# Time Series Predictions Based on PCA and LSTM Networks: a Framework for Predicting Brownian Rotary Diffusion of Cellulose Nanofibrils *

Federica Bragone[0000−0003−4132−3175], Kateryna Morozovska[0000−0002−4065−715X], Tomas Rosén[0000−0002−2346−7063], Daniel Söderberg[0000−0003−3737−0091], and Stefano Markidis[0000−0003−0639−0639]

KTH Royal Institute of Technology, Stockholm, 100-44, Sweden
{bragone, kmor, trosen, dansod, markidis}@kth.se

**Abstract.** As the quest for more sustainable and environmentally friendly materials has increased in the last decades, cellulose nanofibrils (CNFs), abundant in nature, have proven their capabilities as building blocks to create strong and stiff filaments. Experiments have been conducted to characterize CNFs with a rheo-optical flow-stop technique to study the Brownian dynamics through the CNFs' birefringence decay after stop. This paper aims to predict the initial relaxation of birefringence using Principal Component Analysis (PCA) and Long Short-Term Memory (LSTM) networks. By reducing the dimensionality of the data frame features, we can plot the principal components (PCs) that retain most of the information and treat them as time series. We employ LSTM by training with the data before the flow stops and predicting the behavior afterward. Consequently, we reconstruct the data frames from the obtained predictions and compare them to the original data.

**Keywords:** Principal Component Analysis · Long Short-Term Memory · Time Series · Cellulose Nanofibrils

## 1 Introduction

Cellulose nanofibrils (CNFs), the fundamental building block of all plants and trees, have extraordinary mechanical properties in terms of strength and stiffness and are considered one of the major materials in terms of providing sustainable options to many advanced materials used today [16]. Through controlled alignment and assembly of dispersed CNFs, very strong and stiff filaments can be created by means of flow-focusing wet spinning, as described in [20]. Successful spinning relies on a delicate balance of timescales and, in particular, the competing effects from hydrodynamic forcing (causing alignment through shear and extensional flow) and Brownian rotary diffusion (causing de-alignment) [21]. These

* Supported by VINNOVA.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

effects, in turn, depend heavily on how the CNFs are extracted from biomass and the raw material source. Even the same raw material and extraction protocol can yield completely different CNF behavior in flowing processes. Therefore, developing methodologies for quick quality determination of CNF dispersions is crucial to assess their suitability in material processes. A potential method is the rheo-optical flow-stop experiment described by Rosén *et al.* [22]. A flow cell is used to align the dispersed CNFs by means of flow-focusing (see Fig. 1). When placed between two cross-polarized linear polarization filters, the transmitted intensity of light through the filters and flow cell will be a measurement of the CNF alignment as the system becomes birefringent. When the flow is instantly stopped, the decay of birefringence will thus be a measurement of the Brownian rotary diffusion of the CNFs. Both the behavior during flow, as well as the timescales of decay after stop, become a unique fingerprint of the CNF dispersion that can be classified and used to determine the CNF quality. In this work, we will explore the possibility of predicting the behavior after stopping the flow by training our model on the information before the flow stop and for some frames after the flow stop. This would allow us to monitor a material process *in operando* and perform classification without actually stopping the flow.

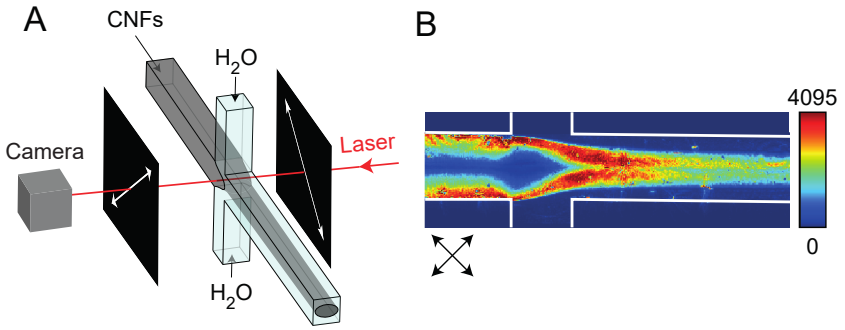



Fig. 1: Illustration of the rheo-optical flow-stop experiment as described by Rosén *et al.* [22]; (A) a core flow of dispersed CNFs (flow rate 23.4 ml/h) is focused by two sheath flows of water (each flow rate 13.5); the setup is illuminated by laser light and placed between cross-polarized linear polarization filters; the transmitted light intensity, corresponding to CNF alignment, is recorded with a camera; channel width is 1 mm; (B) a typical camera image, measuring intensities from 0 to 4095 (12 bit).

