applied sciences

MDPI

Article

Tool Wear Classification in Chipboard Milling Processes Using 1-D CNN and LSTM Based on Sequential Features

Jarosław Kurek [1,*], Elżbieta Świderska [2] and Karol Szymanowski [3]

[1] Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences, Nowoursynowska 159, 02-776 Warsaw, Poland
[2] Faculty of Biology and Environmental Protection, University of Lodz, Stefana Banacha 12/16, 90-237 Łódź, Poland; ela.swid22@gmail.com
[3] Department of Mechanical Processing of Wood, Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences, Nowoursynowska 159, 02-776 Warsaw, Poland; karol_szymanowski@sggw.edu.pl
* Correspondence: jaroslaw_kurek@sggw.edu.pl

Abstract: The paper presents the comparative analysis of Long short-term memory (LSTM) and one-dimensional convolutional neural networks (1-D CNNs) for tool wear classification in chipboard milling processes. The complexity of sequence data in various fields makes selecting the right model for sequence classification very important. This research aims to show the distinct capabilities and performance nuances of LSTM and 1-D CNN models, leveraging their inherent strengths in understanding temporal dependencies and feature extraction, respectively. Through a series of experiments, the study unveils that while both models demonstrate competencies in handling sequence data, the 1-D CNN model, with its superior feature extraction capabilities, achieved the best performance, boasting an accuracy of 94.5% on the test dataset. The insights gained from this comparison not only help to understand LSTM and 1-D CNN models better, but also open the door for future improvements in using neural networks for complex sequence classification challenges.

Keywords: tool condition monitoring (TCM); milling processes; tool wear classification; 1-D CNN; long short-term memory (LSTM); sequential features

Citation: Kurek, J.; Świderska, E.; Szymanowski, K. Tool Wear Classification in Chipboard Milling Processes Using 1-D CNN and LSTM Based on Sequential Features. *Appl. Sci.* **2024**, *14*, 4730. https://doi.org/10.3390/app14114730

Academic Editor: Andrea Prati

Received: 11 May 2024
Revised: 26 May 2024
Accepted: 28 May 2024
Published: 30 May 2024

1. Introduction

This paper focuses primarily on the milling segment of the manufacturing process, where the repercussions of imprecise decisions are particularly significant. A novel perspective is introduced through the application of sensor-based techniques, enhancing both the efficiency and accuracy of the process. While the manual inspection of tools is feasible, it is not only time-consuming but also interrupts production. An automated assessment system that alerts the operator when a tool requires inspection or replacement marks a significant advancement in the field. Finding the exact moment when the tool needs replacement is a broad research area with various approaches and solutions available [1–6].

The use of sensor-based methods for monitoring a tool's condition is not new, and it is still widely discussed [7–9]. As tools wear down during use, it can negatively affect product quality. It is important to avoid unplanned production stops and delays in replacing tools, as these can lead to financial losses due to downtime or material waste. Therefore, any automated solution should carefully balance making accurate decisions with providing useful feedback to the operator. Integrating specialized sensors is an effective approach to achieving this balance [10,11].

The literature has many studies on similar challenges [12,13], but the approach to using signals depends on the specific task. It is important to evaluate how useful the signals are in identifying tool conditions during different machining stages [3,14–18]. Despite existing research, there is a need for an automated solution that can easily be integrated into

production and offers both accuracy and ease of implementation. This paper introduces a new solution that uses sensor-based data to address these challenges.

We can use machine learning (ML) to monitor the condition of tools in manufacturing or even deep learning methods based sequential data. By applying ML/DL algorithms to image and sensor-based systems [16,19–22], we can improve how we track the condition of tools. This approach shows that ML can handle even the toughest tasks if we have the right data and training methods.

When it comes to monitoring a tool's condition, the type of data you use—whether it is images or signals—is really important. Using images with convolutional neural networks (CNNs) [1,23,24] has its advantages, but it requires a lot of consistent training data and strict data collection. On the other hand, using signals for monitoring is precise, but managing and processing the large volume of varied sensor data can be challenging.

Traditional approaches in tool wear classification have often relied on singular value features extracted from sensor data, such as the mean, standard deviation, or maximum values within a given aspect, e.g., time window. While useful, these methods can overlook the rich temporal dynamics inherent in sequential data, which can be crucial for accurately predicting tool wear. This paper explores the shift towards leveraging sequential features for tool condition monitoring [25–28], emphasizing the advantages of this approach over traditional singular value feature extraction methods.

Singular value features, though easy to compute and interpret, may not fully capture the intricate behavior of tools under varying operational conditions. Such features tend to provide a snapshot rather than a continuous narrative of the tool's wear process, potentially leading to the oversimplification of complex phenomena. This limitation becomes particularly evident in scenarios where the temporal progression of wear significantly influences the tool's performance and life expectancy.

Sequential features, in contrast, offer a more nuanced and comprehensive representation of tool wear. By preserving the temporal order and dependencies within the data, sequential features enable the modeling of wear progression over time, providing valuable insights into the tool's condition and the dynamics of machining processes. This approach allows for a more detailed and accurate analysis, facilitating early detection of wear and more informed decision-making regarding tool maintenance and replacement.

The extraction of sequential features involves analyzing sensor data over time to identify patterns, trends, and anomalies that may indicate tool wear. Techniques such as time-series analysis, sliding windows, and feature engineering can be applied to transform raw sensor data into a format suitable for analysis. Once extracted, these features can be utilized in various deep learning models, including long short-term memory (LSTM) networks [29,30] and 1D convolutional neural networks (1-D CNNs) [31,32], which are inherently designed to handle sequential data.

The shift from singular value features to sequential features represents a significant advancement in the field of tool wear classification. By fully exploiting the temporal information contained within sensor data, sequential features provide a more accurate and detailed understanding of tool wear dynamics, leading to better predictive models and more effective maintenance strategies. As technology and data analysis methods continue to evolve, the potential for further improvements in this area remains vast, promising even greater gains in manufacturing efficiency and tool life optimization.

A recent study by Li et al. (2024) underscores the effectiveness of data-driven approaches in predicting the remaining useful life (RUL) of tools. They applied a hybrid model combining bidirectional long short-term memory (BLSTM) and convolutional neural networks (CNNs) for metal-forming processes. The CNN-based WearNet was retrained to classify the wear states of workpiece surfaces, and these classifications were then used as inputs for a BLSTM-based regression model to estimate RUL. Their approach achieved high accuracy and low root mean square error (RMSE), highlighting the robustness of data-driven techniques in tool wear prediction [33].

The primary objective of this article is to present a novel, sensor-based diagnostic system designed for the real-time monitoring and classification of tool wear in chipboard milling processes based on sequential features. By leveraging advanced deep learning techniques, specifically 1-D convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, this study aims to enhance the accuracy and efficiency of tool condition monitoring. By providing an in-depth analysis of sequential features derived from a comprehensive set of sensors, this research seeks to not only improve the predictive capabilities of tool wear classification but also to significantly reduce downtime and operational costs in manufacturing settings. Furthermore, this work contributes to the body of knowledge by offering insights into the practical implementation of deep learning models in industrial applications, thereby paving the way for more intelligent and autonomous manufacturing processes.

2. Materials and Methods

2.1. Data Acquisition

This research focused on developing a system to monitor the wear and tear of tools in real time during manufacturing, eliminating the need for production stops. The system's success was measured by looking at the manufacturing signals gathered.

The testing was conducted on a Busellato Jet 130 CNC machining center located in Thiene, Italy. This machine featured a 40 mm cutter head equipped with a single, replaceable carbide blade provided by Faba SA from Baboszewo, Poland.

A chipboard sheet, sized 300 × 150 mm, was used for the tests. It was firmly mounted on a measuring stand, where a groove of 6 mm was milled. The milling process was carried out at a spindle speed of 18,000 rpm and a feed rate of 0.15 mm per tooth, chosen based on an extensive literature review and practical know-how in processing chipboard.

