Microchemical Journal 214 (2025) 113902

Contents lists available at ScienceDirect

Microchemical Journal

journal homepage: www.elsevier.com/locate/microc

Qualitative recognition of waste textiles based on near-infrared spectroscopy and ModernTCN

Cong Shi [a,*], Junfeng Sang [b]

[a] Beijing Dongjin Electromechanical Technology Co., Ltd., Beijing 100029, China
[b] School of Materials Design and Engineering, Beijing Institute of Fashion Technology, Beijing 100029, China

ARTICLE INFO

Keywords:
Waste textile recognition
Qualitative analysis
ModernTCN
InceptionTime
MiniRocket
Near infrared spectroscopy

ABSTRACT

To address the challenge of online recognition of multi-class waste textiles in their sorting process, the Fabric-ModernTCN, a CNN architecture with strong fabric recognition performance and real-time inference capability, is proposed in this paper. The design of Fabric-ModernTCN is inspired by the state-of-the-art ModernTCN in the field of time series analysis. Using the Fabric-NIR-Dataset, a raw near-infrared spectral dataset including the spectra of 18 categories of fabrics, and its four preprocessed versions generated by applying the standard normal variate transformation, SG smoothing, min–max normalization, and arPLS baseline correction to Fabric-NIR-Dataset, respectively, five categories of Fabric-ModernTCN models are trained, and their recognition performances are evaluated. The most effective preprocessing method, SG smoothing, and its corresponding Fabric-ModernTCN model are selected. Further performance evaluation experiments of this model are conducted, demonstrating its classification accuracy of 93.28 % and F1-score of 94.47 %. The comparative experiments are conducted against two baseline models, InceptionTime and MiniRocket + MLP (MiniRocket coupled with a linear classifier). The results reveal that the Fabric-ModernTCN model outperforms both baseline models across five performance metrics: classification accuracy, precision, recall, F1-score, and inference time per sample. Specifically, the Fabric-ModernTCN model achieves improvements of 2.97 % and 1.82 % in classification accuracy, and 2.52 % and 1.31 % in F1-score, respectively, compared to the baseline models. Regarding computational efficiency, the inference time per sample of the Fabric-ModernTCN model is only 0.0025612 s, corresponding to an FPS of 390.4448, which highlights its strong real-time performance. The ablation experiment results further validate the rationality of the structural design and parameter selection of Fabric-ModernTCN.

1. Introduction

Approximately 100 million tons of waste textiles are generated annually, with the United States and China being the largest contributors [1]. Landfilling and incineration remain the most common methods for handling waste textiles worldwide. Landfilling not only occupies large areas of land but also leads to the release of landfill gases and harmful substances during the degradation process. Incineration, while reducing waste textiles volume and generating energy, releases toxic gases and microplastics, contributing to long-term ecological pollution. The recycling and reuse of waste textiles can effectively address the negative environmental, economic, and social challenges posed by waste textiles. It can reduce the consumption of natural resources and minimize the environmental pollution caused by the incineration or landfilling of waste textiles [2,3].

As waste textiles contain various components, different technologies are required for their processing, making sorting the critical first step in their recycling process [3–5]. With continuous advances in textile research and development, textiles have become increasingly functional, resulting in a wider variety of categories and components of waste textiles, along with more complex structures. This complexity has made the rapid and accurate automatic recognition and sorting of waste textiles a significant technical challenge, leading to the low current recycling and reuse rates.

Many technologies have been developed to promote the advancement of waste textile sorting, including hyperspectral imaging, spectral analysis, machine learning, and mechatronic systems [6–12], some of which leverage the integration of near-infrared spectroscopy (NIRS) and artificial intelligence (AI). NIRS is a powerful analytical technique that measures the interaction between light and matter, generating a

[*] Corresponding author.
E-mail address: leonshi47@hotmail.com (C. Shi).

https://doi.org/10.1016/j.microc.2025.113902
Received 30 January 2025; Received in revised form 30 April 2025; Accepted 7 May 2025
Available online 9 May 2025

spectrum that provides information about a sample's chemical and physical properties [13]. It operates in the near-infrared (NIR) region, with a wavelength range from 780 nm to 2500 nm, where it measures the light absorption at various wavelengths. This absorption is directly related to the functional groups present in the sample, such as C-H, N-H, O-H, and S-H. The absorption spectrum of the sample can be obtained by measuring its degree of light absorption at these specific wavelengths. The positions and intensities of the absorption peaks in the spectrum reflect its composition structure and the contents of its various components. NIRS has been successfully applied in many fields such as medicine, agriculture, petroleum, and the textile industry [14].

There have been several achievements in the application of near-infrared spectroscopy combined with artificial intelligence techniques, particularly deep learning, to textile recognition and sorting [10–12,15]. Valvan has developed Fibersort® [10], a low-cost automated sorting line combining NIRS and RGB camera technology to identify and separate textiles based on fiber composition and color properties. Fibersort® utilizes AI models to predict the concentration of fibers, covering both pure fibers and fiber blends.

Riba et al. [11] proposed an algorithm for classifying and separating 100 % pure samples and binary mixtures of the most common textile fibers using near-infrared spectroscopy combined with mathematical treatment of the spectra through convolutional neural networks. The results in their paper show that classification accuracies reach 100 % for seven categories of pure fibers (cotton, linen, wool, silk, polyester, polyamide, and viscose), and 90–100 % for binary mixtures (mixture of viscose and polyester, mixture of cotton and polyester).

Becker et al. [12] researched the detection, prediction, and sorting of polyester-containing waste textiles using advanced near-infrared spectroscopy. Their research demonstrated that the polyester content in polycotton or a blend of polyester and viscose, acrylic, or wool could be identified using SQALAR software. Additionally, their work examined the effects of wrinkles and moisture in the textile samples on their NIR spectra.

Xia et al. [15] designed a deep learning model, an enhanced hybrid network combining the architectures of the convolutional neural network (CNN) and long short-term memory (LSTM), to predict cotton content in polycotton fabrics using near-infrared spectroscopy. After applying the MSC preprocessing method to the spectral data, this model achieved a root mean square error (RMSE) of 0.75 % on the test dataset.

Despite the progress achieved, including the advancements mentioned above, several challenges that require further investigation remain. The NIR spectra of fabrics contain not only information about the chemical composition of different fiber components, but also interferences from the reflection, scattering, and absorption of NIR light due to the complex structural characteristics of fabrics, such as surface morphology, porosity, and thickness. Additionally, composition-related interferences, such as those caused by dyes in the dyeing process or water content due to humidity, further complicate the spectra. To enable accurate fabric recognition, further research should be conducted to develop more advanced methods that effectively mitigate these interferences and reliably extract fiber composition information from complex NIR spectra.

Furthermore, waste textiles contain various categories of fabrics constituting many fiber components, not limited to single-component or dual-component fabrics. Therefore, additional research is needed to design models or algorithms that can accurately recognize various categories of textiles simultaneously, with strong generalization ability, good real-time performance, and compatibility for deployment on edge devices.

In recent years, numerous advanced deep-learning-based models have emerged in the field of time series analysis [16–23]. Although these models were originally developed to process ordered temporal data (such as meteorological and financial time series), their inherent strengths—such as the adaptive capture of dependencies and sequential patterns, the ability to model nonlinear relationships, multi-scale feature extraction, and enhanced robustness—make them highly promising for spectral data analysis. By leveraging these capabilities, complex features in near-infrared spectral data can be effectively extracted, thereby improving the accuracy and efficiency of spectral classification, regression, and related analytical tasks.

Given that traditional machine learning algorithms' recognition accuracy and generalization ability have increasingly been surpassed by deep learning algorithms, this paper focuses on the design of deep learning models for the direct and non-invasive recognition of various categories of waste textiles based on their NIR spectra. Through extensive experimentation, a novel deep neural network model, Fabric-ModernTCN, is developed based on ModernTCN [16]. This model demonstrates strong performance and fast inference capability. Using a NIR spectral dataset of waste textiles comprising 18 categories of fabrics, four spectral data preprocessing methods are applied, and corresponding Fabric-ModernTCN models are trained and evaluated to select the most effective data preprocessing method. Comparative performance evaluations are conducted between Fabric-ModernTCN and two baseline models—InceptionTime and MiniRocket (coupled with a linear classifier)—on the fabric classification task. The experimental results show that Fabric-ModernTCN outperforms the baseline models across five key performance metrics: classification accuracy, precision, recall, F1-score, and inference time per sample. Furthermore, a series of ablation studies on the design choices of Fabric-ModernTCN are conducted. The results of ablation experiments confirm that the model's architecture and parameter configuration are rationally designed.

This paper is organized into five sections. The first section introduces waste textiles' primary processing methods and associated environmental challenges, emphasizes the importance of recycling, and highlights the critical role of sorting. It also reviews several state-of-the-art NIRS-based textile sorting technologies and recognition approaches while summarizing the main contributions of this work. The second section elaborates on the architectural design of Fabric-ModernTCN. The third section introduces the sources, categories, and compositions of the fabric samples. It also describes the methods used to determine the sample compositions and the procedures for collecting the spectral data. Furthermore, this section provides a detailed explanation of the four data preprocessing methods employed. The fourth section presents the experimental evaluations, including the descriptions of the dataset and experimental settings, the selection of the most effective data preprocessing method among four candidates, the training, testing, and performance evaluation of the Fabric-ModernTCN model, comparative experimental evaluations between Fabric-ModernTCN and two baseline models, as well as ablation studies on Fabric-ModernTCN. Finally, the fifth section presents the conclusions of the study.

2. Fabric-ModernTCN

ModernTCN is a one-dimensional convolutional neural network architecture for time series analysis, which maintains the efficiency advantage of a convolution-based model and provides a better balance of performance and efficiency. ModernTCN has shown outstanding performances on multiple downstream tasks in time series analysis, demonstrating strong generalization ability. Building on the core component of ModernTCN, ModernTCN Block [16], a fabric classification network, Fabric-ModernTCN, is designed.

