Information Theory Inspired Pattern Analysis for Time-Series IoT Data

Yushan Huang, Yuchen Zhao, Alexander Capstick, Francesca Palermo,
Hamed Haddadi, and Payam Barnaghi

Abstract—Current methods for pattern analysis in time series mainly rely on statistical features or probabilistic learning and inference methods to identify patterns and trends in the data. Such methods do not generalize well when applied to multivariate, multi-source, state-varying, and noisy time-series data. To address these issues, we propose a highly generalizable method that uses information theory-based features to identify and learn from patterns in multivariate time-series data. To demonstrate the proposed approach, we analyze pattern changes in human activity data. For applications with stochastic state transitions, features are developed based on Shannon's entropy of Markov chains, entropy rates of Markov chains, entropy production of Markov chains, and von Neumann entropy of Markov chains. For applications where state modeling is not applicable, we utilize five entropy variants, including approximate entropy, increment entropy, dispersion entropy, phase entropy, and slope entropy. The results show the proposed information theory-based features improve the recall rate, F1 score, and accuracy on average by up to 23.01% compared with the baseline models and a simpler model structure, with an average reduction of 18.75 times in the number of model parameters.

Index Terms—entropy, IoT, time-series data, pattern analysis

I. Introduction

With the development of small-scale and low-cost network-connected devices, large volumes of data is generated [1]. In particular, the Internet of Things (IoT) provides us with an unprecedented ability to capture real-world information. By integrating the real world with the digital world, IoT enables us to analyze and mine useful information based on collected data. These technologies have been widely applied across several fields such as healthcare [2].

Yushan Huang is with the Department of Computing, Imperial College London, and Care Research and Technology Centre, The UK Dementia Research Institute, London, UK (e-mail: yushan.huang21@imperial.ac.uk).

Yuchen Zhao is with the Department of Computer Science, University of York, York, UK (e-mail: yuchen.zhao@york.ac.uk).

Alexander Capstcik, Francesca Paleromo, and Payam Barnaghi are with the Department of Brain Sciences, Imperial College London, and Care Research and Technology Centre, The UK Dementia Research Institute, London, UK (e-mail: alexander.capstick19, f.palermo, p.barnaghi@imperial.ac.uk).

Hamed Haddadi is with the Department of Computing, Imperial College London, UK (e-mail: h.haddadi@imperial.ac.uk).

Payam Barnaghi is also with the Great Ormond Street Institute of Child Health, University College London.

Time-series data is critical in the real world, as it contains key information on relationships from a temporal perspective. Analyzing time-series data facilitates the development of effective methods for observing the raw data and also allows us to understand relationships within the data. It also enables us to uncover the various patterns that exist in the data, determine the relationships between these patterns, analyze the trends, and make predictions. Unfortunately, the analysis of time-series data is very challenging, as such data (*e.g.*, human activity data) is often multivariate [3], multi-source [4], rapidly state-varying [5], and noisy [6], which is difficult to mine the potential information and can be easily affected by noise.

There are several well-established methods for pattern and trend analysis applied to time-series data [7]. These methods can be classified into four categories based on their data mining approaches: statistical methods, statistical and probabilistic learning and inference methods, deep neural networks, and information theory-driven techniques. However, these methods have been limited in their applicability to multivariate, multi-source, rapidly state-varying, and noisy time-series data. Recently, deep neural network (DNN) models have attracted a great deal of attention. Such models can learn spatio-temporal properties of data, extract features automatically, and analyze patterns to predict outcomes or changes, such as state transitions[8]. Although deep neural network models can be effective in analyzing complex datasets, these models and the features they extract are often difficult to be interpreted. Interpretable features such as the features extracted by information theory-driven techniques can make a learning model surpass the performance of deep neural network models [9], while also improving our ability to explain the inference process of machine learning models.

Our previous works include the Blocks of Eigenvalues algorithm for time series segmentation [10] as a method to represent time-series data, a pattern representation method based on mutual information and entropy [11], and preliminary experiments and analysis of three Markov chain-based entropy features via heat maps [12]. These studies highlight the potential of entropy features in analyzing time-series data. How-

ever, these works do not present a complete pipeline for analyzing time series data and do not validate the results of the methods by machine learning and deep learning models. These works have demonstrated the potential of using entropy when handling data that is multivariate, multi-source, rapidly state-varying, and noisy. Thus, inspired by information theory and entropy, in this paper we propose a pipeline to extract interpretable features in multivariate time-series data, which will enhance the performance of machine learning and deep learning models. The primary contributions of this paper are as follows:

(1) We introduce different entropy-based methods to derive engineered features from time-series data. We then propose a pipeline for extracting interpretable, higher-level features that are highly generalizable and applicable to processing multivariate, multi-source, rapidly state-varying and noisy time-series data.

(2) We apply our information theory-based models to one human activity dataset (from a clinical study for remote healthcare monitoring) and two publicly available datasets (Gait in Aging and Disease Database [13], and PTB Diagnostic ECG Database [14]) to demonstrate the applicability of this approach in different settings and for different applications.

(3) We evaluate the effectiveness of the extracted features using different models such as logistic regression, Support Vector Machines (SVM), Multi-Layer Perception (MLP), and Long Short Term Memory (LSTM) neural networks. Our experimental results show that, for the three different types of datasets, compared to the baseline methods, the information theory-based features can significantly improve the accuracy, recall, and F1 scores of the models by an average of 10%-25%.

In conclusion, we present a general pipeline for processing multivariate, multi-source, rapidly changing, and noisy time-series data. Our approach provides a comprehensive description of the creation, selection, and modeling of entropy features, offering a new perspective for analyzing this type of data. We also evaluate the effectiveness of our information theory-based pipeline using various datasets, showcasing its versatility and generalizability. Our approach has the potential to enhance the performance of machine learning models for time-series data analysis, making it a useful tool for real-world applications.

The remainder of this paper is organized as follows. In Section II, we review the state-of-the-art works in pattern analysis for time-series data. In Section III, we introduce and analyze the original data from three datasets, which are multivariate, multi-source, rapidly state-varying and noisy. In Section IV, to process this type of time-series data, we provide a technical description of the entropy techniques and their variants in detail to mine the potential information of the time-series data. In Section V, we demonstrate the evalua-

tion results of the three datasets on different machine learning and deep learning models. Finally, in Section VI, we conclude our studies and discuss future work.

The source code, constructed models and links to the public datasets are made available via self-explanatory code with mark-up on a GitHub repository [15].

II. Related Works

There are four main approaches to mining information from time-series data: statistical methods, statistical and probabilistic learning and inference methods, deep neural network models, and information theory-driven techniques.

Classical statistical methods primarily focus on feature selection rather than data mining. However, with the increase in the amount and complexity of data, it becomes challenging to apply classical statistical theory-based techniques as they assume that the data is statistically uncorrelated. These techniques tend to perform poorly when applied to multivariate, multi-source, rapidly changing, or noisy time-series data [16].

DNN-based techniques are popular for mining information from time-series data due to their ability to extract features and yield optimal results for large datasets. Feature extraction methods such as convolutional neural networks (CNN) and long short-term memory (LSTM) are typically used in the design of the DNN structure. In recent years, researchers have continuously carried out innovative research on the basis of these representative methods. For example, Xia et al. combined CNN and LSTM to create an eight-layer CNN-LSTM model that considers both spatial and temporal embedded information of the original data [17]. Singh et al. added a self-attention mechanism to CNN-LSTM for better performance [18]. Despite the convenience of feature extraction using DNNs for time-series data, understanding and interpreting the extracted features is still a significant challenge due to the "black box" nature of DNNs. Furthermore, DNNs can only automatically extract simple features and not more complex features.

To mine features that are both interpretable and more complex from time-series data, some researchers have begun to develop research from the perspective of information theory. Shannon first proposed the concept of entropy, to measure the uncertainty of information, establishing the scientific theoretical basis of modern information theory [19]. Based on Shannon's entropy, several entropy variants such as spectral entropy [20] and sample entropy [21] have been proposed. Nurwulan et al. compared traditional features with multi-scale entropy (MSE) features extracted from 3-axis acceleration data and showed that MSE outperformed traditional features in KNN and random forest (RF) classification [22]. Bao et al. extracted frequency-domain entropy features from original acceleration
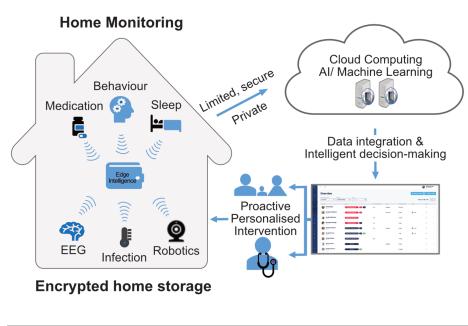
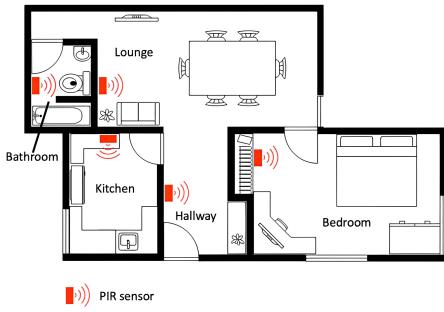Fig. 1. An overview of the healthcare monitoring IoT platform.