These flow-stop experiments produce large amounts of datasets. In particular, the datasets comprise data frames with specific intensities. One dataset consists of the flow of one CNF material at a particular concentration. Several materials and concentrations are tried out, resulting in large quantities of data. Dimensionality reduction techniques can then help analyze the data with smaller amounts of features. In particular, Principal Component Analysis (PCA) [30],

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

[12] is one of the major techniques to reduce data dimensions in the area of Machine Learning (ML). PCA transforms the original dataset into a new set of orthogonal features, a linear combination of the original variables defined as principal components (PCs). The transformation accounts for the maximum variance of the data stored in the first PC, and consequently, the other PCs add up cumulatively in descending order.

As these PCs create a new set of coordinates retaining most of the variance, they can be visualized as time series over the data frames of our dataset. Working with these time series, we aim to create an automatic model that predicts the behavior of the CNFs dynamics after the flow is stopped. If successful, it would allow the creation and running of a monitoring system at the production line without involving any physical stop. Consequently, we would have an artificial stop capable of predicting the Brownian dynamics of the different CNFs, given their behavior, before the flow is stopped. To achieve an automatic prediction of the lower dimensional PCs, we adopt an ML technique capable of working with sequences and time series: Long Short-Term Memory (LSTM) networks [11]. LSTM is a type of Recurrent Neural Network (RNN) [29], [19] designed to overcome the vanishing gradient problem that RNNs often encounter [1]. The class of RNNs is distinguished from classical Artificial Neural Networks (ANNs) as they present recurrent connections in their hidden layers, meaning that their looping constraints capture the sequential information stored in the data. LSTM are more complex structures that can learn complicated and long patterns, given their capability of selectively remembering or forgetting significant information. For this reason, LSTM networks are suitable for time series predictions [23].

Our work aims to implement an LSTM for each time series created by PCs: each network is trained on the time steps before the flow-stop, including a few data frames after the stop, and then predicts the following behavior. Collecting all the results, we can reconstruct the data frames by inverting the PCA transformation and finally compare the predictions with the original data. This study aims to create an automatic method that can capture the dynamic behavior of the CNFs after the flow is stopped. In return, it can help further studies on the characterization of the materials and concentrations of CNFs. Moreover, it could simulate other materials using the proposed method rather than physically setting up and running the experiment. The overall goal is to create a model that simulates an artificial flow stopping capable of predicting the Brownian dynamics of any CNF after being trained on the flow before the flow stop on different CNF samples and concentrations. In this way, the physical stopping experiments could be avoided and implemented easily in the production line without waiting a long time to produce several experiments in the labs. This paper is the first step towards the broader goal and has the following contributions:

- We propose a method to automatically predict the CNF's behavior after the flow is stopped in the rheo-optical flow-stop technique. The model will possibly simulate and characterize further combinations of materials and concentrations without requiring physical experiments.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

- Our model combines a dimensionality reduction technique, like PCA, and an ML method, like LSTM, to predict sequential data created by the PCs. The results show promising behavior predictions, especially after the flow stop.
- The reconstruction from the predicted PCs is already accurate; however, PCA did not capture certain particle-like CNF clusters completely.

The paper is structured as follows. Section 2 presents related work using similar methods and techniques for time series predictions. In Section 3, the methods used for our model are introduced. Section 4 describes the data utilized and the simulation's architecture and details. The results are shown in Section 5. Finally, Section 6 includes a discussion of the results and closing conclusions.