Fabric-ModerntTCN consists of three parts: Embedding, Backbone, and Classification Head. The structure diagram of Fabric-ModernTCN is illustrated in Fig. 1(a), and the structure diagram of the Backbone is illustrated in Fig. 1(b). Let us make the following assumptions: the input to the Fabric-ModernTCN is $X_{in} \in R^{B \times M \times L}$, which denotes the input spectral sequences tensor with batch size B and length L and M variables. X_{in} is also the input to the Embedding, and the output of the Embedding is $X_{em} \in R^{B \times M \times D \times L}$, where D denotes the number of feature channels. X_{em} is the input to the Backbone, and the output of the Backbone is

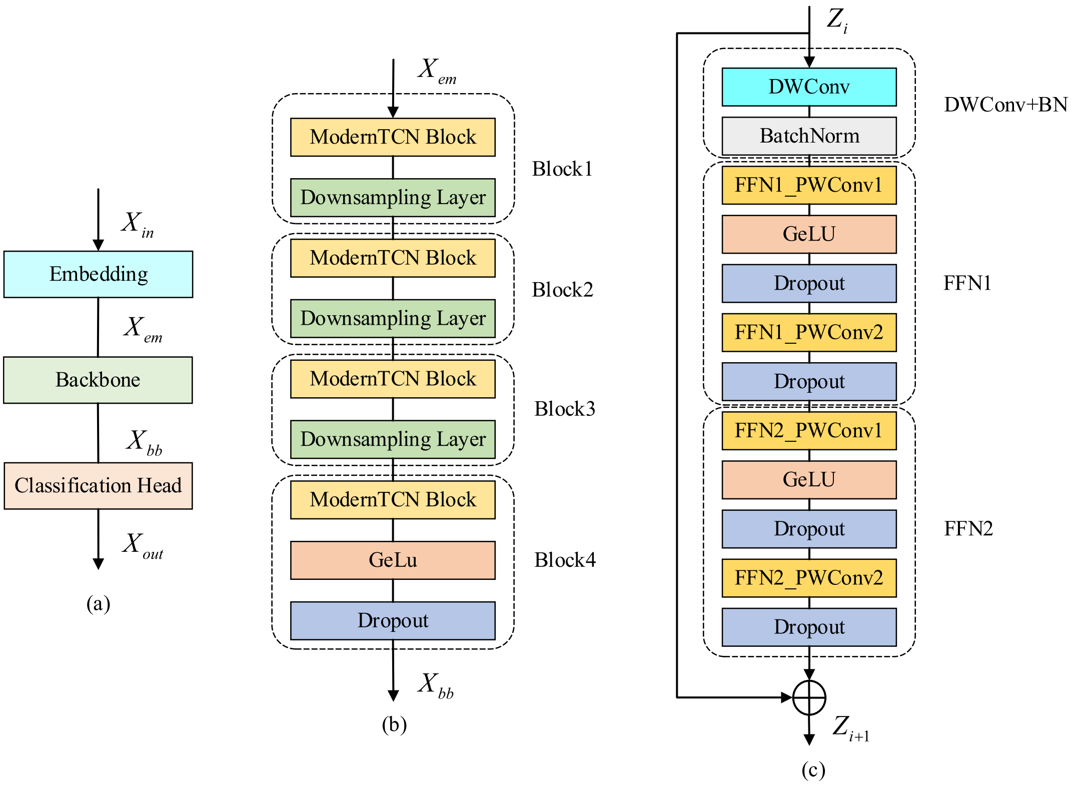

Fig. 1. The structural diagrams of Fabric-ModernTCN (a), its Backbone (b), and its constituent ModernTCN Block (c).

$X_{bb} \in R^{B\times M\times D\times V}$, where V denotes the sequential length of the feature map. X_{bb} is the input to the Classification Head, and the output of the Classification Head is $X_{out} \in R^{B\times C}$, where C is the number of fabric categories.

2.1. Embedding

The function of the Embedding is to extract features from the input X_{in} to obtain its feature representation X_{em}. In the implementation of Embedding, the input X_{in} is first reshaped to $X'_{in} \in R^{B\times M\times 1\times L}$ using the unsqueezing operation. Next, a 1D grouped convolution and batch normalization (BN) are applied to X'_{in}, converting the number of channels from 1 to D while preserving the independence between the M univariate sequences, i.e., maintaining the variable dimension. This enables the Backbone network to capture information from the additional variable dimension.

2.2. Backbone

As shown in Fig. 1(b), the Backbone consists of four sub-blocks: Block1, Block2, Block3, and Block4. Each of the first three sub-blocks is composed of a ModernTCN Block (see Fig. 1(c)) and a Downsampling Layer (strided convolution layer), and the fourth sub-block, Block4, is composed of a ModernTCN Block, a GeLU activation layer (GeLU activation function), and a Dropout layer.

2.2.1. ModernTCN Block

The ModernTCN Block is derived from the work presented in reference [16], and its components are shown in Fig. 1(c). Overall, the ModernTCN Block is a residual block, with its residual mapping consisting of three convolutional modules: DWConv + BN, FFN1, and FFN2. Assuming that the input and output of the ModernTCN Block in Fig. 1(c) are $Z_i \in R^{B\times M\times D\times L}$ and $Z_{i+1} \in R^{B\times M\times D\times L}$, respectively, where $i = 1,2,3,4$, and L_i represents sequential length ($L_4 = V$), the relationship between Z_i and Z_{i+1} is formulated by $Z_{i+1} = Block(Z_i) + Z_i$, where $Block$ denotes the operation of the residual mapping. The parameters of convolution layers in the ModernTCN Block are listed in Table 1.

In the following part of this subsection, DWConv + BN, FFN1, and FFN2 are described in detail. As shown in Fig. 1(c), DWConv + BN consists of a depthwise convolution layer DWConv followed by a batch normalization layer BatchNorm. Both input channels and output channels of the DWConv are $M\times D$, and the number of groups (group number) of the DWConv is also $M\times D$. This type of depthwise convolution is channel-independent (feature-independent) and variable-independent, making it learn the sequential dependency of each univariate sequence independently and capture corresponding feature representations. A large kernel is adopted in the DWConv to increase ERFs and improve the sequential modeling ability. The BatchNorm performs batch normalization, which preserves the D feature channels.

FFN1 and FFN2 are modules consisting of two group-convolution-based pointwise convolution layers, one GeLU activation layer, and two Dropout layers, as shown in Fig. 1(c). The two pointwise convolution layers in FFN1, namely FFN1_PWConv1 and FFN1_PWConv2, are both group convolutions, with the kernel size and stride set to 1, and the group number set to M. The input and output channels of FFN1_PWConv1 are $M\times D$ and $r\times M\times D$, respectively, while the input and output channels of FFN1_PWConv2 are $r\times M\times D$ and $M\times D$, where r is the expansion factor. These two pointwise convolution layers adopt

Table 1
Parameters of convolution layers in ModernTCN Block [16].

Layer	input channels	output channels	kernel size	stride	group number
DWConv	$M\times D$	$M\times D$	large size	1	$M\times D$
FFN1_PWConv1	$M\times D$	$r\times M\times D$	1	1	M
FFN1_PWConv2	$r\times M\times D$	$M\times D$	1	1	M
FFN2_PWConv1	$D\times M$	$r\times D\times M$	1	1	D
FFN2_PWConv2	$r\times D\times M$	$D\times M$	1	1	D

an inverted bottleneck structure, where the hidden channels in the FFN1 module are r times wider than the input channels. The structure of FFN2 is similar to that of FFN1, except that the group numbers of FFN2_PWConv1 and FFN2_PWConv2 are both D, and the input and output channels of FFN2_PWConv1 are $D \times M$ and $r \times D \times M$, respectively, while the input and output channels of FFN2_PWConv2 are $r \times D \times M$ and $D \times M$, respectively. Apart from these differences, the rest of the structure of FFN2 is the same as that of FFN1. The FFN1 is used for learning the new feature representations per variable and the FFN2 for capturing the cross-variable dependency per feature channel. In the ModernTCN Block, each of DWConv + BN, FFN1, and FFN2 only fuses information across one dimension—sequential, feature, or variable dimension, which demonstrates the decoupling design in modern convolution as described in reference [16].

Both the ModernTCN Block and the fourth sub-block (Block4) in the Backbone employ the GeLU activation layer. The choice of GeLU is motivated by its distinctive characteristics [24]. GeLU is slightly more computationally expensive than ReLU, activating neurons based on a Gaussian distribution and retaining negative learning with a smoother non-linearity. GeLU's smooth nature and probabilistic treatment of negative values make it particularly effective for complex, deep networks in tasks requiring sophisticated feature extraction. The Dropout layers used in both the ModernTCN Block and the fourth sub-block (Block4) are the same.

2.2.2. Downsampling layer

The Downsampling Layer consists of a BN layer and a strided convolution layer. The convolution of the strided convolution layer is performed with no padding and its kernel size is equal to the stride length. Unlike the pooling layer, which discards part of the pixels directly, the strided convolution layer retains the spatial information of the entire local region by performing weighted computations with the convolving kernel, thereby effectively reducing information loss. Compared to the pooling operation, strided convolution not only downsamples the feature map (reducing its spatial dimension) but also simultaneously learns feature representations while preserving spatial information. This makes the strided convolution more effective in maintaining important spatial details and enhancing the network's overall feature representation ability.

2.3. Classification Head

In Classification Head, the flatten layer or reshape operation is used for changing the Backbone's output $X_{bb} \in R^{B\times M\times D\times V}$ to $X_{hcin} \in R^{B\times(M\times D\times V)}$ (X_{bb}'s shape is (B, M, D, V) and X_{hcin}'s shape is $(B, M\times D\times V)$). Then a linear layer with softmax activation is to map X_{hcin} to the final classification result $X_{out} \in R^{B\times C}$.

