

Intelligent analysis of heterogeneous data in proactive energy management systems of smart objects

Olena Arsirii^{1,†}, Vasyly Arsirii^{2,†}, Andrii Zasidko^{1*,†} and Illia Krantovskyi^{1,†}

¹ Odesa Polytechnic National University, Shevchenko Ave. 1, 65044, Odesa, Ukraine

² Odesa State Academy of Civil Engineering and Architecture, Didrihsona str. 4, 65029, Odesa, Ukraine

Abstract

This article examines energy-state classification as a data-analytic component of proactive energy consumption management (PECM) for cyber-physical smart objects operating under changes in external-grid conditions. The study proposes a mathematical representation of the object's energy state and formulates state identification as a multi-criteria classification problem. The feature-formation method combines meter readings with weather, temporal, and building-context variables. Statistical features from sliding observation windows and K-Means clustering are used to form data-derived labels for three relative consumption states. Logistic Regression, SVM, Naive Bayes, Soft Voting, and Stacking classifiers are compared on a stratified sample of 60,000 daily aggregated records from Building Data Genome 2 (BDG2). Stacking provided the strongest results among the evaluated models (Accuracy = 64.58%, F1-score = 66.21%, AUC = 83.43%). The experiment evaluates the state-classification component only. The cost objective and the mapping from classified states to control actions remain conceptual and are not assessed for cost reduction, comfort, or grid-capacity compliance. The results indicate moderate feasibility for data-derived state identification, with the greatest overlap in the Medium class.

Keywords

Data Mining, Heterogeneous Data, Proactive Energy Management, Cyber-Physical Systems, Smart Objects

1. Introduction

The current state of the global energy sector, especially in the context of Ukraine's energy infrastructure, necessitates a fundamental transformation of energy consumption management methods at the end-user level. The growing instability of power grids, combined with the increase in the cost of energy resources and the share of stochastic renewable energy sources used (solar and wind generation), requires a transition to adaptive intelligent systems capable of self-regulation based on machine learning models [1]. This is critically important for smart facilities – a category covering hospitals, schools, commercial buildings, offices, residential units and apartment buildings, where simple shutdown of electrical loads is not possible due to operational requirements and comfort standards. Energy-management practice has developed from static consumption schedules toward flexible Demand Response (DR) strategies [2]. Traditional DR uses two main mechanisms: price-based response, in which dynamic tariffs encourage load redistribution, and incentive-based response, in which consumers receive payments for managed load reduction. Both mechanisms can support grid balancing, but their operation is usually triggered by a price signal, an operator request, or an emerging grid constraint.

ADT-ISMDDC 2026: Advanced Digital Technologies: Intelligent Systems, Modeling, and Data-Driven Control, September 22–24, 2026, Odesa, Ukraine.

* Corresponding author.

† These authors contributed equally.

✉ e.arsirii@gmail.com (O. Arsirii), vasyly.arsirii@gmail.com (V. Arsirii), andzasidko@gmail.com (A. Zasidko); illya20020814@gmail.com (I. Krantovskyi)

ORCID: 0000-0001-8130-9613 (O. Arsirii), 0000-0003-3617-8487 (V. Arsirii), 0000-0001-5197-0435 (A. Zasidko), 0009-0008-4801-2453 (I. Krantovskyi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The resulting response may arrive after the building load has already changed, and abrupt intervention can conflict with equipment schedules or occupant comfort. Proactive Energy Consumption Management (PECM) extends DR by estimating an upcoming shortage, peak, or change in operating conditions before a response is required [3]. A complete PECM system would combine data-driven state estimates with a multi-criteria optimization procedure. Such a controller could adjust energy-intensive equipment with reference to weather forecasts, facility schedules, behavioural patterns, tariffs, and technical limits. This broader architecture can make the facility an active participant in demand management rather than a passive recipient of grid signals. The present article addresses one prerequisite for that architecture: forming heterogeneous features and assigning them to data-derived energy-consumption states. It does not implement the subsequent controller.

The aim of this work is to develop and evaluate data-mining models for identifying energy-consumption states of smart objects from heterogeneous data. The resulting classifier is considered an information component of a broader PECM architecture rather than a complete energy controller.

2. Literature overview

The current state of the energy sector is characterized by a rapid increase in the share of renewable energy sources, decentralization of generation and the integration of new types of loads, such as charging stations for electric vehicles [2]. These factors cause high stochasticity and instability in distribution networks. The traditional consumption management paradigm, where generation changes according to the load, exhausts its potential due to the limited possibilities of maneuverable generation and the high cost of maintaining power reserves.

As noted in the works of researchers [4,5], the key tool for ensuring the sustainability of energy systems is a fundamental transformation of the role of the end consumer. Smart objects (in particular, residential and commercial buildings, which consume more than 40% of the world's electricity) should turn from passive consumers into active participants in balancing the energy system. This requires the implementation of intelligent Building Energy Management Systems (BEMS) capable of adapting load schedules to the current state of the grid in real time.

Historically, the energy management of buildings has been based on static scenarios, such as nighttime temperature drops, fixed time schedules for equipment operation, and the classic deviation adjustment rule. However, such approaches do not take into account the dynamics of external factors. An evolutionary step in solving this problem was the development of the concept of Demand Response (DR) [6]. DR strategies allow end users to adjust their load in response to market price signals, such as Real-Time Pricing (RTP) or direct requests from the system operator. The scientific literature describes in detail various models of DR, in particular: Direct Load Control, when the operator has the ability to forcibly turn off certain consumer devices during peaks, or Price-Based DR, where the consumer independently optimizes consumption, focusing on tariffs that change during the day [7].

Recent reviews [8] describe conventional DR models as predominantly reactive. Such systems respond to price changes or restrictions after they occur and may not account adequately for individual habits or thermal comfort. Fixed schedules and aggregate grid commands may also overlook differences between hospitals, offices, educational buildings, and residential facilities. Simultaneous reconnection of high-power equipment after a restriction period can create a new peak, known as the rebound effect. A model that identifies the facility's state before a control decision may reduce this timing problem, although the state estimate must still be linked to a separately validated control policy. These limitations motivate more flexible, human-oriented, and predictive management models [9].

The concept of Proactive Energy Consumption Management (PECM) addresses these limitations by basing decisions on estimates of future conditions in the smart object and its external environment [10].

