Breast Cancer Early Detection with Time Series Classification

Haoren Zhu
Hong Kong University of Science and Technology
Hong Kong
hzhual@cse.ust.hk

Pengfei Zhao*
BNU-HKBU United International College
Zhuhai, China
ericpfzhao@uic.edu.cn

Yiu-Pong Chan
Hong Kong Bio-rhythm R&D Company Limited
Hong Kong
wyliechan@biorhythm.hk

Hong Kang
Hong Kong Bio-rhythm R&D Company Limited
Hong Kong
kanghong@biorhythm.hk

Dik Lun Lee
Hong Kong University of Science and Technology
Hong Kong
dlee@cse.ust.hk

ABSTRACT

Breast cancer has become the leading cause of women cancer death worldwide. Despite the consensus that breast cancer early detection can significantly reduce treatment difficulty and cancer mortality, people still are reluctant to go to hospital for regular checkups due to the high costs incurred. A timely, private, affordable, and effective household breast cancer early detection solution is badly needed. In this paper, we propose a household solution that utilizes pairs of sensors embedded in the bra to measure the thermal and moisture time series data (*BTMTSD*) of the breast surface and conduct time series classification (*TSC*) to diagnose breast cancer. Three main challenges are encountered when doing *BTMTSD* classification, (1) small supervised dataset, which is a common limitation of medical research, (2) noisy time series with unique noise patterns, and (3) complex interplay patterns across multiple time series dimensions. To mitigate these problems, we incorporate multiple data augmentation and transformation techniques with various deep learning *TSC* approaches and compare their performances for the *BTMTSD* classification task. Experimental results validate the effectiveness of our framework in providing reliable breast cancer early detection.

CCS CONCEPTS

• **Applied computing** → **Bioinformatics**; • **Computing methodologies** → *Artificial intelligence*.

KEYWORDS

Breast cancer early detection, time series classification, convolutional neural networks

ACM Reference Format:
Haoren Zhu, Pengfei Zhao, Yiu-Pong Chan, Hong Kang, and Dik Lun Lee. 2022. Breast Cancer Early Detection with Time Series Classification. In *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557107

*Corresponding author

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557107

1 INTRODUCTION

Breast cancer has grown into one of the most prevalent cancers worldwide. As disclosed by the World Health Organization (WHO),[1] 2.3 million women were diagnosed with breast cancer in 2020 with 685000 deaths globally, making it the world's most prevalent cancer (both male and female, surpassing lung cancer) [59].

With the ever-growing need for precaution and treatment, there has been a large amount of research in the past decades studying the pathology of breast cancer and developing effective treatment. Among these works, one important consensus is that early detection of breast cancer is most effective [8, 12, 66]. Breast cancer can be classified into Stage 0-IV [58] depending on the size of the tumor and whether or not it has invaded surrounding areas and lymph nodes, etc. If breast cancer is detected early and is in the localized stage, the 5-year relative survival rate is 99%.[2] Most current breast cancer detection methods utilize screening techniques to produce scanning images of the breast or invasive diagnosis methods. The specialist needs to analyze the results to identify if there exist any breast abnormalities. The examination equipment is mostly deployed in hospitals and charged at a high price [47]. For example, nowadays the most widely used tool for breast cancer diagnosis is mammography, which has significant limitations including radiation exposure, cost, patient discomfort, and more importantly, a high false positive rate [5]. Women in many settings face complex barriers to early detection, including social, economic, geographic, and other interrelated factors, which can limit their access to timely, affordable, and effective breast health care services [12] and make them reluctant to go to the hospital for regular checkup [22].

To overcome the limitations of the existing techniques, one desirable solution is to develop a low-cost, household breast cancer early detection solution that can facilitate women to monitor the health of their breasts at home. The recent advent of high-precision sensors and research on thermal properties of cancer cells open up a new promising solution for breast cancer early detection. Subsurface breast cancer lesions generate more heat and have increased blood supply when compared to healthy tissue, and this temperature rise is reflected in the skin surface temperature [5, 30]. The additional heat generates more sweat which raises the humidity of the breast skin surface. Moreover, during the early stage of breast

[1] https://www.who.int/news-room/fact-sheets/detail/breast-cancer.
[2] https://www.nationalbreastcancer.org/early-detection-of-breast-cancer/

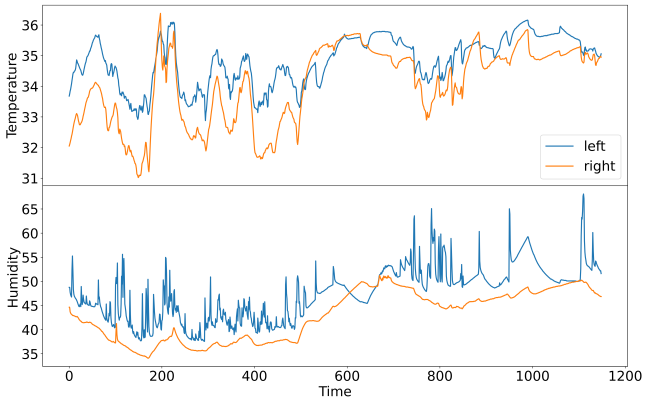

Figure 1: Example of breast thermal and moisture time series for a patient diagnosed with breast cancer.

(a) Device image. **(b) Device position.**

Figure 2: Illustration of the wearable sensor device.

cancer, the tumor is mostly found on only one side of the breast, and it may spread to another side in later stages. Thus, the specificity of breast cancer cells can be identified by comparing the breast surface temperature and humidity levels of the left and right breasts.

Figure 1 displays an example of the breast time series recorded from a patient confirmed with breast cancer. We can observe from the upper plot that the temperature of the left breast is significantly higher than the right one despite the turbulence, and the lower plot demonstrates distinct humidity patterns for the two sides. We develop a wearable device that integrates both thermal and moisture sensors, shown in Figure 2(a), to continuously collect the breast surface temperature and humidity data by embedding and positioning the devices in the patient's bra as shown in Figure 2(b). Most of the lymph nodes drain out of the breast in the upper outer quadrant hence this is the best position to detect temperature change in the breast. For convenience, we name the collected breast surface thermal and moisture time series data as *BTMTSD*. The recorded *BTMTSD* is uploaded to the server for automatic analysis, and a digital diagnostic report including the breast cancer level classification result will be returned to the user for self-examination through a mobile phone application. Manual classification of *BTMTSD* is costly due to the lack of specialists with professional backgrounds considering the tremendous potential user base. Thus, by combining with machine learning techniques, the automatic and accurate *BTMTSD* classification provides an easy-to-use, low-cost, non-invasive, and timely household solution for breast cancer early detection.

In recent years, a great number of works have been proposed to solve the problem of Time Series Classification (*TSC*) [48]. Even though many *TSC* approaches have been proven effective on various time series datasets, *BTMTSD* classification brings several unique challenges: (1) like most medical datasets, the small scale of *BTMTSD* hinders the mining of the specificity signals of breast cancer cells, (2) the raw *BTMTSD* has great noises and some have unique noise patterns due to device displacement, exceptional environment temperature, and humidity levels (see Section 3 for the detail) and the capability of identifying and filtering the noise directly influences the classification accuracy, (3) *BTMTSD* has multiple dimensions, how to identify the unique impact of each dimension and model their complicated interactions is challenging. To mitigate the first two problems, we propose a data augmentation framework based on the properties of *BTMTSD* and embed domain-specific knowledge in the data transformation process. We then apply various state-of-the-art deep learning time series classification (*DTSC*) approaches to model the complex patterns encoded in the multivariate *BTMTSD*. The contributions of this paper are multi-folds:

- We propose a new household solution for breast cancer early detection, design and manufacture a wearable device integrated with thermal and moisture sensors to collect *BTMTSD*, and apply deep learning algorithms to perform diagnosis automatically.
- Different from existing detection methods based on screening thermal images, we use the thermal and moisture time series data and model breast cancer detection as a time series classification problem. We design a *BTMTSD* specific data augmentation framework and use domain knowledge in data transformation to mitigate the insufficiency of labeled data so that the specificity features of breast cancer can be better revealed. According to the authors' best knowledge, this work is the first to apply *DTSC* on *BTMTSD* for breast cancer early detection.
- Based on the *BTMTSD* collected from 98 users with biopsy results, we conduct experiments to comprehensively evaluate the model performance. Experimental results show that the proposed framework can accurately detect breast cancer in its early stage.