Fig. 2. An example of PIR sensor installation in the study.

data, which were combined with mean, energy, and correlation of the original data as inputs to build a model with ideal results [23]. While the above entropy-based methods offer new avenues for time-series data analysis, they also have certain limitations. Many existing studies only utilize a single entropy feature or use entropy features as supplementary to traditional features. Furthermore, these methods are task-specific and do not form a comprehensive pipeline based on entropy methods. Additionally, there is a lack of a clear explanation for the selection and calculation of entropy features.

A similar study to this paper is Howedi et al. 's entropy measurement model [24], which uses approximate entropy (ApEn), sample entropy (SampEn), and fuzzy entropy (FuzzyEn) to detect visitors in a home environment. However, this study does not select entropy features based on the data characteristics, such as Markovian systems and stochastic state transitions, and does not provide a justification for the selection of entropy features.

III. DATASETS

In this paper, we apply our information theory-based pipeline to three datasets, one human activity data collected from the in-home healthcare monitoring IoT platform of our ongoing Minder study, as well as
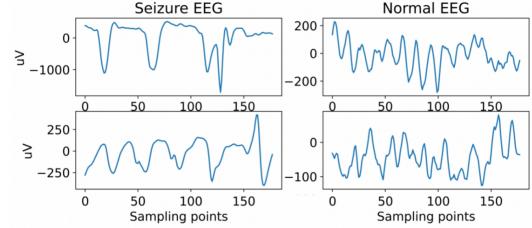

Fig. 3. Visualisation of ESRD. The x-axis represents sampling points, and the y-axis represents EEG signals (μV).
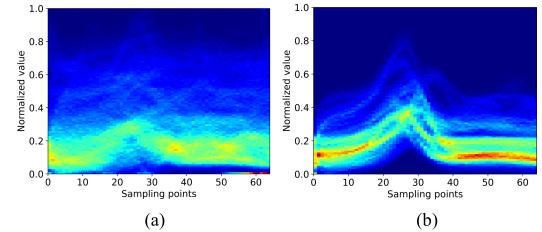

Fig. 4. The histogram color maps for PTBDB marked as abnormal (a) and normal (b). The x-axis represents sampling points, and the y-axis represents the normalized value of the heartbeat.

two publicly available EEG signal datasets, providing information on epileptic seizure and heart disease, respectively.

A. Minder Dataset

We have developed an in-home healthcare monitoring IoT platform (illustrated in Fig. 1), called Minder, to support people living with dementia (PLWD) [25]. The Minder platform collects various digital markers, including activity data, home device usage, and clinical information. It comprises four main parts: 1) device-independent sensors installed in participants' homes to collect original data, 2) a back-end system with cloud infrastructure, storage, and analysis tools to analyze the data and install machine learning algorithms, 3) a user interface presenting clinical and environmental information, as well as alerts generated by the system, and 4) clinical intervention involving healthcare professionals using the system/alerts to communicate with participants and caregivers to address their medical needs.

The Minder study protocol received ethical approval from the London-Surrey Borders Research Ethics Committee and South West London Ethics Committee (see link here) and we obtained informed written consent from all study participants.

The dataset is labeled by our monitoring team in response to alerts generated on the Minder platform, which operates 24/7. These alerts are verified with the people living with dementia (PLWD) or their caregivers, and provide information on potential healthcare-related events such as falls, abnormal motor

TABLE I
ALL IoT DEVICES USED IN THE MINDER PLATFORM

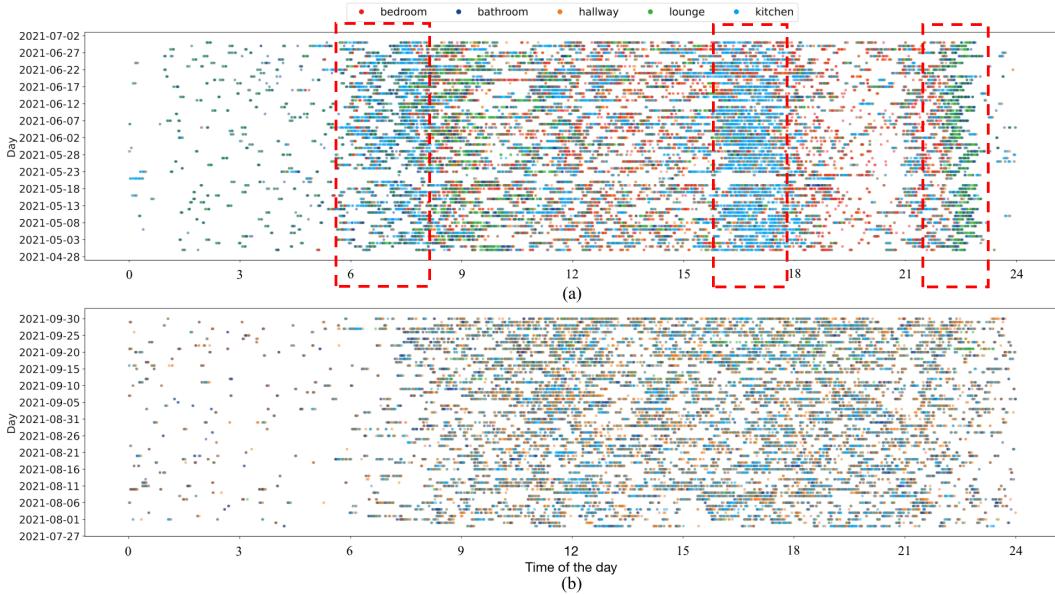| Digital Marker	Monitoring Device	Frequency
Human activity	Passive infrared sensors	Triggered by movement
Home device usage	Smart plugs	Triggered by device use
Body temperature	Smart temporal thermometers	Twice daily or continuous
Blood pressure and heart rate	Wearable devices	Twice daily
Weight and heart rate	Smart scale	Once a day
Respiratory and heart rate during sleep	Sleep mat	Once a minute
Environmental light	Light sensors	Every 15 minutes
Environmental temperature	Temperature sensors	Once an hour

Fig. 5. An example of a PLWD with clear routine activities (a), and another PLWD with fewer routine activities (b). The participant with more routine activities tends to have a more consistent daily activity pattern at the same time each day, as shown in the red boxes. The x-axis shows the time of the day, the y-axis shows different days, and the different colors represent different locations in the house.

function behavior, hospital admissions, Urinary Tract Infections, anxiety and depression, agitation, confusion, and disturbed sleep patterns. Participants who have experienced such events will have labeled data for that adverse health event.

In this study, we focus on the activity data of Minder only. This includes 3762 person-weeks of data collected between December 2020 and March 2022. The mean age of participants is 79. All of the data presented here has been anonymized.

Activity data in the Minder platform is collected using PIR sensors installed in various locations, including the kitchen, bathroom, bedroom, lounge, and hallway, as shown in Fig. 2. The PIR sensor logs an event with seconds precision and a 30-second delay when a person passes by. The recorded data is time-series data, showing the household's life patterns over time. We can identify clear differences in behaviors by visualizing the raw data, as shown in Fig. 5, which compares the routine activities of two PLWDs.

B. Epileptic Seizure Recognition Dataset

The ESRD (Epileptic Seizure Recognition Dataset) contains 11,500 time-series EEG signal data samples from 500 subjects and is used to study EEG signal changes during seizures [26]. Each sample consists of 23 segments containing 178 data points over a one-second interval. The UCI preprocessed the original dataset and randomly rearranged the segments to form the 11,500 time-series EEG signal samples from 500 subjects. The dataset includes five different health conditions, including one related to epileptic seizures, and four normal conditions where the subjects do not show symptoms of epilepsy. However, many researchers choose to perform binary classification to distinguish between class 1 (representing epileptic seizures) and other classes. Our goal is also to distinguish between healthy participants and those with epileptic seizures.

C. PTB Diagnostic ECG Database

The PTB Diagnostic ECG Database (PTBDB) is a collection of 549 records from 290 subjects (209 male,

and 81 female) [13], [14]. The age range of participants is 17 to 87 years old, with an average age of 57.2. The sampling frequency is 125Hz. The Diagnostic class includes myocardial infarction, cardiomyopathy/heart failure, bundle branch block, dysrhythmia, myocardial hypertrophy, valvular heart disease, myocarditis, miscellaneous, and healthy controls. In this study, we extract heartbeat signals and only use ECG lead 2 [27]. We focus on the myocardial infarction and healthy control categories, with a total of 14552 samples in the dataset. The histogram color maps for the PTB data marked as abnormal and normal are shown in Fig. 4.

