iBuilding: Artificial Intelligence in Intelligent Buildings

Will Serrano

The Bartlett, University College London, London United Kingdom

w.serrano@ucl.ac.uk

Abstract

This article presents iBuilding: distributed Artificial Intelligence embedded into Intelligent or Smart Buildings in an Industry 4.0 application that enables the adaptation to the external environment and the different building users. Buildings are becoming more intelligent in the way they monitor the usage of its assets, functionality and space. The more efficiently a building can be monitored or predicted, the more return of investment can deliver as unused space or energy can be redeveloped or commercialized therefore reducing energy consumption while increasing functionality. This article proposes distributed Artificial Intelligence embedded into a Building based on neural networks with a Deep Learning structure. 1) Sensorial neurons at the device level are dispersed through the Intelligent Building to gather, filter environment information and predict its next values. 2) Management neurons based on Reinforcement Learning algorithm at the edge level make predictions about values and trends for building managers or developers to make commercial or operational informed decisions. 3) Finally, Transmission neurons based on the Genetic Algorithms and the Genome codify, transmit iBuilding information and also multiplex its data entirely to generate Clusters of Buildings interconnected with each other at the cloud level. The proposed iBuilding based on distributed learning is validated with a public research dataset; the results show that Artificial Intelligence embedded into the Intelligent Building enables real-time monitoring and successful predictions about its variables. The key concept proposed by this article is that the learned information obtained by iBuilding after its adaptation to the environment is never lost when the building changes over time or is decommissioned but transmitted to future generations.

Keywords: Intelligent Building, Smart City, Reinforcement Learning, Smart Energy, Artificial Intelligence.

1 Introduction

Intelligent Buildings enable their owners and tenants to fulfil their objectives by supporting the management of environmental, space and energy resources. The effectiveness and efficiency of the building business eco-system are increased while adapting to social and technological change driven by human and business needs [1]. Similarly, biology organisms are gradually and continuously learning while adapting to the environment using genetic changes to generate new complex structures in organisms [2]. The adaptations learned from the living organisms affect and guide evolution even though the characteristics acquired are not transmitted to the genome [3]. However, its gene functions are altered and transmitted to the new generation. Successful Machine Learning and Artificial Intelligence (AI) models have been based on biology emulating the structures provided by nature during the learning, adaptation and evolution when interacting with the external environment. Neural networks and deep learning are based on the brain structure which is formed of dense local clusters of the same neurons. Dense clusters perform different functions which are connected between each other with numerous very short paths and few long-distance connections [4]. The brain retrieves a large amount of data obtained from the senses; analyses the material and finally selects the relevant information [5] where the cluster of neurons specialization occurs due to their adaption when learning tasks.

Intelligent Buildings are becoming increasingly on-demand due to their potential for applying innovative and sustainable design initiatives. In addition, emerging technologies optimise its users' comfort, well-being, space, energy and operational management [1]. However, various definitions, interpretations and key performance indicators of Intelligent Buildings have been proposed in different contexts [6] where a systems view of a building is the starting point for considering a building business, space and management. Multi-criteria frameworks composed by primary factors such as energy, environment, space flexibility, cost-effectiveness, client comfort, working efficiency, safety, culture, and technology have been proposed as a comprehensive tool for the selective categorization of Intelligent Buildings [7].

Recent years have witnessed the creation and fast development of "coworking" spaces that are disrupting traditional models of workspaces and businesses and the way people work and collaborate. This additional autonomy and empowerment have raised human, social, managerial and organizational issues [8]. Coworking implies a new form of work organization that enables collaboration opportunities and encourages a sense of community inside a shared space, joining together workers from different companies or even freelancers and contractors with different profiles and objectives. Sharing activities requires not only spatial proximity, it also requires enablement, human behaviour and education [9]. The socio-spatial dimensions of sharing space are modelled through three vectors on different spatial scales: urban sharing, sharing a living space, and shared social spaces. Innovation is more likely to materialize when there are shared practices and spaces as it builds on openness and collaboration [10]. However, the process, relation and outcomes between collaboration and innovation, or the "sparking idea generator" is still not very well understood or modelled.

1.1 Research motivation

The emulation of methods that biology has developed to adapt to the environment and transmit information in the Genome are proposed in a neural computing model. The presented Industry 4.0 application is based on Intelligent or Smart Buildings. This research article considers buildings as biological organisms

where their adaptation to their environment is based on their learning from its users' comfort and well-being.

1.2 Research Proposal

The definition iBuilding: distributed Artificial Intelligence embedded into Intelligent Buildings or Smart Buildings in an Industry 4.0 application. IBuilding enables the adaptation to the external environment and the different building users by monitoring its usage and functionality in terms of physical variables such as assets, space and energy. IBuilding is formed of three layers of neural networks with a Deep Learning structure:

- Sensorial neurons at the device level are dispersed through the Building to gather, filter and predict building information such as space utilisation, energy usage, condition monitoring and environmental variables. Long Short-Term Memory (LSTM) neural networks make next value predictions based on time series data. The device layer supports proactive building operations and maintenance rather than reactive.
- Management neurons at the edge level are based on Deep Reinforcement Learning (DRL) algorithms to make predictions in terms of values and trends (upwards, downwards, equal). DRL entirely considers the previous learnings from current and prior rewards, rather than only the actual, therefore incorporating time on different configurations such as a sampling rate and memory duration. The edge layer enables the adaption of the Intelligent Building to the future demand of its space or energy. It also assists building managers or developers to make informed investment or commercial decisions.
- Transmission neurons codify, multiplex and transmit iBuilding data to the cloud. The learned information is never lost when the building changes over time or is decommissioned but transmitted to future iBuilding generations or existing Intelligent Buildings. The cloud layer interconnects buildings with each other to generate a cluster of buildings that supports evolution. Transmission sensors use Genetic Algorithms based on the Genome where information is transmitted to new generations in the network weights through the different combinations of four different nodes (C, G, A, T) rather than the value of nodes themselves. The output layer of nodes replicates the input layer as the genome reproduces organisms.

The key concept proposed by this article is that the learned information obtained by iBuilding after its adaptation to the environment is never lost when the building changes over time or is decommissioned but transmitted to future generations.

1.3. Research structure

Related work covering energy management, usage, building management, space occupancy, building models, Internet of Things (IoT), 5G, Edge computing Machine Learning, and Artificial Intelligence, including Genetics Algorithms are described in Section 2. IBuilding definition based on distributed Artificial Intelligence is detailed in Section 3 with the mathematical model. The key concept of the proposed Genetic algorithm is that information is codified in the network weights rather than the neurons; similar to the Genome to enable the Artificial Intelligence evolution in iBuilding. iBuilding model is validated in Section 4 where experimental results are also presented, finally, Section 5 shares the conclusions of this research.

2 Research background

2.1. Energy optimization models

Occupant behaviour is one of the key factors that influences building energy consumption and contributes to uncertainty in building energy use prediction and simulation [11]. Advancements in data collection techniques, analytical and modelling methods and simulation applications provide insights in modelling occupant behaviour to quantify its impact on building energy use and energy savings. While most studies focus on energy savings during occupied hours, energy is also wasted during non-occupied hours in commercial buildings [12]. More energy is used during non-working hours (56%) than during working hours (44%) mostly from occupants' behaviour and partly due to inadequate zoning and controls. Occupant behaviour is not well understood and it is often oversimplified in the building life cycle due to its stochastic, diverse, complex, and interdisciplinary nature [13]. The use of simplified methods to quantify the impact of occupant behaviour in building performance simulations significantly contributes to performance gaps between simulated models and the actual building energy consumption.

Buildings consume a significant amount of energy, approximately one-third of the total primary energy resources [14]. Building energy efficiency is a complex problem based on the limitation of the occupants' comfort level and user behaviour. Intelligent control systems for energy and comfort management in smart energy buildings include intelligent methods, simulation tools, occupants' behaviour and preferences for various building types. Energy unaware behaviour increases one-third to a building's designed energy performance [15]. User activity and behaviour are considered as a key element on energy saving potential and it has long been used for control of various assets such as artificial light, heating, ventilation, air conditioning and power sockets.

There are opportunities of using occupancy information to develop a more energy efficient building climate control based on office buildings equipped with integrated room automation that integrates the control of heating, ventilation, air conditioning as well as lighting and blind positioning of a building zone or room [16]. The evaluation of the energy-saving potential cover different types of occupancy information used in a model predictive control framework. Commercial office buildings represent the largest floor area in most developed countries, this makes office buildings a target for occupant-driven demand control measures [17]. However, the application of occupant-driven demand control measures in buildings, most especially in the control of thermal, visual and indoor air quality providing systems, is hindered due to the lack of comprehensive detailed occupancy information.

Many retrofit projects are being carried out in existing buildings to reduce energy consumption. Although the energy consumption after retrofit can be determined through measurement, the energy consumption before retrofit is more difficult to assess [18]. Dynamic simulations or regression models could be used to estimate the energy consumption of buildings before retrofit,

however, existing regression models are not able to calibrate the model if it is inaccurate. One of the largest users of electricity in the average household are appliances, which when aggregated, also accounts for approximately 30% of the electricity used in the residential building sector [19]. Modelling the usage energy of appliances and what causes variation in their use is becoming more relevant to control the demand on the electricity grid infrastructure.

2.2. Forecasting models

Several machine learning regression methods that develop a predictive model are examined and applied [20] to predict the hourly full load electrical power output of a combined cycle power plant. The baseload operation is influenced by four main parameters: ambient temperature, atmospheric pressure, relative humidity and exhaust steam pressure; these parameters are used as input variables in the dataset that affect the electrical power output, which is considered as the target variable. The prediction of Building energy usage has an important role in building energy management and conservation as it assists in the evaluation of the building energy efficiency, conduct building commissioning, detect and diagnose building system faults [21]. AI based approach for building energy use prediction applies historical data and methods such as multiple linear regression, artificial neural networks and support vector regression. A statistical machine learning framework studies the effects of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution) on two output variables: heating load and cooling load of residential buildings based on a classical linear regression approach [22]. The model is compared to a state of the art nonlinear non-parametric method based on random forests.

Forecasting the energy consumption in homes is an important aspect for the smart grid where the prediction of energy consumption in housing is very dependent on inhabitants' behaviour. A stochastic prediction method segments data based on patterns in energy consumption and aggregates it using the k-means clustering algorithm [23]. Data-driven predictive models for the energy use of appliances include measurements of temperature and humidity sensors from a wireless network, weather from a nearby airport station and energy use [24]. Data is filtered to remove non-predictive parameters and feature ranking where four statistical models are trained and evaluated with repeated cross-validation: multiple linear regression, support vector machine with radial kernel, random forest and gradient boosting machines.

2.3. Artificial Intelligence in the IoT

Machine Learning and AI have been used in emergency navigation in a cloud environment to reduce device energy consumption [25] and without static ad hoc networks such as wireless sensor network infrastructure [26]. IoT components cover smart objects with data processing and storage, communication networks, system data storage and security. AI is applied at these different levels to enhance the scalability, decision making, cognitive computation and security [27]. The key challenges to implement AI in the IoT are the limited computing, storage and power on the devices whereas processing at the edge or Cloud requires efficient connectivity, security, trust and data interoperability. Secure and energy efficient data collection are key to the success of the IoT. Cognitive sensing uses smart sensors to intelligently perceive inputs from the environment. Cognitive sensing is divided into smart energy management (energy harvesting, duty cycles, battery optimization, wireless energy transfer), self-management (self-configuring, self-healing, self-optimising), cognitive security (security classification, intrusion detection, malware detection, authentication system, attack recovery, privacy and identity management, anomaly detection) and smart data collection (data compression, data encoding, data prediction and reconstruction, data transmission, data fusion aggregation, dimensionality reduction) [28].

Low power loss networks and wireless sensor networks are the basic transmission components of the IoT. An energy balance routing algorithm based on an intelligent chaotic ant colony algorithm is combined with energy factors of wireless sensor network nodes [29]. The algorithm based on AI considers energy and delay to find the optimal solution for the network energy consumption. The IoT provides new data sets collected from sensor devices. This new model requires new data mining algorithms specialised for IoT data. The applicability, effectiveness and efficiency of eight well-known data mining algorithms are analysed in real IoT datasets [30]. These include Support Vector Machine, K-Nearest Neighbours, Linear Discriminant Analysis, Naïve Bayes, C4.5, C5.0, Artificial Neural Networks, and Deep Learning. From the largest megacities to the smallest IoT are joined in a four-layer model that consists on 1) IoT for context awareness, communication capability and AI to learn about these; 2) smart homes based on advanced AI to learn and predict user's behaviours and preferences and managing IoT data patterns; 3) a Cloud of Things that connect smart homes based on 5G; 4) smart city based on Infrastructure Communications Technologies (ICT) [31].

An integrated IoT architecture that collects and performs Big data analytics is implemented for smart meter networks in smart cities [32]. The machine learning algorithm applies unsupervised clustering and model interference based on the extrapolation of future consumption of each meter in the short and medium terms. The key outcomes are alarm activation after an incident and a supervised clustering tool to classify customers according to their consumption pattern. An AI system in the IoT to predict humidity and temperature [33]. The model uses these predictions to control a fan as a feedback effect. Machine learning and deep learning are the algorithms for Big Data processing and analytics. However, IoT devices based on limited energy and low compute impedes these algorithms to be used at the device level. A roadmap that addresses these issues presents a scalable, high-performance and energy efficient architecture for performing machine learning on the edge [34]. The method is based on the alternation of neural network structures, approximate and near threshold computing, emerging technologies and distributed learning and inference.