Tool condition was classified into three levels: green, yellow, and red. 'Green' indicated a new or well-kept tool; 'Yellow' pointed to moderate wear but still usable; 'Red' meant the tool was critically worn and needed immediate replacement. Tool wear was assessed using the maximum flank wear (VB_{max}) standard.

Figure 1 shows a close-up look at how the tools wore out, with a focus on the VB_{max} value to check the tool's shape. To figure out if a tool was in 'Green', 'Yellow', or 'Red' condition, we used a Mitutoyo TM-505 microscope (Mitutoyo Corporation, Kanagawa, Japan). This microscope is known for its sharpness in measuring sizes and angles.

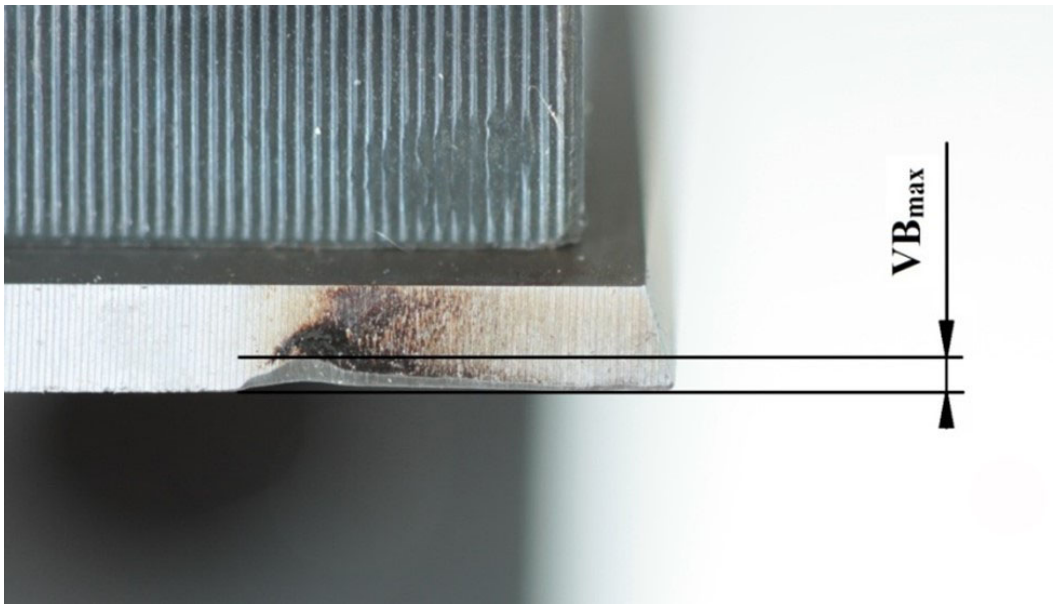

Figure 1. Illustration of tool wear measurement using the VB_{max} parameter.

The geometry of the cutting wedge is depicted in Figure 2.

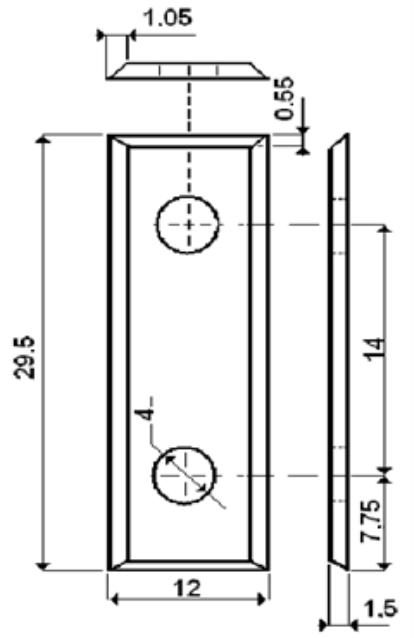

Figure 2. Illustration of cutting edge geometry.

We often stopped the milling to look at the tool's edge under the Mitutoyo TM-505 microscope (Mitutoyo Corporation, Kanagawa, Japan). This microscope can also have a special lens added to check more complex shapes, such as screws and gears. By doing these checks, we could tell how much the tool had worn out and categorize it into one of three wear levels. The specific numbers that defined each category were set as follows:

1. The 'Green' category is for tools with wear and tear (VBmax) from 0 to 0.15 mm, broken down into four smaller wear stages;
2. Tools fall into the 'Yellow' category if their VBmax is between 0.151 mm and 0.299 mm, and this category is divided into two wear stages;
3. The 'Red' category is for tools with a VBmax over 0.299 mm, and this, too, has two stages of wear.

Moreover, the setup for the experiment came with an array of sensors that could capture 11 different types of signals, such as the following:

- Force value in the (1) *X* and (2) *Y*-axes (Kistler 9601A sensor; Impexron GmbH, Pfullingen, Germany);
- (3) Acoustic emission (Kistler 8152B sensor; Kistler Group, Winterthur, Switzerland);
- (4) Noise level (Brüel & Kjær 4189 sensor; Brüel and Kjær, Nærum, Denmark);
- (5) Vibration level (Kistler 5127B sensor; Kistler Group, Winterthur, Switzerland);
- (6) Device-rated current (Finest HR 30 sensor; Micom Elektronika, Zagreb, Croatia);
- (7) Device-rated voltage (Testec TT-Si9001 sensor; Testec, Dreieich, Germany);
- (8) Head-rated current (Finest HR 30 sensor; Micom Elektronika, Zagreb, Croatia);
- (9) Head-rated voltage (Testec TT-Si9001 sensor; Testec, Dreieich, Germany);
- (10) Servo-rated current (Finest HR 30 sensor; Micom Elektronika, Zagreb, Croatia);
- (11) Servo-rated voltage (Testec TT-Si9001 sensor; Testec, Dreieich, Germany).

Data acquisition was facilitated through the use of National Instruments PCI-6111 and PCI-6034E measurement cards (National Instruments Corporation, Austin, TX, USA), respectively, for acoustic emissions and other parameters. This setup ensured comprehensive monitoring, enabling detailed analysis and consistent sensor positioning relative to the workpiece and cutting zone throughout all measurements. The organization and

specific characteristics of the collected data are summarized in the subsequent sections and illustrated with plots of raw signal samples for the selected parameters.

Data was collected by using NI PCI-6111 and PCI-6034E cards, which were responsible for picking up sound emissions and various other measurements. This arrangement allowed for thorough observation, making sure we could analyze the data in depth and keep the sensors in the same spot relative to the workpiece and where the cutting happened for all tests. The following sections will outline how we organized the data and their key features, including charts showing the raw signal examples for the chosen measurements.

We gathered data using a computer that had Lab ViewTM software from National Instruments Corporation (2015 SP1 version, from Austin, TX, USA). For signal capture, we used NI PCI-6034E and NI PCI-6111 (National Instruments Corporation, Austin, TX, USA) cards, also from National Instruments. To get clear sound emission signals, we needed a card that could sample at 2 MHz, allowing us to record for 0.3 s. For other types of signals, a card that could handle a 50 kHz sampling rate and record for 1.1 s was enough. We hooked up each signal to the right card, depending on what frequency we needed to capture, using BNC-2110 connector boxes (National Instruments Corporation, Austin, TX, USA) to make the connections.

To keep the sensor readings stable and reduce the impact of any unwanted noise, we made sure the sensors stayed in the same place next to the workpiece and cutting zone for every measurement. Details about the data we collected are in Table 1.

Table 1. Overview of variable configuration within the datasets.

#	Measured Parameter	Trial Duration	Sampling Rate (Hz)
1	Acoustic Emission	27,999,960	5,000,000
2	X-axis Force	700,000	200,000
3	Y-axis Force	700,000	200,000
4	Noise	700,000	200,000
5	Vibration	700,000	200,000
6	Device Current	30,000	50,000
7	Device Voltage	30,000	50,000
8	Head Current	30,000	50,000
9	Head Voltage	30,000	50,000
10	Servo Current	30,000	50,000
11	Servo Voltage	30,000	50,000

2.2. Sequential Features Generation

The generation of sequential features was meticulously conducted across three distinct domains: time, frequency, and time-frequency. This approach was essential to capture the comprehensive characteristics of the signals under study. Each signal was systematically divided into 100 frames of a predetermined length. For each of these frames, a set of features was extracted, thereby converting the feature representation into sequential data. Consequently, each feature encompasses 100 sequences, effectively capturing the dynamic nature of the signal over time.