The basic principle of PECM is to move from instant response to strategic planning of the smart object's behavior on a certain time horizon (for example, 24 hours in advance). To do this, the PECM system must operate not just with current physical indicators, but with latent states of the system. Assessment of these conditions makes it possible to initiate preventive actions in advance: to accumulate heat or electricity during periods of low tariffs, to gradually reduce the capacity of auxiliary systems before the onset of peak network load, and to flexibly manage resources without violating consumer comfort constraints [11,12].

PECM research is closely connected with Cyber-Physical Systems (CPS) and large heterogeneous datasets such as Building Data Genome 2 [13]. A complete PECM system involves two related tasks. The first is to identify or predict the state of a smart object from meter readings, weather, schedules, and other contextual variables. The inputs differ in units, sampling behaviour, temporal structure, and availability. They are also affected by weather and occupant behaviour, which may be difficult to represent with a single deterministic white-box building model. Data-driven classification can instead summarize the observed combinations of physical and contextual variables. Prior studies have applied methods ranging from linear models and SVM to ensemble methods such as XGBoost, Voting, and Stacking [14,15]. The second task begins after a state has been estimated. It selects a management strategy from the available actions and assesses the consequences of that action over a control horizon. In the wider PECM formulation, the objective can combine financial cost at a dynamic tariff, internal temperature or process-comfort requirements, and a maximum grid-capacity constraint [16]. Classification and optimization are connected because the estimated state can inform the controller, but they remain distinct methods. Each task requires its own variables, algorithm, baselines, and experimental criteria.

This article focuses on the first task: forming heterogeneous feature vectors and classifying data-derived energy-consumption states. The optimization layer is retained only to show how classification can connect to the broader PECM architecture; its algorithmic implementation and performance are outside the present experiment.

3. Research aim statement

The purpose of the study is to develop and evaluate data-mining models for the classification of smart-object energy states as part of a broader PECM framework. The study addresses the following tasks. To develop a mathematical model of a smart object's energy state from heterogeneous data and formulate state identification as a multi-criteria classification problem. To develop a feature-vector formation method that combines meter readings with dynamic contextual variables, including weather and schedules. To define the conceptual interface between the classified energy state and a subsequent cost-based control objective, without claiming implementation of the optimization layer. To evaluate the state-classification component using a daily aggregated sample from the Building Data Genome 2 (BDG2) dataset.

4. Development of a mathematical model of the energy state of a smart object based on intelligent analysis of heterogeneous data

Smart facilities include hospitals, schools, commercial buildings, offices, private residences, and apartment buildings. Their electrical loads cannot always be disconnected because equipment operation and occupant comfort impose constraints. Hospitals and educational facilities, for example, contain services whose operation cannot be reduced in the same way as discretionary residential loads. A PECM framework therefore requires a formal representation of the facility, its energy streams, and the external conditions associated with each observation. Deterministic models remain useful, but on their own they may not capture stochastic variation caused by weather, schedules, equipment use, and consumer behaviour.

We propose a mathematical model of a smart object's energy state based on the analysis of heterogeneous data. The model places state identification in a multi-criteria classification

framework: at each time point, the state is inferred from meter readings and contextual variables rather than from one consumption measurement. This representation allows energy carriers, weather, calendar variables, and externally defined economic signals to be considered in one feature space after normalization and encoding. Statistical summaries over an observation window describe both consumption level and recent variation. The classification stage then produces a discrete state label that can be passed to another PECM component. It provides structured information for a possible downstream control objective, but it does not itself choose equipment actions, minimize operating cost, or enforce comfort and capacity constraints.

A smart object is represented as a cyber-physical system (CPS) with a physical layer comprising engineering networks, equipment, environmental systems, and sensors, and a cyber layer comprising data-analysis and control algorithms. The physical layer produces electricity, heating, cooling, steam, and water measurements together with information about the condition of equipment. The cyber layer aligns these measurements with weather and calendar context, constructs features, and estimates the current state. Physical measurements are therefore converted into digital streams that support state assessment and, in a complete implementation, subsequent control. Unlike an isolated automation rule, this representation allows the same physical load to be interpreted in relation to changing external conditions. We formalize the smart object as follows:

$$O=\{P, C, A\}, \quad (1)$$

where P represents the structural and physical state of the equipment, C is the contextual environment, including weather and externally supplied tariff information, and A is the set of possible management actions. The experiment reported in this article uses physical and contextual data to classify energy states; it does not generate or evaluate the actions in the third component.

Let the functioning of a smart object over time t , as formalized in (1), be described by a set of states S that combine readings from physical devices P and the contextual environment C . At each time t , we then obtain a system snapshot: $x_t=\{p_t, c_t\} \in \mathbb{R}^m$.

System states are given by time intervals (or discrete time points) for which we perform analysis

$$S=\{S_1, S_2, \dots, S_n\}. \quad (2)$$

where each S_i contains the combined information from the window w :

$$S_i=\text{ExtractFeatures}(x_{i-w+1}, \dots, x_i), \quad (3)$$

where $x_i=\{x_{i,1}, x_{i,2}, \dots, x_{i,m}\}$ is the feature vector (the system state at time t_i), m is the number of input parameters (IoT, tariff, and weather data), and w is the multi-scale sliding-window size, which ranges from 3 to 168 hours depending on the overall proactive-management objective. We can then write:

The concept of a window is introduced to justify calculating the statistical features S_i .

$$S_i=\{\bar{x}_w, \sigma_w, CV_w, \text{trend}_w, \max_w, \min_w\}, \quad (4)$$

where, for each variable in a window of length w , $\bar{x}_w$ is the mean value defining the overall energy-consumption level (base load); σ_w is the standard deviation (oscillation amplitude), characterizing absolute energy-consumption volatility; CV_w is the coefficient of variation, a normalized variability measure that enables comparison of consumption instability across energy carriers (for example, stable steam consumption versus a fluctuating electrical load); trend_w is the linear trend, indicating the rate of load increase or decrease (a leading indicator of peaks); and $\max_w, \min_w$ are the extreme values defining the system-load bounds within the window.