3. Data collection and data preprocessing methods

3.1. Data collection

This work is based on the near-infrared spectra of 11,147 fabric samples across 18 categories. These categories include Polyester, Cotton, Polyester/Cotton, Polyester/Spandex, Nylon/Spandex, Polyester/Wool, Wool, Cotton/Spandex, Special Class, Polyester/Nylon, Nylon, Silk, Silk/Cotton, Polyester/Viscose, Nylon/Cotton, Acrylic, Viscose, and Nylon/Viscose. Among the 18 categories of fabric samples, seven categories of fabric samples are single-component fabrics (pure fiber), ten categories of fabric samples are dual-component fabrics (blends of two types of fibers), and the remaining category is classified as Special Class. Within the dual-component fabrics of the same category, the mass fractions of the same components exhibit variations. For instance, two samples classified under the Polyester/Spandex category could have different proportions of the spandex component. In a few categories, such as Polyester/Cotton and Polyester/Wool, the mass fractions of one component span a wide range from about 5 % to 90 %, reflecting diverse compositional ratios. In several categories, the mass fractions of the minor component in dual-component fabrics are relatively low, often falling below 10 %. The varying mass fractions appearing in dual-component fabrics significantly increase the difficulty and complexity of fabric recognition. Additionally, high-gloss samples, such as those subjected to special finishing processes like calendaring, tend to produce significant specular reflection rather than diffuse reflection during near-infrared spectral measurements. This phenomenon leads to severe signal loss and results in spectra with non-obvious characteristic peaks. Such samples with non-obvious characteristic peaks are categorized as Special Class to facilitate analysis.

The category names explicitly specify their compositions for all samples except those in the Special Class category. For example, the samples in the Polyester/Cotton category consist of two types of fibers: polyester and cotton; the samples in the Polyester category consist of one type of fiber: polyester. The samples categorized under the Special Class exhibit heterogeneous and non-fixed compositional characteristics, representing high-gloss samples with variable fiber constituents.

Approximately 1,700 fabric swatches were collected to acquire NIR spectral data of all fabric samples mentioned above. Among these fabric swatches, dual-component fabric swatches were selectively collected to ensure that, within each fabric category, the mass fractions of the same components were as uniformly distributed as possible across their actual variation range. These fabric swatches originated from three main sources: pre-consumer textile waste (production remnants from factories), post-consumer textile waste (end-of-life textiles collected after consumer use), and commercial textiles, with the majority derived from pre-consumer textile waste.

All fabric swatches were analyzed on a Nicolet iS10 FTIR spectrometer equipped with a Smart iTR accessory, under the following parameter settings: 8 cm^{-1} resolution, 4000–400 cm^{-1} scanning range, and 32 co-added scans. The FTIR spectra of all fabric swatches were acquired and qualitatively analyzed for fiber-type classification. This spectral analysis enabled definitive compositional classification of all fabric swatches, with each subsequently assigned a category name and a category label.

Each fabric swatch was folded into four layers using both face-side and reverse-side folding methods. For each folded swatch, NIR spectral data were collected from its different regions using the NIRQuest512-2.5 spectrometer. This spectrometer is primarily equipped with a fiber optic probe, a tungsten-halogen light source, and a white polytetrafluoroethylene (PTFE) reference standard. It operates over a wavelength range of 900 to 2500 nm with an optical resolution of less than 10.0 nm. During the NIR spectral data acquisition process, the spectrometer was configured with an integration time of 10 ms and a scan to average of 1. Environmental conditions were controlled, with ambient temperature maintained between 10 °C and 30 °C, and relative humidity kept within the range of 30 % to 70 %.

For each fabric swatch, four to eight NIR spectra were acquired using the aforementioned procedure, yielding the NIR spectral data of four to eight fabric samples as described in this study. If a specific fabric category had fewer swatches, more spectra were collected per swatch to compensate for the limited sample size; conversely, fewer spectra were collected for categories with a larger number of swatches. Using the methods described above, the NIR spectra of 11,147 fabric samples across 18 categories were acquired. The corresponding category names and category labels of these fabric samples were also assigned.

3.2. Data preprocessing methods

The near-infrared spectrum of a sample can be influenced by factors such as the measurement instrument, the measurement environment, and the physical state of the sample itself. As a result, noise or interference is inevitably introduced [25]. For example, optical components in a near-infrared spectrometer, such as lenses, gratings, and detectors, can introduce noise due to their inherent imperfections. The physical properties of the sample, such as uniformity, surface smoothness, and density, can affect the stability of the spectral signal. Additionally, variations in temperature and humidity can distort the measurement results.

In the raw spectrum of a sample, in addition to the near-infrared characteristic spectral data (true spectral data) that reflect the sample's compositional components and their contents, noise or interference signals may also be present due to the aforementioned factors. Effectively denoising and eliminating interference from raw NIR spectral data remains a challenge, as preprocessing these signals, such as through filtering or smoothing, carries the risk of distorting the true spectral information.

In the case of fabric samples, in addition to the factors mentioned above, the spectral data measurements are also influenced by factors such as inconsistent fabric thickness and variability in the fabric content ratio within the same category. These factors lead to poor consistency in the spectral data. As a result, the preprocessing methods that rely on parameter estimation from the sample spectral matrix (i.e., methods that derive parameters from multiple samples) may not be suitable. Building on this, four preprocessing methods are investigated to determine whether preprocessing the NIR spectra of fabric samples is necessary and, if so, to identify the most effective method. These four preprocessing methods are standard normal variate (SNV) transformation, Savitzky-Golay (SG) smoothing, min–max Normalization (MMN), and arPLS baseline correction.

3.2.1. *Standard normal variate transformation*

The SNV transformation method normalizes each spectrum by obtaining the difference between this spectrum and its mean and then dividing this difference by its standard deviation. After SNV transformation, each spectrum will have a mean of 0 and a standard deviation of 1. The SNV transformation was introduced by Barnes et al. [26] to reduce multiplicative interferences from scattering and particle size effects. SNV transformation can be used to make all spectra comparable in terms of intensity, correct for variations in optical path length and light scattering, and compensate for changes in the material's surface roughness.

The mathematical expressions for SNV normalization are shown in the Equations. (1)-(3), where $\boldsymbol{x}$ is the spectrum of a sample with P data points, $\overline{x}$ is the mean of $\boldsymbol{x}$, s is the standard deviation of $\boldsymbol{x}$, x_k is the spectral data at the point k (the k-th spectral value of the sample), $\boldsymbol{x}^{snv}$ is the transformed spectrum of the sample.

$$\overline{x} = \left(\sum_{k=1}^{P} x_k\right)/P \tag{1}$$

$$s = \sqrt{\sum_{k=1}^{P} (x_k - \overline{x})^2/(P-1)} \tag{2}$$

$$\boldsymbol{x}^{snv} = (\boldsymbol{x} - \overline{x})/s \tag{3}$$

3.2.2. *Savitzky-Golay smoothing*

The Savitzky-Golay smoothing (SG smoothing) [27] is based on local polynomial fitting, it performs local weighted polynomial fitting on the signal. This mobile polynomial smoothing consists of identifying a polynomial of order q in a window of length $2m+1$ centered on point k. The value at the point k is replaced by the value computed using the polynomial. The window is then moved to the next point, and the same calculation is repeated. The polynomial order and the smoothing window length are two key parameters that determine the complexity and accuracy of the fitting. Generally, the order and window length should be adjusted according to the characteristics of the signal to achieve the best filtering effect.

The specific calculation formula for the Savitzky-Golay smoothing is given by Eq. (4).

$$x_k^{sg} = \left(\sum_{j=-m}^{m} w_j^{sg} x_{k+j}\right)/A \tag{4}$$

where $\boldsymbol{x}$ is the spectrum of a sample with P data points, x_k^{sg} is the filtered spectral data at the point k, the smoothing window is centered at point k and has a length of $2m+1$, x_k is the original data of the sample at the point k, w_j^{sg} are the weight coefficients, A is the normalization coefficient. When the polynomial order and smoothing window length are set, A and w_j^{sg} can be determined using the method proposed by Savitzky and Golay [27].

3.2.3. *Min-max normalization*

Min-max normalization (MMN) is one of many normalization methods for near-infrared spectra. It normalizes each spectrum by obtaining the difference between the spectrum and its minimum value and then dividing this difference by the range, which is the difference between its maximum and minimum values. By doing so, MMN mitigates the effects of baseline shift and intensity variation, facilitating more effective comparison and analysis of spectra. However, it can alter the relative intensities of different peaks, potentially distorting the spectral shape and leading to a degradation in model performance in certain applications.

The calculation formula for MMN is given by Eq. (5).

$$\boldsymbol{x}^{mmn} = (\boldsymbol{x} - x_{min})/(x_{max} - x_{min}) \tag{5}$$

where $\boldsymbol{x}$ is the spectrum of a sample with P data points, $\boldsymbol{x}^{mmn}$ is the normalized spectrum of the sample, x_{max} and x_{min} are the maximum and minimum values, respectively, among the P spectral values of the sample.

3.2.4. *arPLS baseline correction*

The near-infrared spectrum collected in practice is complex in composition and frequently considered as the superposition of a slowly varying baseline, random noise, and true spectral data, which consist of a few narrow signal peaks that reveal the physical and chemical properties of the measured sample. Baseline drift and offset in the spectral signals typically reduce the accuracy of both qualitative and quantitative analysis of samples. Therefore, as a commonly used preprocessing method, baseline correction adjusts the slowly varying baseline to zero, eliminating its influence on the true spectral data. However, in some cases, baseline removal techniques can distort the real proportions between absorbance peaks, and therefore, caution should be used when interpreting the results. Baseline correction methods are generally classified into two types: one requiring manual parameter selection and the other with inherent learning capability. The former relies on expert experience for parameter selection, and improper parameter choices can lead to poor baseline correction performance. The latter does not require manual parameter selection, but iterative updates of parameters typically incur a large computational cost. With the advancement of computer technology, the latter has gradually become a research focus, giving rise to many methods such as deep-learning-based methods [28] and PLS-based methods [29–31].