In the time domain, feature extraction plays a pivotal role, especially in applications such as tool condition monitoring during milling processes. For this study, 11 unique time-domain features were meticulously extracted for each of the 11 signals. These features include the following [34–36]:

1. Mean: Represents the average value of the signal, providing a basic measure of its central tendency;
2. RMS (root mean square): Offers a measure of the signal's magnitude, accounting for both positive and negative values;
3. Standard deviation: Indicates the variability or dispersion of the signal values from the mean;

4. Shape factor: Describes the shape of the signal's distribution, offering insights into its characteristics;
5. SNR (signal-to-noise ratio): Quantifies the ratio of signal power to noise power within the signal;
6. THD (total harmonic distortion): Measures the distortion caused by harmonics in the signal;
7. SINAD (signal-to-noise and distortion ratio): Combines noise and distortion measurements to assess signal quality;
8. Peak value: Identifies the maximum absolute value within the signal, highlighting extreme events;
9. Crest factor: Compares the peak amplitude of the signal to its RMS value, which is useful for identifying signal peaks;
10. Clearance factor: Provides insights into the peakiness of the signal relative to its overall power;
11. Impulse factor: Assesses the impulsiveness of the signal, indicative of sudden changes in the signal's behavior.

For each of the 11 signals and the 11 features proposed above, the extraction process produced a matrix of size 100 × 121, resulting in a total of 121 potential sequential features from the time domain.

In the frequency domain, a total of seven distinctive features were generated to enhance the analysis. These features were carefully selected for their ability to provide insightful information about the frequency composition and characteristics of the signal. The defined features include the following [37–40]:

1. Mean frequency: Provides an average frequency value weighted by amplitude, offering a centroid measure of the frequency distribution;
2. Median frequency: Represents the frequency at which the power spectrum is divided into two equal halves, indicating the balance of the frequency content;
3. Band power: Measures the total power within a specific frequency band, which is useful for understanding the energy distribution across frequencies;
4. Occupied bandwidth: Defines the bandwidth containing a specified percentage of the total power, highlighting the frequency range most significant to the signal's power;
5. Power bandwidth: Identifies the frequency range within which a certain percentage of the signal's power is concentrated, offering insights into the spread of the signal's energy;
6. Peak amplitude: Captures the maximum amplitude within the frequency spectrum, pointing to the dominant frequency component;
7. Peak location: Specifies the frequency at which the peak amplitude occurs, aiding in the identification of the principal frequency component.

For each of the 11 signals and the seven features proposed above, the extraction process produced a matrix of size 100 × 77, resulting in a total of 77 potential sequential features from the frequency domain.

In the time-frequency domain, the MeanEnvelopeEnergy feature was generated to analyze tool wear in milling processes. This variable captures the mean energy of the upper and lower envelopes of each intrinsic mode function (IMF), providing insight into the tool's condition during operation [41–44].

The generation of the MeanEnvelopeEnergy involved computing the mean envelope energy of the IMFs derived from the input time-domain signals. The empirical mode decomposition (EMD) method was applied to decompose the signals into their IMFs. This is a critical step in capturing the intrinsic oscillatory modes embedded within the data. The MeanEnvelopeEnergy was then calculated by averaging the energy of the upper and lower envelopes for each IMF, thus encapsulating the signal's energy distribution in a comprehensive manner.

For each of the 11 signals under investigation, the process produced a 10-element vector, resulting in a total of 110 potential features (100 × 110 matrix) (MeanEnvelopeEnergy) for subsequent analysis. This rich feature set served as the foundation for the development of predictive models aimed at enhancing tool condition monitoring in milling processes, ultimately contributing to the optimization of manufacturing operations.

The practical implications of using the MeanEnvelopeEnergy are significant, providing a detailed understanding of tool wear dynamics. By integrating these features into deep learning algorithms, the study aims to achieve high-accuracy predictions of tool wear, thereby facilitating timely maintenance interventions and reducing the likelihood of tool failure. This approach not only promises to elevate the precision of tool condition monitoring but also contributes significantly to the sustainability and efficiency of manufacturing workflows.

The mathematical formulation for the MeanEnvelopeEnergy feature is defined as follows:

$$MeanEnvelopeEnergy = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{M_n}\sum_{m=1}^{M_n}\sqrt{U_m^n(t)^2 + L_m^n(t)^2}\right) \quad (1)$$

where

- N is the total number of IMFs considered for each signal;
- M_n represents the number of discrete time points in the n-th IMF;
- $U_m^n(t)$ and $L_m^n(t)$ denote the upper and lower envelope values at time t for the m-th time point in the n-th IMF, respectively.

This formula encapsulates the essence of the MeanEnvelopeEnergy feature by averaging the energies of the upper and lower envelopes across all IMFs and time points. The resulting feature provides a compact yet informative representation of the signal's energy characteristics, pivotal for assessing tool wear in the milling processes.

In the process of generating sequential features, a total of 308 features were produced, represented in the form of a matrix with dimensions 100 × 308. These features can be categorized into three distinct domains, each contributing a specific set of features:

1. Time domain features: A total of 121 features were generated within the time domain, organized into a matrix of dimensions 100 × 121. These features are primarily derived from the time-based analysis of the sequential data;
2. Frequency domain features: In the frequency domain, 77 features were identified and encapsulated within a matrix of 100 × 77. These features are obtained through the analysis of the frequency components of the sequences;
3. Time-frequency domain features: The time-frequency domain contributed 110 features, represented through a matrix with dimensions 100 × 110. This category of features encompasses the characteristics that are derived from both time and frequency domain analyses, providing a comprehensive view of the sequential data across both domains.

This structured approach in feature generation allows for a thorough exploration of the sequential data from multiple perspectives, enhancing the potential for insightful analysis.

2.3. Long Short-Term Memory (LSTM) Network

Long short-term memory (LSTM) networks are a specialized form of recurrent neural networks (RNNs), renowned for their ability to capture long-term dependencies in sequential data. LSTMs address the vanishing gradient problem commonly encountered in traditional RNNs, making them highly effective for tasks involving long sequences.

2.3.1. Experimental Setup

The experiment utilizes a 10-fold cross-validation approach, ensuring a comprehensive evaluation of the LSTM model's performance across different subsets of the data. This method involves dividing the dataset into 10 parts, using nine for training and one for

testing in each fold. This strategy enhances the reliability of the performance metrics, providing a holistic view of the model's generalizability.

2.3.2. Network Architecture of LSTM Model

The LSTM network designed for this experiment comprises the following layers (Table 2):

- Input layer: A sequence input layer that accepts sequences of a specified number of features, preparing the data for LSTM processing;
- LSTM layer: An LSTM layer with 750 units, which was configured to output only the last element of the sequence. This setup is crucial for sequence classification tasks where the final state often contains information about the entire sequence;
- Output layers: A fully connected layer that maps the LSTM outputs to the desired number of classes, followed by a softmax layer for probabilistic classification.

Table 2. Layer information of LSTM architecture.

#	Name	Type	Activations	Learnable Sizes
1	sequenceinput (Sequence input with 308 dimensions)	Sequence Input	308 (C) × 1 (B) × 1 (T)	-
2	lstm (LSTM with 750 hidden units)	LSTM	750 (C) × 1 (B)	InputWeights: 3000 × 308 RecurrentWeights: 3000 × 750 Bias: 3000 × 1
3	fc (3 fully connected layer)	Fully Connected	3 (C) × 1 (B)	Weights: 3 × 750 Bias: 3 × 1
4	softmax (softmax)	Softmax	3 (C) × 1 (B)	-

2.3.3. Training Configuration and Hyperparameteters

Training the LSTM model involves iterating over each fold of the cross-validation setup, separating the data into training and test sets accordingly. The model applies the Adam optimizer with an initial learning rate of 0.01 and is set to train for a maximum of 600 epochs. To mitigate overfitting and enhance model generalization, training incorporates early stopping based on validation performance, specifically using a validation patience of three epochs.