2 RELATED WORK

2.1 Breast Cancer Early Detection

There exist multiple solutions for diagnosing breast cancer, including mammography, magnetic resonance imaging (MRI), ultrasonography, and positron emission tomography (PET). Most detection methods are done with screening techniques to produce scanning images, based on which the specialist makes a diagnosis for the patients. Various machine learning techniques are applied to image processing for the diagnosis of breast cancer [23, 49]. The above noninvasive modalities were then validated using biopsy as the gold standard. Although existing methods provide an effective solution for breast cancer detection, most equipment is installed in hospital and have limitations such as being expensive, time-consuming, and radiated, hindering women go to the hospital for regular checkup and detecting breast cancer in *early* stage.

In recent years, investigators have paid attention to the development of different biosensors to detect breast cancer. Thermal sensors have also been studied as the diagnostic tool for rapid and cost-effective early-stage breast cancer detection. Previous studies [28, 30] indicate that tumors generate more heat than healthy tissues, which can be identified by using thermal imaging. Recently researchers have built computational thermal models to relate the surface temperature distribution to tumor size and location for

breast cancer [5], where most models are built based on thermal images, like high-resolution infrared images [23, 40, 42, 57]. There are mainly two aspects distinguishing our work from existing image-based solutions. (1) We focus on thermal and moisture time series data and model breast cancer detection as time series classification problem, instead of thermal imaging processing and modeling. (2) We try to identify the specificity signals of breast cancer cells from the long-term and dynamic *BTMTSD*, instead of the short-time snapshot of breast thermal imaging.

2.2 Time Series Classification

There have been extensive studies for time series classification. Following the common taxonomy of previous reviews [9, 48], they can be broadly categorized as follows.

Distance-based methods quantify the similarity between time series using various distance measures [35], based on which the classifier can be built, for example Nearest Neighbor (*NN*) [64] and decision trees [7]. *1NN-DTW* [44, 54] is a typical distance-based classification method frequently used as a baseline for comparison, where the class label of a testing instance depends on the label of its nearest neighbor measured by dynamic time warping distance (*DTW*) distance. Despite that the distance-based methods have suboptimal performance compared to state-of-the-art deep learning techniques [48], their simple design provides good explainability and is suitable for small supervised datasets.

Feature-based (transformation based) methods are based on the representative features extracted by different transformations from the substructures of time series. Two typical approaches are *Shapelet* [16, 36] that maps time series to a distance vector to multiple representative subsequences and *Bag-of-Patterns* (BOP) [33, 34] that discretizes time series and computes the frequency of *BOP* words. Recent advancements include improving feature representations [3, 6, 51, 52, 63, 67], extending to multivariate *TSC* [2, 43, 53], and ensembles of multiple classifiers [1, 37, 38, 56, 63]. Feature-based methods can generate good visualization by relating the discovered features (e.g. shapes) to the classification results. However, many of them may suffer from low efficiency because the feature discovery process can be computationally expensive.

Deep learning methods mitigate the inefficiency of distance-based and feature-based methods due to the high dimensions and their difficulty in capturing multi-dimensional time series patterns. Deep learning methods can directly learn low-dimensional feature representations from raw time series and have achieved top performance in various time series datasets. Many works have been proposed to adapt different deep learning architectures to TSC, including Convolutional Neural Network (*CNN*) [6, 29, 31, 75], recurrent architecture [11, 20, 25], Residual Network (*ResNet*) [68], encoder [39, 55], and other advanced hybrid structure [10, 70, 72, 73].

3 PROBLEM DEFINITION

For ease of understanding, we first introduce the notations used in the framework. Given a user u, the wearable sensor devices record the u's breast surface temperature and humidity over a time period with interval ω. For example, if $\omega = 60000$ (ms) and the time period lasts for an hour, the output time series will have a length $T = 60$. Denote the multivariate time series of user u as X_u and its j-th dimension series as X_u^j. The *BTMTSD* of user u can be written as:

$$\begin{aligned} X_u &= \{X_u^1, X_u^2, ..., X_u^M\} \\ X_u^j &= \{x_{u,1}^j, x_{u,2}^j, ..., x_{u,T}^j\},\ j = 1, 2, ...M \end{aligned} \tag{1}$$

where M denotes the total number of dimensions and T denotes the length of the time series.

The raw dataset used in this work contains 4 dimensions, which are the temperature and humidity of the left and right breast. A dataset $D = \{(X_1, y_1), (X_2, y_2), ..., (X_u, y_u), ...)\}$ contains a collection of tuples (X_u, y_u) where X_u is the sensor output time series and y_u is the corresponding class label. Note that y_u is either equal to 0 (i.e. without cancer) or 1 (i.e. with cancer) and is obtained through biopsy, which is treated as the gold standard in medical society. Finally, the breast cancer *TSC* problem can be formulated as $\hat{y}_u = \mathcal{F}(X_u|\Theta_{\mathcal{F}})$, where $\Theta_{\mathcal{F}}$ denotes the model parameters of method $\mathcal{F}$ and $\hat{y}_u$ denotes the predicted result.

We further introduce the following characteristics of *BTMTSD* which distinguish it from other time series data.

- **Sensor noise**. The temperature time series contain unique noises due to displacement of the devices, high humidity, and internal measurement errors. While the latter two might have mild effects on the output, the former can result in significant deformation of temperature curves, often in the form of a "∧-shape".
- **Pairwise**. The four-variate time series can be divided into a pair of two-variate time series that measures the thermal and moisture environment of the left and right breast, respectively. For ease of reference, we denote them as *l-BTMTSD* and *r-BTMTSD*. To most breast cancer patients, only one side of the breast exist cancer tumors, leading to differences in *l-BTMTSD* and *r-BTMTSD*.
- **Abnormal pattern**. The *BTMTSD* of a user with a regular life schedule should display periodic patterns due to the biological rhythm. However, breast tumors can distort these patterns, manifesting as abnormal substructures in the time series.
- **Personalized pattern**. *BTMTSD* can have different shapes and patterns even for those without breast cancer because users may have different biological rhythms. The time of wearing the sensor device can also influence *BTMTSD*. For example, *BTMTSD* recorded during the sleeping time will be more stable compared to those recorded in the daytime due to physical activities.
- **Interplay of dimensions**. The specificity signal of breast cancer cells may come from the interaction of different time series dimensions. For example, the co-occurrence of high humidity and high temperature on one side of the breast may be considered an indicative feature of cancer tumors.

4 METHODOLOGY

The proposed framework for breast cancer *TSC* consists of three major components: data augmentation module, data transformation module, and *DTSC* module. Figure 3 illustrates the idea. Note that the augmentation module works only on the training dataset, while the transformation module works on both the training and testing dataset. Component details are introduced in the following.