## 2 Related Work

The first RNN was introduced in 1989 [29], and consequently, partially connected RNNs were developed in [7] and [13]. They focus on time series by discovering and modeling their relations and information. Several works afterward extended these preliminary implementations, applying them to several problems. The main drawback of RNNs is that they suffer from the vanishing gradient problem. Therefore, RNNs cannot fully capture the non-stationary dependencies over long periods and multiple time dependencies [18]. Gating mechanisms are introduced to substitute the classical activation functions to overcome this problem. LSTM is one of the models that, with its three gates, can update a cell state by capturing the long-term dependencies [11]. In 2014, the Gated Recurrent Unit (GRU) was introduced [4], a variant of the LSTM network. This mechanism improves short-term information integration but also predicts long-term dependencies. GRU comprises a gating system with a simplified cell structure compared to the LSTM. In [5], an evaluation of gated recurrent neural networks on sequencing models is presented. In particular, a comparison between LSTM and GRU networks is given, and their predictions provide related results. A large-scale analysis on several network architectures is performed in [3], where LSTM outperformed GRU networks. Several LSTM architectures were developed, including Bidirectional LSTM [10], hierarchical and attention-based LSTM [32], [27], Convolutional LSTM [15], LSTM autoencoder [9], Grid LSTM [14], and cross-modal [26] and associative LSTM [6]. Sequence-to-Sequence (Seq2Seq) networks, introduced in [24], also work on using an input sequence to predict output sequences. More findings and comparisons of the network and architecture variants to predict nonlinear time series are presented in [18].

More recently, transformers have been considered for time series analysis and predictions, and several studies have been made to investigate their effectiveness [28], [33]. Transformers are a machine learning architecture based on multi-head self-attention mechanisms that work with sequential data [25]. They became popular in applications like natural language processing (NLP), speech recognition, and computer vision. In [17], a temporal-fusion transformer (TFT) is introduced to forecast multi-horizon and multivariate time series. TFT can

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

interpret the temporal dependencies with different time scales by merging the LSTM Sequence-to-Sequence and the transformers' self-attention mechanism.

In computational science, many problem domains produce large amounts of data, which is not always essential for analysis. For this reason, reduced order models and dimensionality reduction techniques become powerful tools for reducing the number of features while retaining most information. Several works utilize PCA to apply their models then. In [8], the authors apply PCA and LSTM for the short-term power system load forecast. In the context of time series predictions, the work in [31] presents a PCA-LSTM model for anomaly detection and prediction of time series power data. Stock prices rely on time series forecasting, and in [2], the authors first use PCA to reduce the data features and then apply RNNs to predict stock prices. In [34], it also works on stock price prediction but applies a PCA-LSTM model instead.

To the best of our knowledge, our model is the first to consider the time dependencies of the CNF flow by considering the principal components and applying LSTM networks to predict the behavior after the flow is stopped.

## 3 Methods

### 3.1 Dimensionality Reduction

Dimensionality reduction techniques are fundamental methods that reduce the number of attributes of a dataset while preserving most of the variation of the original dataset. One of these techniques is Principal Component Analysis (PCA), which transforms the data into a new set of uncorrelated variables called principal components (PCs). The first PCs contain most of the variance present in the dataset. The first step of performing PCA involves the standardization of the dataset; each variable is scaled using its corresponding mean and standard deviation values. The covariance matrix of the variables is computed, identifying the correlations between features. The following step involves computing the eigenvectors and the eigenvalues of the covariance matrix to identify the PCs. Finally, the PCs are created and arranged according to their eigenvalues, from the highest to the lowest.

### 3.2 Long Short-Term Memory

Recurrent Neural Networks (RNNs) are artificial neural networks that capture sequential data dependencies. Long Short-Term Memory (LSTM) networks [11] are a type of RNN capable of dealing with long-term dependencies. The LSTM network comprises the input gate $i$, the forget gate $f$, and the output gate $o$. The input gate gives information on the new inputs loaded in the cell state $C_t$. The forget gate highlights the information that should not be kept in the cell state. Finally, the output gate gives the final output of the LSTM block, $h_t$, at time step $t$. The equations representing the gates at time step $t$ are as follows:

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

where $\sigma$ represents the sigmoid function; $W_f$, $W_i$, and $W_o$ are the input weights for the forget, input, and output gates, respectively; $U_f$, $U_i$, and $U_o$ are the recurrent weights for the forget, input, and output gates, respectively; $h_{t-1}$ is the output of the previous LSTM block at time step $t-1$; $x_t$ is the input at the current time step $t$; and $b_f$, $b_i$, and $b_o$ are the biases of the input, forget and output gates, respectively. The following equations represent the cell state $C_t$, the candidate cell state $\tilde{C}_t$ and the final output of the LSTM block $h_t$:

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$
$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

where $W_c$ and $U_c$ represent the candidate cell's input and recurrent weights, respectively; the operator $\odot$ represents the element-wise product, and the tanh is the hyperbolic tangent activation function. The cell state $C_t$ includes both the information that needs to be forgotten from the previous cell state $C_{t-1}$ and the information that needs to be looked at in the current time step from the candidate cell state $\tilde{C}_t$. Finally, we define $h_t$ as the output of the current LSTM block, which includes the current cell state passed through the activation function to decide the block's output.

## 4 Input Data and Experiment Description

The data used in this study is generated from the cellulose spinning process and subsequent flow-stop experiments designed to capture changes in the process. We consider one experiment generated by one carboxymethylated (CM) CNF at a concentration of 3.0 g/l. The choice of the sample used for this study was purely arbitrary. The dataset contains 15000 frames of size $640 \times 100$ at 1000 fps recorded by a 12-bit camera with intensity values between 0 and 4095. The experiment shows the flow-stop experiment, meaning the flow is stopped around frames 5000-6000. For our work, we consider the data frames around the flow stop. In particular, we consider a total of 5000 frames: 3000 and 2000 frames before and after the flow-stop, respectively. We are interested in finding an automatic model that can predict what happens after the flow is stopped without actually stopping the flow. In this way, the physical experiments would not need to be stopped to characterize the materials, which is fundamental given the complexity of the experiment. Despite considering a smaller amount of data frames, the dimension of the data is still significant. For this reason, we can apply PCA, a

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

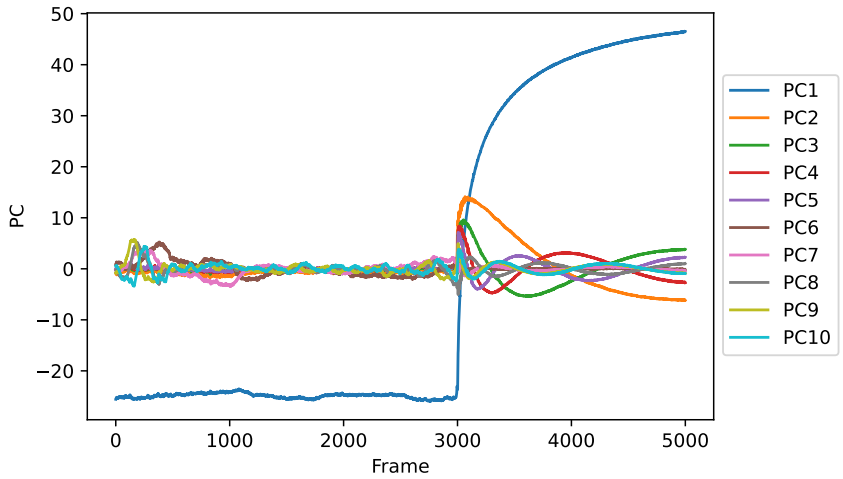



Fig. 2: Plot of the 10 PCs.

dimensionality reduction technique: by keeping most of the information available in the dataset, we reduce the amount of variables in the dataset. First, the data is normalized in the range of (0, 1), then PCA is performed. Only 10 PCs are considered for this work, which already retains 85.19%. The first PC keeps most of the information, precisely 82.24%. 10 PCs can already identify the flow well: using fewer PCs, certain particle-like CNF clusters would be missed; using more PCs, we risk introducing more noise. If we plot the resulting PCs over the frames, we can observe that they create some time series. It is also visible in the plots where the flow is stopped, as shown in Fig. 2.

Working on the lower-dimensional space created by the PCs, we could treat these components as time series, train our model on the first part, and predict when the flow is stopped.

### 4.1 LSTM Architecture

The LSTM model is created using the open-source libraries Keras and TensorFlow in Python. The model consists of 10 LSTM networks, one for each PC. For a better generalization, each network has the same hyperparameters and structure. The only difference arises when using `EarlyStopping` for the number of the argument `patience` for a few PCs.