In summary, this LSTM experiment, characterized by its robust cross-validation approach and meticulously defined network architecture, showcases the potential of LSTM networks to handle complex sequence classification tasks. Through this setup, the model is rigorously evaluated, ensuring its effectiveness and adaptability across varied data segments.

2.4. One-Dimensional Convolutional Neural Network (1-D CNN)

One-dimensional convolutional neural networks (1-D CNNs) are a variant of the traditional CNNs tailored for processing sequential data. Unlike their 2D counterparts, which are well suited for image data, 1-D CNNs excel in handling temporal sequences such as time-series data, audio signals, and any form of data where the sequence order is vital.

2.4.1. Network Architecture of 1-D CNN Model

The architecture of the 1-D CNN under consideration was meticulously designed for sequential data processing. The network comprises the following layers (Table 3):

- Input layer: A sequence input layer that accepts data with a specified number of channels, which was tailored to the dimensions of the input sequences;

- Convolutional layers: Two 1-D convolutional layers, each followed by a rectified linear unit (ReLU) activation function. The first convolutional layer utilizes a filter size of $filterSize = 5$ and $numFilters = 128$ filters. To preserve temporal causality, padding is set to "causal". The second convolutional layer doubles the number of filters to $2 \times numFilters$, maintaining the same filter size and padding strategy;
- Normalization layers: Layer normalization is applied following each ReLU activation to stabilize and accelerate the training process;
- Pooling layer: A global average pooling layer is applied to reduce the dimensionality of the feature maps, summarizing the most salient features;
- Output layers: The network concludes with a fully connected layer that maps the pooled features to the desired number of output classes, followed by a softmax layer to facilitate a probabilistic interpretation of the class predictions.

Table 3. Layer information of 1-D CNN architecture.

#	Name	Type	Activations	Learnable Sizes
1	sequenceinput (Sequence input with 308 dimensions)	Sequence Input	308 (C) × 1 (B) × 1 (T)	-
2	conv1d_1 (128 5 × 308 convolutions with stride 1 and padding 'causal')	1-D Convolution	128 (C) × 1 (B) × 1 (T)	Weights: 5 × 308 × 128 Bias: 1 × 128
3	relu_1 (ReLU)	ReLU	128 (C) × 1 (B) × 1 (T)	-
4	layernorm_1 (Layer normalization with 128 channels)	Layer Normalization	128 (C) × 1 (B) × 1 (T)	Offset: 128 × 1 Scale: 128 × 1
5	conv1d_2 (256 5 × 128 convolutions with stride 1 and padding 'causal')	1-D Convolution	256(C) × 1(B) × 1(T)	Weights: 5 × 128 × 256 Bias: 1 × 256
6	relu_2 (ReLU)	ReLU	256 (C) × 1 (B) × 1 (T)	-
7	layernorm_2 (Layer normalization with 256 channels)	Layer Normalization	256 (C) × 1 (B) × 1 (T)	Offset: 256 × 1 Scale: 256 × 1
8	globalavgpool1d (1-D global average pooling)	1-D Global Average Pooling	256 (C) × 1 (B)	-
9	fc (3 fully connected layer)	Fully Connected	3 (C) × 1 (B)	Weights: 3 × 256 Bias: 3 × 1
10	softmax (softmax)	Softmax	3 (C) × 1 (B)	-

2.4.2. Training Configuration and Hyperparameteters

The training of the 1-D CNN was configured to optimize performance and accuracy. The key settings include the following:

- Optimizer: The Adam optimizer was selected for its adaptive learning rate properties, aiding faster convergence;
- Execution environment: Training was expedited by utilizing a GPU, harnessing its computational power to process data in parallel;
- Epochs and learning rate: The network was trained for a total of 600 epochs, with an initial learning rate of 0.01, offering a balance between learning efficiency and the opportunity for fine-tuning;
- Data padding and validation: Sequences were left-padded to ensure uniformity in length, with a separate validation dataset provided to monitor generalization capabilities during training;
- Performance monitoring: Training progress and accuracy metrics were tracked in real time, enabling immediate adjustments and insights into the model's learning trajectory.

This 1-D CNN architecture and its training regimen exemplify the application of convolutional neural networks to sequence data, highlighting their potential to capture the temporal dependencies and features critical for accurate sequence classification.

2.4.3. Validation Technique and Dataset Description

An essential aspect of the experimental setup is the application of a 10-fold cross-validation technique to assess the model's performance and generalizability. Cross-validation is a statistical method used to estimate the skill of deep learning models. It is commonly used in applied deep learning to compare and select a model for a given predictive modeling problem.

The dataset used in this study consists of 75 samples, each representing a different instance of tool wear condition. These samples were collected from a Busellato Jet 130 CNC (Busellato, Thiene, Italy) machining center, where each sample corresponds to a specific milling operation involving chipboard. The dataset was gathered using multiple sensors, including force sensors, acoustic emission sensors, noise sensors, vibration sensors, and electrical sensors, capturing a wide range of signal parameters during the milling process.

Each data sample is characterized by its sequential nature, comprising 100 sequences, with each sequence containing 308 features. These features were extracted from the raw sensor data using techniques such as time-domain analysis, frequency-domain analysis, and time-frequency analysis. The preprocessing steps included normalization and standardization to ensure that all features were on a comparable scale, enhancing the performance and convergence of the deep learning models.

In the 10-fold cross-validation approach, the dataset was partitioned into 10 equal-sized subsets. During the cross-validation process, each subset was used once as the validation set, while the remaining nine subsets formed the training set. This procedure was repeated 10 times, with each of the 10 subsets used exactly once as the validation data. Thus, each training session involved 68 samples for training and seven samples for testing, ensuring that all samples were tested for validation.

This method provides a comprehensive evaluation of the model's performance across the entire dataset, ensuring that the results are not biased by the selection of training and validation samples. The cross-validation technique helped to assess the model's ability to generalize itself to unseen data, providing a robust estimate of its real-world performance.

By utilizing the detailed dataset and rigorous cross-validation approach, the study ensureed a thorough evaluation of the model's capabilities in classifying tool wear conditions, leveraging the rich sequential data to improve accuracy and reliability.

3. Results for LSTM and 1-D CNN

3.1. Results for LSTM

The performance of the long short-term memory (LSTM) model was evaluated using a confusion matrix, as shown in Figure 3.

The rows of the matrix correspond to the true classes, while the columns correspond to the predicted classes. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while the off-diagonal elements are those that are mislabeled by the classifier.

From the confusion matrix, we can observe the following:

- The class 'Green' was correctly predicted 15 times but was confused with 'Red' seven times and 'Yellow' three times;
- The class 'Yellow' had 14 correct predictions, with seven instances misclassified as 'Red' and four as 'Green';
- The class 'Red' showed the highest number of correct predictions at 23, with only one instance for each misclassified as 'Green' and 'Yellow'.