IV. METHODOLOGY

The pipeline proposed in this paper is mainly composed of three parts: data preprocessing, feature construction, and modeling.

The data preprocessing phase includes missing value processing, data resampling, and label encoding. The missing values are forward-filled with the last valid value, then backfilled with the next valid value. Data resampling is determined by the characteristics of the data as well as the requirements of the target. For example, if a dataset has a low sample size, but narrowing the time window has little impact on the target results, then resampling will be performed to expand the dataset.

In the modeling stage, we use classical machine learning and deep learning models such as Logistic Regression (LR), Support Vector Machine (SVM), Multilayer perceptron (MLP), Convolutional neural network (CNN), and Long Short-term Memory (LSTM).

The following introduces the feature construction stage, including the entropy and entropy variants used in this study, and the feature selection methods.

A. Entropy and Entropy Variants

1) Shannon's Entropy of a Markov chain: Assuming that a certain human activity (*e.g.*, a sequence of locations) can form a Markov chain, then we can regard the occurrence of these activities as random events, and measure the extent of occurrence of these random events. We apply Shannon's entropy of a Markov chain to represent pattern changes in human activity data. Suppose that there are n locations $X = x_1, x_2, ..., x_n$ in a participant's activity, then the Shannon's entropy of a Markov chain $H(x)$ can be described as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{1}$$

In which $P(X_i)$ is the probability of activity x_i. When the frequency of a participant's activity changes, $H(x)$ will change accordingly to represent the change in activity pattern.

2) Entropy Rate of a Markov Chain: Shannon's entropy of a Markov chain does not link the activities in a Markov chain together, but only treats each activity as a separate individual. However, if we utilize the first-order Markov chain to profile the human activities and collect these activities together, we can get the corresponding transitions, where the current activity event of a participant is only dependent on the preceding activity event [28]. Suppose that $X = \{x_1, x_2, ..., x_n\}$ represents n states in a Markov chain. Let $x_i, x_j \in X$, represent the previous state and the current state, respectively. Then the probability P_{ij} of the route from x_i to x_j can be represented as:

$$P_{ij} = P(x_j \mid x_i) \tag{2}$$

Where x_i and $x_j \in X$. Suppose that there are n states in a Markov chain, then the Markov chain can be represented as $n \times n$ matrix $P_{ij_{i,j \in X}}$, called Transition Matrix T, an example is shown in Fig. 6. From Markov chains, stationary distributions π can be calculated, which represent:

$$\pi = \pi T \tag{3}$$

In which, π is an n-dimension vector associated with a Markov chain with n states. Using this, the entropy rate of a Markov chain can be expressed as [25]:

$$\xi = -\sum_{ij}^{n} \pi_i P_{ij} \log P_{ij} \tag{4}$$

In which, π_i is the probability in the stationary distribution associated with activity $x_i \in X$ in a Markov chain with the stationary distribution. When calculating the entropy rate of a Markov chain, there are two time-windows that need to be set, one time-window is used to calculate P_{ij} for target time-series data, and the other is used to calculate π_i to represent the characteristics of time-series data. The time window to calculate P_{ij} is set by the mission objective. And it has to be noted that the time window to calculate the stationary distribution π_i is important, as it should reflect the stationary pattern of the participant. For example, participants' routines may be affected by the seasons, then we need to avoid the possible effects of the seasons when setting up the time windows to calculate the stationary distribution, such as setting the time windows to override the seasonal variations. The complete procedure for calculating the Entropy Rate of a Markov Chain is shown in the Algorithm 1.

3) Entropy Production of a Markov Chain: Entropy Production (EP) is a description of the diverse non-equilibrium principle [29], which is intended to describe physical processes. Physical processes can be described by stochastic processes, such as Markov chains and diffusion processes. The Markov chains generated by human activity data can be regarded as a stochastic
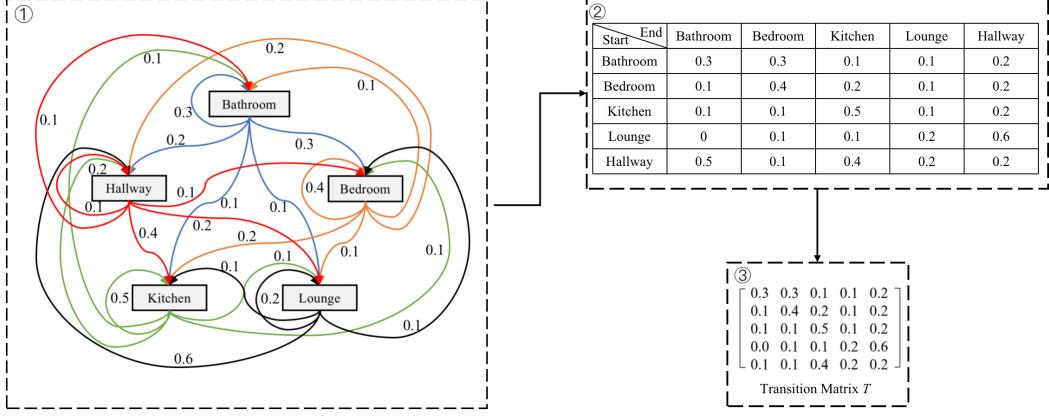
Fig. 6. An example of the Entropy rate of a Markov chain. In ①, the rectangular boxes represent the locations (states) in the Markov chain, and the arrows represent the routes between locations in the house. Different colors represent different start locations (blue: bathroom, orange: bedroom, green: kitchen, black: lounge, and red: hallway). The numbers next to the lines represent route probabilities which correspond to the table ② and Transition Matrix T ③.

Algorithm 1 Entropy rate of a Markov chain

1: Define: $S = \{s_1, s_2, ..., s_L\}$ is a Markov chain trajectory, where L is the length of the trajectory, and $s \in X, X = \{x_1, x_2, ..., x_n\}$, n is the number of states in the Markov chain. TW_1 is the time window required for the stationary distribution, $TW_1 <= L$. TW_2 is the time window required for the target task, $TW_2 <= L$. P_{ij} is the probability from state x_i to state x_j. SP is the start point;

Input: Markov chain trajectory S;
Output: Entropy rate ξ of the Markov chain;

2: Set TW_1 and TW_2;
3: **function** STATIONARYDISTRIBUTION(S, TW_1)
4:     $S_{TW_1} = S[0 : TW_1]$;
5:     $P'_{ij} = P\left(l'_b = x'_j \mid l'_a = x'_i\right)$, where $l'_a, l'_b \in X$, represent the previous state and the current state, $a' \in [2, TW_1], b' \in [1, TW_1 - 1], x'_i \in X, x'_j \in X$;
6:     $\pi = \pi T$;
7:     **return** π;
8: **end function**
9: **function** ENTROPYRATE(π, S, TW_2)
10:     **for** $SP = 0; SP + TW_2 <= L; SP = SP + TW_2$ **do**
11:         $S_{TW_2} = S[SP : TW_2]$;
12:         $P_{ij} = P\left(l_b = x_j \mid l_a = x_i\right)$, where $l_a, l_b \in X$, represent the previous state and the current state, $a \in [2, TW_2], b \in [1, TW_2 - 1], x_i \in X, x_j \in X$;
13:         $\xi_m = -\sum_{x_i, x_j \in X}^{n} \pi_i P_{ij} \log P_{ij}$;
14:     **end for**
15:     **return** $\xi = \{\xi_1, \xi_2, ..., \xi_m\}$;
16: **end function**

process [30]. Therefore, we can apply EP to Markov chains to describe the pattern changes.

EP can be estimated by ML models such as the Neu-

ral Estimator for Entropy Production (NEEP), which can estimate EP of Markovian systems [31]. Given a Markov chain trajectory $S = \{s_1, s_2, ..., s_L\}$ and a function h_θ acting over previous state s_t and the current state s_{t+1} in the Markov chain, where θ denotes the trainable neural network parameters, then the output of NEEP can be defined as [31]:

$$\hat{J}(\theta) = \sum_{t \in L} \left[\Delta S_\theta\left(s_t, s_{t+1}\right) - e^{-\Delta S_\theta(s_t, s_{t+1})} \right] \quad (5)$$

Where ΔS_θ is:

$$\Delta S_\theta\left(s_t, s_{t+1}\right) \equiv h_\theta\left(s_t, s_{t+1}\right) - h_\theta\left(s_{t+1}, s_t\right) \quad (6)$$

The procedure for training NEEP is shown in Algorithm 2 and the model structure of NEEP is shown in Fig. 7. In NEEP, an embedding layer is used to transform the discrete state into a trainable continuous vector [31], then the embedded data is input into a hidden MLP layer. It has to be noted that, the length of the time series data is very important when training NEEP, as we need to ensure that the data for this period of time is sufficient for training and can reflect the participant's characteristics.