Thus, we replace the raw window x_i with the vector of statistical features S_i . To account for the temporal dynamics of the object, we move from the space of instantaneous observations $X \subset \mathbb{R}^m$ to the space of dynamic states $S \subset \mathbb{R}^{m \times 6}$, where each state S_i results from the statistical convolution of data in a window of length w . The set of observations is therefore converted into the sequence $S_i=\Phi(W_i)$, where Φ is the feature-extraction operator and W_i is the sliding data window.

On the basis of (1)-(4), the energy-state classification task is defined as follows.

Let $Y=\{y_1, y_2, \dots, y_K\}$ be the set of energy-state classes, where y_k denotes one class. In the evaluated task, y_1 , y_2 , and y_3 correspond to Low, Medium, and High relative consumption states. These classes are descriptive data-derived labels and are not control strategies.

We seek a mapping $f: S \rightarrow Y$ that assigns each observed state S_i to the most probable class $y^* \in Y$:

$$y^* = \underset{y \in Y}{\operatorname{argmax}} P(y|S_i, \theta), \quad (5)$$

where $P(y|S_i, \theta)$ is the probability assigned to class y for state S_i under model parameters θ .

The classification objective minimized during training is:

$$J_{\text{class}}(\theta) = \sum_{i=1}^n L(y_i, f(S_i, \theta)) + \lambda \Omega(\theta), \quad (6)$$

where y_i is the reference energy-state label for state S_i , L is a classification loss such as cross-entropy, $\Omega(\theta)$ is a regularizer that limits overfitting, and λ is its coefficient.

The estimated model parameters are obtained by minimizing this objective:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} J(\theta), \quad (7)$$

The classifier minimizes the loss L . Economic costs, comfort requirements, and grid constraints are not terms in this experimentally evaluated classification loss.

The time series is therefore represented as a sequence of system states rather than isolated raw measurements. Classification assigns energy-state label y_j to multidimensional state vector x_i . A separate control layer would be required to translate that label into an action.

Figure 1 presents the broader PECM architecture built around the energy-state model. It follows the transformation of physical measurements P and contextual variables C , through statistical convolution in observation window W_i , to an estimated state class y^* . The figure also situates this classification output within a possible control workflow; only the classification component is evaluated in the present experiment.

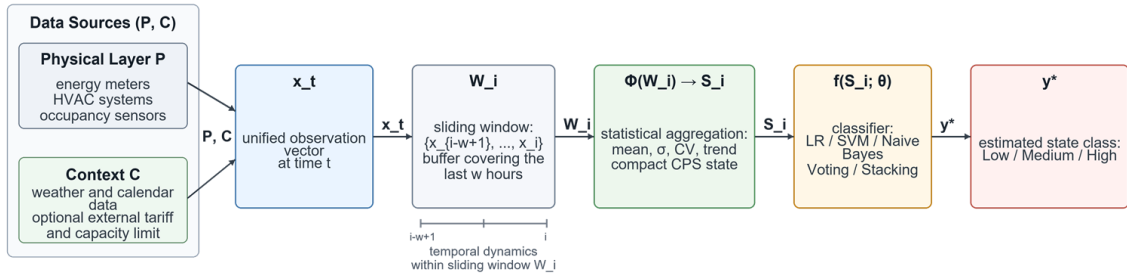


Figure 1: Structural diagram of energy-state classification within the broader PECM architecture

5. Development of an input vector formation method for intelligent analysis of PECM system state data

State-classification quality depends on the information contained in the input data. The feature vector combines measurements from several sources and converts them to a common time-series structure. Meter data describe the physical load, while weather, calendar information, and building metadata provide context for interpreting the same measured value. The sources differ in sampling conventions, units, and numerical scale. Variables measured in kWh, cubic metres, and degrees Celsius cannot be compared directly in distance-based clustering. Z-normalization therefore limits the influence of scale, while one-hot or cyclic encoding represents categorical and periodic variables without imposing an unsupported linear order. The proposed formation method has four stages. First, a heterogeneous dataset is selected and its variables are related to the mathematical state model. Second, the streams are cleaned, synchronized, normalized, and encoded. Third, sliding observation windows replace individual readings with summaries of level, variability, trend, and range. Fourth, cluster analysis assigns data-derived labels where no operational state annotation is available. This

sequence converts raw time series into structured vectors suitable for classifier training and comparison. It does not assume that a cluster is already a management action. In a complete PECM system, the classified state could inform a separate control strategy that also considers tariffs, comfort, and equipment constraints. The present experiment ends at the classification stage.

At the first stage, the open Building Data Genome 2 (BDG2) dataset was selected to develop and evaluate the energy-state model (<https://github.com/buds-lab/building-data-genome-project-2>) [13]. Its structure is summarized in Table 1. The dataset covers 19 sites in North America and Europe and contains electricity, heating, cooling, steam, solar, and water-use measurements. It also provides building metadata and weather data that can be joined to the meter records. These characteristics make BDG2 suitable for the present task because the feature-formation method must handle several energy carriers, different building functions, and changing external conditions. The dataset does not contain verified labels for operational states, control strategies, tariffs, or grid-capacity limits. It is therefore used to evaluate heterogeneous feature preparation and data-derived state classification, not the complete PECM control process.

Table 1 Characteristics of the BDG2 input data for the PECM model

Data category	Features	Units	Data Type	Description
Energy consumption	$P_{elec}, P_{heat}, P_{cool}, P_{steam}$	kWh, kW	Numerical	Hourly energy measurements
Water use	P_{water}	m ³	Numerical	Water consumption (irrigation is a separate meter type)
Meteorological	$C_{ext} = \{T_{out}, H_{out}, I_{sol}\}$	°C, %, W/m ²	Numerical	Temperature, humidity, insolation
Temporal (calendar)	$C_{temp} = \{H_{day}, W_{day}, IsHoliday\}$	int / bool	Categorical	Hour of day, day of the week, holiday
Economic	$C_{econ} = \{Price, Limit\}$	monetary units, kW	Numerical	Externally supplied tariff and network-capacity limit (conceptual PECM inputs)