The asymmetric reweighted penalized least squares smoothing (arPLS) [31] is one of the PLS-based baseline correction methods. The baseline $\boldsymbol{z}$ can be estimated using the following algorithm. Suppose $\boldsymbol{x}$ is the spectrum of a sample with P data points. In Eq. (6), $\boldsymbol{W}$ is a diagonal matrix with an asymmetric weight array $\boldsymbol{w}$ of P data points on its diagonal. $\boldsymbol{U}$ is the second-order difference matrix with $P-2$ rows and P columns. The parameter λ adjusts the balance between fidelity and smoothness. A larger λ leads to a smoother $\boldsymbol{z}$. $\boldsymbol{U}$ is expressed in Eq. (7).

$$\boldsymbol{z} = (\boldsymbol{W} + \lambda \boldsymbol{U}^T \boldsymbol{U})^{-1} \boldsymbol{W}\boldsymbol{x} \tag{6}$$

$$U = \begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{pmatrix} \tag{7}$$

Based on arPLS, the baseline $\boldsymbol{z}$ is obtained through the iterative algorithm. First, $\boldsymbol{w}$ is set to an array of all ones, and the initial baseline $\boldsymbol{z}$ is obtained using Eq. (6). Then, by computing $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{z}$, the P data points in $\boldsymbol{d}$, i.e., $d_1, d_2, \cdots, d_P$, can be divided into two regions: the positive region $\boldsymbol{d}_+$, where $\{d_i | d_i = x_i - z_i \geq 0, d_i \in \boldsymbol{d}\}$, and the negative region $\boldsymbol{d}_-$, where $\{d_i | d_i = x_i - z_i < 0, d_i \in \boldsymbol{d}\}$. The weight array $\boldsymbol{w}$ for the next iteration is constructed based on Eq. (8), where m_- and σ_- represent the mean and standard deviation of $\boldsymbol{d}_-$, respectively. The baseline $\boldsymbol{z}$ is updated by using this new $\boldsymbol{w}$ and solving Eq. (6). The iteration ends when the number of iterations *itermax* reaches the maximum pre-set value, or when $(|w^t - w^{t-1}|/|w^t|) < radio$, where t is the iteration step and *radio* is the pre-set threshold. The baseline correction is achieved by subtracting $\boldsymbol{z}$ from $\boldsymbol{x}$.

$$w_i = \begin{cases} \dfrac{1}{1 + \exp[2((x_i - z_i) - (-m_- + 2\sigma_-))/\sigma_-]} & x_i \geqslant z_i \\ 1 & x_i \leqslant z_i \end{cases} \tag{8}$$

4. Experimental evaluations

4.1. Dataset

The NIR spectral data derived from the 18 categories of fabric samples described above constitute the near-infrared spectral dataset upon which this study is based. This dataset is henceforth referred to as Fabric-NIR-Dataset, which consists of three subsets: the training, validation, and test sets. The sample counts of the training, validation, and test sets are 8,084, 2,021, and 1,042, respectively, with an approximate ratio of 8:2:1.

During the construction of the training, validation, and test sets, all NIR spectra derived from the same fabric swatch were assigned to the same subset. This ensured that the three subsets were mutually independent. This independence is essential for realistic model evaluation and generalization assessment. In addition, when constructing the training set, dual-component fabric samples were preferentially selected such that, within each fabric category, the mass fractions of the shared components were as evenly distributed as possible across their respective ranges. This sampling strategy was designed to improve both the classification accuracy and the generalization capability of the recognition model.

Given the class imbalance in the dataset, the differences in data distribution among the training, validation, and test sets, and the considerable variation in mass fractions of identical components within the same fabric category, the validation set was deliberately enlarged to approximately twice the size of the test set. A larger validation set provides a more representative sampling of the diverse spectral patterns within each fabric category, especially for dual-component fabrics with varying mass fractions. This adjustment enables the validation set to more effectively reflect the complexity of the data distribution, thereby facilitating a more robust and reliable selection of the best-performing model during training. Additionally, the test set contains sufficient data to enable a robust and reliable evaluation of the recognition model's performance.

To illustrate the spectrum characteristics of the samples in the dataset, five samples from each category are randomly selected from the training set, and their NIR spectra are shown in Fig. 2, each spectrum of

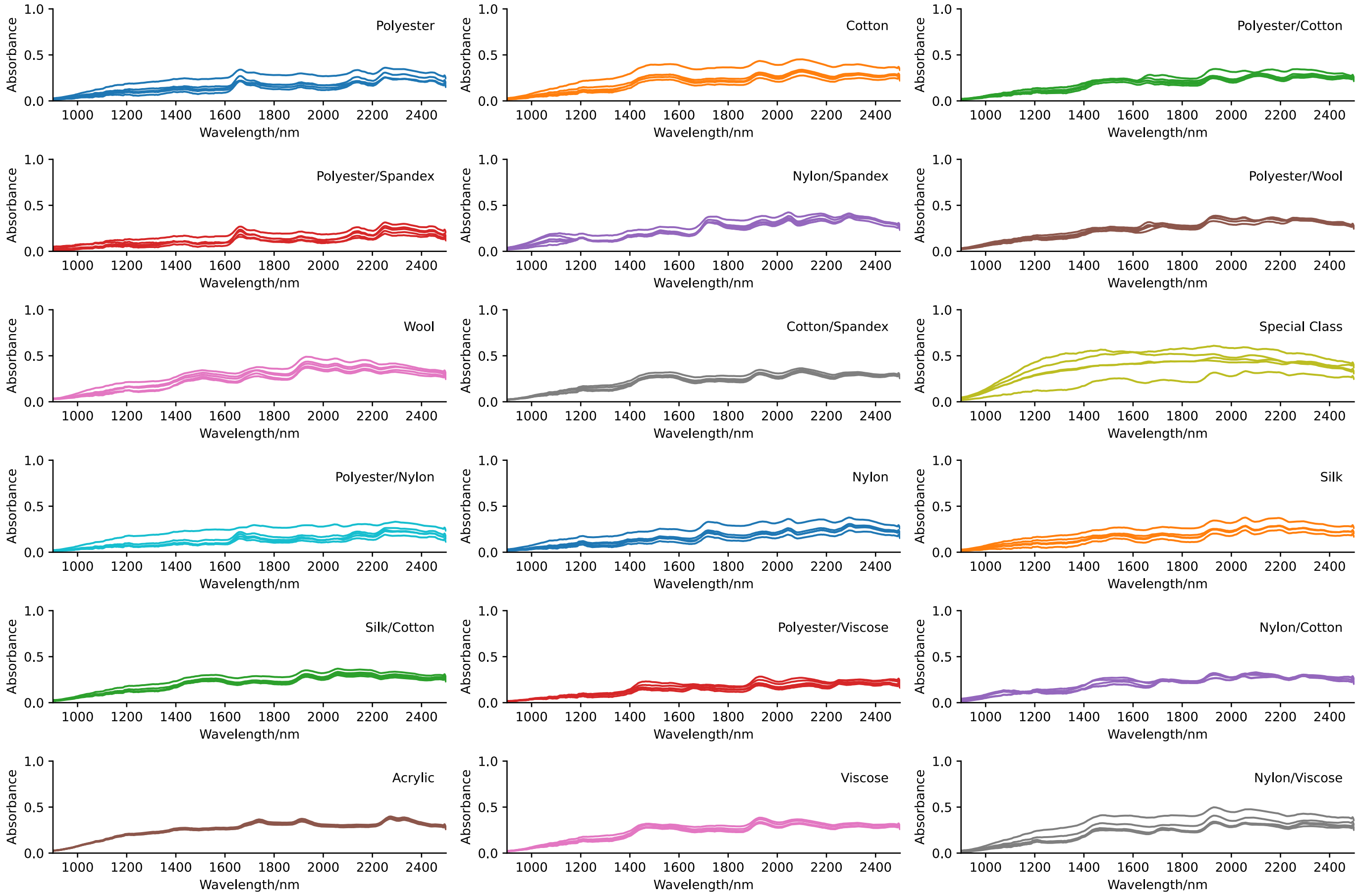

Fig. 2. Raw NIR spectral graphs of samples in the training set of Fabric-NIR-Dataset.

which has 1600 data points.

4.2. Experimental settings

All models mentioned in this paper run on the PyTorch framework, and the experimental platform is configured as follows: 64-bit Windows 10 operating system, 11th Gen Intel(R) Core(TM) i5-11600KF @ 3.90 GHz (CPU), Nvidia GeForce GTX 1080Ti (GPU), with code written in a Python 3.9 environment.

As described in Section 3.1, the NIR spectrum of each fabric sample is a one-dimensional sequence with a length of 1600. The parameters of Fabric-ModernTCN are set as follows: $M = 1$, $L = 1600$, $D = 256$, the expansion factor r is set to 2, the dropout rates for all Dropout layers are set to 0.1, and both the kernel size and stride in the Downsampling Layers are set to 3. The four large kernel sizes of four depthwise convolution layers (DWConv) in Backbone are 31, 29, 27, and 13 (large size in Table 1), respectively.

The parameters of SG smoothing are set as follows: the order of the polynomial is 3, and the length of the smoothing window is 11. An arPLS baseline correction with $\lambda = 10000$,$ratio = 0.05$, and $itermax = 100$ is employed.

All models are trained under the following conditions: the SGD optimizer is used with a momentum of 0.9, and its other parameters are set to their default values; the learning rate is set to 0.001, no learning rate adjustment strategy is employed, and the maximum number of training epochs is set to 100, and the batch size is set to 32, and the loss function used for training all models is the cross-entropy loss function. During training, with the number of epochs ranging from 1 to 100, the best-performing model is selected when its classification accuracy on the validation set reaches its maximum.

Considering the impact of factors such as random initialization of the neural network, random data batching, stochastic optimization steps (using SGD), and the random operation of Dropout during the training process, all model comparison experiments are repeated five times. The averaged classification performance metrics are calculated as the mean of the results from the corresponding five independent runs, ensuring a robust evaluation of model performance. This will not be repeated for the rest of the paper.