An AI system detects and corrects errors in multimedia transmission for a surveillance IoT environment connected through a Software Defined Network (SDN). The AI algorithm consists on a classifier that detects critical multimedia traffic sent through the network and an estimator that informs the SDN controller about the next action to guarantee the Quality of Service (QoS) and Quality of Experience (QoE). Quality is based

on jitter, latency and losses [35]. The application of AI and the IoT in an unstaffed retail shop consists of consumer identification and commodity recognition [36]. The AI model applies image processing, localisation and classification of objects and final screening to determine the final sample library.

Advanced AI in the IoT (AAIoT) is a method that allocates the inference computation of each network layer to each device in a multi-layer IoT system. The inference process is optimised by a neural network segmentation method that can be deployed in multi-layer IoT architectures without knowing the details inside the system [37]. The computing and communication features of each device are the model input whereas the output is the quantitative method in computational allocation for deep learning inference tasks that reduce response time. IoT and AI are combined to reduce traffic congestions in a smart city environment [38]. The model analyses the traffic system in order to find suitable re-routing strategy that minimizes congestion. The backend AI search algorithm calculates the shortest possible path from a given source to a given goal depending on a given cost function to the destination by finding the best route on a macro level and then expanding the graph to obtain best sub-goals for the final solution. The method includes optimization and load balancing so that new congestions are not being created due to the re-routed traffic.

Human-centric AI field generates explanations about the knowledge learned by a multilayer perceptron neural network from IoT environments [39]. Users confirm or refute the decision based on whether the given reasons make sense within the given context. Humans are given the opportunity to understand the reasons behind AI decisions to enhance the supervision process. IoT systems learn from users while providing explanations about decisions or estimations based on two techniques. These two methods consist of the analysis of artificial neuron weights and the estimation based on the analysis of training cases artificial neuron weights in the multilayer perceptron trained with backpropagation. This approach has been applied in a smart IoT kitchen that detects the user depression based on the food used for each meal. Several neural networks with different depth and input size are trained for object detection in IoT devices. Time of execution, temperature, CPU load, RAM load and power consumption are analysed based on the frames per second using an embedded system with an external low-power graphics processing unit [40].

An integrated IoT platform in cloud computing takes information from sensors that monitor air's quality of different environments to alert people are near to the polluted space through wearable smart devices [41]. The platform uses AI to analyse data transmitted via different protocols such as Bluetooth 5.0, WiFi, ZigBee and LoRa where MQTT is used to communicate with other clouds. The intelligent system finds the closest air quality monitoring network within the user usual route, learns the routes with different degrees of pollution and finally selects the best one. A cyber-physical system based on a trusted robust intelligent control strategy and a trusted intelligent prediction model relays on the high concentration of information resources retrieved from IoT devices [42]. The model evaluates, analyses and quantifies the reliability of the system using machine learning to determine the spatio-temporal correlation detection within the analysis strategy of the reliability model. The full integration of AI in every device and stage of manufacturing systems is currently not implemented. holistic integration of AI promotes collaboration as a multi-dimensional conceptual term that covers entirely the important enablers for AI adoption in manufacturing contexts [43]. Collaboration provides business intelligence optimization, "human in the loop" and secure federation across manufacturing sites. The presented model is built on three technical pillars: (1) components that extend the functionality of the existing layers in a reference architectural model for Industry 4.0; (2) definition of new layers for collaboration by means of "human in the loop" and federation; (3) security concerns with AI powered method.

IoT data analytics has gained significant importance and attention due to the 1) high volume of data generated from distributed IoT devices; 2) high variability of data types from heterogeneous data sources; 3) uncertainty in the IoT data streams; 4) balancing scalability with the efficiency. AI analytics consist on descriptive analytics that processes and summarizes raw data while providing actionable insights and predicting analytics that model the behaviour or pattern to predict the likelihood of possible future trends or patterns in data [44]. Prescriptive analytics suggests how to respond to any future events based on data analysis. Adaptive analytics adjusts or optimises the process outcome based on the recent history of the data and its correlations. AI in the IoT termed as the Cognitive IoT will make IoT more sophisticated, intelligent, intuitive and interactive. Cognitive IoT augments the current IoT with the added cognitive ability similar to human cognition by learning, sensing and taking decisions as humans. Cognitive computing enables IoT to interact dynamically with other connected objects, as well as to adapt to the present context through continuous learning from the environment based on networking, behavioural and data analytics [45]. AI in Cognitive IoT is based on machine learning, computer vision and natural language processing.

Integration of cloud, IoT and home appliances defines a new automated environment. AI focused on machine learning and natural language processing interact with humans as the interface between users and smart homes [46]. Machine learning trains itself iteratively to make responses accurate to the commands said by the humans whereas natural language processing understands and speaks the language easily.

2.4. Edge Computing in IoT

Fog is an emergent architecture for computing storage, control, and networking that distributes these services closer to end-users. Fog as a Service (FaaS) supports new business models to deliver services to customers as companies will be enabled to deploy and operate private or public computing, storage, and management services at variable scales [47]. Fog and cloud complement each other to form a continuous service by providing mutually beneficial and interdependent services based on cognition, efficiency, agility and latency. A forward central dynamic and available approach optimises the duty cycle and transmission power levels of IoT-based portable devices during sensing, processing and transmission tasks [48]. The method applies AI at the edge to model the batteries of the system by evaluating the energy dissipation in IoT devices. In addition, AI at the edge models the data reliability over hybrid transmission power control and duty-cycle network.

AI technology provides an efficient and intelligent dynamic path solving model in a mobile terminal fire evacuation system built for large public buildings [49]. The method is based on a dynamic evacuation where implemented flows are derived from an ant colony algorithm. The system framework is composed of data acquisition equipment, RFID tags, building data model and personnel location information. A method of feature dimension reduction for wireless communication signals from IoT devices is applied to a power amplifier radio frequency (RF) fingerprinting [50]. The high dimensionality of RF fingerprint features and the uncorrelated or redundant attributes in the feature space are reduced. The three applied AI algorithms are based on the principal component analysis, linear discriminant analysis, and autoencoder that use the distance ratio criterion to evaluate the separability of the features. The large amount of data generated by the IoT has created bottlenecks in its transmission to the cloud. An IoT architecture uses AI at the edge to provide real-time prediction and advanced data analytics. The method addresses the issues of latency and jitter while filtering data [51].

Edge computing performs parking occupancy detection using real-time video feeds for smart parking surveillance [52]. The solution is based on AI via an enhanced single shot multi-box detector. Three metrics are defined to measure the performance of the system: 1) smartness as the automatic detection and pattern recognition; 2) efficiency as the processing in a real-time and online manner; 3) reliability as consistent detection performance in various environmental conditions. Multi-access edge computing improves the Quality of Experience (QoE) of AI applications in the evolution towards IoT infrastructure based on two heuristic algorithms: greedy minimum dominating set and greedy cover for QoE-aware [53]. The solution is based on SDN and serverless technology that provides a unified service calling interface and schedules the resources automatically to satisfy the QoE requirements of users.

2.5. Cybersecurity in the IoT

Attack and anomaly detection in IoT infrastructure is a rising issue. The prediction performance of different machine learning models has been compared based on attacks and anomalies in IoT systems [54]. The analysed machine learning algorithms are logistic regression, support vector machine, decision tree, random forest, and artificial neural network. A machine learning algorithm based on an artificial neural network is embedded within a gateway to secure IoT systems [55]. The network learns the healthy state of a system and connected devices to detect anomalies in the data sent from the edge devices as invalid data points.

The attack model for IoT systems is investigated where IoT security solutions based on machine learning techniques that include supervised learning, unsupervised learning, and reinforcement learning are reviewed [56]. Machine learning IoT authentication, access control, secure offloading, and malware detection schemes protect data privacy. The challenges that need to be addressed to implement these machine learning security schemes in practical IoT systems are their partial state observation and learning, computation and communication overhead, a large number of training data required and an extensive feature extraction processes. Backup security solutions reinforcement learning based security methods require the exploration of the “bad” security policy that could potentially cause a network disaster for IoT systems at the beginning learning stage. The intrusion detection schemes based on unsupervised learning techniques have misdetection rates that are non-negligible for IoT systems. Supervised and unsupervised learning fail to detect attacks due to oversampling, insufficient training data, and bad feature extraction. A hybrid intelligent classic control approach reconstructs and compensates cyber attacks launched on cyber-physical systems and industrial IoT [57]. Gaussian radial basis neural networks are designed as an intelligent estimator for attack estimation and a classic nonlinear control system based on the variable structure control method is designed to compensate the effect of attacks and control the system performance in tracking applications.

The use of machine learning and deep learning techniques in IoT applications bring multi-faceted challenges such as 1) the development of suitable models; 2) labelling heterogeneous data effectively; 3) minimum labelled data in the learning process; 4) the deployment on resource-constrained IoT devices; 5) anomalies in critical infrastructure and real-time applications. Supervised learning algorithms based on classification and regression work with labelled data and are utilized in IoT networks for spectrum sensing, channel estimation, adaptive filtering, security, and localization problems [58]. Most of the traditional machine learning techniques are not inherently efficient and scalable enough to manage IoT data and thus need considerable modifications. Conventional authentication and authorization schemes are inefficient to overcome the intrinsic IoT security risks due to their dependence on static digital processes and computational complexity. Furthermore, the stand-alone security designs for different layers and link segments do not provide the overall protection, leading to cascaded security risks and growing communication latency and overhead [59]. An AI enabled security method provides fast authentication and progressive authorization. Specifically, a lightweight intelligent authentication and authorization algorithm at the edge identifies prearranged access time sequences, frequency bands or codes used in IoT devices where an online distributed machine learning and trust management process are adopted for achieving an adaptive and holistic access control. Artificial neural networks distinguish variations in the information sent from edge devices after learning the secure environment of an IoT network and associated devices [60]. The AI algorithm scans for IoT gadgets, extracts ordinary patterns and detects abnormal behaviours.

2.6. 5G in the IoT

5G cellular networks provide key enabling technologies for the global deployment of the IoT technology. These include carrier aggregation, massive multiple-input multiple-output, coordinated multipoint processing, device-to-device communications, centralized radio access network, software-defined wireless sensor networking, network function virtualization and cognitive radios. The combination of 5G, IoT and AI supports intelligent decisions in real-time to decrease latency, improve link capacity and enhance security by modelling data packets to predict traffic patterns on the network [61].

5G Intelligent Internet of Things (5G I-IoT) process Big Data intelligently and optimizes communication channels. Sensors collect and transmit data to an object processor at the base station that processes it with Big Data mining algorithms in terms of

integration, cleansing and redundancy [62]. Deep learning and reinforcement learning optimize the entire system online based on temporally inactivating the channel to allow the reception of more relevant data in a shorter time slot. The lower channel occupancy reduces network load latency and energy consumption. There is a need for AI in the future IoT-based 5G networks based on self-organization, learning from the environment and optimum decision making. Machine learning, game theory and optimization algorithms can be applied in the context of IoT 5G dynamic spectrum management, structuring of Big Data, integration of heterogeneous devices, ultra densification of devices, interoperability, and energy management [63]. The incorporation of AI close to the environment at the fog and at the edge reduces the latency, enhances the link capacity and improves the security of the network.

5G edge computing architecture addresses the requirements of IoT networks based on fast and reliable communication with reduced overhead using clustering and AI methods [64]. 5G edge computing supports the offload of device and cloud computation, reduces to the minimum link costs and it enables better management of the entire IoT infrastructure due to its support to decentralization while reducing the need to communicate with data centres in the cloud.

3 IBuilding Model

IBuilding abstracts the underlying digital infrastructure of the Intelligent Building physical infrastructure providing a higher layer that enables distributed Artificial Intelligence to measure, manage and virtualize any Intelligent Building. IBuilding is accessed via a common platform tailored and adapted to the Building functionality and role enabling a flexible, expandable and modular interoperable solution where independent Physical Buildings (PB) and their variables are gradually added and integrated. IBuilding model (Fig. 1) is defined as:

- PB = $\{P_{Building\text{-}1}, P_{Building\text{-}2}, \dots P_{Building\text{-}m}\}$ as a set of m Physical Buildings that represent the real estate.
- AI = $\{ai_1, ai_2, \dots ai_n\}$ as a set of n Artificial Intelligence vectors associated with one or several Physical Buildings $P_{Building\text{-}m}$. AI maps the different building variables that PB is formed with their relevant AI algorithms. Examples are temperature, humidity occupancy associated with feedforward neural networks, recurrent neural networks, convolutional neural networks.
- ai = $\{i_{Sensor}, i_{Management}, i_{Transmission}\}$ as a set that consists of the three virtual layers of Artificial Intelligence that will monitor and manage a Physical Building, $P_{Building\text{-}m}$, based on the set of variables defined by AI.
- iSensor = $(i_{Sensor\text{-}1}, i_{Sensor\text{-}2}, i_{Sensor\text{-}p})$ as a P dimensional vector that represents the device layer. Sensorial neurons collect information $P_{Building\text{-}m}$ related to variable ai_n such as temperature or humidity and make predictions about its future value.
- iManagement = $(i_{Management\text{-}1}, i_{Management\text{-}2}, i_{Management\text{-}s})$ as a S Dimensional vector that represents the edge layer. Artificial Intelligence management algorithms and autonomous decision-making methods make predictions for each $i_{Sensor\text{-}p}$ in terms of trends (upwards, downwards, equal).
- iTransmission = $(i_{Transmission\text{-}1}, i_{Transmission\text{-}2}, i_{Transmission\text{-}q})$ as a Q Dimensional vector that represents the cloud layer. Artificial Intelligence data transmission methods filter, compress, codify and finally transmit $P_{Building}$ information to the cloud. ITransmission will effectively multiplex $P_{Building\text{-}m}$, therefore, creating iBuilding: a cluster of Intelligent Buildings.