First, we split the time series into 67% training and 33% test for each PC, giving 3350 data frames for training and 1650 data frames for testing. The network predicts the time series using a window. Considering the data between $t-w$ and $t$, for some value $w$, we would like to predict one time step forward, i.e., $t+1$. The value $w$ defines the size of the window, specifying how many previous time steps we are looking to predict the future value. In particular, it is defined as the look-back period. For our model, we chose a look-back period of three time steps, considering three previous values to predict the following time step.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

The network has an input layer with one input, two hidden layers with 20 LSTM blocks each, and an output layer with a single-value prediction. The activation function used is the rectified linear unit (ReLU). The model is compiled using the mean squared error (MSE) loss function and the Adam optimizer with a default learning rate of $10^{-3}$. The model is fitted and trained for 100 epochs, and an `EarlyStopping` is placed to monitor the test loss with a `patience` value of 50. This function stops the training when the test loss no longer improves, specifically for 50 consecutive epochs. For two cases, in particular PCs 2 and 3, the `patience` value is set up to be 5 and 10, respectively, since it was noticed that the model was starting to overfit at early epochs. Finally, the model is used to generate predictions for the training and the test datasets.

## 5 Results

This section reports the results of training an LSTM network for each PC. The corresponding predictions are then stacked and reversed to the original data format to compare the results. We train each network for 100 epochs, possibly

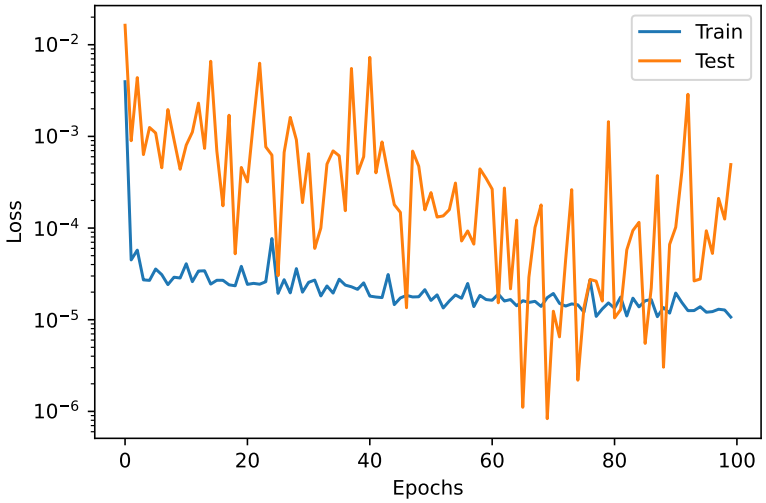



Fig. 3: Loss functions for training (blue) and test (orange) for the first PC.

stopping the training earlier when no improvement in the minimization of the test loss is observed. Fig. 3 shows the evolution of the training loss, the blue line, and the test loss, the orange line, for the first PC.

Fig. 4 shows the training and test predictions of the LSTM networks for the 10 PCs. In particular, the light blue line is the original data, the dashed darker blue line is the LSTM training predictions, and the dashed red line is the

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

LSTM test predictions. The plots show that the LSTM model can capture the component values evolving over the data frames. In particular, the test score of the first PC is of the most interest, as it captures most of the information stored in the original dataset. As for the second and third components, we can notice some discrepancies between the original data and the test predictions. It might be due to more noise in these components, hence more abrupt changes in the data. The train and test scores for the 10 PCs are presented in Table 1. The