4.1 Data Augmentation

To mitigate the problem of insufficient labeled data, we design a *BTMTSD* specific data augmentation module so that the breast

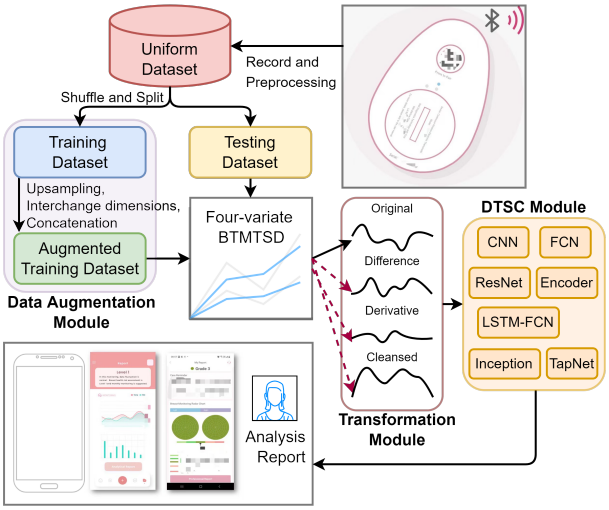

Figure 3: Overall framework for breast cancer TSC.

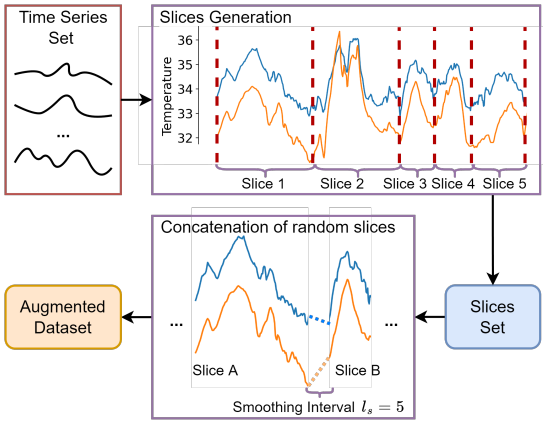

Figure 4: Data augmentation through concatenation of random slices.

cancer specificity features hidden in the original training dataset can be better revealed and preserved.

4.1.1 *Upsampling*

Firstly, we handle the imbalanced labeled data through upsampling. We observe that the uniform training dataset (see Section 5.1 for detailed construction process) has a medium degree of imbalance where the ratio of positive samples to negative samples is approximately $7:3$.[3] We balance the ratio to $1:1$ by duplicating randomly selected negative samples.

4.1.2 *Interchange of Dimensions*

We further augment the dataset through the interchange of dimensions in the following two ways. The first way interchanges the left and the right *BTMTSD* for each individual user. As introduced in Section 3, *BTMTSD* are pairwise, each of which can be divided into *l-BTMTSD* and *r-BTMTSD*. Denote user A's *l-BTMTSD* and *r-BTMTSD* as $X_A = (A_l, A_r)$. Through exchanging their positions we can obtain $X'_A = (A_r, A_l)$, by which the number of data can be doubled without affecting the actual class label.

The second way augments *negative* (without breast cancer) samples by randomly mixing two negative users' *BTMTSD*. For user A and B's *BTMTSD* X_A and X_B, four additional samples (A_l,B_l), (A_l,B_r), (A_r,B_l), (A_r,B_r) can be constructed, each of which is of negative label. However, due to the personalized property of *BTMTSD* introduced in Section 3, two negative samples can have distinctive shapes, and consequently, their hybridization may cover the original samples' left-right *BTMTSD* difference feature for classifying positive samples. Therefore, we employ a distance threshold θ which is obtained by calculating the average L2 distance of all negative users' left-right *BTMTSD*, to filter out the misleading augmented samples whose left-right *BTMTSD* distance is larger than θ. In summary, we repeatedly select two random negative samples, perform hybridization, and add the derived sample to the augmented dataset if the pairwise distance is smaller than θ. Note that this technique is not suitable for positive samples because their *l-BTMTSD* and *r-BTMTSD* can have different class labels.

4.1.3 *Concatenation of Random Slices*

We produce more samples by concatenating random *BTMTSD* slices of the same class label, which are generated based on the unique noise pattern of *BTMTSD* due to device displacement.

The normal breast surface temperature should be at least $35°C$. Under the ideal case where the user wears the device correctly throughout the time, the recorded temperature value should display a surge from the room temperature (i.e. $25°C$) to the actual surface temperature at the beginning, then undergo a slow change in most time following the biological rhythm, and ends with a drop when the user takes off the device. However, in reality, *BTMTSD* is always involved with a great number of noises due to device displacement. The upper right rectangle of Figure 4 displays a noisy example. Transitory displacement of sensor devices (e.g. caused by turning over during sleep) commonly results in a quick drop of temperature values and then a soon recovery, leading to multiple connected "$\wedge$-shape" sub-series where the upper point of "$\wedge$" are more close to the actual temperature while the two bottom points of "$\wedge$" can be regarded as inaccurate measurement. These bottom points generally have temperature satisfying the following conditions: (1) below $35°C$, (2) local minima, and (3) minimum value within a large nearby area. We cut the original *BTMTSD* at these bottom points of the "$\wedge$-shape" to produce multiple time series slices (e.g. 5 slices are generated from the example series in Figure 4).

The proposed slicing method brings two advantages. Firstly, since the endpoints of time series slices are most probably inaccurate measures, concatenation of two slices will not yield significant effects on the original nature of data. On the other hand, it mitigates the discontinuity issue of concatenation. For example, if fixed-length windowing is used instead to extract time series slices, a large gap may exist between the endpoints of two random slices. Concatenation will result in discontinuous time series which may mislead the classification model. In comparison, the end points of the slices extracted using the proposed method mostly range from $31°C$ to $33°C$. By concatenating multiple random slices and interpolating a short smoothing interval between each pair (length

[3]The data was collected through a clinical trial, thus a large portion of positive samples are obtained.

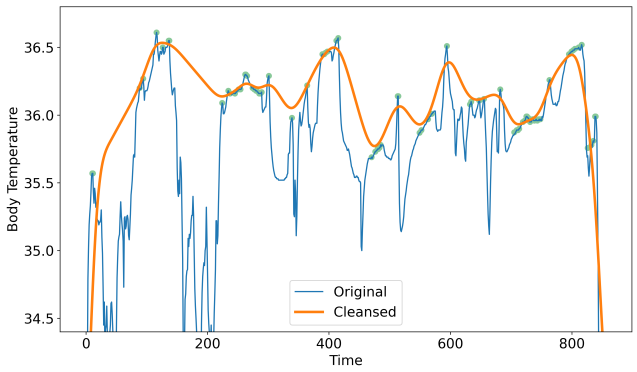

Figure 5: Generate cleansed time series from the original temperature time series.

5 interval is used in Figure 4), we can generate continuous data samples without changing the data properties.

Figure 4 summarizes the overall process. Firstly, time series slices are extracted from the training dataset and assigned the same class label as the original *BTMTSD*. We then concatenate multiple random slices of the same class label to produce new data samples and add them to the augmented dataset.

4.2 Transformation

To better reveal the patterns encoded in *BTMTSD*, in addition to the four-variate *BTMTSD* mentioned in Section 3, we add **five** extra dimensions based on domain-specific knowledge.

- **Difference** dimension is generated by computing the pointwise distance between the left and right temperature time series. As introduced in Section 3, breast cancer tumors mostly exist on only one side of the breast and the breast with cancer tumors should have higher temperature values than the other.
- **Derivative** dimensions contain two new time series which are the derivatives of the left and right temperature time series, respectively. Calculating derivatives is a common practice adopted in *TSC* [13, 53] to better exploit the latent features.
- **Cleansed** dimensions are constructed from the smoothing of the left and right temperature time series. As introduced in Section 4.1.3, device displacement incurs the unique "∧-shape" in temperature where the upper point of the "∧-shape" is more accurate. Based on this observation, we first extract the local maximal points from the original time series. We then produce cleansed temperature time series by performing Gaussian smoothing [61] based on these points. Figure 5 illustrates the idea. Note that we do not remove the original temperature time series because their noisy interval may still contain useful information.