Sensor (i_{Sensor})	Management ($i_{Management}$)	Transmission ($i_{Transmission}$)
$ai=\{i_{Sensor}, i_{Management}, i_{Transmission}\}$		
Artificial Intelligence $\{ai_1, ai_2, ai_n\}$		
Physical Building $\{P_{Building\text{-}1}, P_{Building\text{-}2}, P_{Building\text{-}m}\}$		

Fig. 1. iBuilding Mathematical Definition

IBuilding model for this research proposal (Fig. 2) consists of a layer of sensor neurons at the device level that collects Intelligent Building information about their physical variables and transmits predicted values to their respective management sensor (iSensor). A management layer makes predictions about the iSensor trends at the edge level (iManagement). Finally, a layer of neurons codifies and transmits iBuilding information at the cloud level emulating the Genome (iTransmission). Furthermore, iTransmission also multiplexes its data entirely to generate clusters of buildings interconnected with each other.

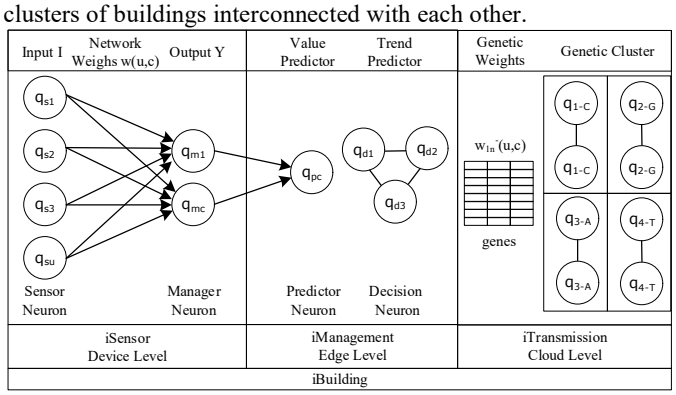


Fig. 2. iBuilding neural model

3.1. ISensor: Neuron Management at the device

ISensor consists of u input neurons that take iBuilding measurements of a specific variable such as temperature or humidity and make local predictions on their values based on a Long Short Term Memory (LSTM) network. These input neurons are related to a precise area or floor and are connected to their management sensor by network weights. iSensor defines:

- I = $(q_{s1}, q_{s2}, \dots, q_{su})$, a U-dimensional vector I Є $[0,1]^U$ that represents the input state q_{su} for the prediction sensor neuron u;
- w(u,c) is the U x C matrix of network weights from the U sensor neurons q_{su} to their C manager neurons q_{mc};
- Y = $(q_{m1}, q_{m2}, \dots, q_{mc})$, a C-dimensional vector Y Є $[0,1]^C$ that represents the output state q_{mc} for the manager neuron c.

The manager neuron q_{mc} aggregates and stores the prediction values from the local sensors q_{su} based on w(u,c). There are several options for the w(u,c): fixed to store the average q_{su} value, variable to include feedback on performance or a function learned with a Machine Learning algorithm. The motivation to

combine sensor information is to generate a higher level of variable representations.

3.1.1 LSTM Mathematical Model

Long Short-Term Memory (LSTM) networks are widely used in time series data as their learning algorithm does not present exploding and vanishing gradient descent issues while traditional recurrent Neural Networks with backpropagation Learning Algorithms do (Fig. 3).

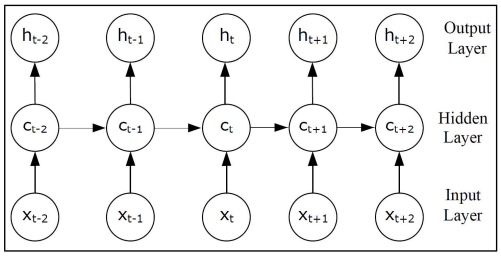


Fig. 3. Long Short-Term Memory Network

LSTM networks are a type of artificial recurrent neural network composed of the following elements:

- The input layer x_t that represents the time-series signal;
- the cell c_t that provides memory to the neural structure;
- the hidden state h_t that transmits information between cells and provides memory based on time;
- the input gate i_t controls the relevance of the new sensorial activity to the cell;
- the forget gate f_t manages the relevance of existing sensorial information stored in the cell;
- the output gate o_t modulates the stimuli the current cells transmit to the next cell in the neural chain.

LSTM network enables a constant error flow through self-connected memory cells (Fig. 4), where input and output gates manage its transmission while protecting it from perturbations [65], although error signals contained within the memory cell can not be altered. Specifically, an LSTM network with P input cells and Q output cells is defined as:

- The forget vector $f_t \in R^Q$:

$$f_t=\sigma_g\left(\sum_{p=1}^{P} w_{fp}x_t+\sum_{q=1}^{Q} u_{qf}h_{t-1}+b_f\right) \tag{1}$$

- The input activation vector $i_t \in R^Q$:

$$i_t=\sigma_g\left(\sum_{p=1}^{P} w_{ip}x_t+\sum_{q=1}^{Q} u_{iq}h_{t-1}+b_i\right) \tag{2}$$

- The output activation vector $o_t \in R^Q$:

$$o_t=\sigma_g\left(\sum_{p=1}^{P} w_{op}x_t+\sum_{q=1}^{Q} u_{oq}h_{t-1}+b_o\right) \tag{3}$$

- The cell state vector $c_t \in R^Q$:

$$c_t=f_t\circ c_{t-1}+i_t\circ\sigma_c\left(\sum_{p=1}^{P} w_{cp}x_t+\sum_{q=1}^{Q} u_{cq}h_{t-1}+b_c\right) \tag{4}$$

- The hidden state vector $h_t \in R^Q$:

$$h_t=o_t\circ\sigma_c(c_t) \tag{5}$$

where:

- $x_t \in R^P$ is the input vector to the LSTM network;
- $w \in R^{QxP}$ is the weight matrix for the input vector;
- $u \in R^{QxQ}$ is the weight matrix for the hidden state vector;
- $b \in R^Q$ is the bias vector;
- σ_g represents the sigmoid function;
- σ_c represents the hyperbolic tangent function.

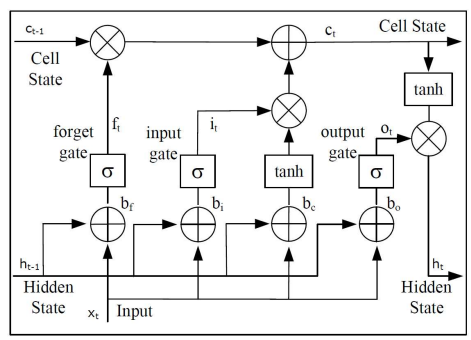


Fig. 4. Long Short-Term Memory Cell

3.1.2 LSTM Learning Model

Each q_{su} has an associated LSTM network. The input x_t at time t for each LSTM network corresponds to the value of its associated q_{su}, where c_t is the predicted next value $t+1$. As time increases, $t=t+1$ the LSTM network includes the newest value, removes its oldest value, learns the new input and finally predicting the next value at $t+2$. This method iterates for the entire time-series data:

- time t, LSTM network learns x_t
- time t, LSTM window slides to include x_t and predicts x_{t+1}
- time t+1, LSTM network learns x_{t+1}
- time t+n, LSTM network slides to include x_{t+n} and predicts x_{t+n+1}
- time t+n+1, LSTM network learns x_{t+n+1}

3.2. IManagement: Deep Reinforcement Learning for predictions at the edge

iManagement takes predictions on iSensor trends via a Deep Reinforcement Learning (DRL) algorithm based on the Random Neural Network (RNN) [66-67]. DRL entirely considers the previous learnings from current and prior rewards, rather than only the actual, therefore incorporating time on different configurations such as a sampling rate and memory duration. This algorithm is used as it includes The number of neurons of the neural network matches the number of decisions to be made by iManagement (Fig. 5). Neurons are designated as 1, …, j, …, n; therefore for any taken decision i, there is some associated neuron i.

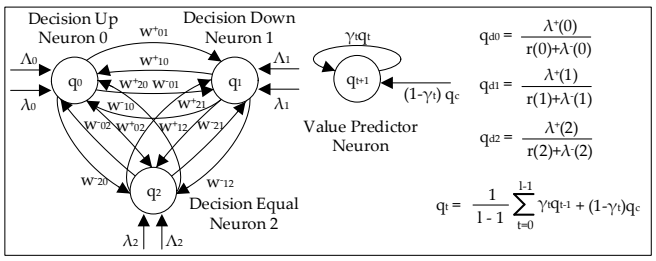


Fig. 5. Deep Reinforcement Learning model

Decisions in the DRL algorithm are selected by taking the option j that corresponds to the most excited neuron, j, which is one which has the greatest value of q_j (Fig. 6). The state q_j is defined as the probability that neuron j is excited, these values meet the following system of non-linear equations:

$$q_j = \frac{\lambda^+(j)}{r(j)+\lambda^-(j)} \tag{1}$$

$$\lambda^+(j) = \sum_{i=1}^{n}\left[q_i r(i) p^+(i,j)\right] + \Lambda(j) \tag{2}$$

$$\lambda^-(j) = \sum_{i=1}^{n}\left[q_i r(i) p^-(i,j)\right] + \lambda(j) \tag{3}$$

where:

- q_j is the probability neuron j is excited;
- $\lambda^+(j)$ is the rate of excitatory signals arriving at neuron j;
- $\lambda^-(j)$ is the rate of inhibitory signals arriving at neuron j;
- r(i) is the firing rate of neuron i;
- $p^+(i,j)$ is the probability positive spikes will go out from neuron i to neuron j;
- $p^-(i,j)$ is the probability negative spikes will go out from neuron i to neuron j;
- $\Lambda(j)$ is the rate of external excitatory signals arriving at neuron j;
- $\lambda(j)$ is the rate of external inhibitory signals arriving at neuron j;

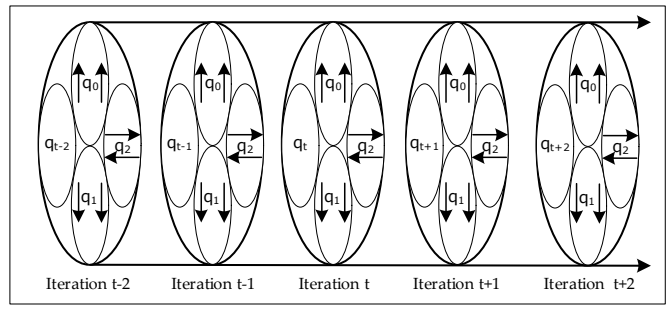


Fig. 6. Deep Reinforcement Algorithm

3.2.1 Decision neurons

Decision Neurons make choices about the predicted or forecasted trend based on the measured Reward on the DRL. In particular, iManagement decision neurons consider if the trend of the next reward is equal, downwards or upwards. As well as trend prediction, the presented model consists of a predictor neuron that forecasts the figure of future rewards. DRL algorithm modifies the Reward and Threshold based on a similar approach as defined in the Cognitive Packet Network [68-69]. IManagement is given a Goal G that has to achieve as a function to be optimized where Reward R is a consequence of the interaction with the environment. Sequential values of the measured R are represented by R_l, l = 1, 2, … these values are used to calculate a decision threshold T_l according to the following equations:

$$R_l = \beta(Y_l - Y_{l-1})$$
$$T_l = \alpha T_{l-1} + (1-\alpha)R_l \tag{4}$$

where β defines the Learning Gradient and α defines the Threshold Memory, 0 < α < 1. These parameters could be dynamically revised following the measurements from external observations or statically assigned. IManagement takes the l_{th} decision based on the most excited neuron, let's say neuron j. After the interaction with the environment, the l_{th} Reward R_l is measured and its associated T_{l-1} is computed where neural network weights are revised according to the following equations for all neurons i ≠ j. The DRL algorithm rewards the neural network weights if the trend decision made by iManagement is right; R_l>0 and j=0 (upwards) or R_l<0 and j=1 (downwards) or R_l=0 and j=2 (equal):

$$w_l^+(i,j) = \sum_{t=0}^{l-1}\delta_t w_t^+(i,j) + T_l$$
$$w_l^-(i,k) = \sum_{t=0}^{l-1}\delta_t w_t^-(i,k) + T_l \quad \text{if } k \neq j \tag{5}$$

Otherwise, it punishes the neural network weights by:

$$w_l^+(i,k)=\frac{1}{l-1}\sum_{t=0}^{l-1}\delta_t w_t^+(i,k)+T_l \text{ if } k\neq j$$

$$w_l^-(i,j)=\frac{1}{l-1}\sum_{t=0}^{l-1}\delta_t w_t^-(i,j)+T_l \tag{6}$$

where:

- δ_t defines a weighting factor that is variable and depends on t, $0 < \delta_t < 1$,
- l is the stage decision
- w^+_{ij} represents the rate at which neuron i emits excitation spikes to neuron j
- w^-_{ij} represents the rate at which neuron i releases inhibitory spikes to neuron j, neuron i must be excited to transmit excitation on inhibition spikes.