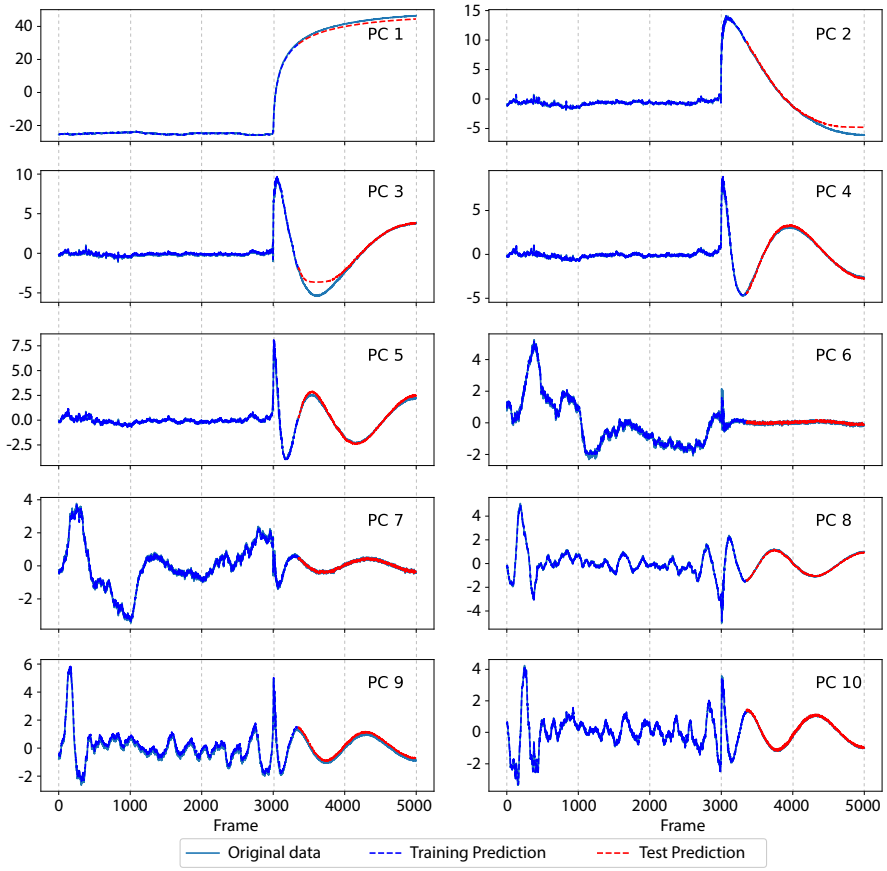



Fig. 4: LSTM predictions on the 10 PCs.

scores capture the root mean squared error (RMSE) between the original data and the predictions. All the results were tested multiple times with different random seeds to assess the robustness of the model in terms of initialization of the network. For the first PC, a standard deviation of the scale of $10^{-2}$ for training and $10^{-1}$ for test was noticed between different runs. While for the other PCs, a scale of $10^{-3}$ and $10^{-1}$ for training and test, respectively, was observed.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

Table 1: RMSE scores for each PC for training and test datasets of LSTM.

| **PC** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Train score** | 0.26 | 0.15 | 0.14 | 0.12 | 0.14 | 0.14 | 0.11 | 0.10 | 0.19 | 0.10 |
| **Test score** | 1.61 | 0.62 | 0.70 | 0.12 | 0.16 | 0.07 | 0.06 | 0.04 | 0.17 | 0.05 |

The predicted data is then reversed using the `inverse_transform` function available for the `sklearn.decomposition.PCA` from the `sklearn` library. Finally, the reconstructed predictions are compared to the original data frames. Fig. 5 shows the snapshots of the original data, the predicted data using PCA and LSTM, and the pointwise absolute error for certain data frames. The choice of the specific data frames to display was purely arbitrary. In particular, the plots in Figs. 5a and 5b display some results before the flow is stopped, precisely for data frames 1100 and 2980. They represent the predicted results of the training set used for the model. We noticed that the predictions already captured the flow well for both cases. However, some parts are poorly captured, as seen in their respective error evaluations. This could relate to using a smaller dataset created by only 10 PCs, which retains 85.19% of the total information. Therefore, it is assumed that the considered PCs might not have captured entirely some CNFs. Instead, the plots in Figs. 5c and 5d represent the flow after the stop. Snapshot 3340 refers to the training dataset; however, snapshot 5000 is part of the test data, where LSTM networks validated the model. By looking at the error plots for data frames 3340 and 5000, the results improved compared to the data frames before the flow stop. The intensity of the errors is lower, meaning that both PCA and LSTM can capture most of the CNFs flow in each data frame. Fig. 6 shows the mean absolute error (MAE) over the intensity of the images between the original data and the LSTM predictions for each frame. It can be noticed that the predictions for the test set (red line) are more accurate than the predictions for the training set (blue line).