4.3. Evaluation of data preprocessing methods

The four preprocessing methods described in Section 3.2—SNV, SG smoothing, MMN, and arPLS baseline correction—are applied to preprocess the spectra data of all fabric samples in the raw Fabric-NIR-Dataset, resulting in four corresponding datasets, each consisting of samples whose spectra have been preprocessed using one of these methods. Subsequently, the Fabric-ModernTCN network is sequentially trained using the training sets from the raw dataset and the four preprocessed datasets. The Fabric-ModernTCN model that achieves the highest classification accuracy on the corresponding validation set is selected and saved as the output model. This process results in five categories of Fabric-ModernTCN models: Fabric-ModernTCN(Raw), Fabric-ModernTCN(SNV), Fabric-ModernTCN(SG), Fabric-ModernTCN (MMN), and Fabric-ModernTCN(arPLS). After training, these models are evaluated on their respective test sets, and their classification performance metric values, which are the values of accuracy, precision, recall, and F1-score, are calculated. The experiments are repeated five times to obtain the averaged classification performance metric values for the five models in each category and the classification performance metric values for the best-performing model in each category. These classification performance metric values are presented in Table 2.

The data in Table 2 illustrate that when MMN, SNV, and arPLS are applied to the fabric sample spectra, no improvement in classification performance is observed. However, when SG smoothing is used, the model's classification performance improves, albeit to a limited extent, with the classification accuracy increasing by 1.44 % (for the best model). Given the superior performance of the SG smoothing method, only the SG smoothing preprocessing method is applied to the sample spectra when evaluating the Fabric-ModernTCN model, comparing it with the baseline models, and conducting ablation experiments in the subsequent analysis.

4.4. Evaluation of Fabric-ModernTCN

As shown in Table 2, the Best Fabric-ModernTCN (SG) model achieves classification accuracy, precision, recall, and F1-score of 93.28 %, 94.62 %, 94.80 %, and 94.47 %, respectively, on the test set of the SG smoothing-preprocessed Fabric-NIR-Dataset. Based on the test set and the Best Fabric-ModernTCN (SG) model, the classification accuracy for each of the 18 fabric categories is calculated. The classification confusion matrix of the Best Fabric-ModernTCN (SG) model on the test set is presented in Fig. 3. A bar chart corresponding to these 18 classification accuracies is created and shown in Fig. 5.

As depicted in Fig. 3, the Best Fabric-ModernTCN (SG) model accurately classifies samples from eight fabric categories—Polyester/Spandex, Polyester/Nylon, Silk/Cotton, Polyester/Viscose, Nylon/Cotton, Acrylic, Viscose, and Nylon/Viscose. For the remaining ten fabric categories, the classification accuracies exceed 80 % but are below 100 %. The confusion matrix reveals that the model has lower accuracies in recognizing samples from the Polyester, Cotton, Polyester/Cotton, and Special Class categories. Specifically, the model misclassifies 10 Polyester samples as Polyester/Spandex samples; 8 Cotton samples are misidentified as Cotton/Spandex samples, 2 Cotton samples are misclassified as Viscose samples, and 2 Cotton samples are misclassified as Polyester/Cotton samples. Additionally, 8 Polyester/Cotton samples are incorrectly classified as Polyester/Viscose samples, and 5 Polyester/Cotton samples are wrongly identified as Cotton samples. Furthermore, 13 Special Class samples are misclassified as Polyester/Spandex samples.

The primary reasons for these misclassifications are summarized as follows: firstly, the model's classification performance still requires improvement. Additionally, spandex, also known as elastane or polyurethane fiber, constitutes a low mass fraction in the dual-component fabric samples, which increases the likelihood of misclassification. Furthermore, cotton and viscose, both cellulose fibers, share certain spectral similarities, which may lead to the misclassification of the samples from the Cotton category or Viscose category. As the mass fractions of polyester in the fabrics of the Polyester/Cotton category span a wide range, misclassification is more likely to occur among samples from the Cotton, Polyester, and Polyester/Cotton categories.

Table 2
Performance comparison of the Fabric-ModernTCN models using different preprocessing methods. Best in bold.

Model	Accuracy		Precision		Recall		F1-score	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
Fabric-ModernTCN(Raw)	0.9109	0.9184	0.9278	0.9356	0.9337	0.9387	0.9265	0.9334
Fabric-ModernTCN(SG)	**0.9167**	**0.9328**	**0.9343**	**0.9462**	**0.9367**	**0.9480**	**0.9328**	**0.9447**
Fabric-ModernTCN(MMN)	0.8866	0.8916	0.9131	0.9218	0.9172	0.9207	0.9106	0.9180
Fabric-ModernTCN(SNV)	0.8893	0.9127	0.9090	0.9321	0.9203	0.9366	0.9103	0.9300
Fabric-ModernTCN(arPLS)	0.9002	0.9175	0.9212	0.9328	0.9258	0.9372	0.9176	0.9319

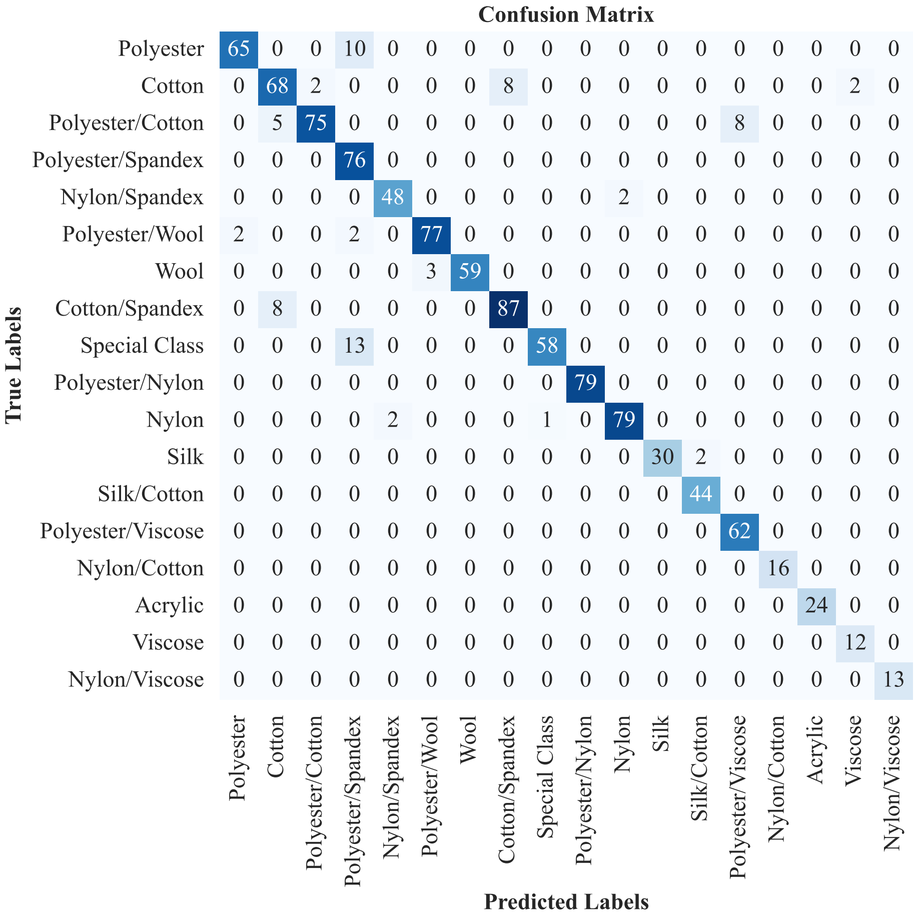

Fig. 3. Confusion matrix of the Best Fabric-ModernTCN (SG) model on the SG smoothing-preprocessed test set.

This is because the varying mass fractions lead to overlapping spectral features, making it difficult to distinguish between these categories accurately. Moreover, during dataset construction, the degree to which samples with high gloss and non-obvious spectral peaks should be classified into the Special Class category is also a matter that requires further investigation and careful handling.

4.5. Comparison with baselines

To further evaluate the performance of Fabric-ModernTCN, comparative experiments are conducted to compare it with the two baselines: InceptionTime [32] and MiniRocket (coupled with a linear classifier) [33]. The choice of these two baselines is motivated by the design consideration for Fabric-ModernTCN: achieving high classification accuracy while maintaining a lightweight architecture and fast inference speed. Middlehurst et al. [17] highly acknowledged the advancement of InceptionTime in Section 4.7 of their review article published in 2024. Similarly, Farahani et al. [21] identified InceptionTime as one of the top-performing time series classification algorithms in the abstract of their paper published in 2025. MiniRocket achieves an excellent balance between classification accuracy and computational efficiency. This balance makes MiniRocket particularly well-suited for real-time or large-scale time series classification tasks, where speed and accuracy are both critical factors. Given that the fabric recognition task explored in this study requires both high classification accuracy and fast inference speed, InceptionTime and MiniRocket are selected as the baselines for comparison with Fabric-ModernTCN in this comparative analysis.

The comparative experiments are conducted under the same conditions and hyperparameter settings as those used for Fabric-ModernTCN (as described in Section 4.2), using the same training set, validation set, test set, and the same evaluation metrics. The classification performance metric values for InceptionTime and MiniRocket are obtained in the same manner as described in Section 4.3. For clarity, the architectures of InceptionTime and MiniRocket, as employed in this study, are introduced first.