The fundamental property of the DRL algorithm is that learning considers time, therefore memory, when it updates the neural network weights; values consist on the entire previous measurements rather than the only is the previous neural state.

3.2.2 Predictor Neuron

In addition to the Reinforcement Learning algorithm for trend decisions, iManagement uses the DRL algorithm to make predictions on the quantitative value of future rewards. The value predictor neuron is based on the current measurement q_c and the entire previous predictions q_t:

$$q_{t+1}=\frac{1}{l-1}\sum_{t=0}^{l-1}\gamma_t q_t+(1-\gamma_t)q_c \tag{7}$$

where γ_t represents the prediction memory, a weighting factor that is variable and depends on t, $0 < \gamma_t < 1$; this parameter can be dynamically updated based on the measurements from external observations or statically assigned.

3.3. ITransmission: Genetic Algorithms at the cloud

ITransmission is a layer of neurons that emulates the way the Genome codifies and transmits information to further generations based on the Random Neural Network Genetic Algorithm. Each Physical Building $P_{Building\text{-}m}$ of iBuilding will be formed of n management layers (iManagement) that correspond to each ai_n Intelligent Building variable. IManagement is formed of predictor and decision neurons which values are codified and transmitted using iTransmission.

In addition, iTransmission multiplexes its data entirely to generate clusters of buildings interconnected with each other. Information in the proposed Genetic Algorithm is transmitted to new generations in the network weights through the different combinations of four different Genetic Clusters or nodes (C, G, A, T) rather than the value of neurons themselves where each network weigh is considered as iBuilding gene (Fig. 7). The proposed algorithm applies an ELM autoencoder as it codes the replica of the organism that contains it where the output layer of nodes replicates the input layer as the Genome reproduces organisms.

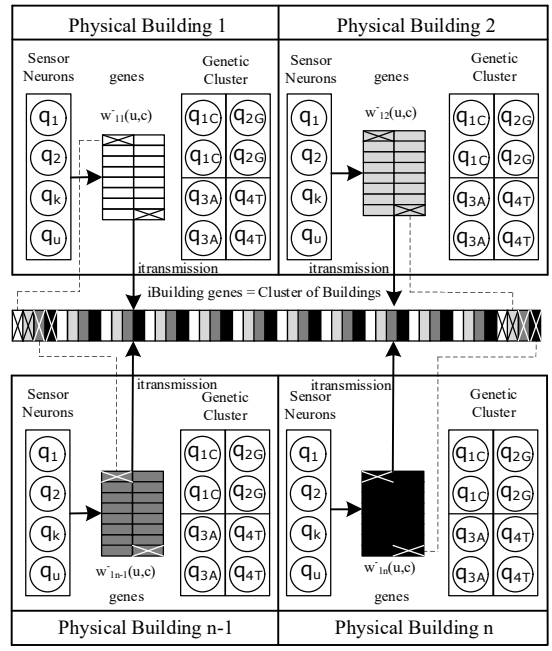


Fig. 7. ITransmission

3.3.1 Extreme Learning Machine

The proposed iTransmission Genetic learning algorithm on this paper is an autoencoder based on the Extreme Learning Machine (ELM) [70-71] for Single Layer Feedforward Networks (SLFN). For N arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, \dots x_{in}]^T$ R^n and $t_i = [t_{i1}, t_{i2}, \dots t_{im}]^T$ R^m, an standard SLFN with hidden nodes and activation function g(x) is mathematically modelled as:

$$f_{N'}(x_j)=\sum_{i=1}^{N'}\beta_i g_i(x_j)=\sum_{i=1}^{N'}\beta_i g_i(w_i\cdot x_j+b_i)=t_j \quad \text{for } j=1,\dots N \tag{1}$$

where $w_i = [w_{i1}, w_{i2}, \dots w_{in}]^T$ is the weight vector connecting the i_{th} hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots \beta_{im}]^T$ is the weight vector connecting the i_{th} hidden node and the output nodes, b_i is the threshold of the i_{th} hidden node and g(x) activation function of hidden nodes. The above N equations can be written as:

$$h(x)\beta=t_j$$
$$H\beta=T \tag{2}$$

where $T= [t_{i1}, t_{i2}, \dots t_{im}]^T$ are the target outputs and $H= [g_i(w_1\cdot x_1+b_1), g_i(w_2\cdot x_2+b_2), \dots g_i(w_{n'}\cdot x_{n'}+b_{n'})]^T$. The output weights β can be calculated by equation 3:

$$\beta=H^{\dagger}T$$

(3)

where $H^{\dagger}$ is the Moore–Penrose generalized inverse of matrix H.

Extreme Learning Machine [70-71] proves that the input weights and hidden layer biases of SLFNs can be randomly assigned if the activation functions in the hidden layer are infinitely differentiable. In addition, SLFNs can be considered as a linear system where the output weights can be analytically determined through the simple generalized inverse operation of the hidden layer output matrices. The proposed Genetic learning algorithm is based on an ELM autoencoder that models the genome as it codes the replica of the organism that contains it. It consists of two instances of the Network described in the next section.

3.3.2 The Random Neural Network with Deep Learning clusters

Deep Learning with Random Neural Networks [72-73] is based on the generalized queuing networks with triggered customer movement (G-networks) where customers or tasks are either "positive" or "negative" and customers or tasks can be moved from queues or leave the network. This algorithm is used due to its minimal codification and decodification error. The model considers a special network M(n) that contains n identically connected neurons, each which has a firing rate r and external inhibitory and excitatory signals λ^- and λ^+ respectively (Fig 6). The state of each neuron is denoted by q, and it receives an inhibitory input from the state of some neuron u which does not belong to M(n). Therefore, for any neuron i Є M(n) There is an inhibitory weight $w^-(u) \equiv w^-(u, i) > 0$ from u to i.

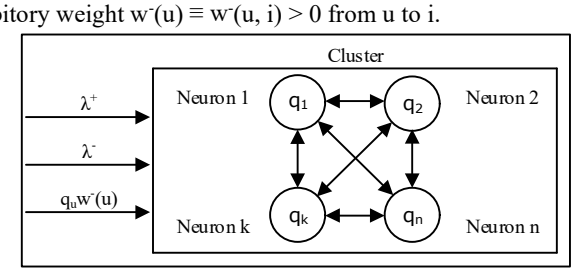


Fig. 8. Clusters of Neurons

The Deep Learning Architecture is composed of C multiple clusters, each of which is made up of an M(n) cluster each with n hidden neurons (Fig 9). For the c-th such cluster, c = 1, ..., C, the state of each of its identical neurons is denoted by q_c. In addition, there are U input neurons which do not belong to these C clusters, and the state of the u-th neuron u=1, ..., U is denoted by q_u. The cluster network has U input neurons and C clusters. The Deep Learning clusters model defines:

- $I = (i_1, i_2, \dots, i_u)$, a U-dimensional vector I Є $[0,1]^U$ that represents the input state for the neuron u;
- $w^-(u,c)$ is the U x C matrix of weights from the U input neurons to the neurons in each of the C clusters;
- $Y = (y_1, y_2, \dots, y_c)$, a C-dimensional vector Y Є $[0,1]^C$ that represents the neuron state q_c for the cluster c.

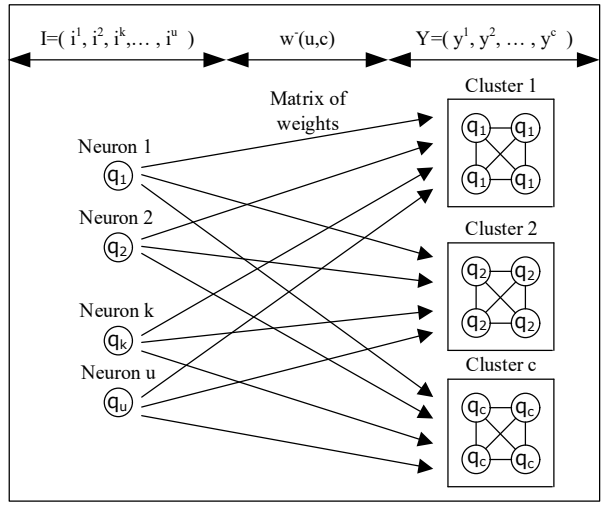


Fig. 9. The Random Neural Network with multiple clusters

The network learns the U x C weight matrix $w^-(u,c)$ by calculating new values of the network parameters for the input I and output Y using Gradient Descent learning algorithm which optimizes the network weight parameters $w^-(u,c)$ from a set of input-output pairs (i_u, y_c).

3.3.3. Genetic Learning Algorithm

The Genetic learning algorithm is based on an ELM autoencoder that models the genome as it codes the replica of the organism that contains it [74-75]. It consists of two instances of the Network described in the previous section. Network 1 is formed of U input neurons and C clusters and Network 2 has C input neurons and U clusters (Fig 8). The organism is represented as a set of data X which is a U vector X Є $[0,1]^U$. The Genetic Learning algorithm fixes C to 4 Genetic clusters that represent the four different nucleoids G, C, A and T and it also fixes W_1 to generate 4 different types of neurons rather than random values as proposed by the ELM theory. Network 1 encodes the organism is defined as:

- $q_1=(q^1_1, q^1_2, \dots, q^1_u)$, a U-dimensional vector q_1 Є $[0,1]^U$ that represents the input state q_u for neuron u corresponding to the iManagement prediction neuron q_{pc} and decision q_{dc} neurons;
- W_1 is the U x C matrix of weights $w_1^-(u,c)$ from the U input neurons to the neurons in each of the C clusters corresponding to the iBuilding genes;
- $Q^1=(Q^1_1, Q^1_2, \dots, Q^1_c)$, a C-dimensional vector Q^1 Є $[0,1]^C$ that represents state q_c for the Genetic cluster c where $Q^1=\zeta(W_1X)$.

Network 2 decodes the genome, as the pseudo inverse of Network 1, it is defined as:

- $q_2=(q^2_1, q^2_2, \dots, q^2_c)$, a C-dimensional vector q_2 Є $[0,1]^C$ that represents the input state q_c for neuron c with the same value as $Q^1=(Q^1_1, Q^1_2, \dots, Q^1_c)$;
- W_2 is the C x U matrix of weights $w_2^-(c,u)$ from the C input neurons to the neurons in each of the U neurons;
- $Q^2=(q^2_1, q^2_2, \dots, q^2_u)$, a U-dimensional vector Q^2 Є $[0,1]^U$ that represents the state q_u for the neuron u where $Q^2=\zeta(W_2Q^1)$ or $Q^2=\zeta(W_2\,\zeta(XW_1))$.

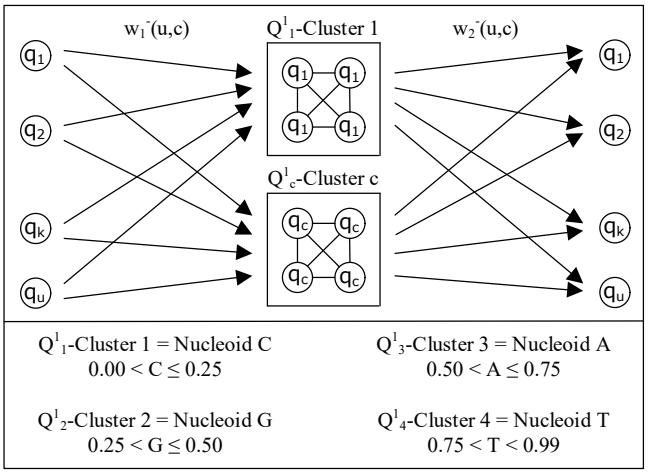


Fig. 10. Genetic Learning Algorithm

The learning algorithm is the adjustment of W_1 to code the organism X into the four different neurons or nucleoids and then calculate W_2 so that resulting decoded organism Q_2 is the same as the encoded organism X:

$$\min\left\|X-\zeta\left(W_2\zeta(XW_1)\right)\right\| \text{ s.t. } W_1 \geq 0 \text{ } (W_1 \text{ positive definite}) \tag{4}$$

Following the Extreme Learning Machine model; W_2 is calculated as:

$$\zeta(XW_1)W_2 = X \tag{5}$$

we have:

$$W_2=\text{pinv}\left(\zeta(XW_1)\right)X \tag{6}$$

where pinv is the Moore-Penrose pseudoinverse:

$$\text{pinv}(x)=(x^Tx)x^T$$

4 IBuilding Validation

IBuilding is validated as a proof of concept with a research dataset [24] based on a house with electric metering with M-BUS energy counters that measures the energy consumption of appliances, electric baseboard heaters and lighting (Fig 11). The house temperature and humidity were monitored with a ZigBee wireless sensor network located in 9 different zones, in addition, the temperature and humidity of an external weather station are also included. Information was collected every ten minutes for 137 days (4.5 months), from 11/01/2016 17:00:00 to 27/05/2016 18:00:00 with 19736 measurements in total.

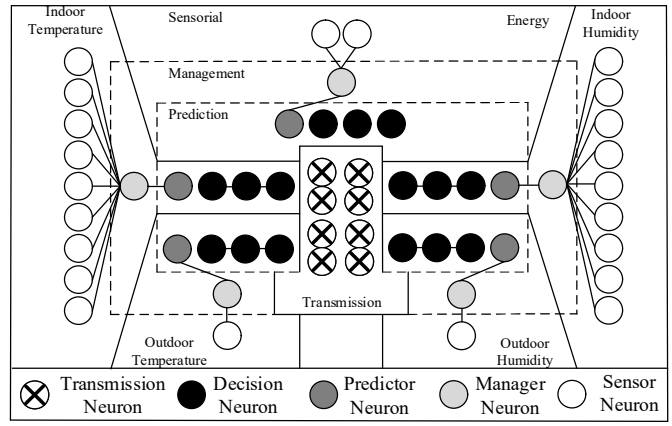


Fig. 11. iBuilding Validation

IBuilding involves a network of 22 distributed sensor neurons. The key five variables measured by the sensor neurons are energy consumption, internal and external temperature and humidity respectively (Table 1).