The transformation module output time series contains 9 dimensions, which are the left and right temperature, the left and right humidity, the left and right temperature derivative, the left and right cleansed temperature, and the temperature difference, respectively. Finally, the output time series is fed to the *DTSC* module.

4.3 Deep Learning Time Series Classification

We introduce the deep learning based time series classification algorithms used in *DTSC* module below.

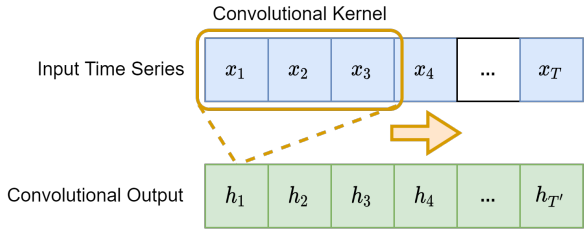

Figure 6: Example of convolutional operation on time series. The input time series $\{x_1, x_2, ..., x_T\}$ contains T timestamps and each x_t is a vector of size M (i.e. number of dimensions). Here convolutional kernel has filter size 3 and it can be regarded as a moving window over the input time series to extract hidden feature h_t from x_t, x_{t+1}, and x_{t+2}. Subject to the output channel c_{out}, h_t can be either a vector ($c_{out} > 1$) or a single value ($c_{out} = 1$).

4.3.1 *Basic Convolutional Neural Network (CNN)*

Convolutional Neural Networks (*CNN*) are first introduced to tackle the image modeling problem [45] and have been extensively applied in different domains [32]. The success of *CNN* lies in that it significantly reduces the number of model parameters to be learned by restricting the connections between neurons within a local area (i.e. convolutional kernel), thus effectively mitigating the over-fitting problem and accelerating the model convergence.

In *TSC*, the convolutional operation can be regarded as sliding a 1D (time) kernel over time series, compared to sliding a 2D (height and weight) kernel over images. The number of time series dimensions (9 in our setting) is similar to the channel for images. Figure 6 illustrates the idea. A general form of performing the convolutional operation at time t can be formulated as:

$$C(t) = f\Big(\sum_{i=1}^{l}\sum_{j=1}^{M} x_{i+s(t-1),j}\omega_{i,j} + b\Big) \tag{2}$$

where $x \in \mathbb{R}^{T\times M}$ denotes the input time series, M denotes the number of input channels, l denotes the filter size, s denotes the stride, $\omega \in \mathbb{R}^{l\times M}$ and b are learnable parameters, respectively, and f refers to the activation function. Commonly, *Rectified Linear Unit* (*ReLU*) is selected as the activation function. The output of the convolutional operation is sometimes referred to as *feature map*.

The pooling layer is another important component in *CNN* which can further compress the convolutional output. The pooling operation at time t can be formulated as:

$$P(t) = g(C((t-1)l+1), C((t-1)l+2), ..., C(tl)) \tag{3}$$

where $C(*)$ is the convolutional output, l denotes the kernel size, and g refers to the pooling strategy. For example, if $l = 2$, pooling layer output will have a size half of the convolutional output. The convolutional layer and pooling layer can be stacked multiple times to extract the latent representations from original time series data.

A typical *CNN* framework [74] stacks two convolutional layers and two pooling layers to obtain the feature map, after which a fully connected layer is used to generate the classification.

4.3.2 Fully Convolutional Neural Networks (FCN)

Another approach proposes *Fully Convolutional Neural Networks* (*FCN*) [68] which removes the pooling layer between convolutional layers to prevent over-fitting. The *FCN* architecture contains three consecutive convolutional blocks, each of which is composed of a convolutional layer, a batch normalization (*BN*) layer [21], and a *ReLU* activation function. The network ends with a global average pooling (*GAP*) layer, followed by a softmax classifier.

4.3.3 Residual Network (ResNet)

Viewing the great success of residual network in solving the image recognition task [15], *Residual Network* (*ResNet*) [68] proposes to combine deep *CNN* with residual structure. *ResNet* contains three residual blocks and ends with a *GAP* layer followed by a softmax classifier. Each residual block is composed of three convolutional blocks with the same design as those used in *FCN* [68]. Specifically, *ResNet* adds shortcut connections between consecutive residual blocks, which enables the gradient to bypass the bottom layers and effectively alleviate the gradient descending problem.

4.3.4 Encoder

Encoder [55] first introduces the attention mechanism to deep *CNN* for *TSC*. The attention mechanism computes a weight tensor (i.e. attention map) in the same shape as the input time series, which enables the model to perceive which part of the input is more important in generating the classification results. Similar to the architecture of *FCN* [68], the first three layers of the encoder are convolutional blocks with minor adaptions. Each convolutional block is composed of a convolution layer, an instance normalization (*IN*) layer [65], and a parametric rectified linear unit (*PReLU*) [14] activation function. Particularly, the third convolutional block is followed by an attention layer instead of a *GAP* layer. Finally, the attention layer output is fully connected to a softmax classifier.

4.3.5 LSTM-FCN

LSTM-FCN [24, 25] first combines *FCN* with *Long Short Term Recurrent Neural Network* (*LSTM*). The main architecture of *LSTM-FCN* composes of two paralleled modules: (1) *Fully convolutional* (*FC*) module extracts features through three convolutional blocks and utilizes the *squeeze-and-excite* [19] block to capture the interdependencies between dimensions. (2) *LSTM* module applies dimension shuffle to the input and then uses an (attention) LSTM block to learn temporal dependencies. Finally, the outputs of two modules are concatenated and fully connected to a softmax classifier.

4.3.6 InceptionTime

InceptionTime [10] further boosts the performance by incorporating *Inception* module [60] with *ResNet*. The outer structure of *InceptionTime* is similar to *ResNet* except that it replaces the residual block with *Inception* block and reduces the number of blocks to *two*.

Inside the *Inception* block, the input multivariate time series of the shape $T \times M$ is first compressed by a bottleneck layer which applies a convolutional operation with kernel size and stride 1, resulting in an output of the shape $T \times M'$ where $M' < M$. The bottleneck layer reduces the number of parameters to be learned and mitigates the over-fitting problem. Then convolutions of different sizes (e.g. 10, 20, and 40 are used in the original paper) are applied to capture time series patterns lasting for different periods. In addition, the input time series is also fed to a paralleled max pooling layer followed by another bottleneck layer, which enables the model to be invariant to small perturbations. Finally, it concatenates the results of multiple convolutions by the channel axis.

4.3.7 Time Series Attentional Prototype Network (TapNet)

Time Series Attentional Prototype Network (TapNet) [73] combines the strength of traditional methods by embedding domain-specific knowledge and the advantage of deep learning methods in capturing low-dimension features. The *TapNet* architecture can be divided into three modules. Firstly, *Random Dimension Permutation* module is applied to the input multivariate time series to generate multiple groups comprising different combinations of dimensions. This design aims at modeling how different combinations of dimensions may affect the prediction. Then, a low-dimension representation is extracted using the *Multivariate Time Series Encoding* module, which contains two paralleled components: (1) an *LSTM* component takes the original time series as input, and (2) a complex convolution component accepting the groups generated by the previous module. The outputs of these components are concatenated and fed to a fully-connected layer to compute an embedding for the input time series. Finally, the *Attentional Prototype Learning* module generates a prototype candidate for each class based on the embedding. In training, *TapNet* aims at minimizing the intra-class distance (i.e. distance between class members and the class prototype) while maximizing the inter-class distance (i.e. distance between different class prototypes). In testing, the model classifies an instance based on its similarity to each class prototype.