4.5.1. InceptionTime

InceptionTime is an Inception-based architecture for time series classification (TSC). It incorporates residual blocks, which help mitigate the vanishing gradient problem by facilitating a direct flow of gradients. As shown in Fig. 4, the InceptionTime architecture employed in this study consists of two distinct residual blocks—Residual Block1 and Residual Block2— and Classifier Head. The residual mapping section of each residual block is composed of three Inception modules, each containing four parallel branches. The feature maps from the four branches are concatenated along the channel dimension, which is then processed by a BN layer followed by a ReLU activation layer. The residual connection section of the Residual Block1 consists of a 1D pointwise convolution layer (PWConv in Fig. 4(a)) followed by a BN layer (BatchNorm in the upper part of Fig. 4(a)). The numbers of input and output channels of this 1D pointwise convolution layer are 1 and 128, respectively. The residual connection section of the Residual Block2 consists of only a BN layer (BatchNorm in the lower part of Fig. 4(a)). The input and output channels of the Residual Block1 are 1 and 128, respectively; the input and output channels of the Residual Block2 are 128 and 128, respectively. The Classifier Head is composed of a global average pooling (GAP) layer and a final fully connected softmax layer with the number of neurons equal to the number of classes in the dataset. The GAP layer averages the output of each feature channel across the

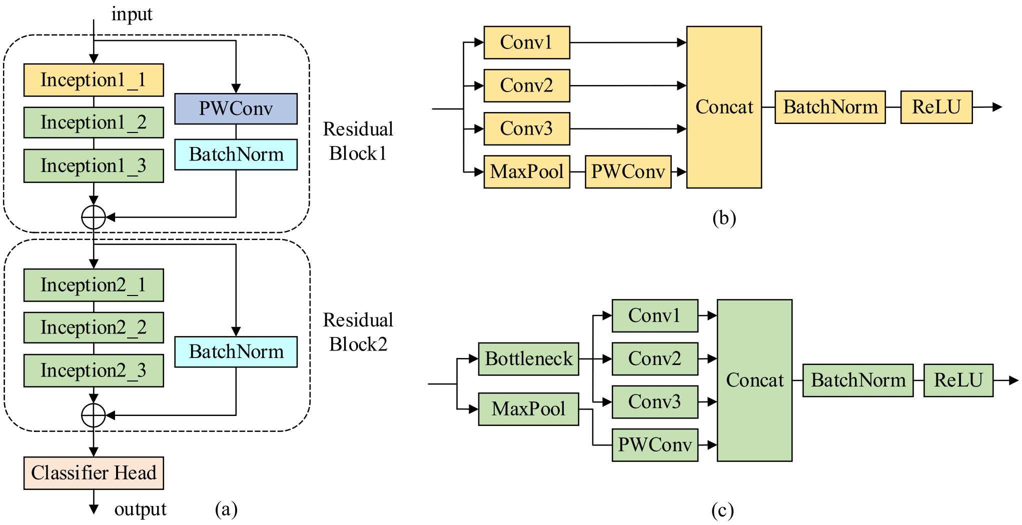

Fig. 4. The structure diagrams of the InceptionTime and its two different Inception modules.

entire temporal dimension. The final fully connected softmax layer converts the output feature map from the GAP layer into the final output. For clarity, all Inception modules are detailed below.

As shown in Fig. 4(b), the first Inception module (Inception1_1 in Fig. 4(a)) in the Residual Block1 consists of four parallel branches: three 1D convolution layers and one 1D max-pooling layer that is followed by a 1D pointwise convolution layer (PWConv in Fig. 4(b)). The kernel sizes of the three 1D convolution layers are 39, 19, and 9, with strides of 1 and padding values of 19, 9, and 4, respectively, to preserve the temporal dimensions of the feature map before and after convolution. The numbers of input and output channels of each 1D convolution layer and the PWConv in Fig. 4(b) are 1 and 32, respectively. The 1D max-pooling layer has a kernel size of 3, a stride of 1, and paddings of 1, ensuring that the temporal dimensions of the feature map remain unchanged during pooling. The feature maps from the four branches are concatenated into a 128-channel feature map, which is sequentially processed by a BN layer and a ReLU activation layer.

As shown in Fig. 4(c), the second Inception module (Inception1_2 in Fig. 4(a)) in the Residual Block1 differs from the Inception1_1, although it also consists of four branches. The first three branches share a bottleneck layer (Bottleneck in Fig. 4(c)), which is a 1D pointwise convolution layer, with the numbers of input and output channels being 128 and 32, respectively. The fourth branch contains a max-pooling layer, followed by a 1D pointwise convolution layer. The three 1D convolution layers in parallel after the Bottleneck are configured similarly to the three 1D convolution layers in Inception1_1, except that the number of their input channels is 32. The max-pooling layer in the fourth branch is nearly identical to the max-pooling layer in Inception1_1, except that the number of its output channels is 128. The numbers of input and output channels of the following 1D pointwise convolution layer (PWConv in Fig. 4(c)) are 128 and 32, respectively.

The third Inception module (Inception1_3 in Fig. 4(a)) in the Residual Block1 is identical to the Inception1_2. The three Inception modules in the Residual Block2 —Inception2_1, Inception2_2, and Inception2_3— are the same as the Inception1_2.

4.5.2. *MiniRocket*

MiniRocket [33], Rocket [34], and MultiRocket [35] are kernel-based (convolution-based) algorithms for TSC. These approaches transform the input time series data into fixed-dimensional feature vectors that effectively represent the temporal patterns inherent in the data. Subsequently, classifiers—such as a simple ridge regression or a neural network—are trained on these feature vectors to perform the classification task.

MiniRocket is an optimized and streamlined version of the original Rocket, to achieve high accuracy while significantly reducing computational expense and increasing speed. MiniRocket leverages a convolutional architecture to efficiently extract discriminative features from time series data through a specialized approach.

MultiRocket builds upon the MiniRocket framework by enhancing the feature extraction process through the use of multiple sets of convolutional kernels, thereby generating a richer and more diverse set of features. This expansion typically results in improved classification accuracy, as it captures a broader range of temporal patterns in the data. However, this increase in feature diversity comes at the cost of higher computational demands, leading to longer training and inference times. Consequently, MultiRocket generally takes more time to process time series data compared to MiniRocket, making it less suitable for applications where inference speed is critical.

The specific implementation scheme of the MiniRocket algorithm in this study is from tsai.models.MINIROCKETPlus_Pytorch [36] in tsai, an open-source deep learning package built on top of Pytorch & fastai, focused on state-of-the-art techniques for time series tasks. It is briefly outlined below.

In terms of the selection of convolution kernels, MiniRocket differs from Rocket. Rocket utilizes convolution kernels with lengths randomly chosen from {7, 9, 11}, weights sampled from $N(0,1)$, bias terms drawn from $U(-1, 1)$, along with random dilation and padding values. In contrast, MiniRocket employs a fixed set of 84 convolution kernels. Each convolution kernel has a length of 9, with the 9 weights constrained to one of two values: $\{-1, 2\}$. Specifically, 6 weights are set to $\alpha = -1$, and 3 weights are set to $\beta = 2$. The specific values of α and β are not critical; what is important is that $\beta = -2\alpha$, ensuring that the sum of all weight values equals zero. In the MiniRocket scheme, dilated convolutions are applied by assigning each kernel the same set of dilations rates, D'. The generation of D' follows the three key conditions detailed below.

(1) **Adjustment of Dilation Rates:** The dilation rates are determined based on the length of the input time series and are constrained to the range $\{2^0, \cdots, 2^{max}\}$, where the exponent values are uniformly distributed between 0 and $max = log_2((L_{input} - 1)/(L_{kernel} - 1))$, with L_{input} representing the input sequence length and L_{kernel} representing the kernel length (which is 9). The dilation rates are selected in this manner because the effective size of a dilated convolution kernel is given by $k' = k + (k-1) \times (d-1)$, where k is the kernel size and d is the dilation rate. This ensures that the maximum size of the dilated kernel does not exceed the length of the input sequence.

(2) **Distinct Dilation Rates:** The dilation rates in D' must be distinct integers.
(3) **Limit on the Number of Dilation Rates:** The maximum number of dilation rates assigned to each convolution kernel is limited to 32.

Subsequently, the count of each unique dilation rate is proportionally allocated based on the total number of features (9996/84 = 119), resulting in the number of features to be computed for each dilation rate such that the sum of features across all dilation rates equals 119. This process completes the generation of D'.

Once the set of fixed dilation rates D' for each convolution kernel is determined, the corresponding padding for each dilated convolution is calculated using the standard padding formula. Based on the number of features to be computed for each dilation rate in D', the corresponding quantiles are computed. The number of quantiles for each dilation rate equals the number of features for that rate. Subsequently, dilated convolution operations are performed on the input samples with a given batch size (referred to as "*chunksize*", a user-defined parameter) to obtain the dilated convolution responses of these input samples. The biases are calculated based on the quantiles of the single-channel convolution responses from 84 samples, which are randomly and repeatedly selected from the input samples with a size of *chunksize*. Using PyTorch's broadcasting mechanism, the dilated convolution responses are compared with the biases to calculate the Positive Predictive Value (PPV) features. For a detailed description of PPV calculation, refer to the relevant reference [33]. Using a single convolution kernel results in 119 PPV features, and using 84 kernels yields 9,996 PPV features, which form a 9,996-dimensional feature vector.

The fabric classification network based on the MiniRocket comprises two parts: the MiniRocket feature extractor and the MiniRocketHead classifier. The MiniRocketHead is implemented as a multilayer perceptron network (MLP), which consists of a batch normalization layer followed by a fully connected (FC) layer. The FC layer is designed with 9,996 input neurons, corresponding to the feature dimensionality of the output of the MiniRocket feature extractor, and 18 output neurons, matching the number of fabric categories. The output of the MiniRocket feature extractor is first processed through the BN layer and then transformed into a probability distribution by the subsequent FC layer using the softmax function. This probability distribution indicates the likelihood of the fabric sample belonging to each of the 18 categories.