Table 1. iBuilding Sensorial Neurons

Variable	Type	Unit	Sensorial Neurons
Appliances	Electricity	Wh	1
Lights	Electricity	Wh	1
Internal Temperature	Environment	°C	9
Internal Humidity	Environment	%	9
External Temperature	Environment	°C	1
External Humidity	Environment	%	1

4.1 iSensor validation – Device layer

4.1.1 Sensor neuron validation

LSTM networks that predict the measurement of each sensor neuron are validated where P and S are the same figure. The quality of the prediction LSTM network is compared against traditional commercial spreadsheet software forecast formulas based on Linear Regression (LR) equations:

$$y=a+bx \tag{1}$$

where the constant *a* represents the intercept:

$$a=\bar{y}-b\bar{x} \tag{2}$$

and the *b* coefficient represents the inclination of the line:

$$b=\frac{\sum(x-\bar{x})\,(y-\bar{y})}{\sum(x-\bar{x})^2}$$

(3)

The performance of the predictor models is calculated via the Root Mean Square Error (RMSE):

$$\text{RMSE}=\sqrt{\sum_{i=1}^{N}\frac{(x_i-y_i)^2}{N}} \tag{4}$$

where x_i are the predicted values, y_i are the measured values and N the total number of measurements. Table 2 shows the RMSE for the LSTM network and Linear Regression (LR) models for the next value on the time series that correspond to the 22 distributed sensor neurons.

Table 2. Sensor Neuron – Next prediction value – Average RMSE

Variable	Type	5 Cells	10 Cells	20 Cells
Appliances Energy (Wh)	LSTM	8.06E+01	8.03E+01	8.02E+01
	LR	1.02E+02	9.55E+01	9.14E+01
Lights Energy (Wh)	LSTM	5.43E+00	5.40E+00	5.38E+00
	LR	6.01E+00	5.99E+00	6.39E+00
Kitchen Temp (°C)	LSTM	6.10E-02	6.11E-02	6.16E-02
	LR	6.16E-02	8.67E-02	1.63E-01
Kitchen Hum (%)	LSTM	6.36E-01	6.33E-01	6.32E-01
	LR	8.78E-01	1.03E+00	1.23E+00
Living Room Temp (°C)	LSTM	1.32E-01	1.33E-01	1.35E-01
	LR	1.40E-01	2.29E-01	4.35E-01
Living Room Hum (%)	LSTM	3.60E-01	3.62E-01	3.63E-01
	LR	4.75E-01	6.63E-01	9.72E-01
Laundry Temp (°C)	LSTM	8.87E-02	8.91E-02	8.95E-02
	LR	1.16E-01	1.71E-01	2.50E-01
Laundry Hum (%)	LSTM	2.55E-01	2.55E-01	2.55E-01
	LR	3.65E-01	5.02E-01	6.26E-01
Office Room Temp (°C)	LSTM	1.06E-01	1.07E-01	1.07E-01
	LR	1.42E-01	2.05E-01	2.94E-01
Office Room Hum (%)	LSTM	2.18E-01	2.19E-01	2.19E-01
	LR	2.97E-01	4.20E-01	5.71E-01
Bathroom Temp (°C)	LSTM	1.50E-01	1.50E-01	1.49E-01
	LR	1.99E-01	2.23E-01	2.84E-01
Bathroom Hum (%)	LSTM	2.81E+00	2.82E+00	2.81E+00
	LR	3.84E+00	4.78E+00	6.01E+00
Outdoor Temp (°C)	LSTM	3.24E-01	3.25E-01	3.27E-01
	LR	3.67E-01	5.22E-01	9.01E-01
Outdoor Hum (%)	LSTM	1.71E+00	1.71E+00	1.72E+00
	LR	2.02E+00	2.99E+00	4.79E+00
Ironing Room Temp (°C)	LSTM	5.52E-02	5.51E-02	5.56E-02
	LR	6.02E-02	8.36E-02	1.53E-01
Ironing Room Hum (%)	LSTM	2.27E-01	2.26E-01	2.26E-01
	LR	2.80E-01	3.90E-01	5.96E-01
Teenager Room Temp (°C)	LSTM	6.37E-02	6.39E-02	6.44E-02
	LR	6.75E-02	9.64E-02	1.70E-01
Teenager Room Hum (%)	LSTM	2.86E-01	2.88E-01	2.89E-01
	LR	3.38E-01	5.13E-01	8.23E-01
Parents Room Temp (°C)	LSTM	5.30E-02	5.29E-02	5.29E-02
	LR	7.24E-02	8.67E-02	1.02E-01
Parents Room Hum (%)	LSTM	2.52E-01	2.53E-01	2.55E-01
	LR	3.14E-01	4.51E-01	7.13E-01
External Temp (°C)	LSTM	2.20E-01	2.21E-01	2.23E-01
	LR	2.25E-01	3.73E-01	6.71E-01
External Hum (%)	LSTM	1.21E+00	1.23E+00	1.24E+00
	LR	1.39E+00	2.29E+00	3.86E+00
Average values	LSTM	4.33E+00	4.31E+00	4.31E+00
	LR	5.42E+00	5.35E+00	5.52E+00

Figure 12 shows the average RMSE for the LSTM and LR models. Similar to the previous validation, the number of memory cells does not affect the quality of the prediction for the LSTM model. A larger memory size increases the RMSE of the LR model, the opposite effect as the previous validation. LSTM outperforms LR on this validation.

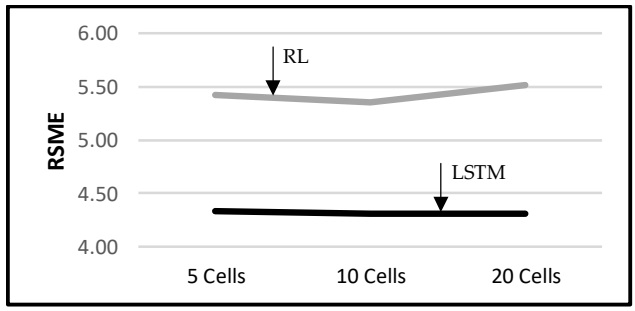


Fig. 12. Sensor Neuron- Next Prediction Value- Average RMSE.

Figure 13 shows the next predicted value of the appliances energy variable (Wh) for a window of ten data points.

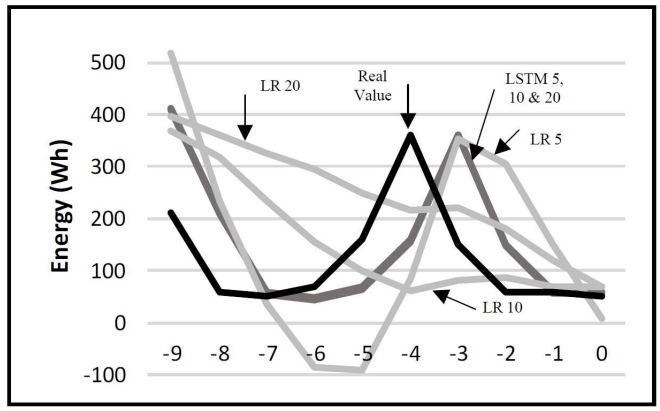


Fig. 13. iSensor - Next Prediction Values.

Table 3 shows the RMSE for the predictions for the next ten time series related to the appliances energy variable (Wh) only. The error of the LSTM predictions remains constant and independent from the number of neurons. A medium memory size provides optimum predictions in the LR model with RMSE values similar to the LSTM model.

Table 3. Sensor Neuron – Next ten prediction values – RMSE

Time	Type	5 Cells	10 Cells	20 Cells
t=1	LSTM	9.83E+00	6.69E+00	9.39E+00
	LR	1.61E+02	4.85E+01	1.38E+01
t=2	LSTM	9.75E+00	5.24E+00	9.13E+00
	LR	1.88E+02	6.07E+01	7.28E+00
t=3	LSTM	7.94E+00	5.90E-01	6.25E+00
	LR	2.44E+02	4.53E+01	2.01E+01
t=4	LSTM	8.20E+00	1.34E+01	6.62E+00
	LR	3.32E+02	7.57E+00	4.56E+01
t=5	LSTM	2.85E+00	5.92E+00	2.52E-01
	LR	3.84E+02	1.89E+01	5.23E+01
t=6	LSTM	4.02E+00	2.20E+00	5.93E+00
	LR	4.24E+02	3.58E+01	7.14E+01
t=7	LSTM	2.15E+01	2.06E+01	2.20E+01
	LR	4.74E+02	8.45E+00	9.56E+01

t=8	LSTM	9.42E+00	8.20E+00	8.71E+00
	LR	5.47E+02	1.33E-01	1.17E+02
t=9	LSTM	1.24E+01	1. 38E+01	2.49E+01
	LR	6.26E+02	3.34E+01	1.48E+02
t=10	LSTM	2.40E+01	3.03E+01	2.79E+01
	LR	6.98E+02	6.04E+01	1.64E+02

Figure 14 and Figure 15 show the predicted and real appliances energy measurements (Wh) for the next ten prediction with the associated RMSE respectively.

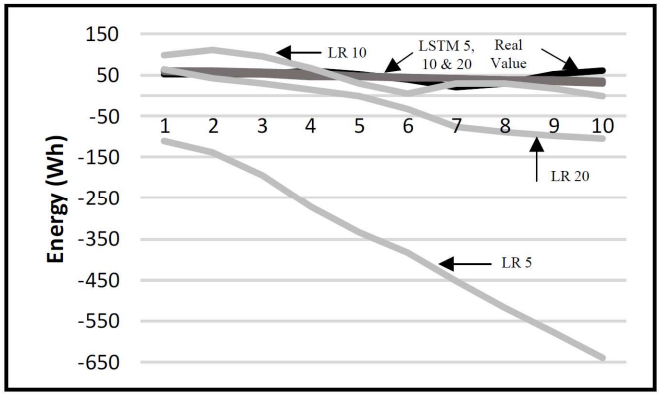


Fig. 14. Sensor Neuron - Next Ten Prediction Values.

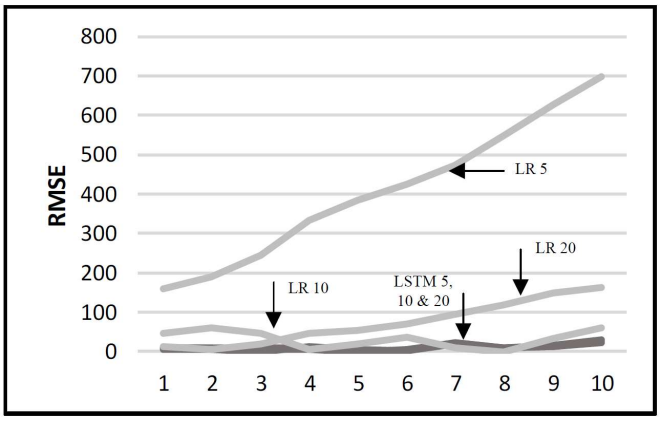


Fig. 15. Sensor Neuron- Next Ten Prediction Values - RMSE.

The number of memory cells does not have a great impact on the prediction accuracy due to its time series model. On the other hand, the memory size has a large impact on the prediction quality for the Linear Regression model. The LSTM model has performed consistently although the average RMSE values have a significant difference between each due to the normalization of the variables.

4.1.2 Manager neuron validation

IBuilding consists of a configuration of five iSensor networks, therefore five manager neurons associated with the energy consumption, indoor and outdoor temperature and humidity variables. The manager neuron collects the potential of their respective sensor neurons. On this validation, w(u,c) is fixed in order to either add the energy or make the average of the temperature and humidity sensors. iSensor parameters are shown in Table 4.

Table 4. iSensor Parameters

iSensor	Measurement	Sensor I u	w(u c)	Sensor Y c
1	Energy (W)	2	(1.0,1.0)	1
2	Indoor Temperature (C)	9	(1/9,1/9,1/9,1/9,1/ 9,1/9,/1/9,1/9,1/9)	1
3	Indoor Humidity (H)	9	(1/9,1/9,1/9,1/9,1/ 9,1/9,/1/9,1/9,1/9)	1
4	Outdoor Temperature (C)	1	(1.0)	1
5	Outdoor Humidity (H)	1	(1.0)	1

Figure 16 shows the value of the five iSensor neuron managers q_{mc} for the complete dataset (top) and zoomed in to only one day (bottom).

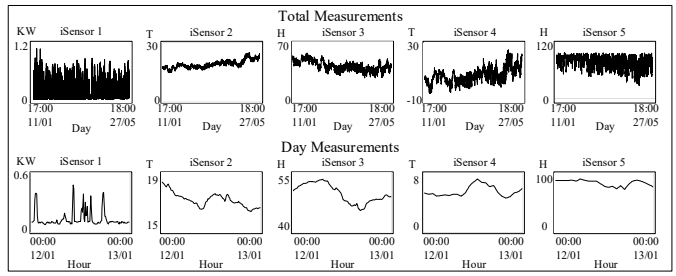


Fig. 16. iSensor Validation

ISensors gather information from the predicted measurements and merge it into a single value. As expected, the value of the q_{mc} for Temperature and Humidity follows a continuous trend however, the q_{mc} for energy consumption is alternating. Table 5 shows the average values for the 19736 measurements with the Standard Deviation σ and 95% Confidence Range.