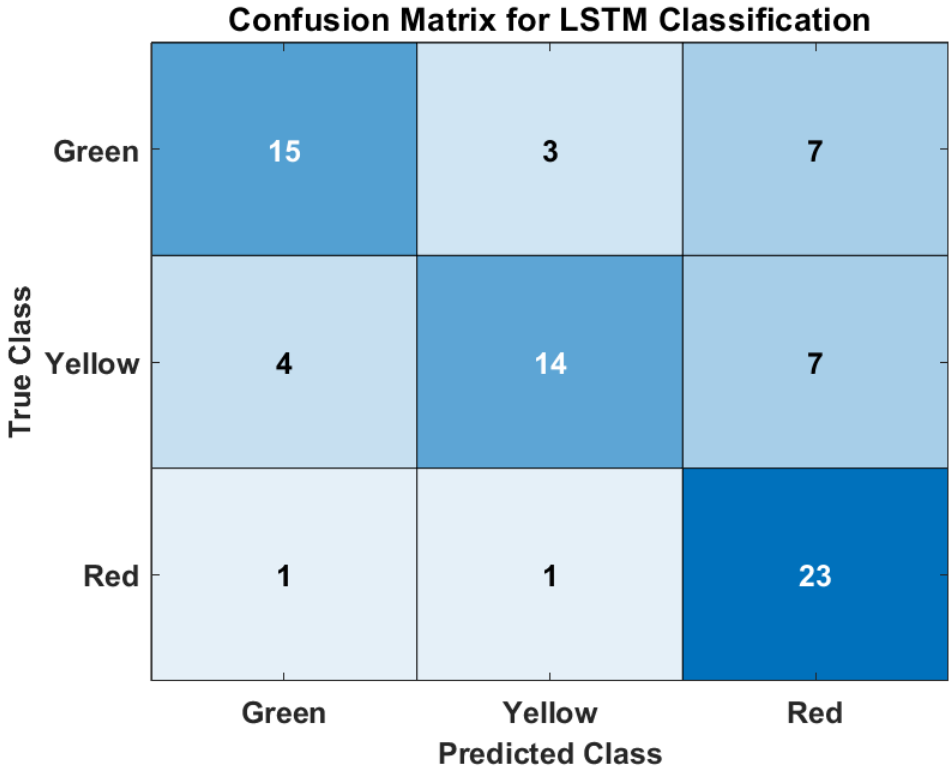

Figure 3. Confusion matrix for the LSTM classification.

In addition to the confusion matrix, a detailed statistical analysis was performed to evaluate the precision, sensitivity, specificity, F1 score, and overall accuracy of the LSTM model. These metrics are summarized in Table 4.

Table 4. Performance metrics of LSTM.

Class	Precision	Sensitivity	Specificity	F1Score	Accuracy
Green	75.00%	60.00%	90.00%	66.66%	80.00%
Yellow	77.77%	56.00%	92.00%	65.11%	80.00%
Red	62.16%	92.00%	72.00%	74.19%	78.66%
Overall					69.33%

Precision measures the model's accuracy in predicting positive instances for each class. The LSTM model shows high precision in identifying the 'Green' and 'Yellow' classes, with 75.00% and 77.77%, respectively, indicating a lower rate of false positives for these classes. The 'Red' class has a slightly lower precision of 62.16%, suggesting more false positives in comparison.

Sensitivity, or recall, reflects the model's ability to correctly identify true positives. The 'Red' class has a notably high sensitivity at 92.00%, showing that the LSTM model is particularly effective at identifying 'Red' instances. However, the 'Green' and 'Yellow' classes have lower sensitivity rates of 60.00% and 56.00%, respectively, which indicates that the model misses a fair number of actual positive instances for these classes.

Specificity measures the model's ability to correctly identify true negatives. The 'Yellow' class exhibits the highest specificity at 92.00%, followed closely by 'Green' with 90.00%, suggesting that the model is generally reliable in recognizing instances that are not of these classes. The 'Red' class, however, has a lower specificity of 72.00%, pointing towards a higher rate of false negatives.

The F1 Score provides a balance between precision and sensitivity. A high F1 Score for the 'Red' class at 74.19% suggests a balanced performance between precision and

sensitivity, despite the lower precision for this class. 'Green' and 'Yellow' classes have F1 Scores of 66.66% and 65.11%, respectively, indicating room for improvement in balancing the precision and sensitivity for these classes.

The overall accuracy of the model stands at 69.33%, which considers the correct predictions made over all classes. While this indicates generally acceptable performance, the variation across different classes suggests that the model's performance is not uniform and could be optimized further, especially for the 'Green' and 'Yellow' classes.

In conclusion, the LSTM model demonstrates a strong ability to identify the 'Red' class with high sensitivity but shows variability in performance across other classes.

Figure 4 shows the training and validation accuracy and loss of the LSTM model during the training process.

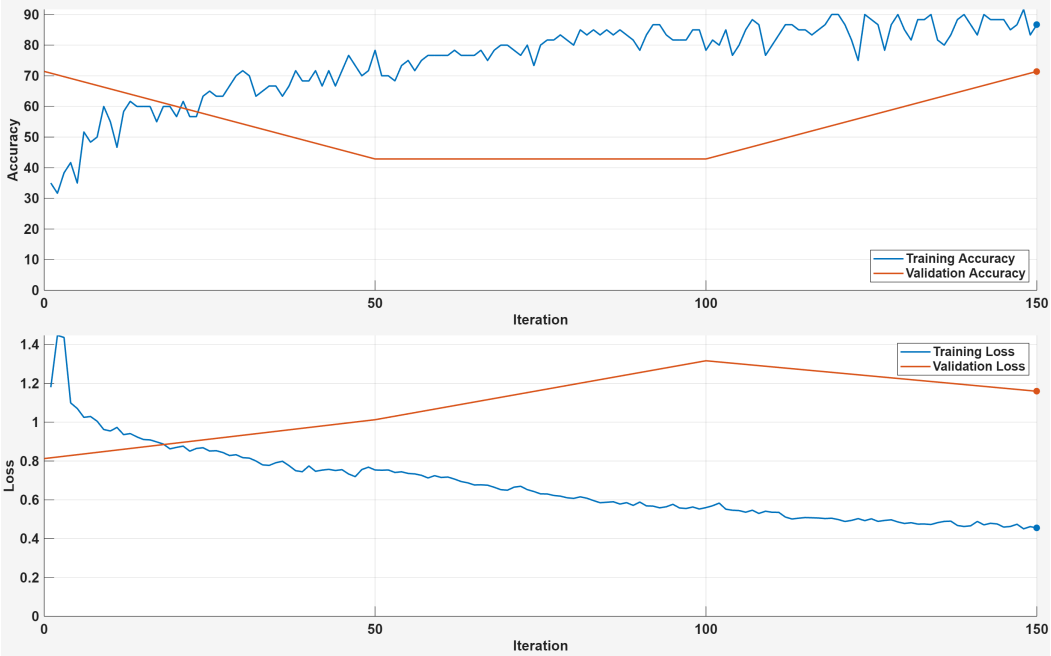

Figure 4. Training and validation accuracy and loss of the LSTM model over iterations.

In the top graph, the blue line represents the training accuracy, while the red line denotes the validation accuracy. Initially, the training accuracy sharply increases, indicating rapid learning in the early stages. As iterations progress, the training accuracy shows a gradual upward trend, eventually stabilizing. The validation accuracy, however, starts off high but shows a slight decreasing trend, indicating potential overfitting as the model learns to perform well on the training data at the expense of its generalization to new data.

The bottom graph illustrates the loss during training (blue line) and validation (red line). The training loss decreases sharply at first and then gradually levels off, suggesting that the model is effectively learning from the training data. However, the validation loss decreases initially but then begins to increase, which is a typical sign of overfitting. The model's increasing loss on the validation set implies that the adjustments made to the model parameters fit the training data more closely, but do not improve its performance on unseen data.

Overall, these trends suggest that while the LSTM model is learning and improving its performance on the training set over time, it is also becoming more specialized to the training data. This specialization reduces its ability to generalize, as evidenced by the increasing validation loss and decreasing validation accuracy after a certain number of iterations. It may be beneficial to introduce regularization techniques, adjust the model's complexity, or provide more varied training data to improve the model's generalization capabilities.

3.2. Results for 1-D CNN

The performance of the one-dimensional convolutional neural network (1-D CNN) in the classification task is summarized in the confusion matrix presented in Figure 5. The matrix shows the number of predictions made by the model for each class compared to the true classes.