5 EXPERIMENT

To validate the effectiveness of the proposed framework, we compare multiple *TSC* approaches' performance on *BTMTSD* with standard classification evaluation metrics.

5.1 Dataset

To collect the data, we invited 98 participants (aged 23 - 75, women) to join the experiment. The participants are newly diagnosed with unilateral BI-RADS 1, 2, or 3 breast lesions in B ultrasound and molybdenum target inspections. They are required to wear the sensor device for at least 8 hours from 20:00 to 6:00. Specifically, the sensor device is fixed on the breast surface via stickers as shown in Figure 2(b) and the wearing process is monitored by nurses to ensure the data is collected objectively. We then perform an anonymous labeling process on the collected raw data and finally obtain 98 time series with different lengths. We generate the uniform training and testing dataset as follows.

Firstly, the time series segments with temperature values below $27°C$ for more than 30 timestamps (i.e. half an hour) are removed. The measured temperatures are much lower than regular body temperatures, which means the participants did not wear the devices. The minimum duration is chosen because sometimes the displacement of devices may result in a quick drop in temperature, but generally the state would not last for more than half an hour. After this step, we obtain 105 valid continuous time series (some users' raw time series are broken into two parts, see step 1 in Figure 7), among which the shortest one has length 607 and the longest one has length 2367, and the average length is approximately 1200.

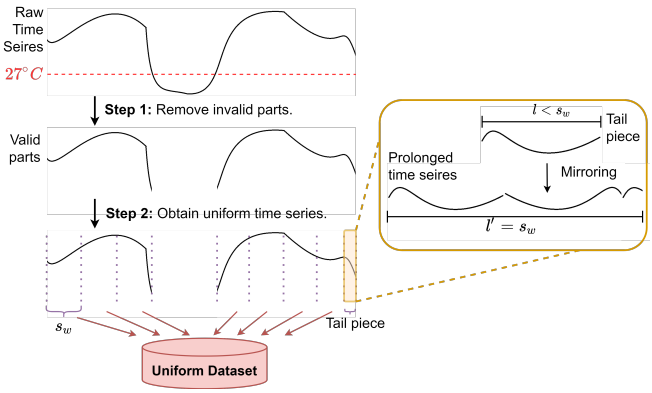

Figure 7: Overall flow of dataset construction.

Secondly, we obtain the uniform time series data using windowing. We define the window size s_w to 240 (i.e. four hours) and uniformly cut the valid time series obtained in the previous step into multiple non-overlapping pieces. Since most time series' lengths are not divisible by s_w, the end part of these time series may have lengths $l < s_w$. We name these parts as *tail pieces* and process them in the following manner: (1) if the length $l < 30$, we discard it directly; (2) otherwise, we prolong the tail piece through mirroring, as illustrated in the orange box of Figure 7. The reason for using mirroring instead of stretching is that the stretching operation may interfere with the features contained in the derivatives of *BTMTSD* (e.g. sensitivity of the temperature sensor).

Finally, the derived uniform time series are assigned the same class label as in the original time series and added to the uniform dataset. Figure 7 displays the overall flow of dataset construction. We shuffle the uniform time series data and split them into training and testing sets by the ratio 0.7, which results in 375 training instances and 161 testing instances. We further expand the training set to the scale of 10k through the data augmentation techniques introduced in Section 4.1. After the data augmentation module, the positive/negative ratio of the training set decreases from 7:3 to 1:1. The testing set is not augmented, thus the positive/negative ratio remains 7:3. All deep learning frameworks are tuned based on the augmented training dataset and tested on the testing dataset.

5.2 Baselines

Apart from the deep learning approaches introduced in Section 4.3, we also compare our framework to the following baseline models.

- **Ridge-DTW**. We build a ridge classifier [17] based on the dynamic time warping distance between the left and right temperature curves. Since subsurface breast cancer lesions generate more heat and during the early stage of breast cancer the tumor is mostly found on only one side of the breast, it's a natural idea to utilize left-right breast temperature difference for classification.
- **WEASEL+MUSE** [53]. This is a state-of-the-art bag-of-pattern (*BOP*) approach that is frequently compared to deep learning methods. It first applies Symbolic Fourier Approximation (*SFA*) [50] to discretize the *BTMTSD* and utilizes sliding windows to extract both unigram and bigram words. It then filters out less important features using the *Chi-Squared* test and constructs the classifier based on the selected discriminative features.
- **gRSF** [26]. This is a recent benchmark shapelet-based model for *TSC*. A *shapelet* is defined as a subsequence of time series that is "maximally representative" of a class [16, 36, 71]. *gRSF* produce classification through an ensemble of multiple random shapelet trees (*RST*s), each of which is constructed on a subset randomly sampled from the whole training dataset.

5.3 Experiment Result

In this section, we compare and analyze the results of different *TSC* approaches. To evaluate the model performance, we adopt the following commonly used classification metrics: precision (PR), recall (RE), F1-score (F1), specificity (SP), and accuracy (ACC). $PR = \frac{tp}{tp+fp}$ measures for all the testing instances that are predicted as positive (have breast cancer), how many are really positive, where tp and fp denote true positive and false positive, respectively. $RE = \frac{tp}{tp+fn}$ measures that out of all the positive samples in the testing set, how many are predicted as positive, where fn denotes false negative. $F1$ combines the precision and recall into a single metric by taking their harmonic mean $\frac{2*PR*RE}{PR+RE}$. $SP = \frac{tn}{tn+fp}$ measures out of all women that do not have breast cancer, how many are predicted as negative, where tn denotes true negative. Finally, $ACC = \frac{tn+tp}{tp+fp+tn+fn}$ measures the overall ratio of correct classification. Note that *ACC* is less frequently used in the medical field and we consider $F1$ the most important metric in tuning the models.

All the methods are trained based on the augmented training dataset. For model tuning, we utilize the tool *WandB* [4] to find out the optimal combination of hyperparameters over a predefined parameter pool. Specifically for the deep learning methods, we train the model on one GTX-2080Ti graphical card. We set the initial learning rate to 1e-3 and use a dynamic learning rate scheduler which reduces the learning rate by half if the performance decays for 4 epochs. It takes less than an hour to train a model once for most methods. Finally, we apply the well-trained models to the testing dataset to compare their performance.

5.3.1 Overall Comparison

The overall comparison of different *TSC* approaches is displayed in Table 1. We can see all the deep learning approaches and the other two baselines have significantly outperformed the simple *Ridge-DTW* in terms of all the metrics. It indicates that dimensions other than temperature (e.g. humidity) and interplay of dimensions are important to the classification task and their ability in capturing the patterns across multiple dimensions grants them a performance advantage. Most deep learning approaches except *ResNet* beat the three non-deep baselines. The poor performance of *ResNet* in all three metrics may be due to the overfitting problem. Even though we have augmented the dataset by different means, the size of labeled data may still be too limited to train such as a deep model.

Comparing deep learning approaches, *Encoder* has achieved the best performance regarding all the three metrics and *TapNet* has achieved the second best performance in precision and F1-score. We find that more complicated architecture does not necessarily yield a better result. For example, despite the advanced design of *InceptionTime* compared to *Encoder*, it has inferior performance with low precision and F1-score close to *WEASEL+MUSE*. In contrast, a simple model *FCN* can have comparable performance to

Table 1: Overall comparison. Bold number indicates the optimal performance while * indicates the second best performance.