Table 5. iSensor Validation Overall Average Values

iSensor	Measurement	Value	σ	95% CR
1	Energy (W)	101.5453	104.6441	1.4599
2	Indoor Temperature (C)	19.3818	2.1455	0.0299
3	Indoor Humidity (H)	42.7096	5.9615	0.0832
4	Outdoor Temperature (C)	7.4120	5.3195	0.0742
5	Outdoor Humidity (H)	79.7486	14.9063	0.2080

As expected, outdoor temperature and humidity measurements have more deviation that the indoor values as these can be adjusted by the user. The energy consumption follows a dispersed pattern therefore its standard deviation is quite large. Table 6 shows the average values with the Standard Deviation σ and 95% Confidence Range for every day at 21:00 hours, these represent 137 measurements for each iSensor. The 21:00 hours was chosen as the users are normally at home therefore values are expected to be consistent.

Table 6. iSensor Validation Average Values

iSensor	Measurement	Value	σ	95% CR
1	Energy (W)	122.0438	79.3460	13.287

2	Indoor Temperature (C)	19.6128	1.9512	0.3267
3	Indoor Humidity (H)	42.8909	6.1288	1.02628
4	Outdoor Temperature (C)	7.5956	5.2246	0.8749
5	Outdoor Humidity (H)	79.2554	12.4150	2.0789

The standard deviation decrease when the measurements are taken on the same time slot rather than accounting all measured values except for the indoor humidity. The main reason is indoor humidity does not greatly fluctuate (iSensor 3 Fig 16) therefore a reduced sample increases its standard deviation.

4.2 IManagement validation – Edge layer

As there are five iSensor Networks, correspondingly there are associated five iManagement networks that take trend decisions and value predictions based on Deep Reinforcement Learning algorithm. IManagement has been validated with several memory parameters covering the rate of sampling and the duration of memory as represented in Table 7.

Table 7. Deep Reinforcement Learning Memory Parameters

Type	Memory	Description
MP1-0M	t=l-1	Learning only applies to the previous day
MP2-FM	t=0	Learning applies from day 1
MP3-1D	t=l-1-144	Learning applies only last day
MP4-7D	t=l-1-(144×7)	Learning applies only last week
MP5-DD	t=Δ144	Learning applies the same time for all previous days
MP6-WW	t=Δ(144×7)	Learning applies the same time for all previous weeks

The quantity of Rewards (R) or successful decisions, Penalizations (P) or wrong decisions, and Accuracy (A) for the different iManagement networks are shown in Table 8. This covers the 19736 data measurements. Table 8 only represent MP1-0M No Memory and MP2-FM Full Memory parameters with a medium Threshold Memory (α=0.5) and medium Learning Gradient (β =1K) for simplicity.

Table 8. iManagement Validation: Trend Decision Neuron

Variable	DRL	
	MP1-0M	**MP2-FM**
Energy	R: 6503 P: 13233 A:32.95%	R: 6624 P: 13112 A:33.56%
Indoor Temperature	R: 14190 P: 5546 A: 71.90%	R: 14640 P: 5096 A:74.18%
Indoor Humidity	R: 14208 P: 5528 A: 71.99%	R: 14734 P: 5002 A: 74.66%
Outdoor Temperature	R: 17606 P: 2130 A: 89.21%	R: 18016 P: 1720A: 91.28%
Outdoor Humidity	R: 17187 P: 2549 A: 87.08%	R: 17524 P: 2212 A: 88.79%
Total Values	R: 69694 P: 28986 A:70.63%	R: 71538 P: 27142 A:72.49%

Energy iManagement provides worse results due to its alternating values. The introduction of DRL algorithm provides a moderate increment of accuracy although the values between different memory parameters are not highly dissimilar (Fig. 17). Figures represent the six different types with variations on the Threshold Memory (α) and medium Learning Gradient (β).

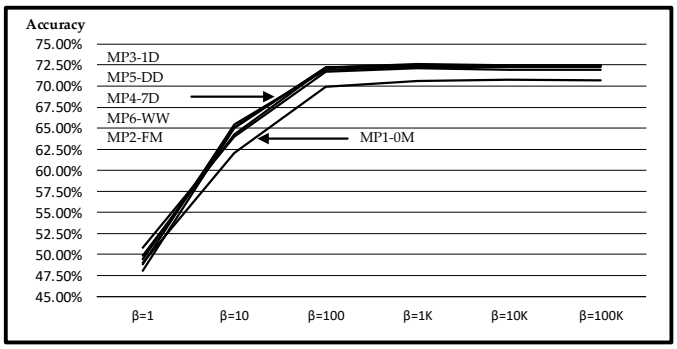


Fig. 17. iManagement Validation – β value

The Learning Gradient β has an impact in the accuracy where its optimum value is 1K whereas the Threshold Memory α has not a great impact in the accuracy, although it peaks at 0.25 value (Fig. 18).

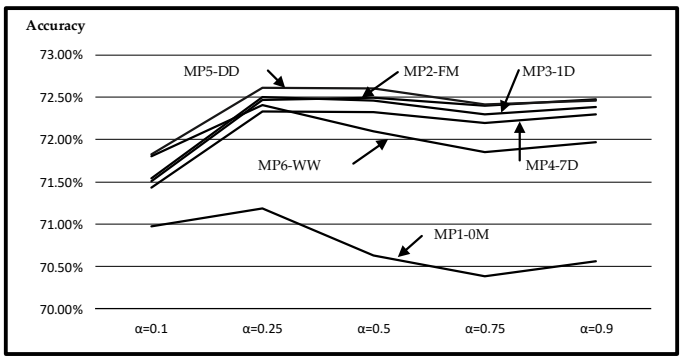


Fig. 18. iManagement Validation – α value

The root-mean-square error of the predicted values by the iManagement Predictor Neuron against the real measurements for the MP1-0M No Memory and MP2-FM Full Memory parameters respectively are shown in Table 9. These figures cover the 19736 data measurements with a low Prediction Memory (γ =0.1) for simplicity.

Table 9. iManagement Validation: Value Predictor Neuron

Variable	DRL	
	MP1-0M	**MP2-FM**
Energy	5.14E-01	7.42E-01
Indoor Temperature	3.34E-04	1.26E-02
Indoor Humidity	2.36E-03	3.49E-02

Outdoor Temperature	1.23E-03	3.43E-02
Outdoor Humidity	6.85E-03	1.03E-01
Total Values	5.25E-01	9.27E-01

The addition of DRL algorithm does not reduce the value prediction error. Figure 19 shows the total error values for different prediction memories and DRL memory parameters.

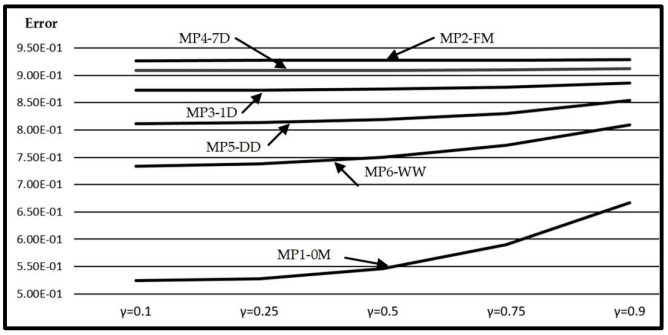


Fig. 19. iManagement Validation – γ value

The smallest prediction memory generates the lowest error. Figure 20 represents the iManagement 1 energy Value Predictor Neuron q_{pc} for only one day (144 data measurements) covering MP1-0M No Memory and MP2-FM Full Memory across different values of γ where γ=0.1 provides the closest prediction to the real measurement.

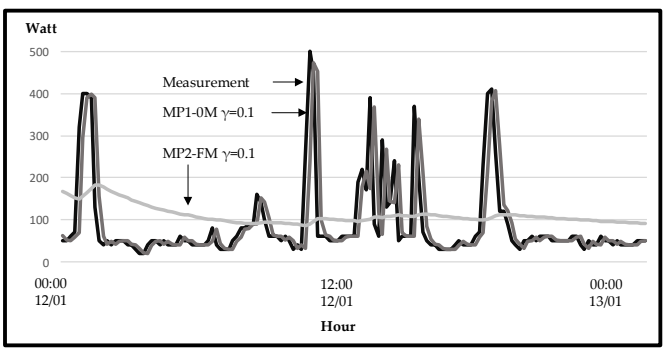


Fig. 20. iManagement Validation: value predictor – Deep Reinforcement Learning

4.3 ITransmission validation – Cloud layer

4.3.1 Specific validation

iTransmission Genetic Algorithm validation for the four different Nucleoids (C, G, A, T) during the 19736 data points is shown in Table 10 with the Random Neural Network Genetic Algorithm RMSE Error, the value of the different Genetic Clusters or Nucleoids, iteration and time. The results for the Smart House validation are shown in Table 2, including its statistical information where feasible. The number of sensorial neurons distributed in the physical building is 22, generating a total of 88 iBuilding genes.

Table 10. iTransmission specific validation

Variable	Value	σ	95% CR
Error	2.29E-31	2.50E-31	3.49E-33
Nucleoid-C	0.2138	1.57E-04	2.19E-06
Nucleoid-G	0.4022	2.25E-04	3.13E-06
Nucleoid-A	0.6420	2.15E-04	3.00E-06
Nucleoid-T	0.9304	6.02E-05	8.39E-07
Iteration	1.00	0.00	0.00
Time (ns)	1.27E+04	1.20E+04	1.68E+02

The proposed Genetic Algorithm successfully codifies iBuilding 22 Sensorial neurons and transmits its 88 genes information to the next generations with a residual error with only one iteration at a reduced time. The Standard Deviation σ and 95% Confidence Range values proof that the results are statistically significant (Fig 21).

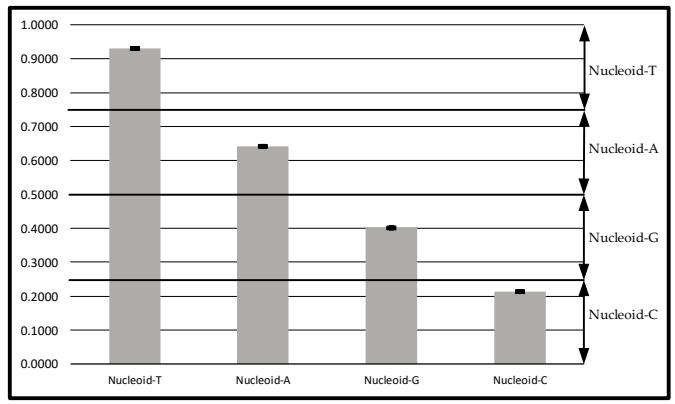


Fig. 21. iTransmission specific validation

4.3.2 General validation

This section presents the general validation of iTransmission where sensorial neurons increase gradually from 1 to 100,000, or equivalently Building genes grow from 4 to 400,000. The potential value of the sensor neuron is normalised at 0.5. The results for the general validation are shown in Table 11, including its statistical information:

Table 11. iTransmission general validation

Variable	Value	σ	95% CR
Error	1.15E-28	1.84E-28	2.54E-30
Nucleoid-C	0.214	8.45E-14	1.17E-15
Nucleoid-G	0.403	1.63E-13	2.26E-15
Nucleoid-A	0.643	2.33E-13	3.23E-15
Nucleoid-T	0.931	1.19E-13	1.64E-15
Time (ns)	1.93E+06	5.46E+06	7.56E+04

Figure 22 shows the error between the iBuilding replica generated from the iBuilding genes against the physical building.

There are some sparks, or genetic mutations, due to the rounding and precision of the gene's values.

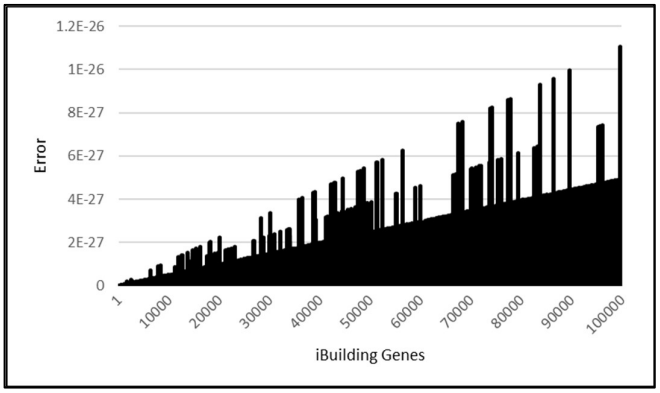


Fig. 22. iTransmission General validation – iBuilding Genes – Error

Figure 23 represents the time on nanoseconds (ns) to codify the physical building into iBuilding genes and decodify its replica from its genes. The multiplexing time of iBuilding is limited by the computer CPU instead of the gradual algorithm complexity due to the larger number of genes.