From the matrix, we can observe the following:

- The model has a high accuracy in classifying the 'Green' class, with 25 true positives and no false positives or false negatives;
- The 'Yellow' class was mostly classified correctly, with 23 true positives. However, there was one instance of misclassification, where a 'Green' was incorrectly labeled as 'Yellow', and one instance where 'Yellow' was incorrectly labeled as 'Red';
- The 'Red' class also shows high accuracy with 25 true positives, and there were no instances of 'Red' being misclassified as either 'Green' or 'Yellow'.
- Overall, the model demonstrates a strong ability to discriminate between the classes, with a total of only two misclassifications out of 75 observations.

The results indicate that the 1-D CNN has learned to classify the three classes with high accuracy, which is evident from the high true positive rates and low misclassification rates. The model's robustness in classification suggests that the features learned during training are highly discriminative for the given task.

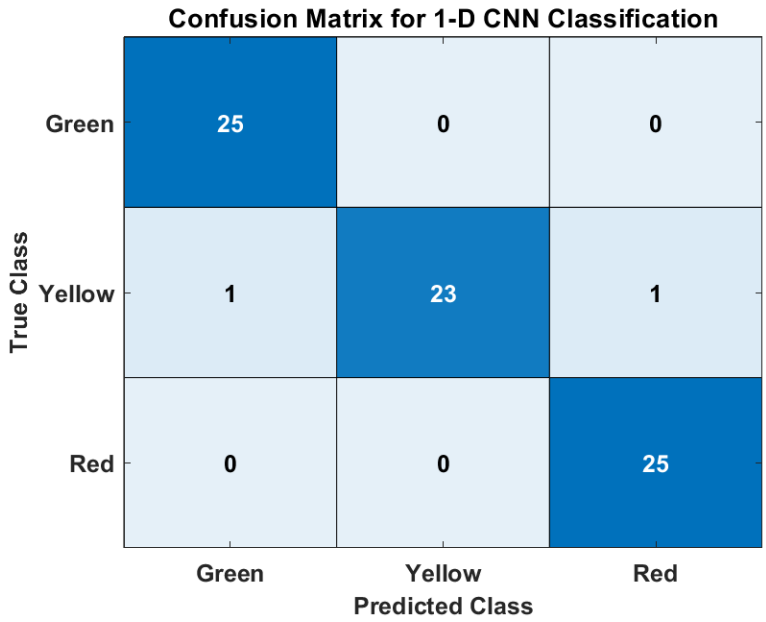

Figure 5. Confusion Matrix for the 1-D CNN classification results.

The statistical performance of the 1-D convolutional neural network (1-D CNN) is quantitatively summarized in Table 5. The evaluation metrics provided are precision, sensitivity (also known as recall), specificity, F1 Score, and overall accuracy, which offer a comprehensive view of the model's performance across the three classes: Green, Yellow, and Red.

Precision quantifies the number of true positive predictions in relation to the number of true positive and false positive predictions. Sensitivity measures the proportion of actual positives correctly identified. Specificity reflects the model's ability to identify actual negatives, while the F1 Score provides a balance between precision and sensitivity. Accuracy is the proportion of true results, both true positives and true negatives, in the dataset.

For the 'Green' and 'Red' classes, the model exhibits high precision of 96.15%, indicating that the model has a high likelihood of correct positive classification. Both classes also have perfect sensitivity at 100.00%, showing that all actual positives were identified.

The specificity for these classes is 98.00%, suggesting that the model is just as robust in correctly identifying negatives. The F1 Score of 98.03% for these two classes indicates an excellent balance between precision and sensitivity.

The 'Yellow' class has an outstanding precision of 100.00%, indicating that every instance classified as 'Yellow' was correct. However, the sensitivity is slightly lower at 92.00%, suggesting that there were some 'Yellow' instances that the model did not identify. The specificity is at a maximum, and the F1 Score is 95.83%, reflecting the slightly lower sensitivity.

The overall accuracy of the model was calculated to be 97.33%, which signifies a high level of correct predictions across all classes. This demonstrates that the 1-D CNN is exceptionally proficient at classifying the dataset with a minimal margin of error.

In conclusion, the 1-D CNN exhibits strong predictive performance across all classes, with high values in precision, specificity, and F1 scores. The slight variance in sensitivity for the 'Yellow' class suggests potential areas for model improvement. The overall metric evaluations illustrate the effectiveness of the model in the given classification task.

Table 5. Performance metrics of 1-D CNN.

Class	Precision	Sensitivity	Specificity	F1 Score	Accuracy
Green	96.15%	100.00%	98.00%	98.03%	98.66%
Yellow	100.00%	92.00%	100.00%	95.83%	97.33%
Red	96.15%	100.00%	98.00%	98.03%	98.66%
Overall					97.33%

The training process of 1-D CNN is illustrated in the graph shown in Figure 6. The graph displays two primary aspects of training progression: the accuracy and loss over iterations.

The top half of the graph depicts the accuracy of training and validation over the course of training iterations. The training accuracy starts at a low value, rapidly increases, and then plateaus, indicating that the model is effectively learning from the data. The validation accuracy closely follows the training accuracy, showing consistent improvement. The slight fluctuations in the training accuracy can be attributed to the inherent stochastic nature of the gradient descent optimization. Notably, the validation accuracy remains slightly below the training accuracy, but it does not decrease over time, which suggests that the model does not overfit to the training data.

The bottom half of the graph represents the training and validation loss. Initially, there is a sharp decline in the training loss, indicating that the model is quickly learning to minimize the error between the predicted and true values. As the iterations proceed, both the training and validation loss curves flatten out, which implies that the model is converging to a solution. The training loss remains consistently lower than the validation loss, which is expected, as the model is directly trained to minimize the training loss. The small gap between the training and validation loss suggests that the model generalizes well to unseen data.

Overall, the convergence of both the accuracy and loss metrics suggests that the model's capacity is appropriate for the complexity of the task. The training process seems stable and does not exhibit signs of divergence or overfitting. This balance between learning from the training data and generalizing to the validation data indicates a well-tuned training regimen, which includes appropriate hyperparameter settings and regularization strategies.

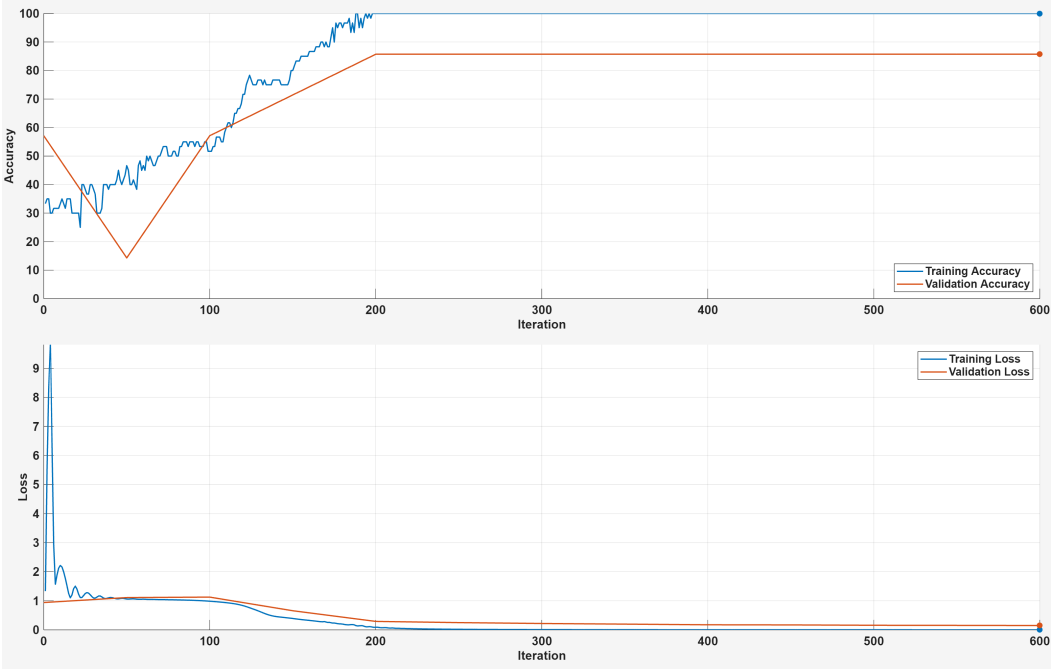

Figure 6. Training and validation accuracy and loss for the 1-D CNN.