Metrics	Deep Learning Approaches							Baselines		
	CNN	FCN	ResNet	Encoder	LSTM-FCN	InceptionTime	TapNet	Ridge-DTW	WEASEL+MUSE	gRSF
PR	0.7946	0.8214	0.6786	**0.8571**	0.8393*	0.7589	0.8214	0.7143	0.8036	0.7232
RE	0.8448	0.8624	0.7549	**0.8974**	0.8447	0.8174	0.8692*	0.6630	0.7944	0.7434
F1	0.8189	0.8414	0.7147	**0.8768**	0.8420	0.7871	0.8446*	0.6877	0.7990	0.7332
SP	0.6735	0.6939	0.4898	**0.7755**	0.6531	0.6122	0.7143*	0.1633	0.5306	0.4286
ACC	0.7578	0.7826	0.6211	**0.8323**	0.7826	0.7143	0.7888*	0.5466	0.7205	0.6335

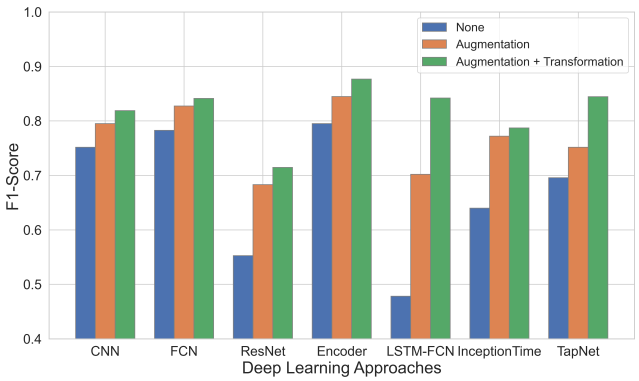

Figure 8: Performance comparison with and without the data augmentation and preprocessing modules. "None" refers to the original model without the two modules.

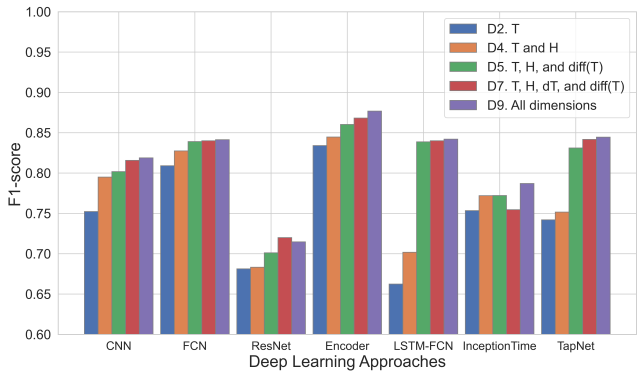

Figure 9: Performance comparison of *BTMTSD* with different combinations of dimensions.

more recent hybrid models such as *LSTM-FCN* and *TapNet*. After investigating the training process of these models, we discover that the more complicated ones generally have lower final training loss and take a longer time to converge. It indicates that these models may still have different degrees of overfitting problems.

In addition, we observe that most approaches have a higher recall compared to the precision and have low specificity, which means that they make more mistakes in classifying negative samples as positive samples. This may be caused by the imbalance of *BTMTSD*. As mentioned in Section 4.1.1, the ratio of positive samples to negative samples for uniform *BTMTSD* dataset is about 7 : 3. Although we have alleviated this problem by upsampling the negative samples, the trained models are prone to classify a testing instance as a positive sample if it falls near the decision boundary.

5.3.2 *Impact of Data Augmentation and Transformation*

To validate the effectiveness of the data augmentation and transformation techniques, we compare the model performance with and without these two modules for the deep learning approaches. Figure 8 displays the comparison results. F1-score is selected as the performance metric because it provides a balanced measure between precision and recall.

In general, both the data augmentation module and the transformation module significantly improve the classification F1-score. However, they can have various performance gains for different approaches. For lightweight architectures which contain fewer parameters to be learned (e.g. *CNN*, *FCN*, and *Encoder*), the "None" version (i.e. original model without the two modules) can achieve a relatively high F1-score and the addition of data augmentation and transformation only boosts the performance by a small fraction. On the contrary, the performance of some complicated architectures is not good if there is not sufficient labeled data. As shown in Figure 8, the "None" version of *ResNet* and *LSTM-FCN* have F1-score below 0.6, which is even worse than the bottom-line method *Ridge-DTW*. It indicates small supervised dataset will constrain the capacity of complex deep learning models and data augmentation techniques can effectively alleviate the insufficient labeled data problem.

5.3.3 *Impact of BTMTSD Dimensions*

We investigate the impact of different *BTMTSD* dimensions in producing the classification results. As introduced in Section 4.2, the transformation module converts the original four-variate time series to nine-variate time series through three transformation functions. Therefore, we compare the performance of the following dimension combinations: (D2) the left and right temperature, (D4) the left and right temperature and humidity, (D5) the left and right temperature and humidity + temperature difference, (D7) the left and right temperature, temperature derivative, and humidity + temperature difference, (D9) all dimensions. Note that the number of dimensions increases from 2 to 9 in these combinations and the gap between two consecutive combinations is often 2 because most dimensions are pairwise (i.e. relative to the left and right breast).

Figure 9 displays the comparison results. All the methods generally perform better as the number of dimensions increases. The left and right temperature (D2) alone can achieve an F1-score of around 0.75 for most methods except *ResNet* and *LSTM-FCN*, which indicates that the thermal pattern recorded by the temperature sensor is indeed important for detecting breast cancer. Integration of humidity further improves the performance, especially for *CNN* and *LSTM-FCN*. These observations are consistent with the theoretical expectations. As for the dimensions related to the transformation

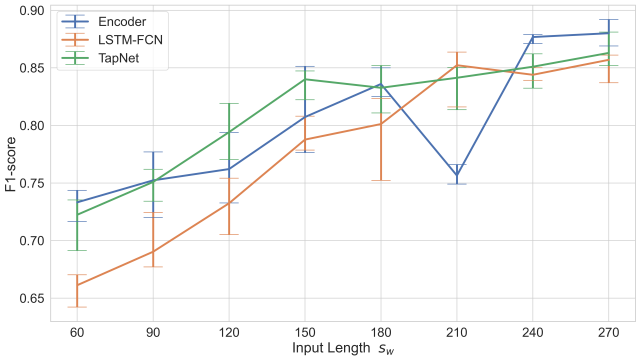

Figure 10: Performance comparison of *BTMTSD* with different lengths. The errorbar indicates the minimum and maximum F1-score of the 5 random train-test splits for each s_w.

module, the addition of temperature difference (D5) significantly improves the performance of all models. Especially, it increases the F1-score by 19.5% for *LSTM-FCN* and 10.6% for *TapNet*. In comparison, the addition of temperature derivative (D7) and cleansed temperature (D9) have relatively marginal impacts. Note that some models have deteriorated performance under some dimension combinations (e.g. D9-*ResNet*). To conclude, by embedding domain knowledge into the transformation module, the derived dimensions are helpful in generating better results for *DTSC* models.