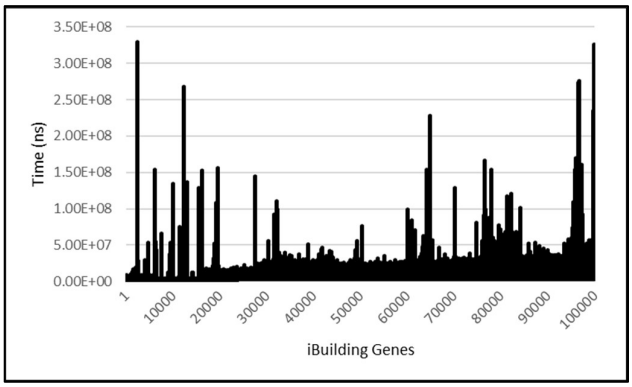


Fig. 23. General validation - iBuilding Genes – Time (ns)

Figure 24 provides the average neuron state, q_c, when the sensorial neurons increase gradually from 1 to 100,000. Statistical information is also shown, although the 95% confident range is almost negligible. The neuron state q_c is also independent of the number of sensorial neurons, or the number of buildings, enabling iBuilding to successfully codify and multiplex any large amount of data.

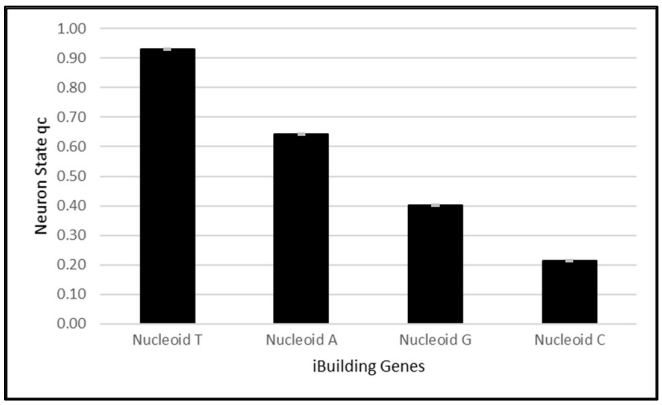


Fig. 24. General validation - iBuilding Genes – Neuron State q_c

The general iTransmision validation confirms the results obtained on previous experiments: the time to codify iBuilding genes increases linearly, although the codification error remains almost constant, therefore independent to the number of iBuilding genes.

4.3.3 Design Considerations

This section analyses the optimum number of nucleoids or clusters for different 10xnx10 iTransmission networks, where n = 1, 2, 4, 10, 20, 50, 100, represent the nucleoids, Q^1_c, that will codify the organism with a minimum error. In addition, the error in the organism when the Genome varies or mutates is examined in this section. The value for Q^1_c is gradually changed with a ± Δ0.001 deviation where the Genetic Algorithm error is measured. The overall error results for the different networks are shown in Table 12.

Table 12. Genetic Algorithm Overall Error

Deviation	± Δ0.10	± Δ0.08	± Δ0.06	± Δ0.04	± Δ0.02	± Δ0.00
10x1x10	7.65E-01	4.90E-01	2.76E-01	1.22E-01	3.06E-02	1.23E-32
10x2x10	1.91E-01	1.22E-01	6.89E-02	3.06E-02	7.65E-03	1.23E-32
10x4x10	4.78E-02	3.06E-02	1.72E-02	7.65E-03	1.91E-03	1.23E-32
10x10x10	7.65E-03	4.90E-03	2.76E-03	1.22E-03	3.06E-04	1.02E-30
10x20x10	1.91E-03	1.22E-03	6.89E-04	3.06E-04	7.65E-05	5.18E-31
10x50x10	3.06E-04	1.96E-04	1.10E-04	4.90E-05	1.22E-05	1.23E-30
10x100x10	7.65E-05	4.90E-05	2.76E-05	1.22E-05	3.06E-06	4.62E-30

The iTransmission incremental and overall error are shown in Figure 25 and Figure 26.

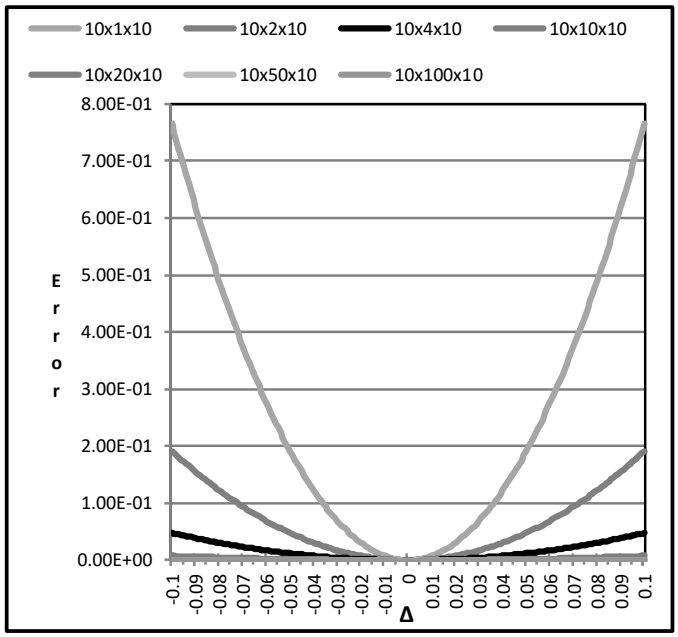


Fig. 25. Genetic Algorithm incremental error

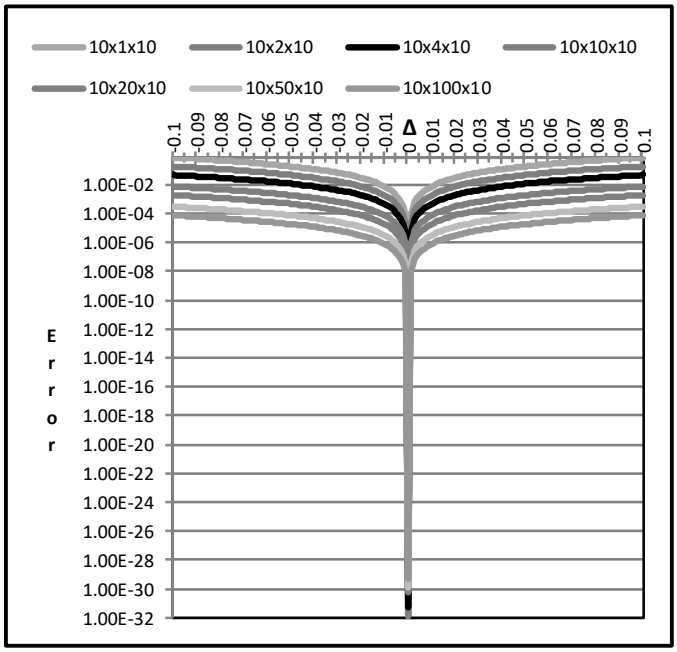


Fig. 26. Genetic Algorithm incremental error – Log Scale

The optimum value between the number of nucleoids, or biological cost, and error to codify an organism or information, is actually four as shown in Figure 27. A minor mutation or variation in a single or double Nucleoid Network has a major error impact in the organism whereas the resilience against mutations or errors achieved with large Nucleoids Genetic Networks from n =10, 20, 50 and 100 does not largely increase after four nucleoids.

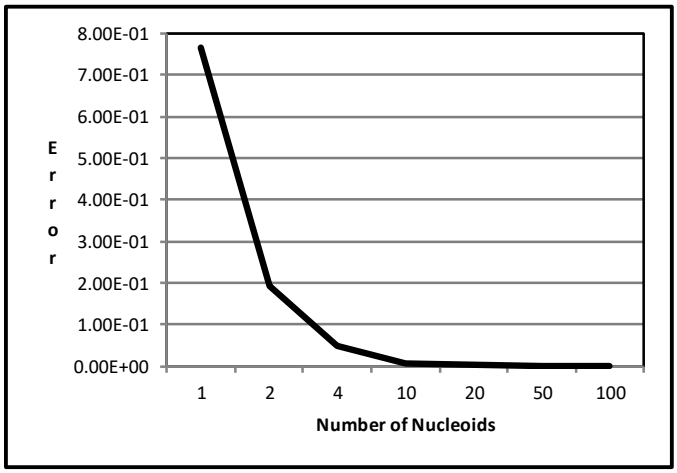


Fig. 27. Genetic Algorithm Overall Error

5 Conclusions

This research has proposed iBuilding: Artificial Intelligence embedded into Intelligent Buildings that enable their adaptation to the external environment, learning from its users and monitoring its functionality in terms of assets, space and energy, therefore, assisting building managers or developers to make commercial or operational decisions.

Sensorial neurons are dispersed through the Building at the device level to gather and filter building environment information whereas Management Sensors based on Deep Reinforcement Learning algorithm at edge level make predictions about values and trends (upwards, downwards and equal) that enable the Intelligent Building to adapt to the future demand of its space, environmental conditions or energy. Finally, the Random Neural Network Genetic Algorithm based on the Genome codifies and transmits Intelligent Building information; furthermore, it also multiplexes its data entirely to generate Clusters of Buildings interconnected with each other at cloud level.

The key concept proposed in this paper is the learned information that iBuilding obtains after its adaptation to the environment is never lost when it is decommissioned but transmitted to future generations. Data is codified in the network weights rather than the neurons; similar to the genome to enable an Artificial Intelligence evolution in iBuilding in distributed organisms.

The results provided show that a possible reason for which the genome is formed of four different nucleoids is that nature has optimized the way information is codified and transmitted with the right balance between biological cost, number of Nucleoids, and resilience to changes or mutations, or autoencoder error.

Compliance with ethical standards

Conflict of interest: The author declares that he has no conflict of interest against any company or institution.

Ethical standards: This research has not involved human participants and/or animals, except for the author contribution.

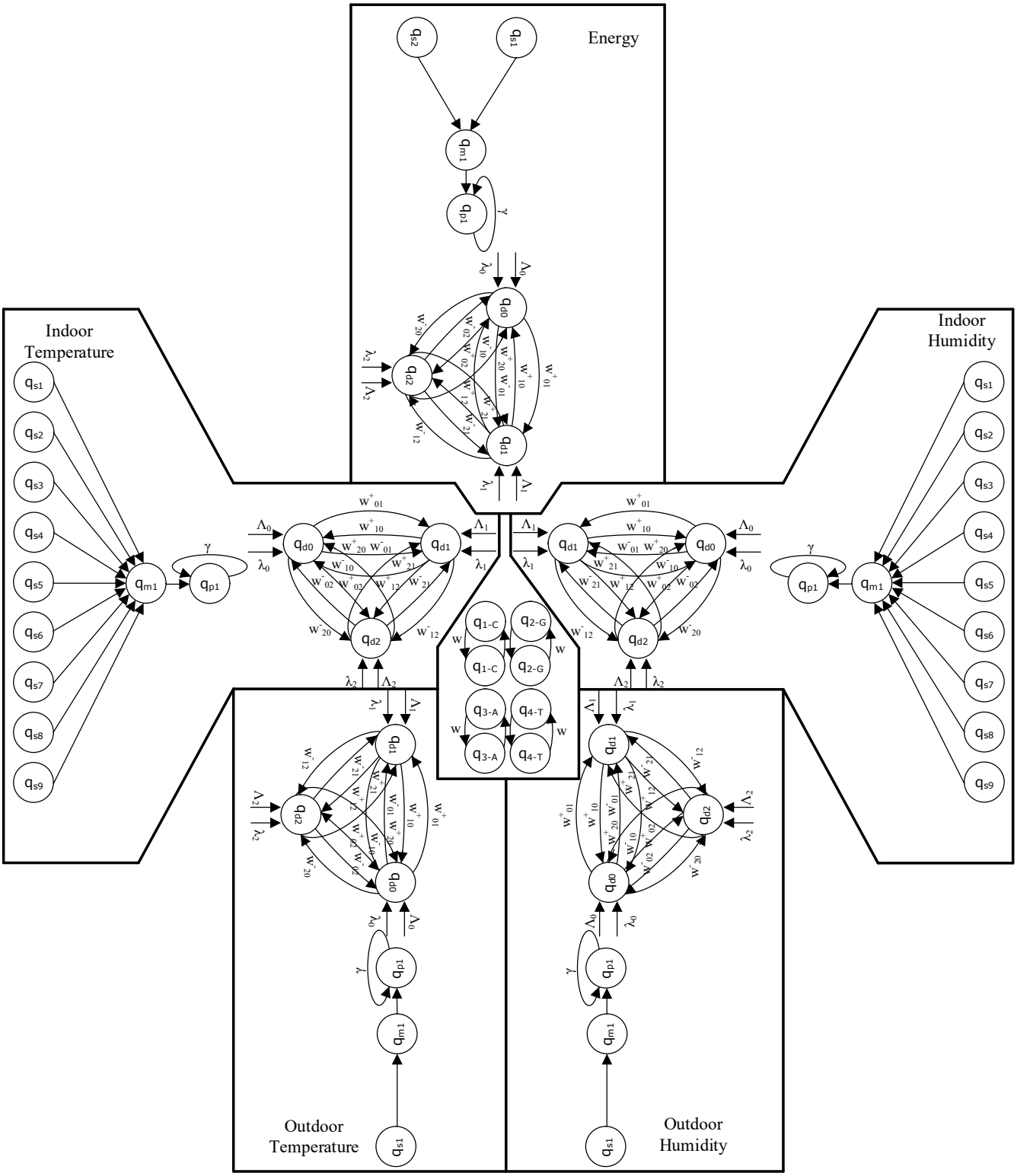
References

1. A. Ghaffarianhoseini, U. Berardi, H. AlWaer, S. Chang, E. Halawa, A. Ghaffarianhoseini, D. Clements What is an intelligent building? Analysis of recent interpretations from an international perspective. Architectural Science Review, 59, 5. (2016), 338-357
2. M. Kirschner, J. Gerhart. The Plausibility of Life Resolving Darwin's Dilemma. Yale University Press. (2005), 1-336
3. G. Hinton, S. Nowlan. How learning can guide evolution. Complex Systems. 1. (1996), 495-502
4. D. Smith, E. Bullmore. Small-World Brain Networks. The Neuroscientist, 12. (2007), 512-523
5. O. Sporns, D. Chialvo, M. Kaiser, C. Hilgetag. Organization, development and function of complex brain networks. Trends in Cognitive Sciences, 8, 9. (2004), 418-425
6. D. Clements. What do we mean by intelligent buildings?. Automation in Construction, 6, 5–6. (1997), 395-400
7. O. Omar. Intelligent building, definitions, factors and evaluation criteria of selection. Alexandria Engineering Journal, 57. (2018), 2903–2910
8. A. Leclercq, H. Isaac. The new office: how coworking changes the work concept. Journal of Business Strategy, 37, 6. (2016), 3-9
9. J.K. Hui, Y. Zhang. Sharing Space: Urban Sharing, Sharing a Living Space, and Shared Social Spaces. Space and Culture. (2018), 1–13
10. G.Yacoub. How do Collaborative Practices Emerge in Coworking Spaces? Evidence from Fintech Start-ups. Academy of Management Proceedings, 1. (2018), 1-10
11. T. Honga, S. Taylor-Langea, S. D'Ocab, D. Yanc, S. Corgnati. Advances in research and applications of energy-related occupant