4) von Neumann Entropy of a Markov Chain: The von Neumann entropy (VNE) quantifies the amount of information present in a system, which can be applied to time-series data to quantify the fluctuation and the correlation of the data [32]. For a density operator ρ with N eigenvalues $\lambda_{1,...,n}$, VN is defined as follows:

$$S(\boldsymbol{\rho}) = -\operatorname{tr}(\boldsymbol{\rho} \log \boldsymbol{\rho}) = -\sum_{j=1}^{N} \lambda_j \log \lambda_j \quad (8)$$

We apply VNE to the human activity data with stochastic state transitions to reflect the pattern change

Algorithm 2 Training process of NEEP

1: Define: $S = \{s_1, s_2, ..., s_L\}$ is a Markov chain trajectory, where L is the length of the trajectory, and $s \in X, X = \{x_1, x_2, ..., x_n\}$, n is the number of states in the Markov chain.

Input: Markov chain trajectory S;

Output: The values calculated by the loss function $\hat{J}(\theta)$;

2: **loop**

3:     Embedding layer;

4:     Objective function

$$\hat{J}(\theta) = \sum_{t \in L} \left[\Delta S_\theta(s_t, s_{t+1}) - e^{-\Delta S_\theta(s_t, s_{t+1})} \right] \quad (7)$$

5:     Compute gradients $\nabla_\theta \hat{J}(\theta)$;

6:     Update parameters θ;

7: **end loop**

Algorithm 3 von Neumann entropy of a Markov chain

1: Define: $S = \{s_1, s_2, ..., s_L\}$ is a Markov chain trajectory, where L is the length of the trajectory, and $s \in X, X = \{x_1, x_2, ..., x_n\}$, n is the number of states in the Markov chain. VN is the von Neumann entropy of a Markov chain. TW_2 is the time window required for the target task, where $TW_2 <= L$. SP is the start point;

Input: Markov chain trajectory S;

Output: The VN;

2: Set TW_2;

3: **for** $SP = 0$; $SP + TW_2 <= L$; $SP = SP + TW_2$ **do**

4:     Calculate original matrix A (e.g., activity frequency matrix);

5:     $N \times N$ Pearson correlation Matrix R of A;

6:     Density operator $\rho \leftarrow \rho = R/N$;

7:     von Neumann entropy $VN(\rho) \leftarrow VN(\rho) = \text{Tr}(\rho \log \rho)$, $log\rho = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(B-I)^k}{k}$;
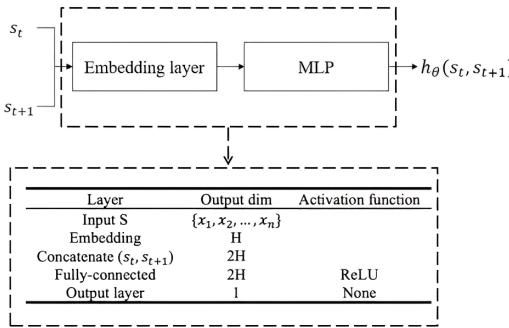
8: **end for**

Fig. 7. The model structure of NEEP of Markovian systems. H is the size of the embedding dimension.

of the data. The human activity data of a Markov chain can be analyzed by VN from spatial and temporal perspectives, for example, illustrated in Fig. 8. One of the key points to calculate VN is to obtain the density operator ρ, which must satisfy (i) be Hermitian, (ii) have unit trace, and (iii) be positive semidefinite. Given $\boldsymbol{R} \in \mathbb{R}^N$, an N-dimension Pearson correlation matrix of the human activity data, then the density operator $\boldsymbol{\rho}$ can be defined as [33]:

$$\boldsymbol{\rho} = \boldsymbol{R}/N \quad (9)$$

The density operator ρ, calculated by Eq. (9) satisfies all the requirements. However, it has to be noted that the density operator ρ, which is calculated from real IoT data, may be sparse and thus there may be anomalies in the calculation of $log\rho$ using standard classical mathematical methods. Therefore, we calculate $\log \rho$ by Mercator's Series. Suppose B is a matrix and sufficiently close to the identity matrix I, and $\|B - I\| < 1$, then a logarithm of B can be computed by means of the following k-power series [34]:

$$\log(B) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(B-I)^k}{k} \quad (10)$$

This means, we can obtain $\log \rho$ by:

$$\log(\boldsymbol{\rho}) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\boldsymbol{\rho} - I)^k}{k} \quad (11)$$

Integrating Eq. (8), Eq. (9) and Eq. (10), the VN can be obtained. The complete procedure for calculating VNE is shown in the Algorithm 3.

5) Approximate Entropy: For Non-Markovian chain systems, Approximate Entropy (ApEn) can be used to quantify the complexity of the system. Given a time series dataset $\{u(i) : 1 \le i \le N\}$ with N samples, form the sequence in order to generate an m-dimension vector:

$$u'(i) = [u(i), u(i+1), \ldots, u(i+m-1)] \quad i = 1, N-m+1 \quad (12)$$

Define the distance between the vectors $u'(i)$ and $u'(j)$ to be the maximum of the differences between the corresponding elements of the two vectors:

$$d[u'(i), u'(j)] = \max_{k=0, m-1} [|u'(i+k) - u'(j+k)|] \quad (13)$$

Given a threshold p, count the number of $d[u'(i), u'(j)] <= p$, denoted as $A_N^m(p)$, and calculate the ratio of $A_N^m(p)$ to $N - m + 1$, denoted as $B_N^m(p)$:

$$B_l^m(p) = \frac{A_l^m(p)}{N-m+1} \quad (14)$$

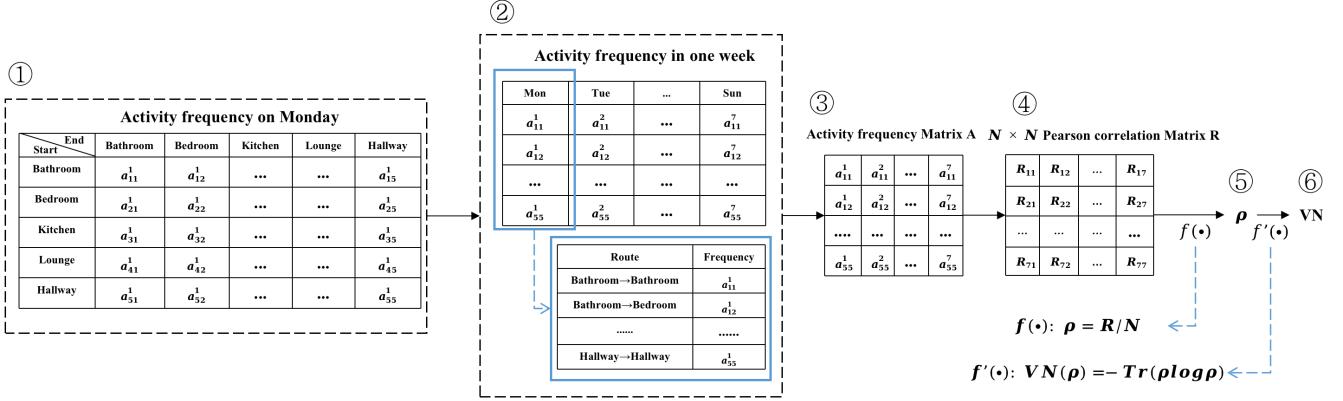Calculate the average value of $B_N^m(p)$:

Fig. 8. An example for the von Neumann entropy of a Markov chain. Suppose that there are five locations (states) in a Markov chain, and we plan to calculate the von Neumann entropy of one week. From the perspective of spatial, ①: count the frequency $a_{i,j}^d$ of different routes in a Markov chain for each day of the week, in which i and j represent the previous location and the current location, respectively; ②: aggregate weekly activity frequency; ③: transfer the weekly activity frequency to the activity frequency matrix A; ④: calculate the Pearson correlation Matrix R between each day; ⑤: calculate the density matrix ρ by $f(\bullet)$; ⑥: calculate the von Neumann entropy by $f'(\bullet)$. And from the perspective of temporal, the only difference is changing the activity frequency to activity duration.