5.3.4 *Impact of BTMTSD Length*

In studying *BTMTSD*, one natural question is that *how long is the time series sufficient to reveal breast cancer?* To answer the question, we investigate how different lengths of input time series (denoted as s_w) may affect the model performance. Since s_w participates in the dataset construction process, different values of s_w result in different sizes of the testing set. Smaller s_w will generate a larger uniform dataset of short-term time series while larger s_w may preserve the long-term patterns at the cost of a smaller dataset. Thus, the question becomes, *the short-term but larger dataset, the long-term but smaller dataset, which one is better for DTSC?*

To provide a fair comparison, we first generate the corresponding uniform dataset for each s_w and produce 5 random train-test splits for each uniform dataset. We then train and test the model on each split independently and compute the average F1-score to evaluate the model performance regarding s_w. We experiment with the top three models regarding F1-score: *Encoder*, *TapNet*, and *LSTM-FCN*. The comparison results of different input lengths s_w are displayed in Figure 10. We can observe large performance gains in all models when s_w increases from 60 (i.e. an hour) to 150. Further growth of s_w leads to different behaviors for the three models. *LSTM-FCN* demonstrates a steady increasing trend in F1-score and reaches its second best performance at $s_w = 210$. In comparison, performance of *Encoder* suddenly drops at $s_w = 210$ and bounces back afterwards, while *TapNet* seems to be less affected by larger s_w. *Encoder* has outperformed the other two models when $s_w \geq 240$. Note that the performance gains for all models are minimal when $s_w \geq 240$, which indicates that four-hour *BTMTSD* might be sufficient to reveal breast cancer for these deep learning models. From the experiment results, we can see long-term but smaller dataset outperforms the small-term but larger dataset, and this shows the importance of modeling the long-term and dynamic patterns of *BTMTSD* in order to extract the specificity of breast cancer cells' thermal behavior.

6 DISCUSSION

In this paper, we provide a low-cost, household, self-exam solution by transforming breast cancer early detection to *BTMTSD* classification problem based on the theoretical research about cancer cells' thermal properties. We develop a breast cancer *DTSC* framework which combines *BTMTSD* specific data augmentation and transformation techniques with the state-of-the-art deep learning models. Real clinical data is collected and used in the experiment, and the results validate the effectiveness of our framework.

We have also manufactured the sensor devices, and the deployment of the proposed algorithm into a mobile application is ongoing. Although it cannot achieve the same high accuracy as the state-of-the-art clinical detection methods (e.g. biopsy), the low cost and convenience of wearable sensor devices make it a suitable household solution for breast cancer early detection, which reduces most constraints and concerns faced by patients in using mainstream image-based detection methods. In addition, the data augmentation and transformation techniques proposed in this paper may also be effective in other medical time series datasets containing similar noises or displaying pairwise properties.

Admittedly, the experiment presented in this paper has limitations. Firstly, some models' performance is still constrained by the limited availability of labeled data due to the costly and complicated data acquisition process, where we have spent great efforts in contacting the patients and the hospital, conducting clinical trials, etc. On the other hand, the sensor device is currently fixed with stickers to guarantee correct positioning, while a more convenient solution is to attach the sensor device to underwear which is less restrictive for the users. The new way may incur more noise and bring more challenges to the problem.

Finally, we identify the following future directions for this new research problem. (1) Enrich the dataset by collecting more *BTMTSD* from patients and adding common factors such as percentage of body fat, age, and nationality that can influence breast cancer patterns. (2) Design a new *Transfer Learning* (*TL*) framework to mitigate the insufficiency of data in a new application scenario [62, 69, 76] and for developing a model with good transferability and pretrained with homogeneous time series data (e.g. body temperature series recorded by sports bracelets). The pretrained model is able to understand the local substructures of time series and is further fine-tuned using *BTMTSD*. (3) Embed *Explainable Artificial Intelligence* (*XAI*) [18, 27, 41, 46] into the model. In research of a mature solution for breast cancer early detection, the ability to identify contributing factors and produce a diagnostic report with good visualization that is convincing and understandable to the public is also important.

ACKNOWLEDGEMENTS

Research reported in this paper was supported by the Hong Kong Bio-rhythm R&D Company Limited, under RDC grant 2021074-0. We appreciate the anonymous reviewers for their helpful comments on the manuscript.

REFERENCES

[1] Anthony Bagnall, Jason Lines, Jon Hills, and Aaron Bostrom. 2015. Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering* 27, 9 (2015), 2522–2535.

[2] Mustafa Gokce Baydogan and George Runger. 2015. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery* 29, 2 (2015), 400–422.

[3] Mustafa Gokce Baydogan and George Runger. 2016. Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery* 30, 2 (2016), 476–509.

[4] Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/ Software available from wandb.com.

[5] Arjun Chanmugam, Rajeev Hatwar, and Cila Herman. 2012. Thermal analysis of cancerous breast model. In *ASME International Mechanical Engineering Congress and Exposition*, Vol. 45189. American Society of Mechanical Engineers, 135–143.

[6] Angus Dempster, François Petitjean, and Geoffrey I Webb. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* 34, 5 (2020), 1454–1495.

[7] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. 2013. A time series forest for classification and feature extraction. *Information Sciences* 239 (2013), 142–153.

[8] Ruth Etzioni, Nicole Urban, Scott Ramsey, Martin McIntosh, Stephen Schwartz, Brian Reid, Jerald Radich, Garnet Anderson, and Leland Hartwell. 2003. The case for early detection. *Nature reviews cancer* 3, 4 (2003), 243–252.

[9] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* 33, 4 (July 2019), 917–963. https://doi.org/10.1007/s10618-019-00619-1 arXiv: 1809.04356.

[10] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. InceptionTime: Finding AlexNet for Time Series Classification. *Data Mining and Knowledge Discovery* 34, 6 (Nov. 2020), 1936–1962. https://doi.org/10.1007/s10618-020-00710-y arXiv: 1909.04939.

[11] Claudio Gallicchio and Alessio Micheli. 2017. Deep echo state network (deepesn): A brief survey. *arXiv preprint arXiv:1712.04323* (2017).

[12] Ophira Ginsburg, Cheng-Har Yip, Ari Brooks, Anna Cabanes, Maira Caleffi, Jorge Antonio Dunstan Yataco, Bishal Gyawali, Valerie McCormack, Myrna McLaughlin de Anderson, Ravi Mehrotra, et al. 2020. Breast cancer early detection: A phased approach to implementation. *Cancer* 126 (2020), 2379–2393.

[13] Tomasz Górecki and Maciej Łuczak. 2015. Multivariate time series classification with parametric derivative dynamic time warping. *Expert Systems with Applications* 42, 5 (2015), 2305–2312.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. 2014. Classification of time series by shapelet transformation. *Data mining and knowledge discovery* 28, 4 (2014), 851–881.

[17] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: applications to nonorthogonal problems. *Technometrics* 12, 1 (1970), 69–82.

[18] Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. 2021. Explainable Multivariate Time Series Classification: A Deep Neural Network Which Learns to Attend to Important Variables As Well As Time Intervals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, New York, NY, USA, 607–615. https://doi.org/10.1145/3437963.3441815

[19] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[20] Michael Hüsken and Peter Stagge. 2003. Recurrent neural networks for time series classification. *Neurocomputing* 50 (2003), 223–235.

[21] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

[22] Rakibul M Islam, Baki Billah, Md Nassif Hossain, and John Oldroyd. 2017. Barriers to cervical cancer and breast cancer screening uptake in low-income and middle-income countries: a systematic review. *Asian Pacific journal of cancer prevention: APJCP* 18, 7 (2017), 1751.

[23] Seyed Hamed Jafari, Zahra Saadatpour, Arash Salmaninejad, Fatemeh Momeni, Mojgan Mokhtari, Javid Sadri Nahand, Majid Rahmati, Hamed Mirzaei, and Mojtaba Kianmehr. 2018. Breast cancer diagnosis: Imaging techniques and biochemical markers. *Journal of Cellular Physiology* 233, 7 (2018), 5200–5213.

[24] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 2017. LSTM fully convolutional networks for time series classification. *IEEE access* 6 (2017), 1662–1669.