BDG2 provides heterogeneous profiles from 3,053 meters, including electricity, heating and cooling ($P_{elec}, P_{heat}, P_{cool}, P_{steam}$), and separate water and irrigation records (P_{water}). It also provides meteorological and building context. Temporal variables are generated from timestamps, whereas the economic component ($C_{ext}, C_{temp}, C_{econ}$) is an external conceptual input rather than a BDG2 measurement. This structure supports joint analysis of energy flows and their physical context. The dataset is also sufficiently diverse for the present proof of concept. It contains 1,636 buildings used as hospitals, offices, educational facilities, and commercial premises. Differences in function, size, construction year, climate, and operating schedule broaden the range of consumption patterns represented in the sample. This diversity is useful for testing whether the same preparation pipeline can be applied across heterogeneous smart objects. The source data have hourly resolution for 2016 and 2017. This temporal coverage makes it possible to derive daily and seasonal features, although the reported experiment aggregates the readings to daily totals and does not evaluate the illustrative hourly windows introduced below. BDG2 measurements resemble data obtained from building-management systems and IoT meters. This makes the dataset appropriate for evaluating synchronization, normalization, feature extraction, and state classification under realistic variation in missingness and consumption. Nevertheless, an archived benchmark cannot reproduce the feedback, communication delay, actuator behaviour, or safety requirements of an operating controller. The experiment should therefore be interpreted as an offline proof of concept rather than

a live PECM deployment. Considering the dataset structure in Table 1 and the formal representation of the smart object in (1), the energy-state vector $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,m}\}$ of the smart object at time i is refined as follows:

$$x_i = [P_{elec}, P_{heat}, P_{cool}, P_{steam}, P_{water}, C_{ext}, C_{temp}, C_{econ}], \quad (8)$$

where $P_{elec}, P_{heat}, P_{cool}, P_{steam}$ denotes the energy-carrier measurements available for the current meter record; P_{water} denotes water use, with general water and irrigation retained as separate BDG2 meter types; C_{ext} contains weather variables; C_{temp} contains temporal context; and C_{econ} denotes optional economic inputs such as a dynamic tariff and grid-capacity limit. These economic inputs are externally defined and are not BDG2 measurements.

The second stage prepares the heterogeneous BDG2 data for clustering and classification through cleaning, synchronization, Z-normalization, and categorical encoding. These operations establish a consistent time index and comparable numerical representation before dynamic features and pseudo-labels are generated.

During data cleaning, anomalous values associated with meter faults are removed. Short gaps are filled by linear interpolation to preserve continuity in the time series.

Measurements and contextual data from different sources are synchronized by timestamp:

$$x_i = [X_{P_{elec}} \cup X_{P_{heat}} \cup X_{P_{cool}} \cup X_{P_{steam}} \cup X_{P_{water}} \cup X_{C_{ext}} \cup X_{C_{temp}} \cup X_{C_{econ}}], \quad (9)$$

Weather context C_{ext} may be obtained from NOAA [17] or the OpenWeatherMap API [18]. Calendar variables C_{temp} are generated algorithmically, with 1 denoting working time and 0 denoting night or weekend periods. In the general PECM formulation, economic context C_{econ} may be supplied externally through a Time-of-Use scenario with higher peak-period prices and lower night prices. This component is not contained in BDG2 and is not evaluated in the reported classification experiment. Aligning the available energy streams $P_{elec}, P_{heat}, P_{cool}, P_{steam}, P_{water}$ and contextual variables $C_{ext}, C_{temp}, C_{econ}$ produces an integrated state vector x_i for time step i .

At the Z-normalization step, the effects of differing units and scales in the numerical energy and contextual inputs are removed. Z-normalization of each numerical component $x_{i,j}$ of the input vector is performed using the following expression:

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \quad (10)$$

where μ_j is the mean of the j -th parameter and σ_j is its standard deviation.

After normalization, each feature has zero mean and unit variance, making scales comparable for clustering and classification.

At the categorical-feature encoding step, one-hot encoding transforms the temporal context C_{temp} , such as day of the week and working day/weekend, into numerical form; cyclic encoding is used for hours and months so that the model can account for periodicity in the object's operation.

Preprocessing converts heterogeneous measurements into a structured feature matrix. This step reduces scale and encoding artefacts that could otherwise dominate the generated dynamic features and state classification. It also separates cleaning decisions from the classifier itself: missing-value treatment, temporal alignment, and categorical representation are completed before model comparison. The resulting matrix retains physical, temporal, and contextual meaning while providing numerical inputs suitable for clustering.

The third stage generates dynamic features from a sliding window [19]. This replaces individual measurements with summaries of recent behaviour and allows the classifier to represent both consumption level and load variation. Mean and extreme values describe the level and range of demand, while standard deviation and the coefficient of variation describe instability. A linear trend distinguishes a rising or falling load from a stationary interval. These summaries reduce the influence of isolated readings and provide a compact description of the historical context associated with each state.

For the general hourly formulation, a buffer is defined at time i from normalized feature vectors over the preceding W_i hours, where w is the window length:

$$W_i = \{x_j | j \in \{i-w+1, \dots, i\}\}, \quad (11)$$

where each vector x_j has the structure given in (8).

$$x_j = [P_{elec,j}, P_{heat,j}, P_{cool,j}, P_{steam,j}, P_{water,j}, C_{ext,j}, C_{temp,j}, C_{econ,j}].$$

Here, index j iteratively denotes the time points within the window: $j=i$ corresponds to the current time, while $j=i-w+1$ marks the beginning of the observation interval. This clearly distinguishes the current system state (i) from the historical context needed to identify dynamic trends. To transform the multidimensional time window W_i into a compact state vector S_i , a statistical convolution Φ is used. The feature vector S_i is formed by calculating the set of statistical metrics in (4) for each component of the window:

$$S_i = \Phi(W_i) = \{\bar{x}_w, \sigma_w, CV_w, \text{trend}_w, \max_w, \min_w\}, \quad (12)$$

The sliding-window size w controls the balance between temporal memory and sensitivity to recent change. In the general hourly framework, $w_{short}=6$ is an illustrative short-term setting for changes associated with the economic context C_{econ} or a brief load surge, whereas $w_{long}=24$ represents a daily cycle and smooths shorter fluctuations. A longer observation period can represent weekly or seasonal structure but may respond slowly to recent change. More generally, window w may be treated as a hyperparameter selected from candidates $\{w_1, w_2, \dots, w_n\}$. For each setting, the statistical operators in (4) and (12) produce a different state vector S_i , and cluster separation can be used as one criterion for comparison. This is a general design procedure rather than a result of the current experiment. The reported data preparation aggregates the hourly meter streams to daily totals before feature calculation. It therefore uses a daily-scale observation window w and cannot demonstrate the response characteristics of the illustrative 6-hour and 24-hour settings. No empirical comparison of alternative window sizes is reported, so the hourly multi-scale formulation should not be interpreted as experimentally validated here. In the general framework, clustering can provide feedback on temporal resolution by showing whether candidate features separate the states. The present experiment does not compare alternative window sizes, so this role remains part of the proposed framework rather than an experimentally validated result.