$$B^m(p) = \frac{1}{N-m+1} \sum_{N=1}^{N-m+1} B_N^m(p) \qquad (15)$$

Increase the dimension from m to $m+1$, and repeat the above steps. For sequences of finite length, an estimate of the sample entropy can be obtained as [35]:

$$\mathrm{ApEn}(m,r,N) = B^m(p) - B^{m+1}(p) \qquad (16)$$

6) Increment Entropy: The Incremental Entropy (In-crEn) algorithm is a method for calculating the entropy of a sequence of data points incrementally, rather than computing the entropy of the entire sequence all at once. Given a time series dataset $\{u(i) : 1 \le i \le N\}$ with N samples. Construct an increment time series $\{v(i), 1 \le i \le N-1\}$ by $v(i) = x(i+1) - x(i)$ from $u(i)$. Hence, for a positive integer m, $N-m$ vectors of dimension m are derived from an incremental time series. These vectors, denoted as $V(k) = [v(k), v(k+1), \ldots, v(k+m-1)], 1 \le k \le N-m$, represent contiguous segments of the time series. Each element in a vector $V(k)$ is mapped onto a word of two letters. The sign of each component is represented by $v'_{k+j} = \mathrm{sgn}(v(k+j)), j = 1\cdots, m-1$, and the magnitude of each component in relation to the other components within the vector is represented by $q_{k+j}, j = 1, \ldots, m-1$ for a quantifying resolution r. As a result, $N-m$ words, $w_k, 1 \le k \le N-m$, are generated. Each word, consisting of $2 \times m$ letters, can have $(2r+1)^m$ variations, depending on the values of m and r. The frequency of occurrence of each unique word w_n is defined as:

$$p(w_n) = \frac{Q(w_n)}{N-m} \qquad (17)$$

where $Q(w_n)$ signifies the count of the unique word w_n within the $\{w_k\}$. The Increment Entropy (IncrEn) of order m (where m is equal to or greater than 2) and resolution R is defined as:

$$IncrEn(m) = -\sum_{n=1}^{(2R+1)^m} p(w_n) \log p(w_n) \qquad (18)$$

7) Dispersion Entropy: Dispersion entropy (DE) can be used to describe the complexity of time series data. For time series with low regularity, DE can reflect the degree of disorder of the series [36]. Given a time series dataset $\{u(i) : 1 \le i \le N\}$ with N samples. Map $u(i)$ to $y(i)$ between 0 and 1 by normal cumulative distribution function (NCDF):

$$y_j = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{u_j} e^{-\left((t-\mu)^2/2\sigma^2\right)dt} \qquad (19)$$

In which, the parameter μ is the expectation of $u(i)$, while the parameter σ is its standard deviation. Map y to the range of $[1, 2, ..., c]$, and obtain a new sequence $z_j^{(c)}$:

$$z_j^{(c)} = \mathrm{int}\,(cy_j + 0.5) \qquad (20)$$

In which, c is the number of categories, and int is the rounding function. Then construct the embedding vector $z_i^{(m,c)}$ by:

$$z_i^{(m,c)} = \left(z_i^{(c)}, z_{i+d}^{(c)}, \cdots, z_{i+(m-1)d}^{(c)}\right),$$
$$i = 1, 2, \cdots, N - (m-1)d \qquad (21)$$

In which, m is the embedding dimension, c is the number of class, d is the time delay. Then each $z_j^{(m,c)}$ is mapped to dispersion pattern $\pi_{v_0 v_1 \cdots v_{m-1}}(v = 1, 2, \cdots, c)$, in which $z_i^{(c)} = v_0$, $z_{i+d}^{(c)} = v_1$, ..., and
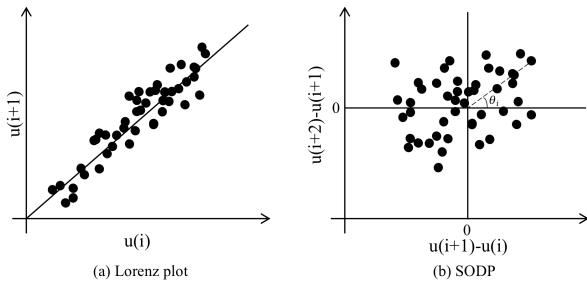
(a) Lorenz plot (b) SODP

Fig. 9. The phase space representation of an HRV signal.

$z_{i+(m-1)d}^{(c)} = v_{m-1}$. The number of possible dispersion of each $z_j^{(m,c)}$ is c^m.

Calculate the relative frequency for each potential dispersion pattern:

$$P\left(\pi_{v_0 v_1,\cdots,v_{m-1}}\right) = \frac{\text{num}\left(\pi_{v_0 v_1,\cdots,v_{m-1}}\right)}{N-(m-1)d} \qquad (22)$$

Finally, based on Shannon's entropy, DE can be obtained by [37]:

$$DE(u,m,c,d) = -\sum_{\pi=1}^{c^m} p\left(\pi_{v_0,...,v_{m-1}}\right) \ln\left(p\left(\pi_{v_0,...,v_{m-1}}\right)\right) \qquad (23)$$

8) Phase Entropy: Phase entropy (PhEn) is developed to detect the complexity of physiological signals. For example, given a time series dataset $\{u(i) : 1 \le i \le N\}$ with N samples, we can represent the data by the Lorenz plot, as Fig. 9 (a) shows. In the Poincaré plot, if we replace the sequence u_i by $u_{i+1} - u_i$, then we can get SODP plot, as Fig. 9 (b) shows. Specifically, from a given time series u_i, we can obtain Y_i and X_i by [38]:

$$\begin{aligned} Y_i &= u_{i+2} - u_{i+1} \\ X_i &= u_{i+1} - u_i \end{aligned} \qquad (24)$$

Then compute the slope angle of each scatter point as shown in Fig (b).

$$\theta_i = \tan^{-1}\frac{Y_i}{X_i} \qquad (25)$$

Then the probability distribution p_i can be calculated by:

$$p_i = \frac{S_{\theta_i}}{\sum_{i=1}^{k} S_{\theta_i}} \qquad (26)$$

Finally, based on Shannon's entropy, the PhEn can be calculated as [38]:

$$\text{PhEn} = \frac{-1}{\log N}\sum_{i=1}^{k} p(i)\log p(i) \qquad (27)$$

9) Slope Entropy: Slope Entropy (SlopEn) is an algorithm to describe the complexity of a time series dataset, which is primarily based on transferring the original time series data to a series of single-threshold and symbolic patterns [39], [40]. Given a time series dataset $\{u(i) : 1 \le i \le N\}$ with N samples. Decompose u into j subsequences according to the embedded dimension m:

$$u_i^m = \{u_i, u_{i+1}, \cdots, u_{i+m-1}\} \qquad (28)$$

In which, $i = \{1,2,...,j\}$, $j = N - m + 1$. Define two soft threshold parameters δ and γ to calculate the symbolic patterns of u_i^m, where $0 < \delta < \gamma$.

Define $d = u_{i+1} - u_i$, and compare d with the two soft threshold parameters δ and γ, then five patterns can be obtained:

$$\begin{cases} \text{pattern} = 2, & \gamma < d, \\ \text{pattern} = 1, & \delta < d \le \gamma, \\ \text{pattern} = 0, & |d| \le \delta, \\ \text{pattern} = -1, & -\gamma \le d < -\delta, \\ \text{pattern} = -2, & d < -\gamma. \end{cases} \qquad (29)$$

Based on the five patterns, we can get 5^{m-1} sequence combinations. The relative frequency p_n of the the combination can be calculated by the number of occurrences f_n of each combination:

$$p_n = \frac{f_n}{j}, n = 1,2,\cdots,5^{m-1} \qquad (30)$$

Finally, SlopEn can be calculated based on the Shannon's entropy:

$$\text{SE}(m,\gamma,\delta) = -\sum_{n=1}^{5^{m-1}} p_n \ln p_n \qquad (31)$$

B. Feature Selection

For feature selection, if the dataset is with stochastic state transitions and can be constructed as a Markov chain, we prioritize the entropy features associated with Markov chains, because linking the time-series data together to form Markov chains can potentially mine more information. For the dataset where state-space modeling is not applicable, we utilize mathematical statistics such as mutual information and the Pearson relationship matrix for filtering.