[25] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. 2019. Multivariate LSTM-FCNs for time series classification. *Neural Networks* 116 (2019), 237–245.

[26] Isak Karlsson, Panagiotis Papapetrou, and Henrik Boström. 2016. Generalized random shapelet forests. *Data mining and knowledge discovery* 30, 5 (2016), 1053–1085.

[27] Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. 2018. Explainable time series tweaking via irreversible and reversible temporal transformations. *arXiv:1809.05183 [cs, stat]* (Sept. 2018). http://arxiv.org/abs/1809.05183 arXiv: 1809.05183.

[28] D. Kennedy, Tanya Lee, and Dugald Seely. 2009. A Comparative Review of Thermography as a Breast Cancer Screening Technique. *Integrative cancer therapies* 8 (04 2009), 9–16. https://doi.org/10.1177/1534735408326171

[29] Bee Hock David Koh, Chin Leng Peter Lim, Hasnae Rahimi, Wai Lok Woo, and Bin Gao. 2021. Deep temporal convolution network for time series classification. *Sensors* 21, 2 (2021), 603.

[30] Ray Lawson. 1956. Implications of surface temperatures in the diagnosis of breast cancer. *Canadian Medical Association Journal* 75, 4 (1956), 309.

[31] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[33] Jessica Lin, Rohan Khade, and Yuan Li. 2012. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems* 39, 2 (2012), 287–315.

[34] Jessica Lin and Yuan Li. 2009. Finding structural similarity in time series data using bag-of-patterns representation. In *International conference on scientific and statistical database management*. Springer, 461–477.

[35] Jason Lines and Anthony Bagnall. 2015. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery* 29, 3 (May 2015), 565–592. https://doi.org/10.1007/s10618-014-0361-2

[36] Jason Lines, Luke M Davis, Jon Hills, and Anthony Bagnall. 2012. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 289–297.

[37] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2016. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1041–1046.

[38] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data* 12, 5 (2018).

[39] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. 2021. Gated Transformer Networks for Multivariate Time Series Classification. *arXiv preprint arXiv:2103.14438* (2021).

[40] Adolfo Lozano, Jody C. Hayes, Lindsay M. Compton, Jamasp Azarnoosh, and Fatemeh Hassanipour. [n.d.]. Determining the thermal characteristics of breast cancer based on high-resolution infrared imaging, 3D breast scans, and magnetic resonance imaging. *Scientific Reports* 10, 1 ([n. d.]). https://par.nsf.gov/biblio/10189740

[41] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[42] Aigerim Mashekova, Yong Zhao, Eddie Y.K. Ng, Vasilios Zarikas, Sai Cheong Fok, and Olzhas Mukhmetov. 2022. Early detection of the breast cancer using infrared technology – A comprehensive review. *Thermal Science and Engineering Progress* 27 (2022), 101142.

[43] Matthew Middlehurst, James Large, and Anthony Bagnall. 2020. The canonical interval forest (CIF) classifier for time series classification. In *2020 IEEE international conference on big data (big data)*. IEEE, 188–195.

[44] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.

[45] Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015).

[46] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. 2021. Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey. *arXiv:2104.00950 [cs]* (April 2021). http://arxiv.org/abs/2104.00950 arXiv: 2104.00950.

[47] Anne F Rositch, Karla Unger-Saldaña, Rebecca J DeBoer, Anne Ng'ang'a, and Bryan J Weiner. 2020. The role of dissemination and implementation science in global breast cancer control programs: frameworks, methods, and examples. *Cancer* 126 (2020), 2394–2404.

[48] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. 2021. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 35, 2 (March 2021), 401–449. https://doi.org/10.1007/s10618-020-00727-3

[49] Farahnaz Sadoughi, Zahra Kazemy, Farahnaz Hamedan, Leila Owji, Meysam Rahmanikatigari, and Tahere Talebi Azadboni. 2018. Artificial intelligence methods

for the diagnosis of breast cancer by image processing: a review. *Breast Cancer: Targets and Therapy* 10 (2018), 219.

[50] Patrick Schäfer and Mikael Högqvist. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th international conference on extending database technology*. 516–527.

[51] Patrick Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery* 29, 6 (Nov. 2015), 1505–1530. https://doi.org/10.1007/s10618-014-0377-7

[52] Patrick Schäfer and Ulf Leser. 2017. Fast and Accurate Time Series Classification with WEASEL. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Nov. 2017), 637–646. https://doi.org/10.1145/3132847.3132980 arXiv: 1701.07681.

[53] Patrick Schäfer and Ulf Leser. 2018. Multivariate Time Series Classification with WEASEL+MUSE. *arXiv:1711.11343 [cs]* (Aug. 2018). http://arxiv.org/abs/1711.11343 arXiv: 1711.11343.

[54] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855, 1-23 (2008), 40.

[55] Joan Serrà, Santiago Pascual, and Alexandros Karatzoglou. 2018. Towards a universal neural network encoder for time series. *arXiv:1805.03908 [cs, stat]* (May 2018). http://arxiv.org/abs/1805.03908 arXiv: 1805.03908.

[56] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I Webb. 2020. TS-CHIEF: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery* 34, 3 (2020), 742–775.

[57] Deepika Singh, Ashutosh Singh, and Sonia Tiwari. 2021. *Thermal Analysis of Realistic Breast Model With Tumor and Validation by Infrared Images*. 208–218.

[58] S Eva Singletary, Craig Allred, Pandora Ashley, Lawrence W Bassett, Donald Berry, Kirby I Bland, Patrick I Borgen, Gary Clark, Stephen B Edge, Daniel F Hayes, et al. 2002. Revision of the American Joint Committee on Cancer staging system for breast cancer. *Journal of clinical oncology* 20, 17 (2002), 3628–3636.

[59] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. 2021. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians* 71, 3 (2021), 209–249.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[61] Gabriel Taubin. 1995. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE international conference on computer vision*. IEEE, 852–857.

[62] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 242–264.

[63] Kerem Sinan Tuncel and Mustafa Gokce Baydogan. 2018. Autoregressive forests for multivariate time series modeling. *Pattern recognition* 73 (2018), 202–215.

[64] Ken Ueno, Xiaopeng Xi, Eamonn Keogh, and Dah-Jye Lee. 2006. Anytime classification using the nearest neighbor algorithm with applications to stream mining. In *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 623–632.

[65] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

[66] Lulu Wang. 2017. Early diagnosis of breast cancer. *Sensors* 17, 7 (2017), 1572.

[67] Xing Wang, Jessica Lin, Pavel Senin, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2016. RPM: Representative Pattern Mining for Efficient Time Series Classification.. In *EDBT*. 185–196.

[68] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 1578–1585. https://doi.org/10.1109/IJCNN.2017.7966039 ISSN: 2161-4407.

[69] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[70] Zhiwen Xiao, Xin Xu, Huanlai Xing, Shouxi Luo, Penglin Dai, and Dawei Zhan. 2021. RTFN: a robust temporal feature network for time series classification. *Information Sciences* 571 (2021), 65–86.

[71] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 947–956.

[72] Chunyong Yin, Sun Zhang, Jin Wang, and Neal N Xiong. 2020. Anomaly detection based on convolutional recurrent autoencoder for IoT time series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 1 (2020), 112–122.

[73] Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. 2020. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (April 2020), 6845–6852. https://doi.org/10.1609/aaai.v34i04.6165 Number: 04.

[74] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. 2017. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics* 28, 1 (Feb. 2017), 162–169. https://doi.org/10.21629/JSEE.2017.01.18 Conference Name: Journal of Systems Engineering and Electronics.

[75] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. 2014. Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management*. Springer, 298–310.

[76] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.