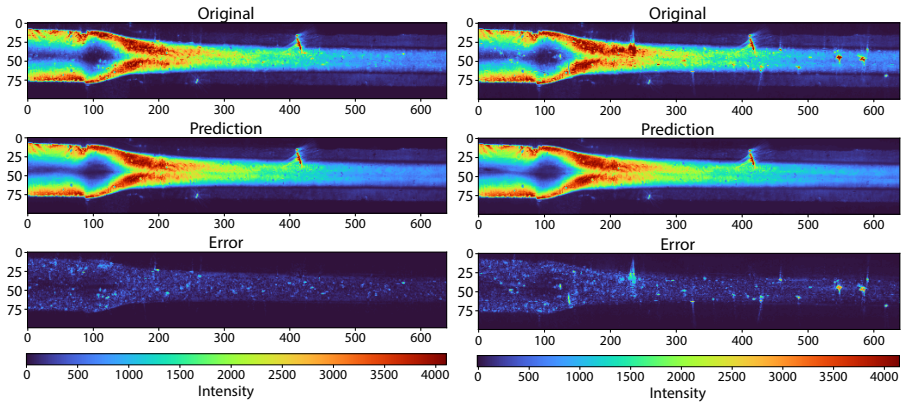
Regarding the computational time, evaluating each LSTM network requires, on average, 280 seconds per network, amounting to around 46 minutes for the whole model. The model was run using an Intel Core i7-1185G7 processor at 3.00GHz $\times$ 8.

## 6 Discussion and Conclusion

CNFs have been proven to be essential building blocks for creating more sustainable materials in the future. Their mechanical strength and stiffness are fundamental properties for achieving environmentally friendly alternatives.

In our work, we propose an ML model that involves dimensionality reduction of the data through PCA and LSTM predictions of the time series created by the principal components. First, we scale down the number of variables of our data frames, keeping 10 PCs for the analysis. The 10 PCs already explain 85.19% of the total variance. The first PC is the most critical as it already retains 82.24%

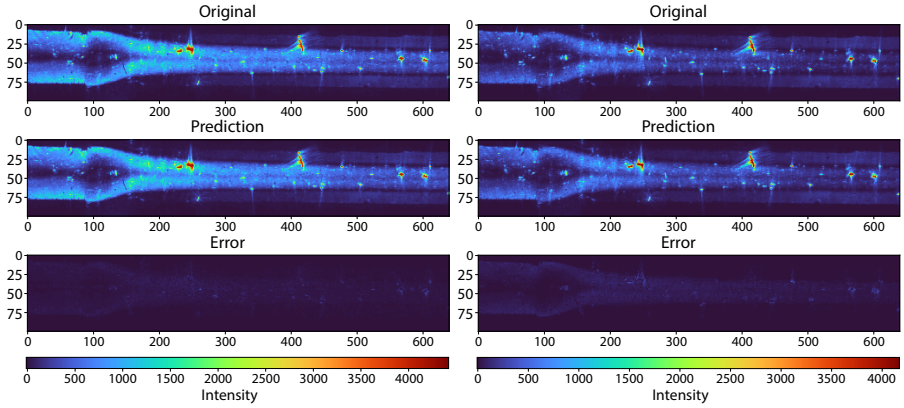
ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15




(a) Frame 1100




(b) Frame 2980




(c) Frame 3340




(d) Frame 5000

Fig. 5: Data frames 1100 (a), 2980 (b), 3340 (c) and 5000 (d). Top: Original data. Middle: LSTM predictions. Bottom: Pointwise absolute error between the original and predicted snapshots. Each frame has a size of 640x100 pixels.

of the information. Representing 85% of the variance, we can already identify the flow well. Using fewer PCs would lead to neglect of some particle-like CNF clusters, especially before the flow stops. On the other hand, using more PCs would introduce more noise. Further analysis should be considered to distinguish the noise from the particle-like CNF clusters, as we would like to retain as much information as possible about the flow. By reducing the dimensionality of the workspace, we are left with new coordinates that, if plotted over the time steps of the data frames, produce ten time series. In the flow-stop experiments, we are interested in analyzing the Brownian dynamics for the given CNF materials and concentrations that emerge after the flow is stopped. The goal is to make this analysis automatically without performing physical experiments by stopping the flow. For this reason, we propose a time series prediction of the PCs after