1) Minder Database: As Fig. 5 shows, the original data of the Minder Database mainly includes the time and location where the infrared sensors were triggered, and the original data can be reconstructed into Markov chains to reflect the activity routes of the participants. Therefore, we prioritize the entropy features associated with Markov chains, including Shannon's entropy, the entropy rate of a Markov chain, the entropy production
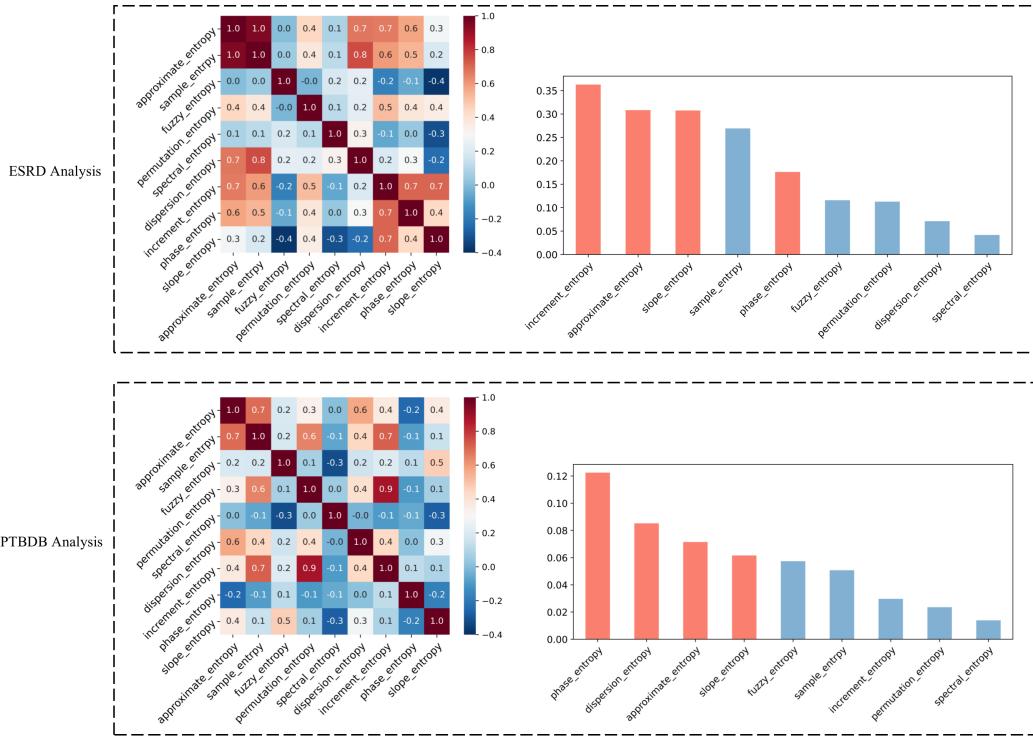
Fig. 10. Feature selection of ESRD and PTBDB, including Pearson relationship matrices (the left) and mutual information (the right). The coordinates of the Pearson correlation matrices and the horizontal coordinates of the mutual information represent some common entropy measures. In mutual information features, the red represents the final selection of entropy features. For ESRD, according to the mutual information, although the most four important features are increment entropy, approximate entropy, slope entropy, and sample entropy, the Pearson correlation coefficient is too high between approximate entropy and sample entropy. Thus we finally select phase entropy to replace sample entropy.

of a Markov chain, and the von Neumann entropy of a Markov chain (from the perspective of spatial and temporal).

2) ESRD and PTBDB: As Fig. 3 and Fig. 4 show, the data from ESRD and PTBDB is collected by wearable sensors, and it is hard to generate Markov chains. Thus we apply mutual information and Pearson relationship matrices to ESRD and PTBDB to select approximate entropy features, as Fig. 10 shows.

V. Modeling and Results

We utilize classical models to evaluate the entropy features, including Logistic Regression (LR), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Convolutional neural network (CNN), and Long-short Term Memory (LSTM).

A. Minder Database

We evaluate the performance of LR, SVM, MLP, and LSTM on the Minder database. Since our focus is on identifying whether a participant has had any non-healthy events, we use recall rate, F1 score, and accuracy as evaluation methods. Additionally, we consider the effect of sundowning and circadian rhythms in

people living with dementia (PLWD) [41] by dividing one day into two time periods: daytime (06:00 - 18:00) and night (18:00 - 24:00 and 00:00 - 6:00). The baseline features are average frequency of bathroom, bedroom, hallway, kitchen, and lounge in each week (daytime and night). The entropy features are Shannon's entropy of Markov chains, Entropy rate of Markov chains, EP

TABLE II
The average performance of the models for the Minder

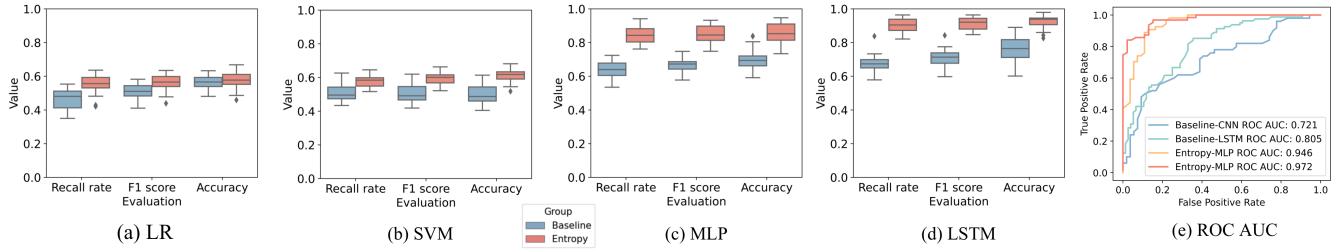|	Evaluation	Baseline	Entropy	Improvement
LR	Recall rate	46.42±6.11%	55.41±5.67%	8.79%
	F1 score	50.76±4.53%	56.59±4.67%	5.46%
	Accuracy	56.52±4.13%	58.03±4.70%	1.52%
SVM	Recall rate	51.11±5.24%	57.84±3.35%	6.73%
	F1 score	50.61±5.48%	59.34±3.65%	8.73%
	Accuracy	50.13±5.76%	60.94±4.03%	10.81%
MLP	Recall rate	63.76±5.03%	84.16±5.21%	20.40%
	F1 score	66.59±3.71%	84.97±4.97%	18.38%
	Accuracy	70.15±5.85%	85.88±5.51%	15.73%
LSTM	Recall rate	67.28±4.99%	**90.29±4.41%**	**23.01%**
	F1 score	71.25±4.99%	**91.29±3.72%**	**20.04%**
	Accuracy	76.06±7.15%	**92.41±4.18%**	**16.35%**
Average	Recall rate	-	-	**14.73**%
	F1 score	-	-	**13.15**%
	Accuracy	-	-	**11.10**%

Fig. 11. The evaluation results of the Minder database. (a), (b), (c), and (d) are the results of LR, SVM, MLP, and LSTM, respectively. The x-axis represents different evaluation methods. And (e) is the ROC-AUC curves for MLP and LSTM models. From (a), (b), (c), and (d), we can find that, for four different models, compared with the baseline features, modeling by the entropy features can improve the recall rate, F1 score, accuracy, and AUC score. Especially for LSTM, the entropy features can improve the recall rate (90.29%), F1 score (91.29%), and accuracy (92.41%) by about 23.01%, 20.04%, and 16.35%.

of Markov chains, VNE of Markov chains (activity frequency), VNE of Markov chains (activity duration), and activity duration difference of Markov chains in each week (daytime and night). The output of the models is healthcare-related events (True or False).

LR: Model parameters of the baseline features: penalty = L2, solver = sag, class weight = balanced, random state = 10, test size = 0.3, repeat times = 30. Model parameters of the entropy features: penalty = L2, solver = sag, class weight = balanced, random state = 10, test size = 0.3, repeat times = 30.

SVM: Model parameters of the baseline features: kernel = linear, test size = 0.3, repeat times = 30. Model parameters of the entropy features: kernel = linear, test size = 0.3, repeat times = 30.

MLP: Model parameters of the baseline features: input layer (10), hidden layer $(10 \times 30, 30 \times 30)$, output layer (30×1), activation functions = $(tanh, tanh, sigmoid)$, epochs = 3000, batch size = 256, learning rate = 0.15, criterion = Binary Cross-Entropy, optimizer = SGD, test size = 0.3, repeat times = 30. Model parameters of the entropy features: input layer (1×12), hidden layer $(12 \times 50, 50 \times 50)$, output layer (50×1), activation functions = $(tanh, tanh, sigmoid)$, epochs = 5000, batch size = 256, learning rate = 0.06, criterion = Binary Cross-Entropy, optimizer = SGD, test size = 0.3, repeat times = 30.

LSTM: Model parameters of the baseline features: input layer (10), hidden layer $(10 \times 30, 30 \times 30)$, output layer (30×1), activation functions = $(tanh, tanh, sigmoid)$, epochs = 5000, batch size = 256, learning rate = 0.15, criterion = Binary Cross-Entropy, optimizer = SGD, timesteps = 3, test size = 0.3, repeat times = 30. Model parameters of the entropy features: input layer (1×12), hidden layer $(12 \times 50, 50 \times 50)$, output layer (50×1), activation functions = $(tanh, tanh, sigmoid)$, epochs = 5000, batch size = 256, learning rate = 0.4, criterion = Binary Cross-Entropy, optimizer = SGD, timesteps = 3, test size = 0.3, repeat times = 30.