behaviour in buildings. Energy and Buildings, 116. (2016), 694–702
12. O.T. Masoso, L.J. Grobler. The dark side of occupants' behaviour on building energy use. Energy and Buildings, 42. (2010), 173–177
13. T. Hong, D. Yan, S. D'Oca, C. Chen. Ten questions concerning occupant behaviour in buildings: The big Picture. Building and Environment, 114. (2017), 518-530
14. T. Nguyen, M. Aiello. Energy intelligent buildings based on user activity: A survey. Energy and Buildings, 56. (2013), 244–257
15. P. Shaikh, N. Nor, P. Nallagownden, I. Elamvazuthi, T. Ibrahim. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. Renewable and Sustainable Energy Reviews, 34. (2014), 409–429
16. F. Oldewurtel, D. Sturzenegger, M. Morari. Importance of occupancy information for building climate control. Applied Energy, 101. (2013), 521–532
17. T. Labeodan, W. Zeiler, G. Boxem, Y. Zhao: Occupancy measurement in commercial office buildings for demand-driven control applications—A survey and detection system evaluation. Energy and Buildings 93. (2015), 303–314
18. J.H. Ko, D.S. Kong, J.H. Huh. Baseline building energy modelling of cluster inverse model by using daily energy consumption in office buildings. Energy and Buildings, 140. (2017), 317–323
19. K.S. Cetina, P.C. Tabares-Velascob, A. Novoselac. Appliance daily energy use in new residential buildings: Use profiles and variation in time-of-use. Energy and Buildings, 84. (2014), 716–726
20. P. Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. Electrical Power and Energy Systems, 60. (2014), 126–140
21. Z. Wang, R.S. Srinivasan. A review of artificial intelligence-based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renewable and Sustainable Energy Reviews, 75. (2017), 796–808
22. A. Tsanas, A. Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy and Buildings, 49. (2012), 560–567
23. N. Arghira, L. Hawarah, S. Ploix, M. Jacomino. Prediction of appliances energy use in smart homes. Energy, 48. (2012), 128-134
24. L. M. Candanedo, V. Feldheim, D. Deramaix. Data-driven prediction models of energy use of appliances in a low-energy house. Energy and Buildings, 140. (2017), 81–97
25. H. Bi, E. Gelenbe. A Cooperative Emergency Navigation Framework Using Mobile Cloud Computing. International Symposium Computer and Information Sciences. (2014), 41-48
26. E. Gelenbe, H. Bi. Emergency Navigation without an Infrastructure. Sensors, 14, 8. (2014), 15142-15162
27. A. Ghosh, D. Chakraborty, A. Law. Artificial intelligence in Internet of things. Chinese Association for Artificial Intelligence. Transactions on Intelligence Technology, 3, 4. (2018), 208-218
28. M. Osifeko, G. Hancke, A. Mahfouz. Artificial Intelligence Techniques for Cognitive Sensing in Future IoT: State-of-the-Art, Potentials, and Challenges. Journal Sensors Actuator Networks, 9, 21. (2020), 1-31
29. J. Zhang. Real-time detection of energy consumption of IoT network nodes based on artificial intelligence. Computer Communications, 153. (2020), 188–195
30. F. Alama, R. Mehmoodb, I. Katiba, A. Albeshri. Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT). International Workshop on Data Mining in IoT Systems. Procedia Computer Science, 98. (2016), 437-442
31. K. Skouby, P. Lynggaard. Smart Home and Smart City Solutions enabled by 5G, IoT, AAI and CoT Services. International Conference on Contemporary Computing and Informatics. (2014), 874-878
32. J. Lloret, J. Tomas, A. Canovas, L. Parra. An Integrated IoT Architecture for Smart Metering. Internet of Things. IEEE Communications Magazine, 54, 12. (2016), 50-57
33. Abdulhafis Abdulazeez Osuwa, Esosa Blessing Ekhoragbon, Lai Tian Fat. Application of Artificial Intelligence in Internet of Things. International Conference on Computational Intelligence and Communication Networks. (2017), 169-173
34. M. Shafique, T. Theocharides, C. Bouganis, M. Hanif, F. Khalid, R. Hafiz, S. Rehman.An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era. Design, Automation and Test in Europe Conference and Exhibition. (2018), 827-832
35. A. Rego, A. Canovas, J. M. Jiménez, J. Lloret. An Intelligent System for Video Surveillance in IoT Environments. IEEE Access, 6. (2018), 31580-31598
36. L. Liu, B. Zhou, Z. Zou, S. Yeh, L. Zheng. A Smart Unstaffed Retail Shop Based on Artificial Intelligence and IoT. International Workshop on Computer-Aided Modelling and Design of Communication Links and Networks. (2018), 1-4
37. J. Zhou, Y. Wang, K. Ota, M. Dong. AAIoT: Accelerating Artificial Intelligence in IoT Systems. IEEE Wireless Communications Letters, 8, 3. (2019), 825-828
38. S. Soomro, M. Miraz, A. Prasanth, M. Abdullah. Artificial Intelligence Enabled IoT: Traffic Congestion Reduction in Smart Cities. Smart Cities Symposium. (2018), 1-6
39. I. Garcia, R. Muttukrishnan, J. Lloret. Human-Centric AI for Trustworthy IoT Systems with Explainable Multilayer Perceptrons. IEEE Access, 7. (2019), 125562-125574
40. V. Prutyanov, N. Melentev, D. Lopatkin, A. Menshchikov, A. Somov. Developing IoT Devices Empowered by Artificial Intelligence: Experimental Study. Global IoT Summit. (2019), 1-6
41. P. Gonzalez, J. Lloret, M. Taha. Architecture to Integrate IoT Networks using Artificial Intelligence in the Cloud. International Conference on Computational Science and Computational Intelligence. (2018), 996-1001
42. Z. Lv, Y. Han, A. Singh, G. Manogaran and H. Lv. Trustworthiness in Industrial IoT Systems Based on Artificial Intelligence. IEEE Transactions on Industrial Informatics, 17, 2. (2021), 1496-1504
43. P. Trakadas et all. An Artificial Intelligence-Based Collaboration Approach in Industrial IoT Manufacturing: Key Concepts, Architectural Extensions and Potential Applications. Sensors, 20, 19, 5480. (2020), 1-20
44. E. Adi, A. Anwar, Z. Baig, S. Zeadally. Machine learning and data analytics for the IoT. Neural Computing and Applications 32. (2020), 16205–16233
45. P. Pramanik, S. Pal, P. Choudhury. Beyond Automation: The Cognitive IoT. Artificial Intelligence Brings Sense to the Internet of Things. Cognitive Computing for Big Data Systems Over IoT. (2018), 1-37
46. L. Gladence, V. Anu, R. Rathna, E. Brumancia. Recommender system for home automation using IoT and artificial intelligence. Journal of Ambient Intelligence and Humanized Computing. (2020), 1-9
47. M. Chiang, T. Zhang. Fog and IoT: An Overview of Research Opportunities. IEEE Internet of Things Journal, 3, 6. (2016), 854-864
48. A. Sodhro, S Pirbhulal, V. Albuquerque. Artificial Intelligence-Driven Mechanism for Edge Computing-Based Industrial Applications. IEEE Transactions on Industrial Informatics, 15, 7. (2019), 4235-4243
49. H. Jiang. Mobile Fire Evacuation System for Large Public Buildings Based on Artificial Intelligence and IoT. IEEE Access, 7. (2019), 64101-64109

50. X. Chen, X. Hao. Feature Reduction Method for Cognition and Classification of IoT Devices Based on Artificial Intelligence. IEEE access, 7. (2019), 103291-103298
51. O. Debauche, S. Mahmoudi, S. Mahmoudi, P. Manneback, P. Manneback, F. Lebeau. A new Edge Architecture for AI-IoT services deployment. International Conference on Mobile Systems and Pervasive Computing. Procedia Computer Science, 175. (2020), 10–19
52. R. Ke, Y. Zhuang, Z. Pu, Y. Wang. A Smart, Efficient, and Reliable Parking Surveillance System with Edge Artificial Intelligence on IoT Devices. IEEE Transactions on Intelligent Transportation Systems. (2020), 1-13
53. S. Yang, K. Xu, L. Cui, Z. Ming, Z. Chen, Z. Ming. EBI-PAI: Towards An Efficient Edge-Based IoT Platform for Artificial Intelligence. IEEE Internet of Things Journal. (2020), 1-15
54. M. Hasan, M. Milon, M. Ishrak, Zarif, M. Hashem. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things, 7, 100059. (2019), 1-14
55. J. Cañedo, A. Skjellum. Using Machine Learning to Secure IoT Systems. Annual Conference on Privacy, Security and Trust. (2016) 219-222
56. L. Xiao, X. Wan, X. Lu, Y. Zhang, D. Wu. IoT Security Techniques Based on Machine Learning. IEEE Signal Processing Magazine, 35, 5. (2018), 41-49
57. F. Farivar, M. Sayad, A. Jolfaei, M. Alazab. Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT. IEEE Transactions on Industrial Informatics, 16, 4. (2020), 2716-2725
58. F. Hussain, R. Hussain, S. Hassan, E. Hossain. Machine Learning in IoT Security: Current Solutions and Future Challenges. IEEE Communications Surveys & Tutorials, 22, 3. (2020), 1686-1721
59. H. Fang, A. Qi and X. Wang. Fast Authentication and Progressive Authorization in Large-Scale IoT: How to Leverage AI for Security Enhancement. IEEE Network, 34, 3. (2020), 24-29
60. R. Ganesh, P. Karthika, V. Aravinda. Secure IoT Systems Using Raspberry Pi Machine Learning Artificial Intelligence. International Conference on Computer Networks and Inventive Communication Technologies. (2019), 797-805
61. K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi and M. Mustaqim. Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios. IEEE Access, 8. (2020), 23022-23040
62. D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, X. Du. From IoT to 5G I-IoT: The Next Generation IoT-Based Intelligent Algorithms and 5G Technologies. IEEE Communications Magazine. (2018), 114-120
63. N. Javaid, A. Sher, H. Nasir and N. Guizani. Intelligence in IoT-Based 5G Networks: Opportunities and Challenges IEEE Communications Magazine, 56, 10. (2018), 94-100
64. P. Kiss, A. Reale, C. J. Ferrari and Z. Istenes. Deployment of IoT applications on 5G edge. IEEE International Conference Future IoT Technologies. (2018), 1-9
65. S Hochreiter, J Schmidhuber - Long short-term memory. Neural computation, 9. (1997), 1735-1780
66. E. Gelenbe. Random Neural Networks with Negative and Positive Signals and Product Form Solution. Neural Computation, 1. (1989), 502-510
67. E. Gelenbe. Learning in the Recurrent Random Neural Network. Neural Computation, 5. (1993), 154-164
68. E. Gelenbe. Cognitive Packet Network. Patent US 6804201 B1. (2004)
69. E. Gelenbe, Z. Xu, E. Seref. Cognitive Packet Networks. International Conference on Tools with Artificial Intelligence. (1999), 47-54
70. G. Huang, Q. Zhu, C. Siew. Extreme learning machine: Theory and applications. Neurocomputing, 70, 1,3. (2006), 489-501
71. G. Huang, L. Chen, C. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks. 17, 4. (2006), 879-892
72. E. Gelenbe, Y. Yin. Deep learning with random neural networks. International Joint Conference on Neural Networks. (2016), 1633-1638
73. Y. Yin, E. Gelenbe. Deep Learning in Multi-Layer Architectures of Dense Nuclei. CoRR abs/1609.07160. (2016), 1-10
74. W. Serrano. The random neural network with a genetic algorithm and deep learning clusters in fintech: smart investment. IFIP International Conference on Artificial Intelligence Applications and Innovations. (2018), 297-310
75. W. Serrano. Genetic and deep learning clusters based on neural networks for management decision structures. Neural Computing and Applications. (2019), 1-25

Appendix


Energy
Indoor Temperature
Indoor Humidity
Outdoor Temperature
Outdoor Humidity