At the fourth stage, state labels are generated because BDG2 contains no verified operational-state or control-strategy annotations. Each feature vector S_i from (12) is processed by K-Means, which groups the vectors S_i into K clusters C_k , where $k=1, \dots, K$. In the reported three-class formulation, C_1 , C_2 , and C_3 are interpreted as Low, Medium, and High relative consumption states, not operating modes or control actions. Their meaning is descriptive: Low identifies the lower-consumption cluster within a meter type, High identifies the upper cluster, and Medium covers the intermediate region. The boundaries therefore depend on the observed distribution and should not be treated as externally verified physical thresholds. The general number of clusters K may be examined by the elbow criterion, although the present experiment reports a fixed three-class result and no comparative cluster-quality analysis. Each vector S_i receives the pseudo-label $y_i \in \{1, \dots, K\}$ used for classifier training. The supervised metrics consequently measure agreement with these generated labels rather than accuracy against documented control states.

In the process of clustering by K-Means, we minimize the target function:

$$J_{clust} = \sum_{k=1}^K \sum_{S_i \in C_k} |S_i - \mu_k|^2, \quad (13)$$

where μ_k is the centroid of the k -th cluster, representing the ideal state profile in feature space.

This stage converts the continuous energy-consumption stream into a sequence of data-derived labels $\{y_1, y_2, \dots, y_n\}$. The theoretical formulation represents a possible next-state target y_{i+1} based on energy features $P_{elec}, P_{heat}, P_{cool}, P_{steam}, P_{water}$ and context $C_{ext}, C_{temp}, C_{econ}$. The reported experiment does not specify a forecast horizon or document an explicit target shift. It also does not report separate metrics for one-step or multi-step prediction. Its results are therefore interpreted as classification of generated consumption states rather than validated prospective forecasting. The next-state notation remains part of the conceptual formulation and requires a separate time-indexed experiment.

6. Energy-state classification method and conceptual control extension

The proposed PECM architecture contains an implemented classification layer and a conceptual control layer. The implemented stages prepare the feature vectors, divide the data into training and test samples, train Logistic Regression, SVM, and Naive Bayes classifiers, and combine them through Voting and Stacking. A future control layer could map a classified state to action a_i by minimizing objective J_{cost} . Such a mapping would require an explicit action set, tariff series, network-capacity value, comfort model, optimization algorithm, and baseline control policy. These elements are not contained in the BDG2 classification sample. The present experiment therefore evaluates only the preparation and classification stages and treats the optimizer as an architectural extension.

For a sequential PECM implementation, a Time-Series Split strategy [21] can divide the state sequence $\{y_1, y_2, \dots, y_n\}$ into training and test periods while preserving temporal order. An Expanding Window can then assess stability as new observations become available. This procedure is suitable when the aim is to test prospective performance and prevent future observations from entering model fitting. It also permits periodic retraining as the historical record grows. The experiment reported below uses a train/test split after daily aggregation and evaluates classification performance on the held-out sample. It does not document a forecast horizon, expanding-window retraining schedule, or online update process. It also does not evaluate control strategies under operating conditions. Logistic Regression, linear SVM, and Gaussian Naive Bayes were selected as established classifiers with different statistical assumptions. Soft Voting tests probability averaging, whereas Stacking allows a logistic-regression meta-model to weight the base predictions. The authors' earlier work [22] informed this ensemble configuration, but validation for the present task is provided only by the BDG2 experiment reported here. In the conceptual extension, a classified state y_i could be mapped to a control action a_i by an optimizer that minimizes J_{cost} :

$$J_{cost}(a_i) = \text{Cost}(E, P_{price}) + \lambda \cdot \text{Penalty}(P_{limit}), \quad (14)$$

In this conceptual objective, Cost denotes direct energy cost based on external economic input C_{econ} , and Penalty penalizes violation of a capacity limit P_{limit} . Coefficient λ sets the relative weight of the constraint term $\lambda \cdot \text{Penalty}$. The tariff, limit, action space, and optimizer are not supplied by BDG2 and are not implemented or evaluated in the present experiment. The classification output may therefore serve as an input to a future objective J_{cost} that selects action a_i . This mapping is an architectural extension rather than a validated result of the current study.

7. Experimental evaluation of energy-state classification using Building Data Genome 2

The experiment evaluates the feature-formation and state-classification components using the open Building Data Genome 2.0 (BDG2) dataset. It tests whether selected classifiers can reproduce three data-derived relative consumption states after daily aggregation and after the contemporaneous meter value has been removed from the classifier inputs. The evaluation compares three basic models with two ensemble constructions using the same prepared sample and a common set of multiclass metrics. It does not test the hourly multi-scale formulation, future-state forecasting, cost optimization, control-action generation, comfort preservation, or grid-capacity compliance. These elements belong to the wider PECM architecture but are outside the reported experiment. The procedure comprised the following steps.

Step 1. Dataset selection and preparation. BDG2 provides eight meter files: electricity, chilled water, gas, hot water, irrigation, solar energy, steam, and water. The wide-format files were reshaped to long format so that each row contains a building identifier, meter type, timestamp, and measured value. Because two years of hourly data for about 1,600 buildings produce tens of millions of rows, readings were aggregated to daily totals immediately after reshaping to reduce memory use. Aggregation changes the temporal unit of the experimental analysis: subsequent historical statistics

describe daily observations rather than the hourly windows in the general formulation. The long-format records from the eight meter types were retained in one prepared dataset. The meter records were then combined with building metadata and daily weather variables so that each consumption observation could be interpreted in relation to the building and external conditions:

- Building metadata include site identifier, primary use, floor area, construction year, number of floors, coordinates, and time zone. These variables describe persistent differences in typical consumption.
- Daily weather variables include air temperature, wind speed, atmospheric pressure, and precipitation, which help represent environmental effects on energy demand.