4.5.3. *Experimental comparison based on performance metrics*

Using the same datasets and identical training and testing procedures as those employed for Fabric-ModernTCN(SG), the InceptionTime and MiniRocket + MLP networks are trained and tested, resulting in two additional model categories—InceptionTime(SG) and MiniRocket + MLP(SG). Their classification metric values, including accuracy, precision, recall, and F1-score, are presented in Table 3. Furthermore, the averaged inference time per sample in milliseconds (ms) and the averaged FPS (samples per second) for each of the three best-performing models—Best Fabric-ModernTCN(SG), Best InceptionTime(SG), and Best MiniRocket + MLP(SG)—are computed and summarized in Table 3. The averaged inference time, with the preprocessing time for SG smoothing accounted for, is calculated as the mean inference time over 300 samples. For Best MiniRocket + MLP(SG), the averaged inference time also includes the time taken to extract 9,996 PPV features using the MiniRocket transformation method. Based on the SG smoothing-preprocessed test set, the three best models' classification accuracies for each of the 18 fabric categories are calculated. A bar chart demonstrating the three best models' classification accuracies for each of the 18 fabric categories is created and shown in Fig. 5. Three best models' classification accuracies on the test set are shown in Fig. 5.

As shown in Tables 3 and Fig. 5, Best Fabric-ModernTCN(SG) demonstrates superior performance compared to two baseline models, Best InceptionTime(SG) and Best MiniRocket + MLP(SG), across five key performance metrics: accuracy, precision, recall, F1-score, and averaged inference time. Specifically, in terms of classification accuracy, Best Fabric-ModernTCN(SG) achieves an improvement of 1.82 % over Best MiniRocket + MLP(SG) and 2.97 % over Best InceptionTime(SG). Regarding the F1-score, Best Fabric-ModernTCN(SG) outperforms Best MiniRocket + MLP(SG) and Best InceptionTime(SG) by 1.31 % and 2.52 %, respectively. In terms of the averaged inference time, the per-sample inference time of Best Fabric-ModernTCN(SG) is only 2.5612 ms, corresponding to an averaged FPS of 390.4448, highlighting its strong real-time performance.

4.6. *Ablation study on design choices*

In machine learning, particularly in the field of deep learning, the term "ablation study" refers to a methodological process that involves removing or modifying a specific component of a deep learning model and evaluating its impact on the model's overall performance by systematically analyzing the model's behavior and performance metrics, either in the absence of the component or with the modified version of it [37]. Ablation study has become a standard methodological tool for analyzing and understanding the contributions of individual components within the complex neural network architecture.

When designing the Fabric-ModernTCN, it is crucial to consider both its recognition performance and efficiency, particularly regarding its lightweight architecture and inference speed. The objective is to achieve not only excellent recognition performance but also fast inference capability. To this end, ablation experiments are conducted on key components of the Fabric-ModernTCN network: the number of sub-blocks in its Backbone, the dimension of the feature channels (denoted as D) in both Embedding and Backbone, and the expansion factor r (referred to as the FFN ratio in reference [16]) of its ModernTCN Block. These experiments are designed to assess the impact of these parameters on the overall model performance and determine their appropriate values. The experiments follow the same procedures as those employed for Fabric-ModernTCN(SG). Table 4 presents the averaged accuracy, averaged precision, averaged recall, and averaged F1-score for each category of the Fabric-ModernTCN model on the test set. These models are intentionally designed with different parameter configurations, allowing for evaluating the impact of these configurations on model performance.

In Table 4, the second row summarizes the performance metric values of Fabric-ModernTCN(SG), as described earlier. As previously mentioned, this category of model is configured with $r = 2$, $D = 256$, and its Backbone consists of four sub-blocks (including Block3). The experimental results for $D = 128$ and $D = 384$ are presented in the third and fourth rows of Table 4, respectively. The classification performance metrics improve when $D = 256$. The fifth row of Table 4 presents the results obtained when Block3 is removed from the Backbone; in this

Table 3
Performance comparison of three categories of models. Best in bold.

Model	Accuracy		Precision		Recall		F1-score		Inference Time (ms)	FPS
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best	Best	Best
Fabric-ModernTCN(SG)	**0.9167**	**0.9328**	**0.9343**	**0.9462**	**0.9367**	**0.9480**	**0.9328**	**0.9447**	**2.5612**	**390.4448**
InceptionTime(SG)	0.8935	0.9031	0.9181	0.9233	0.9194	0.9237	0.9152	0.9195	3.0843	324.2258
MiniRocket + MLP(SG)	0.9042	0.9146	0.9225	0.9353	0.9272	0.9333	0.9210	0.9316	10.0689	99.3156

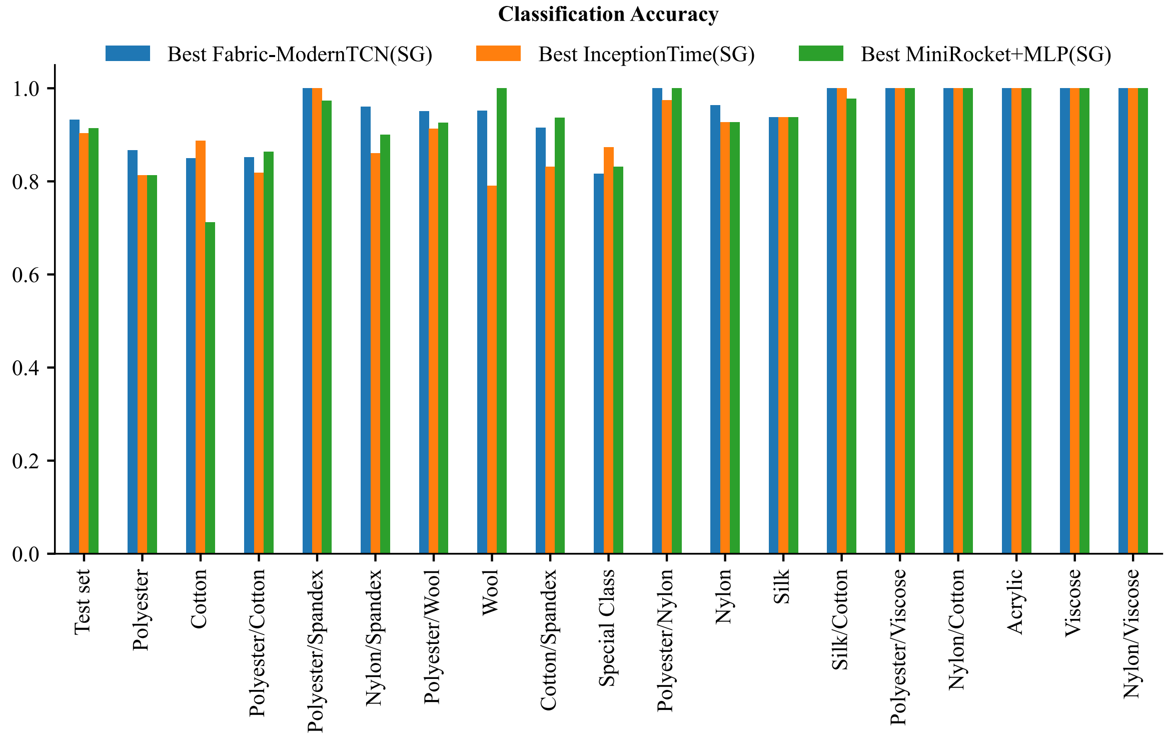

Fig. 5. Classification accuracy comparison of the three best Models on 18 Fabric categories.

Table 4
Results of ablation experiments. Best in bold.

Model	Accuracy	Precision	Recall	F1-score
Fabric-ModernTCN(SG)	**0.9167**	**0.9343**	0.9367	**0.9328**
Fabric-ModernTCN(SG)($D = 128$)	0.9127	0.9281	0.9352	0.9276
Fabric-ModernTCN(SG)($D = 384$)	0.9152	0.9278	**0.9371**	0.9290
Fabric-ModernTCN(SG)(without Block3)	0.9125	0.9251	0.9349	0.9267
Fabric-ModernTCN(SG)($r = 1$)	0.9102	0.9268	0.9331	0.9261

scenario, the classification performance metrics show a decline compared to those of Fabric-ModernTCN(SG). Similarly, a decrease in classification performance metrics is observed when $r = 1$. The results of the ablation experiments suggest that the architecture and parameter settings of Fabric-ModernTCN(SG) are well-designed and effective for achieving both high recognition performance and efficiency in the 18-class fabric classification task.

5. Conclusion

In this study, to address the challenge of online recognition of multi-class waste textiles during their sorting process, the Fabric-ModernTCN model is designed based on the Fabric-NIR-Dataset. This dataset includes the NIR spectra of 11,147 fabric samples from 18 distinct categories. For the dual-component fabric samples within the same category, the mass fractions of their same components exhibit variations. The proposed model demonstrates strong fabric recognition performance and real-time inference capability. The specific research objectives and results presented in this paper are summarized as follows.

To mitigate the interferences present in the spectral data of fabric samples, four preprocessing methods—SNV, SG smoothing, MMN, and arPLS baseline correction—are applied to the raw Fabric-NIR-Dataset, resulting in four corresponding preprocessed versions. Using these five datasets (including the raw dataset), five categories of Fabric-ModernTCN models are trained, and their averaged classification metrics are evaluated. This evaluation leads to the selection of the most effective preprocessing method among four candidates—SG smoothing—and its corresponding optimal model referred to as Best Fabric-ModernTCN(SG). The classification accuracy, precision, recall, and F1-score of the Best Fabric-ModernTCN (SG) model on the test set of the SG smoothing-preprocessed Fabric-NIR-Dataset reach 93.28 %, 94.62 %, 94.80 %, and 94.47 %, respectively. Furthermore, the classification accuracies of the Best Fabric-ModernTCN (SG) model for each of the 18 fabric categories in the test set are calculated. On the test set, the Best Fabric-ModernTCN (SG) model achieves a classification accuracy of 100 % for eight fabric categories—namely, Polyester/Spandex, Polyester/Nylon, Silk/Cotton, Polyester/Viscose, Nylon/Cotton, Acrylic, Viscose, and Nylon/Viscose—while the classification accuracies for the remaining ten categories exceed 80 % but do not reach 100 %. A detailed analysis of these results is presented.

