple statistical methods may be more effective for this type of application in certain contexts.

17.8 Discussion

17.8.1 Energy efficiency and communication overhead

One of the primary advantages of the proposed neuromorphic IoT architecture is its energy efficiency. By processing data locally at the edge and using event-driven communication, the system significantly reduces the need for constant data transmission to centralized servers, thereby lowering energy consumption and communication overhead. This is particularly important in rural areas like Neuhaus, where energy resources may be limited.

17.8.2 System responsiveness and latency

The asynchronous processing and event-driven communication of the proposed architecture also contribute to improved system responsiveness. In real-time applications such as water management, where timely detection of anomalies is crucial, the ability to process data and make decisions quickly can prevent significant water loss and reduce the environmental impact.

17.8.3 Safety & security

Safety: For analog systems it is possible to use continuity properties when pondering system behavior in different points of their state space. If a system exhibits intended behavior in a state A and in a related state B, it can be argued that it will show intended behavior also when $C = \alpha A + (1 - \alpha)B$, where $0<\alpha<1$. So if the system is tested in states A and B, then it can be assumed that it will not change too much in the intermediate states C in between.

Security: It is more difficult to access and modify analog hardware like memristors or reservoirs than to perform a security attack over the internet.

17.8.4 Practical implications

The case study demonstrates that the proposed neuromorphic IoT architecture is not only theoretically sound but also practically viable. The deployment in Neuhaus and preliminary results show that the architecture can handle the complexities of a real-world environment while delivering tangible benefits

in terms of energy savings, reduced communication overhead, and improved system responsiveness.

17.9 Conclusion

The exponential growth of IoT networks demands a new approach to system architecture, one that balances flexibility, learning capability, and energy efficiency. This paper has proposed a neuromorphic IoT architecture inspired by biological systems, designed to meet these challenges in edge computing scenarios. Through a case study on water management in the Carinthian community of Neuhaus, we have demonstrated the practical application and benefits of this architecture. The results show that the proposed architecture can deliver significant improvements in energy efficiency, communication overhead, and system responsiveness, making it a promising solution for the future of IoT networks.

17.10 Future Work

While the proposed architecture has shown promising results, further research is needed to optimize the integration of neuromorphic computing with existing IoT frameworks and to explore its application in other domains. Additionally, the development of more sophisticated anomaly detection methods, potentially integrating neuromorphic principles, could further enhance the system's capabilities. As IoT networks continue to evolve, the principles outlined in this paper will be crucial in guiding the development of next-generation systems that are both efficient and effective

Acknowledgements

The research reported in this paper has been partly funded by the European Union's Horizon 2020 research and innovation program within the framework of Chips Joint Undertaking (Grant No. 101112268) and by the City of Vienna Magistrat 23, FH-Call 29, project 29-22: 'Stiftungsprofessur—Artificial Intelligence' and FH-Call 30, project 30-25: 'Artificial Intelligence and Virtual Reality Lab'. We would also like to thank the municipality of Neuhaus for its great cooperation.

References

[1] S. F. Ahmed, et al., "Towards a secure 5G-enabled Internet of Things: A survey on requirements, privacy, security, challenges, and opportunities," *IEEE Access*, 2024.

[2] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[3] S. Furber, "Large-scale neuromorphic computing systems," *Journal of Neural Engineering*, vol. 13, no. 5, p. 051001, 2016.

[4] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, 2020.

[5] A. Sandberg, "Energetics of the brain and AI," *arXiv preprint arXiv:1602.04019*, 2016.

[6] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, and B. Kay, "Opportunities for neuromorphic computing algorithms and applications," *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, 2022.

[7] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[8] R. A. Adams, S. Shipp, and K. J. Friston, "Predictions not commands: active inference in the motor system," *Brain Structure and Function*, vol. 218, pp. 611–643, 2013.

[9] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[10] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[11] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[12] Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019.

[13] K. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.
[14] T. Parr and K. J. Friston, "Generalised free energy and active inference," *Biological Cybernetics*, vol. 113, no. 5, pp. 495–513, 2019.
[15] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, "Active inference on the edge: A design study," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Mar. 2024, pp. 550–555.
[16] E. Bollt, "On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 1, 2021.
[17] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, "Next generation reservoir computing," *Nature Communications*, vol. 12, no. 1, pp. 1–8, 2021.