4. Discussion

The examination of the LSTM and 1-D CNN models reveals significant insights into their performance, strengths, and limitations in classifying the given dataset. This discussion synthesizes the findings from both models, comparing their outcomes and elucidating the broader implications of these results.

The LSTM model demonstrated commendable capability, particularly in identifying the 'Red' class, with a notable sensitivity of 92.00%. However, its performance varied across the 'Green' and 'Yellow' classes, indicating potential areas for improvement. The confusion matrix and performance metrics suggest that while the model can effectively recognize certain classes, their precision and sensitivity for others could be enhanced. The observed overfitting, as indicated by the divergence between training and validation accuracy, suggests that the model might benefit from regularization techniques or more varied training data to improve its generalization.

Conversely, the 1-D CNN model showcased a robust ability to classify all three classes with high precision and sensitivity, underscored by an overall accuracy of 97.33%. Its performance is particularly noteworthy in the precise classification of the 'Green' and 'Red' classes, with both precision and sensitivity reaching or nearing 100%. The minimal discrepancy between training and validation metrics suggests that the 1-D CNN model has achieved a commendable balance between learning from the training data and generalizing to new, unseen data.

Understanding which features or aspects of the sequence data are most influential in the classification decision is crucial for domain experts and enhances trust in the model's predictions. For the LSTM model, interpretability can be enhanced by examining the attention weights if an attention mechanism is integrated. This would highlight which parts of the sequence the model focuses on when making predictions. For the 1-D CNN model, interpretability can be improved by analyzing the learned filters in the convolutional layers. By visualizing these filters, we can identify which features of the signal, such as specific frequency components or temporal patterns, are most significant for distinguishing between different classes of tool wear. Additionally, techniques such as gradient-weighted class activation mapping (Grad-CAM) can be applied to visualize the regions of the input sequence that contribute most to the model's decisions.

The findings of this study have significant implications and can be extended to other types of manufacturing processes or different types of tools. The methodology and models developed in this study can be adapted for monitoring tool wear in different manufacturing processes such as drilling, turning, or grinding. This would involve collecting appropriate sensor data for these processes and possibly retraining the models to accommodate the unique characteristics of the new datasets. Beyond milling tools, the models can be tested on other types of cutting tools, such as those used in woodworking or metal-cutting industries. This would require validating the models on data from these tools and ensuring that the feature extraction techniques are still relevant.

Future studies could investigate the integration of more advanced regularization techniques, such as dropout or L2 regularization, to enhance the robustness of the LSTM model and reduce overfitting. Data augmentation methods could also be explored to increase the variability of the training data, further improving model generalization. Combining the strengths of LSTM and 1-D CNN models into a hybrid architecture could leverage the sequence processing capabilities of LSTM with the feature extraction prowess of CNNs. Such models could potentially offer improved performance over standalone models.

Developing real-time monitoring systems based on the proposed models would be highly beneficial for industrial applications. This would involve optimizing the models for inference speed and integrating them into production environments with real-time data acquisition and processing capabilities. Implementing explainability techniques such as SHAP (shapley additive explanations) or LIME (local interpretable model-agnostic explanations) could provide further insights into model decision-making processes, thereby enhancing trust and facilitating model validation by domain experts.

In conclusion, while both models exhibit strengths in classifying complex patterns, their varied performance highlights the nuanced decision-making involved in model selection and optimization for specific tasks. The insights gained from this analysis not only contribute to the understanding of LSTM and 1-D CNN models but also pave the way for future advancements in neural network applications. We believe these additional discussions address the reviewer's concerns and provide a more comprehensive assessment of our study's implications and future directions.

5. Conclusions

The study compared LSTM and 1-D CNN models to see how well they classify sequences. The results showed that each model has strengths and limitations depending on the specific task and data characteristics. LSTM models performed well in tasks where understanding long-term patterns is important, while 1-D CNNs excelled in tasks requiring the identification of important features.

These findings highlight the importance of choosing the right model based on the task and data. The study provides insights into the strengths and weaknesses of LSTM and 1-D CNN models and emphasizes the need for a tailored approach in model selection and optimization for sequence classification tasks.

Future research could explore combining LSTM and 1-D CNN models to take advantage of their strengths, leading to more robust solutions for complex classification tasks. Additionally, further research into advanced regularization techniques, data augmentation methods, and hyperparameter tuning might improve model performance and applicability.

In summary, this research contributes to our understanding of neural network applications in sequence classification and provides a nuanced perspective on selecting and optimizing models for specific classification challenges.

Author Contributions: Conceptualization, J.K.; methodology, J.K. and E.Ś.; software, J.K. and E.Ś.; validation, J.K., E.Ś. and K.S.; formal analysis, J.K.; investigation, J.K., E.Ś. and K.S.; resources, J.K.; data curation, K.S.; writing—original draft preparation, J.K.; writing—review and editing, J.K., E.Ś. and K.S.; visualization, E.Ś.; supervision, J.K.; project administration, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to privacy and confidentiality.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, J.; Song, W.; Zhang, W.; Zhao, Y.; Yilmaz, A. Deep learning for use in lumber classification tasks. *Wood Sci. Technol.* **2019**, *53*, 505–517. [CrossRef]
2. Kurek, J.; Antoniuk, I.; Górski, J.; Jegorowa, A.; Świderski, B.; Kruk, M.; Wieczorek, G.; Pach, J.; Orłowski, A.; Aleksiejuk-Gawron, J. Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network. *Mach. Graph. Vis.* **2019**, *28*, 13–23. [CrossRef]
3. Osowski, S.; Kurek, J.; Kruk, M.; Górski, J.; Hoser, P.; Wieczorek, G.; Jegorowa, A.; Wilkowski, J.; Śmietańska, K.; Kossakowska, J. Developing automatic recognition system of drill wear in standard laminated chipboard drilling process. *Bull. Pol. Acad. Sci. Tech. Sci.* **2016**, *64* , 633–640. [CrossRef]
4. Jegorowa, A.; Kurek, J.; Antoniuk, I.; Dołowa, W.; Bukowski, M.; Czarniak, P. Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard. *Wood Sci. Technol.* **2021**, *55*, 271–293. [CrossRef]
5. Kurek, J.; Antoniuk, I.; Świderski, B.; Jegorowa, A.; Bukowski, M. Application of Siamese Networks to the Recognition of the Drill Wear State Based on Images of Drilled Holes. *Sensors* **2020**, *20*, 6978. [CrossRef] [PubMed]
6. Kurek, J.; Antoniuk, I.; Górski, J.; Jegorowa, A.; Świderski, B.; Kruk, M.; Wieczorek, G.; Pach, J.; Orłowski, A.; Aleksiejuk-Gawron, J. Data augmentation techniques for transfer learning improvement in drill wear classification using convolutional neural network. *Mach. Graph. Vis.* **2019**, *28*, 3–12. [CrossRef]
7. Iskra, P.; Hernández, R.E. Toward a process monitoring and control of a CNC wood router: Development of an adaptive control system for routing white birch. *Wood Fiber Sci.* **2010**, *42* , 523–535.
8. Szwajka, K.; Trzepieciński, T. Effect of tool material on tool wear and delamination during machining of particleboard. *J. Wood Sci.* **2016**, *62*, 305–315. [CrossRef]
9. Wei, W.; Li, Y.; Xue, T.; Tao, S.; Mei, C.; Zhou, W.; Wang, J.; Wang, T. The research progress of machining mechanisms in milling wood-based materials. *BioResources* **2018**, *13*, 2139–2149. [CrossRef]
10. Jemielniak, K.; Urbański, T.; Kossakowska, J.; Bombiński, S. Tool condition monitoring based on numerous signal features. *Int. J. Adv. Manuf. Technol.* **2012**, *59*, 73–81. [CrossRef]
11. Świderski, B.; Antoniuk, I.; Kurek, J.; Bukowski, M.; Górski, J.; Jegorowa, A. Tool condition monitoring for the chipboard drilling process using automatic, signal-based tool state evaluation. *BioResources* **2022**, *17*, 5349–5371. [CrossRef]
12. Lemaster, R.L.; Lu, L.; Jackson, S. The use of process monitoring techniques on a CNC wood router. Part 1. sensor selection. *For. Prod. J.* **2000**, *50*, 31–39.
13. Lemaster, R.L.; Lu, L.; Jackson, S. The use of process monitoring techniques on a CNC wood router. Part 2. Use of a vibration accelerometer to monitor tool wear and workpiece quality. *For. Prod. J.* **2000**, *50*, 59–65.
14. Wilkowski, J.; Górski, J. Vibro-acoustic signals as a source of information about tool wear during laminated chipboard milling. *Wood Res.* **2011**, *56*, 57–66.
15. Swiderski, B.; Kurek, J.; Osowski, S.; Kruk, M.; Jegorowa, A. Diagnostic system of drill condition in laminated chipboard drilling process. In Proceedings of the MATEC Web of Conferences, Heraklion, Greece, 14–17 July 2017; EDP Sciences: Paris, France, 2017; Volume 125, pp. 1–6. [CrossRef]
16. Kuo, R. Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Eng. Appl. Artif. Intell.* **2000** , *13*, 249–261. [CrossRef]
17. Panda, S.; Singh, A.K.; Chakraborty, D.; Pal, S. Drill wear monitoring using back propagation neural network. *J. Mater. Process. Technol.* **2006**, *172*, 283–290. [CrossRef]
18. Jegorowa, A.; Górski, J.; Kurek, J.; Kruk, M. Use of nearest neighbors (K-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas Cienc. Tecnol.* **2020**, *22*, 189–196. [CrossRef]
19. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
20. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
21. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends® Signal Process.* **2014**, *7*, 197–387. [CrossRef]
22. Bengio, Y. Learning deep architectures for AI. *Found. Trends® Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
23. Kurek, J.; Swiderski, B.; Jegorowa, A.; Kruk, M.; Osowski, S. Deep learning in assessment of drill condition on the basis of images of drilled holes. In Proceedings of the Eighth International Conference on Graphic and Image Processing (ICGIP 2016), Tokyo, Japan, 29–31 October 2016; SPIE: Bellingham, WA, USA, 2017; Volume 10225, pp. 375–381. [CrossRef]
24. Kurek, J.; Wieczorek, G.; Kruk, B.S.M.; Jegorowa, A.; Osowski, S. Transfer learning in recognition of drill wear using convolutional neural network. In Proceedings of the 2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE), Kutná Hora, Czech Republic, 11–13 September 2017; IEEE: New York, NY, USA, 2017; pp. 1–4. [CrossRef]