The results of the Minder Database are shown in

TABLE III
Comparison of ESRD classification results

	Recall rate	F1 score	Accuracy
Baseline-CNN	77.27±1.99%	80.91±1.54%	84.93±1.38%
Baseline-LSTM	96.01±1.47%	95.21±0.66%	94.46±1.11%
Entropy-MLP	**97.51±0.81%**	**97.80±0.45%**	**98.10±0.72%**
Avg Improvement	**10.87%**	**9.74%**	**8.41%**

Fig. 11 and Table. II. We can find that, compared with the baseline features, modeling with the entropy features can improve the recall rate, F1 score, and accuracy on average by 14.03%, 13.86%, and 11.10%. Especially for LSTM, compared with the model build using baseline features, the recall rate (90.29%), F1 score (91.29%), and Accuracy (92.41%) are improved by 23.01%, 20.04%, and 16.35%.

B. Epileptic Seizure Recognition Dataset

We aim to differentiate between the normal participants and those with epileptic seizures. The baseline models are LSTM and CNN with complete data. The entropy model is MLP with IncrEn, ApEn, SlopEn, and PhEn. The output of the models is Participants with epileptic seizures (True or False).

The Baseline-CNN: Max pooling-1d layer 1 $(-1 \times 89 \times 1)$, Conv-1d layer 1 $(-1 \times 89 \times 16)$, Max pooling-1d layer 2 $(-1 \times 44 \times 16)$, Conv-1d layer 2 $(-1 \times 44 \times 8)$, Flatten layer (-1×352), Dense layer 1 (-1×250), Dense layer 2 (-1×2), activation functions = $Relu$, epochs = 1000, batch size = 256, learning rate = 0.0001, criterion = sparse categorical crossentropy, optimizer = adam, test size = 0.3, repeat times = 30.

The Baseline-LSTM: LSTM layer 1 $(-1 \times 178 \times 64)$, LSTM layer 2 $(-1 \times 178 \times 32)$, LSTM layer 3 $(-1 \times -1 \times 8)$, Flatten layer (-1×8), Dense layer 1 (-1×250), Dense layer 2 (-1×2), activation functions = $Relu$, epochs = 1000, batch size = 256, learning rate = 0.0001, criterion = sparse categorical crossentropy, optimizer = adam, test size = 0.3, repeat times = 30.

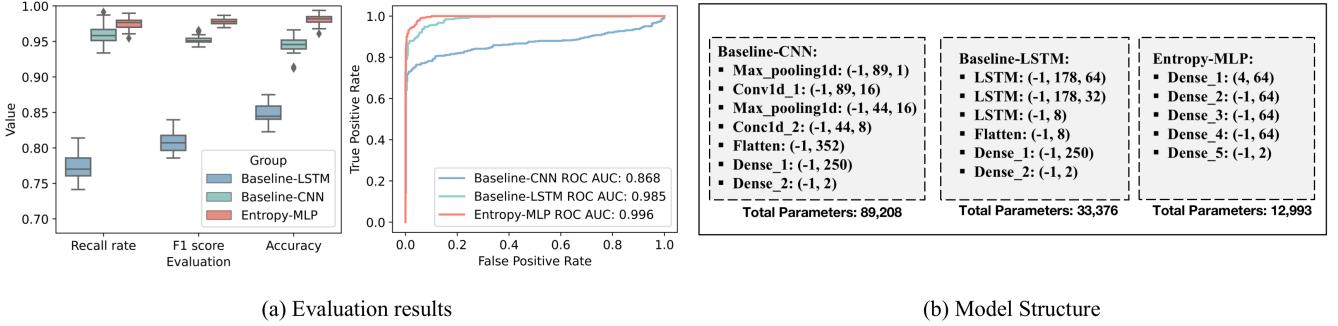(a) Evaluation results

(b) Model Structure

Fig. 12. The results of ESRD. (a) is the evaluation results, including recall rate, F1 score, accuracy, and ROC-AUC. (b) is the comparison of the model structure between the baseline and entropy models. From (a), we can find that the recall rate, F1 score, and accuracy of the entropy-MLP models can be improved to up to 97.51%, 97.80%, and 98.10%. As we used pre-processed data with fewer data noise, the AUC-ROC performances of all the models are ideal. From (b), we can find that the entropy-MLP model is reduced by 5.86 times and 1.59 times in terms of total parameters.

The Entropy-MLP: Dense layer 1 (4×64), Dense layer 2 (-1×64), Dense layer 3 (-1×64), Dense layer 4 (-1×64), Dense layer 5 (-1×2), activation functions = $tanh$, epochs = 2000, batch size = 256, learning rate = 0.3, criterion = Binary Cross-Entropy, optimizer = SGD, test size = 0.3, repeat times = 30.

The results of the ESRD Database are shown in Fig. 12 and Table. III. We can find that, compared with the baseline models, modeling with the entropy features can improve the recall rate, F1 score, and accuracy by up to 10.87%, 9.74%, and 8.41% on average. For the model structure, compared with the Baseline-LSTM and Baseline-CNN, Entropy-MLP can reduce by 5.86 times and 1.59 times.

C. PTBDB

We aim to distinguish the ordinary participants and the participants with any heart disease. The baseline models are MLP and CNN with complete data. The entropy model is MLP with PhEn, DE, ApEn, and FuzzyEn. The output of the models is the participants with any heart disease (True or False).

The Baseline-MLP: Dense layer 1 (187×64), Dense layer 2 (-1×64), Dense layer 3 (-1×64), Dense layer 4 (-1×64), Dense layer 5 (-1×2), activation functions = $Relu$, epochs = 1000, batch size = 256, learning rate = 0.0001, criterion = sparse categorical crossentropy, optimizer = adam, test size = 0.3, repeat times = 30.

The Baseline-CNN: Conv-1d layer 1 ($-1 \times 187 \times 64$), Conv-1d layer 2 ($-1 \times 187 \times 64$), Max pooling-1d layer ($-1 \times 94 \times 64$), Dropout layer ($-1 \times 94 \times 64$), Flatten layer (-1×6016), Dense layer 1 (-1×32), Dense layer 1 (-1×2), activation functions = $Relu$, epochs = 1000, batch size = 256, learning rate = 0.0001, criterion = sparse categorical crossentropy, optimizer = adam, test size = 0.3, repeat times = 30.

The Entropy-MLP: Dense layer 1 (4×64), Dense layer 2 (-1×64), Dense layer 3 (-1×64), Dense layer 4 (-1×64), Dense layer 5 (-1×2), activation functions =

TABLE IV
Comparison of PTBDB classification results

	Recall rate	F1 score	Accuracy
Baseline-MLP	91.68+1.29%	92.54+0.73%	93.42+0.40%
Baseline-CNN	97.14+1.12%	97.13+0.67%	97.13+0.51%
Entropy-MLP	**98.08+0.94%**	**98.37+0.81%**	**98.66+0.79%**
Avg Improvement	**3.67%**	**3.54%**	**3.39%**

$tanh$, epochs = 2000, batch size = 256, learning rate = 0.3, criterion = Binary Cross-Entropy, optimizer = SGD, test size = 0.3, repeat times = 30.

The results of the PTBDB Database are shown in Fig. 13 Table. IV. We can find that, compared with the Baseline-MLP and Baseline-CNN, Entropy-MLP can achieve better performance with a simpler model structure, and reduce the number of model structure parameters by 6.19 times and 61.35 times. And the Entropy-MLP can improve the recall rate, F1 score, and accuracy to 98.08%, 98.37%, and 98.66%.

VI. Conclusions

We propose a novel method for analyzing multivariate time-series data using information theory-based features analysis methods. Our approach utilizes entropy-based features. For applications with stochastic state transitions, we utilize Shannon's entropy of Markov chains, entropy rates of Markov chains, entropy production of Markov chains, and von Neumann entropy of Markov chains to analyze pattern changes in the data. Additionally, for applications where state transition modeling is not applicable, we used five classical entropy and entropy variants, and introduce the entropy feature selection method (by mutual information and Pearson relationship matrix).