Because the prepared dataset contained more than two million rows, a stratified sample of 60,000 records was formed while preserving the proportions of meter types.

Step 2. Preprocessing. Numerical missing values were replaced by medians and numerical features were scaled with StandardScaler. Nominal variables were one-hot encoded. Label encoding was avoided because it would impose an unsupported order on categorical values and could distort distances or linear decision boundaries. Calendar variables were represented in a form suitable for model input, and preprocessing was applied before classifier training. These operations place heterogeneous numerical variables on comparable scales and provide a consistent matrix for the Gaussian Naive Bayes, Logistic Regression, and SVM baselines.

Step 3. Dynamic feature generation. For each building-meter pair, the daily aggregated series was used to calculate the statistics in (4) and (12): mean, standard deviation, coefficient of variation, linear trend, and minimum and maximum values within the observation window. This experiment therefore evaluates daily historical summaries rather than the illustrative hourly multi-scale settings in the general framework.

Step 4. Data-derived state labelling. BDG2 has no categorical target for energy states, so labels were generated separately within each meter type. Values were logarithmically transformed and k -means clustering produced three relative consumption classes: Low, Medium, and High. The labelled records were then pooled for the reported classification task. These labels are relative within a meter type and are not verified operating modes or control strategies. The contemporaneous meter reading was excluded from the classifier's feature space to reduce direct target leakage.

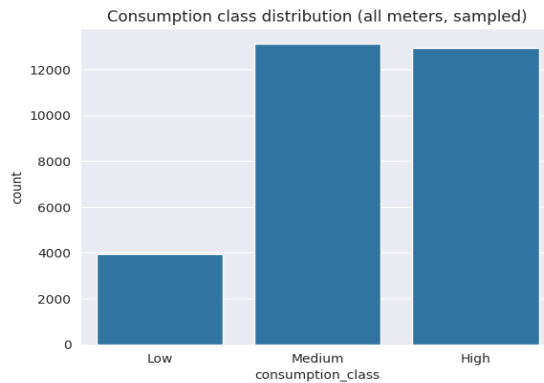


Figure 2: Distribution of formed consumption classes Low, Medium and High

Step 5. Training and test samples. After the train/test split, class imbalance in the training sample was addressed with the Synthetic Minority Over-sampling Technique (SMOTE).

Step 6. Training basic and ensemble classifiers. Three basic classifiers and two ensembles were trained for the multiclass task:

- Logistic Regression was used as a linear multiclass baseline. Its regularization coefficient, penalty type, and solver were selected by randomized search with cross-validation.

- The SVM used a linear kernel because an RBF kernel becomes impractical at the sample size produced after SMOTE. The linear formulation also provides a direct comparison with Logistic Regression in the standardized feature space. Because Soft Voting requires comparable probability outputs from its base models, the fitted SVM was calibrated to provide class-probability estimates.
- Gaussian Naive Bayes was used as a probabilistic baseline that assumes normally distributed numerical features.
- Soft Voting averaged the class probabilities produced by the three basic classifiers.
- The Stacking classifier used Logistic Regression as a meta-model to learn weights for the base predictions rather than averaging them equally.

Step 7. Evaluation. Accuracy, precision, recall, and F1-score were calculated with weighted averaging to account for class imbalance. The area under the ROC curve (AUC) was calculated in a one-versus-rest formulation and then weighted across classes. Weighted metrics preserve the contribution of each class in proportion to its frequency, which is relevant because the generated Low class is smaller than the Medium and High classes. LogLoss evaluates the quality of predicted probabilities, while the Matthews correlation coefficient (MCC) summarizes the full multiclass confusion structure. Specificity has no single direct multiclass definition, so it was calculated for each one-versus-rest class and macro-averaged. These measures describe different aspects of agreement with the K-Means pseudo-labels. They do not measure energy savings, comfort, or control quality. Table 2 summarizes the results.