Comparative experiments are conducted to evaluate the performance of Fabric-ModernTCN against two well-known baselines, namely InceptionTime and MiniRocket + MLP (MiniRocket coupled with a linear classifier). The results demonstrate that the Best Fabric-ModernTCN(SG) model outperforms both the Best InceptionTime(SG) and Best MiniRocket + MLP(SG) models in terms of classification accuracy on the test set, achieving improvements of 2.97 % and 1.82 %, respectively. Furthermore, its F1-score on the test set is higher by 2.52 % and 1.31 % compared to the Best InceptionTime(SG) and Best MiniRocket + MLP(SG) models, respectively. In terms of computational efficiency, the averaged inference time per sample of the Best Fabric-ModernTCN(SG) model is shorter than that of both baseline models, at only 0.0025612 s, corresponding to an averaged FPS of 390.4448, highlighting the model's strong real-time performance. Additionally, the results of the ablation studies validate the rationality of the structural design and parameter selection of Fabric-ModernTCN.

CRediT authorship contribution statement

Cong Shi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Junfeng Sang:** Methodology, Investigation, Data curation, Conceptualization, Writing – review & editing.

Funding

This research did not receive any financial support.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] K.H.D. Tang, State of the Art in Textile Waste Management: A Review, Textiles 3 (2023) 454–467, https://doi.org/10.3390/textiles3040027.

[2] M.S. Abbas-Abadi, B. Tomme, B. Goshayeshi, O. Mynko, Y. Wang, S. Roy, K.M. Van Geem, Advancing Textile Waste Recycling: Challenges and Opportunities Across Polymer and Non-Polymer Fiber Types, Polymers 17 (5) (2025) 628, https://doi.org/10.3390/polym17050628.

[3] M. Tripathi, M. Sharma, S. Bala, V.K. Thakur, A. Singh, K. Dashora, V.K. Gupta, Recent technologies for transforming textile waste into value-added products: A review, Curr. Res. Biotechnol. 7 (2024) 100225, https://doi.org/10.1016/j.crbiot.2024.100225.

[4] Textile World. Fiber-to-Fiber Textile Recycling. Available online: https://www.textileworld.com/textile-world/features/2023/09/fiber-to-fiber-textile-recycling/ (Accessed on 28 April 2025).

[5] Ben Amor, R., Ng, K. T. W., Sithi, T. T., & Mahmud, T. S. (2023, May). Opportunities and Challenges for the Sorting of Post-consumer Textile Waste. In Canadian Society of Civil Engineering Annual Conference (pp. 89-99). Cham: Springer Nature Switzerland.

[6] J.R. Riba, R. Cantero, T. Canals, R. Puig, Circular economy of post-consumer textile waste: Classification through infrared spectroscopy, J. Clean. Prod. 272 (2020) 123011, https://doi.org/10.1016/j.jclepro.2020.123011.

[7] G. Bonifazi, R. Gasbarrone, R. Palmieri, S. Serranti, A characterization approach for end-of-life textile recovery based on short-wave infrared spectroscopy, Waste Biomass Valoriz. 15 (3) (2024) 1725–1738, https://doi.org/10.1007/s12649-023-02413-z.

[8] D. Gupta, A. Dubey, Towards Improved Sustainability in The Textile Lifecycle with Deep Learning, in: Proceedings of the 7th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies, 2024, pp. 220–229, https://doi.org/10.1145/3674829.3675077.

[9] R. Tian, Z. Lv, Y. Fan, T. Wang, M. Sun, Z. Xu, Qualitative classification of waste garments for textile recycling based on machine vision and attention mechanisms, Waste Manag. 183 (2024) 74–86, https://doi.org/10.1016/j.wasman.2024.04.040.

[10] Fibersort® technology. Available online: https://www.fibersort.com/en/technology (Accessed on 28 April 2025).

[11] J.-R. Riba, R. Cantero, P. Riba-Mosoll, R. Puig, Post-Consumer Textile Waste Classification through Near-Infrared Spectroscopy, Using an Advanced Deep Learning Approach, Polymers 14 (12) (2022) 2475, https://doi.org/10.3390/polym14122475.

[12] A. Becker, A. Datko, N. Kroell, B. Küppers, K. Greiff, T. Gries, Near-Infrared-Based Sortability of Polyester-Containing Textile Waste, Resour. Conserv. Recycl. 206 (2023) 107577, https://doi.org/10.2139/ssrn.4663236.

[13] C. Pasquini, Near infrared spectroscopy: A mature analytical technique with new perspectives – A review, Anal. Chim. Acta 1026 (2018) 8–36, https://doi.org/10.1016/j.aca.2018.04.004.

[14] W. Zhang, L.C. Kasun, Q.J. Wang, Y. Zheng, Z. Lin, A Review of Machine Learning for Near-Infrared Spectroscopy, Sensors 22 (24) (2022) 9764, https://doi.org/10.3390/s22249764.

[15] H. Xia, R. Zhu, C. Song, H. Yuan, Rapid Quantitative Analysis of Cotton-Polyester Blended Fabrics Using Near-Infrared Spectroscopy Combined with CNN-LSTM, Microchem. J. 200 (2024) 110391, https://doi.org/10.2139/ssrn.4717135.

[16] Luo, D., & Wang, X. (2024). ModernTCN: A modern pure convolution structure for general time series analysis. In The Twelfth International Conference on Learning Representations (ICLR 2024).

[17] M. Middlehurst, P. Schäfer, A. Bagnall, Bake off redux: a review and experimental evaluation of recent time series classification algorithms, Data Min. Knowl. Disc. 38 (4) (2024) 1958–2031, https://doi.org/10.1007/s10618-024-01022-1.

[18] N. Mohammadi Foumani, L. Miller, C.W. Tan, G.I. Webb, G. Forestier, M. Salehi, Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey, ACM Comput. Surv. 56 (9) (2024) 1–45, https://doi.org/10.1145/3649448.

[19] Wang, Y., Wu, H., Dong, J., Liu, Y., Long, M., & Wang, J. (2024). Deep time series models: A comprehensive survey and benchmark. arXiv preprint arXiv: 2407.13278.

[20] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, Q. Wen, Foundation Models for Time Series Analysis: A Tutorial and Survey, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6555–6565, https://doi.org/10.1145/3637528.3671451.

[21] M.A. Farahani, M.R. McCormick, R. Harik, T. Wuest, Time-series classification in smart manufacturing systems: An experimental evaluation of state-of-the-art machine learning algorithms, Rob. Comput. Integr. Manuf. 91 (2025) 102839, https://doi.org/10.1016/j.rcim.2024.102839.

[22] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, L. Sun, Transformers in Time Series: A Survey, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, pp. 6778–6786.

[23] Y. Liu, S. Wijewickrema, C. Bester, S.J. O'Leary, J. Bailey, Time Series Representation Learning with Supervised Contrastive Temporal Transformer, International Joint Conference on Neural Networks (IJCNN) 2024 (2024) 1–8, https://doi.org/10.1109/ijcnn60899.2024.10650516.

[24] M. Lee, Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning, J. Math. (Wuhan) 2023 (2023) 1–13, https://doi.org/10.1155/2023/4229924.

[25] J.M. Roger, A. Mallet, F. Marini, Preprocessing NIR spectra for Aquaphotomics, Molecules 27 (20) (2022) 6795, https://doi.org/10.3390/molecules27206795.

[26] R.J. Barnes, M.S. Dhanoa, S.J. Lister, Standard Normal Variate Transformation and De-Trending of Near-Infrared Diffuse Reflectance Spectra, Appl. Spectrosc. 43 (5) (1989) 772–777, https://doi.org/10.1366/0003702894202201.

[27] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (8) (1964) 1627–1639, https://doi.org/10.1021/ac60214a047.

[28] Q. Jiao, X. Guo, M. Liu, L. Kong, M. Hui, L. Dong, Y. Zhao, Deep learning baseline correction method via multi-scale analysis and regression, Chemom. Intel. Lab. Syst. 235 (2023) 104779, https://doi.org/10.1016/j.chemolab.2023.104779.

[29] Z.M. Zhang, S. Chen, Y.Z. Liang, Baseline correction using adaptive iteratively reweighted penalized least squares, Analyst 135 (5) (2010) 1138–1146, https://doi.org/10.1039/b922045c.

[30] Q. Wang, X.L. Yan, X.C. Chen, P. Shuai, M. Wang, Y.H. Zhang, Spectral baseline estimation using penalized least squares with weights derived from the Bayesian method, Article 148, Nucl. Sci. Tech. 33 (11) (2022), https://doi.org/10.1007/s41365-022-01132-9.

[31] S.-J. Baek, A. Park, Y.-J. Ahn, J. Choo, Baseline correction using asymmetrically reweighted penalized least squares smoothing, Analyst 140 (1) (2015) 250–257, https://doi.org/10.1039/c4an01061b.

[32] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D.F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, F. Petitjean, InceptionTime: Finding AlexNet for time series classification, Data Min. Knowl. Disc. 34 (6) (2020) 1936–1962, https://doi.org/10.1007/s10618-020-00710-y.

[33] A. Dempster, D.F. Schmidt, G.I. Webb, MiniRocket: A very fast (almost) deterministic transform for time series classification, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 248–257, https://doi.org/10.1145/3447548.3467231.

[34] A. Dempster, F. Petitjean, G.I. Webb, ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels, Data Min. Knowl. Disc. 34 (5) (2020) 1454–1495, https://doi.org/10.1007/s10618-020-00701-z.

[35] C.W. Tan, A. Dempster, C. Bergmeir, G.I. Webb, MultiRocket: multiple pooling operators and transformations for fast and effective time series classification, Data Min. Knowl. Disc. 36 (5) (2022) 1623–1646, https://doi.org/10.1007/s10618-022-00844-1.

[36] timeseriesAI/tsai. Available online: https://github.com/timeseriesAI/tsai/blob/main/tsai/models/MINIROCKETPlus_Pytorch.py (Accessed on 28 April 2025).

[37] Meyes, R., Lu, M., de Puiseau, C. W., & Meisen, T. (2019). Ablation studies in artificial neural networks. arXiv preprint arXiv:1901.08644.