The results show that, compared with the baseline, the entropy-based models improve the recall rate, F1 score, and accuracy on average by up to 23.01%. We
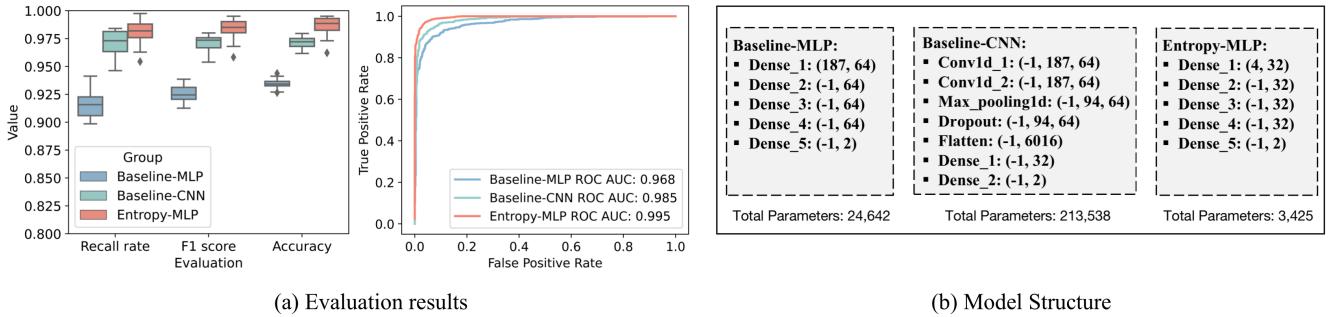
(a) Evaluation results (b) Model Structure

Fig. 13. The results of PTBDB database. (a) is the evaluation results. (b) is the comparison of the model structure between the baseline and entropy models. From (a), we can find that the recall rate, F1 score, and accuracy of the entropy-MLP models can be improved to up to 98.08%, 98.37%, and 98.66%. As we used pre-processed data with fewer data noise, the AUC-ROC performances of all the models are ideal. From (b), we can find that, the entropy-MLP model is reduced by 6.19 times and 61.35 times in terms of total parameters.

also compared the entropy-based model with state-of-the-art deep learning models on ESRD and PTBDB. And the results show that the entropy based model can achieve better performances on the recall rate, F1 score, and accuracy, with an average reduction of 18.75 times in the number of model parameters.

The proposed pipeline offers a versatile, high-precision, and interpretable solution for analyzing time series data from the perspective of information theory, which can be applied to various forms of time series data, such as those in the fields of IoT, intelligent systems, and data security.

References

[1] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.

[2] S. Selvaraj and S. Sundaravaradhan, "Challenges and opportunities in iot healthcare systems: A systematic review," *SN Applied Sciences*, vol. 2, no. 1, pp. 1–8, 2020.

[3] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI Conf. on Artificial Intel.*, vol. 33, 2019, pp. 1409–1416.

[4] F. Piccialli, F. Giampaolo, E. Prezioso, D. Camacho, and G. Acampora, "Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion," *Information Fusion*, vol. 74, pp. 1–16, 2021.

[5] T. Tao, E. Zio, and W. Zhao, "A novel support vector regression method for online reliability prediction under multi-state varying operating conditions," *Reliability Engineering & System Safety*, vol. 177, pp. 35–49, 2018.

[6] G.-J. Both, S. Choudhury, P. Sens, and R. Kusters, "Deepmod: Deep learning for model discovery in noisy data," *Journal of Computational Physics*, vol. 428, p. 109 985, 2021.

[7] J. Zhu, R. San-Segundo, and J. M. Pardo, "Feature extraction for robust physical activity recognition," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–16, 2017.

[8] L. C. Portugal, J. Schrouff, R. Stiffler, M. Bertocci, G. Bebko, H. Chase, J. Lockovitch, H. Aslam, S. Graur, T. Greenberg, *et al.*, "Predicting anxiety from wholebrain activity patterns to emotional faces in young adults: A machine learning approach," *NeuroImage*, vol. 23, 2019.

[9] M. A. Reyna, C. Josef, S. Seyedi, R. Jeter, S. P. Shashikumar, M. B. Westover, A. Sharma, S. Nemati, and G. D. Clifford, "Early prediction of sepsis from clinical data: The physionet/computing in cardiology challenge 2019," in *Computing in Cardiology*, IEEE, 2019.

[10] A. Gonzalez-Vidal, P. Barnaghi, and A. F. Skarmeta, "Beats: Blocks of eigenvalues algorithm for time series segmentation," *IEEE Trans-*

actions on Knowledge and Data Engineering, vol. 30, no. 11, pp. 2051–2064, 2018.

[11] R. Rezvani, P. Barnaghi, and S. Enshaeifar, "A new pattern representation method for time-series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 7, 2019.

[12] Y. Huang, Y. Zhao, H. Haddadi, and P. Barnaghi, "Using entropy measures for monitoring the evolution of activity patterns," *arXiv preprint arXiv:2210.01736*, 2022.

[13] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, e215–e220, 2000.

[14] R. Bousseljot, D. Kreiseler, and A. Schnabel, "Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet," 1995.

[15] Y. Huang, *Entropypipeline*, https://github.com/yushan-huang/EntropyPipeline, 2023.

[16] S. Ding, H. Zhu, W. Jia, and C. Su, "A survey on feature extraction for pattern recognition," *Artif. Intel. Rev.*, vol. 37, no. 3, pp. 169–180, 2012.

[17] K. Xia, J. Huang, and H. Wang, "Lstm-cnn architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.

[18] S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep convlstm with self-attention for human activity decoding using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8575–8582, 2020.

[19] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[20] G. Powell and I. Percival, "A spectral entropy method for distinguishing regular and irregular motion of hamiltonian systems," *Journal of Physics A*, vol. 12, no. 11, 1979.

[21] J. S. Richman, D. E. Lake, and J. R. Moorman, "Sample entropy," in *Methods in Enzymology*, vol. 384, Elsevier, 2004, pp. 172–184.

[22] N. R. Nurwulan and B. C. Jiang, "Multiscale entropy for physical activity recognition," in *Proceedings of the 2020 2nd Asia Pacific Information Technology Conf.*, 2020, pp. 73–77.

[23] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Int. Conf. on Pervasive Comp.*, Springer, 2004, pp. 1–17.

[24] A. Howedi, A. Lotfi, and A. Pourabdollah, "Exploring entropy measurements to identify multi-occupancy in activities of daily living," *Entropy*, vol. 21, no. 4, p. 416, 2019.

[25] S. Enshaeifar, A. Zoha, A. Markides, S. Skillman, S. T. Acton, T. Elsaleh, M. Hassanpour,

A. Ahrabian, M. Kenny, S. Klein, *et al.*, "Health management and pattern analysis of daily living activities of people with dementia using in-home sensors and machine learning techniques," *PloSOne*, vol. 13, no. 5, 2018.

[26] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, 2001.

[27] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "Ecg heartbeat classification: A deep transferable representation," in *IEEE Int. Conf. on healthcare informatics)*, IEEE, 2018, pp. 443–444.

[28] D. Lampard, "A stochastic process whose successive intervals between events form a first order markov chain—i," *Journal of Applied Probability*, vol. 5, no. 3, pp. 648–668, 1968.

[29] R. Dewar, "Information theory explanation of the fluctuation theorem, maximum entropy production and self-organized criticality in non-equilibrium stationary states," *Journal of Physics A*, vol. 36, no. 3, p. 631, 2003.

[30] A. Abdelgawwad and M. Pätzold, "A 3d non-stationary cluster channel model for human activity recognition," in *2019 IEEE 89th Vehicular Technology Conf.*, IEEE, 2019, pp. 1–7.

[31] D.-K. Kim, Y. Bae, S. Lee, and H. Jeong, "Learning entropy production via neural networks," *Physical Review Lett.*, vol. 125, no. 14, 2020.

[32] I. Bengtsson and K. Życzkowski, *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge university press, 2017.

[33] H. Felippe, A. Viol, D. de Araujo, M. da Luz, F. Palhano-Fontes, H. Onias, E. Raposo, and G. Viswanathan, "The von neumann entropy for the pearson correlation matrix: A test of the entropic brain hypothesis," *arXiv preprint arXiv:2106.05379*, 2021.

[34] C. C. MacDuffee, *The theory of matrices*. Springer Science & Business Media, 2012, vol. 5.

[35] X. Chen, I. C. Solomon, and K. H. Chon, "Comparison of the use of approximate entropy and sample entropy: Applications to neural respiratory signal," in *IEEE Conf. Eng. in Medicine and Biology*, IEEE, 2006.

[36] M. Rostaghi and H. Azami, "Dispersion entropy: A measure for time-series analysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 610–614, 2016.

[37] M. Chakraborty, D. Mitra, *et al.*, "Automated detection of epileptic seizures using multiscale and refined composite multiscale dispersion entropy," *Chaos, Solutions & Fractals*, vol. 146, 2021.

[38] A. Rohila and A. Sharma, "Phase entropy: A new complexity measure for heart rate variability," *Physiol. Measureme.*, vol. 40, no. 10, 2019.

[39] Y. Li, L. Mu, and P. Gao, "Particle swarm optimization fractional slope entropy: A new time series complexity indicator for bearing fault diagnosis," *Fractal and Fractional*, vol. 6, no. 7, 2022.

[40] Y. Li, B. Tang, and Y. Yi, "A novel complexity-based mode feature representation for feature extraction of ship-radiated noise using vmd and slope entropy," *Applied Acoustics*, vol. 196, p. 108 899, 2022.

[41] L. Volicer, D. G. Harper, B. C. Manning, R. Goldstein, and A. Satlin, "Sundowning and circadian rhythms in alzheimer's disease," *American Journal of Psychiatry*, vol. 158, no. 5, pp. 704–711, 2001.