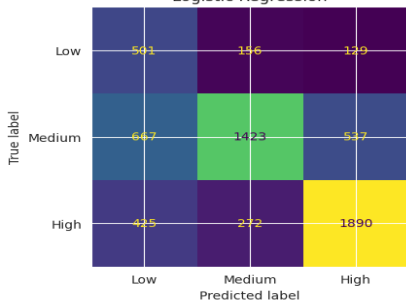
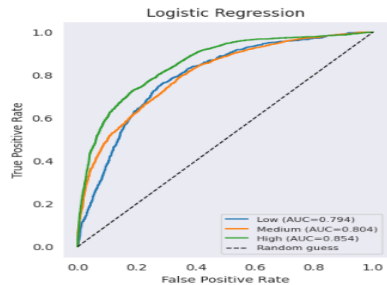
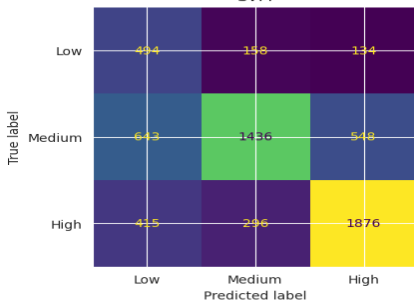
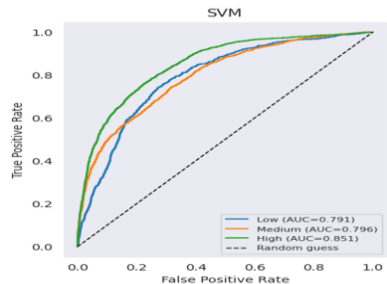
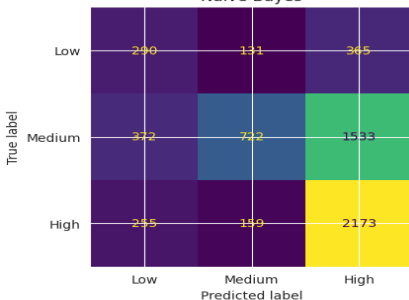
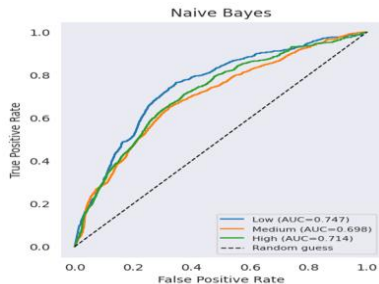
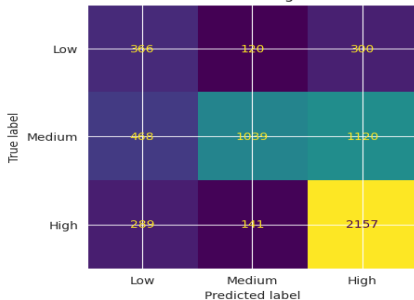
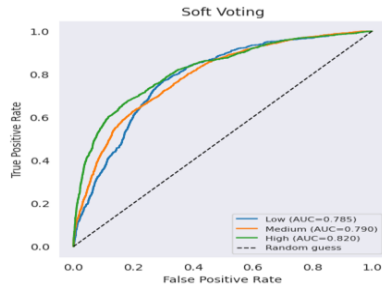
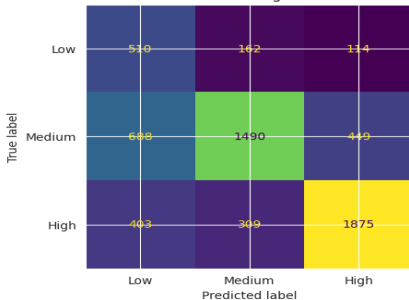
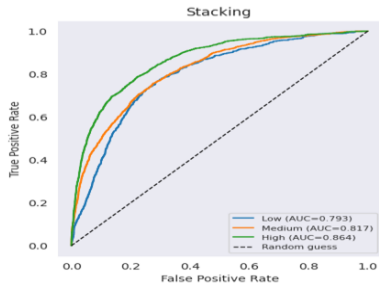
Table 3 also presents a confusion matrix and one-versus-rest ROC curves for each classifier. These plots show which generated classes are most difficult to separate, particularly the Medium class with its diffuse cluster boundaries. The metrics in Table 2 and plots in Table 3 show that Stacking performed best among the five evaluated models (Accuracy = 64.58%, F1 = 66.21%, AUC = 83.43%). Its learned combination of base predictions was more effective than equal probability averaging in Soft Voting. Logistic Regression and linear SVM produced similar results (≈ 0.65), suggesting partial linear separation in the feature space. Naive Bayes performed worst (LogLoss = 4.8586), which is consistent with violations of its distributional assumptions in the heterogeneous feature space. The ensemble advantage is limited: Stacking improves the reported metrics but still misclassifies a substantial proportion of records. Overall performance is therefore moderate. It supports continued evaluation of the state-classification component, but should not be interpreted as evidence of reliable autonomous control or as validation of the conceptual optimization layer.

The confusion matrices and ROC curves in Table 3 show that the Medium class produced the most classification errors. Its boundaries are diffuse under k -means because it lies between the more distinct Low and High relative consumption groups. This overlap is also a limitation of the generated pseudo-labels.

Table 2
Performance metrics of basic and ensemble classifiers

Model	Accuracy	Precision	Recall	F1 Score	AUC	LogLoss	MCC	Specificity
Logistic Regression	0.6357	0.6966	0.6357	0.6503	0.8242	0.7997	0.4488	0.8228
SVM	0.6343	0.6906	0.6343	0.6482	0.8194	0.8052	0.4436	0.8209
Naive Bayes	0.5308	0.5839	0.5308	0.4998	0.7112	4.8586	0.2658	0.7459
Soft Voting	0.5937	0.6526	0.5937	0.5837	0.8023	0.8577	0.3723	0.7871
Stacking	0.6458	0.706	0.6458	0.6621	0.8343	0.7792	0.464	0.8287

Table 3
Confusion Matrices and ROC curves of basic and ensemble classifiers

Model	Confusion Matrices	ROC curves
Logistic Regression	Logistic Regression 	Logistic Regression 
	SVM 	SVM 
Naive Bayes	Naive Bayes 	Naive Bayes 
Soft Voting	Soft Voting 	Soft Voting 
Stacking	Stacking 	Stacking 

The Stacking classifier therefore provides moderate evidence that heterogeneous daily features can identify the generated consumption states. Because the labels are derived from K-Means rather than verified operating conditions, and because no forecast horizon or control layer is evaluated, the results establish a preliminary classification component rather than a reliable decision or control system.

Conclusion

This paper develops and evaluates a data-analytic component for proactive energy-management research: the classification of data-derived energy-consumption states from heterogeneous building information. The results show that meter histories, weather variables, and building metadata contain information associated with three relative consumption classes. The study therefore addresses the state-identification stage that can precede a control decision in the wider PECM architecture. The results support the feasibility of this component but do not establish prospective forecasting, improved energy efficiency, or the performance of a complete proactive controller.

The mathematical model represents the smart object as a CPS that combines physical measurements with contextual variables. Within this representation, heterogeneous observations are converted into feature vectors and assigned to energy-state classes. Sliding-window statistics describe the level, variability, trend, and range of the recent measurements, while normalization makes variables with different units comparable. The classification loss estimates the parameters required to assign a feature vector to Low, Medium, or High relative consumption. This formulation separates state identification from the later selection of a management action. A classified state can be an input to a controller, but the class is not itself an action, operating schedule, or assurance that a cost or comfort objective will be satisfied.

The feature-formation method integrates daily historical meter statistics, weather variables, and building metadata. Logarithmic transformation and K-Means produced three relative consumption classes, Low, Medium, and High, separately within each meter type before pooled classifier evaluation. This per-meter labelling avoids treating electricity, steam, solar, and water values as though they shared a common physical scale. The class names indicate positions within each meter-specific distribution. They are data-derived pseudo-labels rather than verified operating modes or predefined control strategies.

The broader PECM architecture also contains a conceptual mapping from a classified state to a control action a_i through objective J_{cost} . The present work does not specify or test an optimizer, action space, tariff scenario, comfort model, or grid-capacity constraint. The economic variables shown in the general state vector are externally supplied conceptual inputs rather than measurements from BDG2. Equation (14) therefore defines a possible interface between state classification and a future control layer. It is not an algorithmic description of the optimizer and should not be interpreted as an experimentally validated control method.