25. Xu, H.; Zhang, C.; Hong, G.S.; Zhou, J.; Hong, J.; Woon, K.S. Gated Recurrent Units Based Neural Network For Tool Condition Monitoring. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7. [CrossRef]
26. Rifai, A.P.; Briliananda, S.; Aoyama, H. Tool Condition Monitoring with Convolutional Neural Network for Milling Tools and Turning Inserts. *J. Tek. Ind. J. Keilmuan Dan Apl. Tek. Ind.* **2023**, *25*, 1–16. [CrossRef]
27. Zhang, J.; Starly, B. Recurrent Neural Networks with Long Term Temporal Dependencies in Machine Tool Wear Diagnosis and Prognosis. *arXiv* 2019, arXiv:1907.11848.
28. Caggiano, A.; Mattera, G.; Nele, L. Smart Tool Wear Monitoring of CFRP/CFRP Stack Drilling Using Autoencoders and Memory-Based Neural Networks. *Appl. Sci.* **2023**, *13*, 3307. [CrossRef]
29. He, Y.; Liu, J.; Wu, S.; Wang, X. Condition Monitoring and Fault Detection of Wind Turbine Driveline With the Implementation of Deep Residual Long Short-Term Memory Network. *IEEE Sens. J.* **2023**, *23*, 13360–13376. [CrossRef]
30. Jeon, W.S.; Rhee, S.Y. Tool Wear Monitoring System Using Seq2Seq. *Machines* **2024**, *12*, 169. [CrossRef]
31. Choi, J.G.; Kim, D.C.; Chung, M.; Lim, S.; Park, H.W. Multimodal 1D CNN for delamination prediction in CFRP drilling process with industrial robots. *Comput. Ind. Eng.* **2024**, *190*, 110074. [CrossRef]
32. Wang, X.; Yan, J. Deep learning based multi-source heterogeneous information fusion framework for online monitoring of surface quality in milling process. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108043. [CrossRef]
33. Li, W.; Zhang, L.C.; Wu, C.H.; Wang, Y.; Cui, Z.X.; Niu, C. A data-driven approach to RUL prediction of tools. *Adv. Manuf.* **2024**, *12*, 6–18. [CrossRef]
34. Chan, A.D.; Green, G.C. Myoelectric control development toolbox. In Proceedings of the 30th Conference of the Canadian Medical & Biological Engineering Society, Toronto, ON, Canada, 28–30 May 2007; Volume 30 .
35. Ye, J.; Janardan, R.; Li, Q.; Park, H. Feature extraction via generalized uncorrelated linear discriminant analysis. In Proceedings of the Twenty-First International Conference on Machine Learning, New York, NY, USA, 4–8 July 2004; p. 113. [CrossRef]
36. Tsotsopoulou, E.; Karagiannis, X.; Papadopoulos, P.; Dyśko, A.; Yazdani-Asrami, M.; Booth, C.; Tzelepis, D. Time-domain protection of superconducting cables based on artificial intelligence classifiers. *IEEE Access* **2022**, *10*, 10124–10138. [CrossRef]
37. Lin, C.H. Frequency-domain features for ECG beat discrimination using grey relational analysis-based classifier. *Comput. Math. Appl.* **2008**, *55*, 680–690. [CrossRef]
38. Li, D.; Cai, Z.; Qin, B.; Deng, L. Signal frequency domain analysis and sensor fault diagnosis based on artificial intelligence. *Comput. Commun.* **2020**, *160*, 71–80. [CrossRef]
39. Bergmann, S.; Moussa, D.; Brand, F.; Kaup, A.; Riess, C. Frequency-Domain Analysis of Traces for the Detection of AI-based Compression. In Proceedings of the 2023 11th International Workshop on Biometrics and Forensics (IWBF), Barcelona, Spain, 19–20 April 2023; IEEE: New York, NY, USA, 2023; pp. 1–6. [CrossRef]
40. Wang, M.; Fan, P.; Yang, T. Fake face detection based on deep learning and frequency domain processing. In Proceedings of the 2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, 11–13 October 2023; IEEE: New York, NY, USA, 2023; pp. 77–82. [CrossRef]
41. Prince, A.A.; Ganesh, S.; Verma, P.K.; George, P.; Raju, D. Efficient implementation of empirical mode decomposition in FPGA Using Xilinx System Generator. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 24–27 October 2016; pp. 895–900. [CrossRef]
42. Wen, W.; Gao, R.X.; Cheng, W. Planetary Gearbox Fault Diagnosis Using Envelope Manifold Demodulation. *Shock Vib.* **2015**, *2016*, 3952325. [CrossRef]
43. Palani, P.; Sompur, V.; Thondiyath, A. Characterisation of Physiological Tremor using Multivariate Empirical Mode Decomposition and Hilbert Transform. In Proceedings of the 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Sydney, Australia, 24–27 July 2023; IEEE: New York, NY, USA, 2023; pp. 1–4. [CrossRef]
44. Hussein, H.M.; Abdalla, K.K. Seizure prediction algorithm based on simulated annealing and machine learning. *Int. J. Nonlinear Anal. Appl.* **2023**, *14*, 1499–1508. [CrossRef]