The experiment used a stratified sample of 60,000 daily aggregated BDG2 records. Among the five evaluated classifiers, the linear models achieved ≈ 0.65 , while Naive Bayes was least effective (LogLoss = 4.8586). Soft Voting did not improve on the two linear baselines, whereas Stacking learned a more effective combination of their outputs. These results describe agreement with generated consumption classes, not accuracy against externally verified operational states. They are also based on daily summaries and do not establish performance for the illustrative hourly windows.

Stacking achieved the strongest results in the evaluated set (Accuracy = 64.58%, F1-score = 66.21%, AUC = 83.43%). The Medium state remained the most difficult to classify because its cluster overlaps the neighbouring Low and High groups. This overlap is expected for a data-derived intermediate class whose boundaries are less distinct than the lower and upper consumption clusters. The resulting performance is moderate: it indicates useful structure in the heterogeneous features, but approximately one third of records remain misclassified. The result may support further

development of the classification stage, but it is insufficient by itself to demonstrate dependable automated control or practical cost reduction.

Future work should implement optimization objective J_{cost} and evaluate control actions, economic effects, comfort requirements, and grid limits under explicit tariff and capacity scenarios. It should also test prospective prediction horizons and the hourly multi-scale formulation. These extensions are required before conclusions can be drawn about end-to-end proactive energy management.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT 5.6-Sol, Gemini 3.1 Flash-Lite and 3.1 Pro to improve language, readability, and stylistic consistency of the manuscript, as well as for translation. After using these tools, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

References

- [1] T. Tröndle, O. Melnyk, O. Tutova, V. Porieva, F. Neumann, I. Staffell, A. Patt, Rebuilding Ukraine's energy supply in a secure, economic, and decarbonised way, *Environmental Research: Infrastructure and Sustainability* 4 (2024) 031002. doi:10.1088/2634-4505/ad6738.
- [2] P. Palensky, D. Dietrich, Demand side management: demand response, intelligent energy systems, and smart loads, *IEEE Transactions on Industrial Informatics* 7 (2011) 381–388. doi:10.1109/TII.2011.2158841.
- [3] Y. Wei, X. Zhang, Y. Shi, L. Xia, S. Pan, J. Wu, M. Han, X. Zhao, A review of data-driven approaches for prediction and classification of building energy consumption, *Renewable and Sustainable Energy Reviews* 82 (2018) 1027–1047. doi:10.1016/j.rser.2017.09.108.
- [4] 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. doi:10.1016/j.rser.2014.03.027.
- [5] D. Mariano-Hernández, L. Hernández-Callejo, A. Zorita, O. Duque, F. Santos García, A review of strategies for building energy management system: model predictive control, demand side management, optimization, and fault detect & diagnosis, *Journal of Building Engineering* 33 (2021) 101692. doi:10.1016/j.jobe.2020.101692.
- [6] M. H. Albadi, E. F. El-Saadany, A summary of demand response in electricity markets, *Electric Power Systems Research* 78 (2008) 1989–1996. doi:10.1016/j.epsr.2008.04.002.
- [7] P. Charoen, N. Kitbutrawat, J. Kudtongngam, A demand response implementation with building energy management system, *Energies* 15 (2022) 1220. doi:10.3390/en15031220.
- [8] A. Kathirgamanathan, M. De Rosa, E. Mangina, D. Finn, Data-driven predictive control for unlocking building energy flexibility: a review, *Renewable and Sustainable Energy Reviews* 135 (2021) 110120. doi:10.1016/j.rser.2020.110120.
- [9] A. Afram, F. Janabi-Sharifi, Review of modeling methods for HVAC systems, *Applied Thermal Engineering* 67 (2014) 507–519. doi:10.1016/j.applthermaleng.2014.03.055.

- [10] K. Amasyali, N. M. El-Gohary, A review of data-driven building energy consumption prediction studies, *Renewable and Sustainable Energy Reviews* 81 (2018) 1192–1205. doi:10.1016/j.rser.2017.04.095.
- [11] Y. Himeur, M. Elnour, F. Fadli, et al., AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives, *Artificial Intelligence Review* 56 (2023) 4929–5021. doi:10.1007/s10462-022-10286-2.
- [12] C. Miller, P. Arjunan, A. Kathirgamanathan, C. Fu, J. Roth, J. Y. Park, et al., The ASHRAE Great Energy Predictor III competition: overview and results, *Science and Technology for the Built Environment* 26 (2020) 1427–1447. doi:10.1080/23744731.2020.1795514.
- [13] buds-lab, Building Data Genome Project 2, GitHub repository, 2020. URL: <https://github.com/buds-lab/building-data-genome-project-2> (accessed 10 July 2026).
- [14] Z. Wang, R. S. Srinivasan, A review of artificial intelligence-based building energy prediction with a focus on ensemble prediction models, in: *Proceedings of the 2015 Winter Simulation Conference*, 2015. doi:10.1109/WSC.2015.7408504.
- [15] F. Divina, A. Gilson, F. Gómez-Vela, M. García Torres, J. F. Torres, Stacking ensemble learning for short-term electricity consumption forecasting, *Energies* 11 (2018) 949. doi:10.3390/en11040949.
- [16] Y. Thangaraj, D. Buvana, M. Thirumalai, S. Venkatesan, M. Bajaj, V. Blažek, L. Prokop, Artificial intelligence-driven cyber-physical energy resilience framework for secure and sustainable smart distribution networks, *Energy Strategy Reviews* 64 (2026) 102168. doi:10.1016/j.esr.2026.102168.
- [17] National Oceanic and Atmospheric Administration, National Centers for Environmental Information climate data. URL: <https://www.ncei.noaa.gov> (accessed 10 July 2026).
- [18] OpenWeather, OpenWeatherMap API documentation. URL: <https://openweathermap.org/api> (accessed 10 July 2026).
- [19] I. M. Guyon, A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research* 3 (2003) 1157–1182. URL: <https://www.jmlr.org/papers/v3/guyon03a.html>.
- [20] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035. doi:10.1145/1283383.1283494.
- [21] R. J. Hyndman, G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., OTexts, Melbourne, 2021. URL: <https://otexts.com/fpp3/> (accessed 10 July 2026).
- [22] O. O. Arsirii, I. O. Krantovskyi, O. V. Rudenko, M. G. Glava, Machine learning models for predicting payment status on an online car rental platform, *Applied Aspects of Information Technology* 8 (2025) 13–23. doi:10.15276/aait.08.2025.1.