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

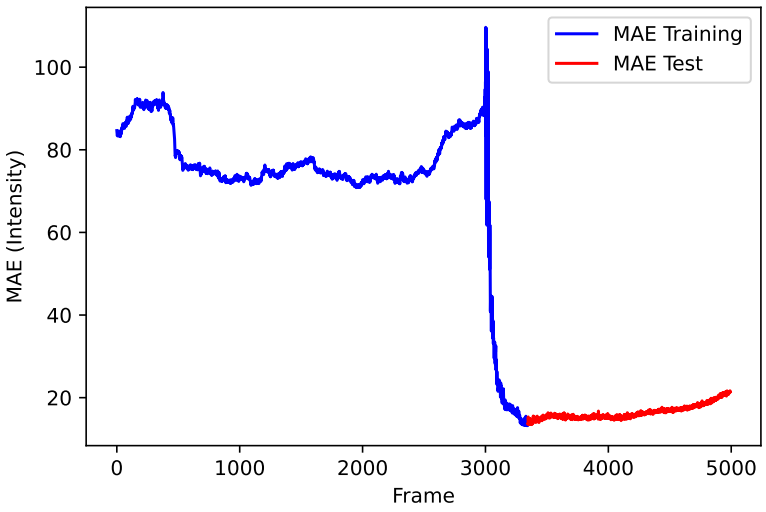



Fig. 6: MAE between the original data and the LSTM predictions for each frame.

the flow is stopped. First, by reducing the dimensionality, we reduce the large amount of data available and the computational costs. Secondly, we can exploit ML models' capabilities in predicting and forecasting sequential data and time series.

Classic time series forecast methods include Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and Seasonal Autoregressive Integrated Moving Average (SARIMA) models. They combine the autoregressive model to make predictions based on the data observed in the previous steps and the moving average model, which regulates the model by monitoring the average prediction errors of earlier predictions. While ARMA is the base model, ARIMA addresses time series trends, and SARIMA focuses on seasonality. Given that we are considering time series that do not have proper trends and seasonality and are looking at longer time frames, these models were not considered appropriate for our problem. Machine learning models are identified to be suitable for the kind of problem we are addressing. Especially given the number of data frames we are taking into account. Our choice of LSTM is because we can use a relatively more straightforward model with accurate results. Several networks and models are mentioned in Section 2. However, the basic LSTM network is a good candidate for the work without using more complex architectures that would require more parameter tuning and computational power. The trade-off between accuracy and simplicity makes us choose a model that stands in between. Without giving up on the efficiency of the predictions, we can build a model that uses few layers and neurons but can guarantee a precise result within minutes of training. A choice of transformer architecture would have possibly increased the accuracy of the predictions. However, the improve-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

ment would not have been so high as to disregard the simplicity of the LSTM that was just implemented. Moreover, when we want to achieve a broader goal using more datasets simultaneously, we believe that the computational time and power will increase, leading to the choice of a model like LSTM. However, variants of LSTM networks like Bidirectional LSTM and GRU were tried without gaining any benefit in accuracy and computational execution.

The results obtained using LSTM are already satisfactory. First, comparing the LSTM model to the PCA result for each PC shows that LSTM performs accurately both for the training and test. The most critical PC is the first, and the corresponding RMSE test score is 1.61 over the intensity. Considering that it is the component retaining most of the information, it follows that it requires more hyperparameter tuning. By looking at the accuracy of the LSTM predictions, the results are particularly accurate for the data after the flow stop. In particular, analyzing both the pointwise absolute error of the data frames and the mean absolute error shows that the accuracy in the predictions for the test dataset is higher than in the training dataset. This follows from the fact that for the data frames before the flow stops, higher intensities are present, showing the flow of CNFs. PCA has already struggled to capture a few of the particle-like CNF clusters, transferring some errors to the LSTM predictions. Therefore, the predictions for the data frames before the flow stop are less accurate. However, as the flow is stopped and less intensity is present in the images, PCA is capable of capturing most of the particle-like CNF clusters. As a consequence, the LSTM model can predict more precise results. The model takes about 45 minutes to run on a laptop. However, as we would like to implement this approach on a real-time production line, we will consider different computational approaches for future work, like running the code on GPUs and parallel computing, since the outcome of one network does not affect the others.

The work presented is the first step that allows for extending more experiments. It helps to focus and understand the flow behaviors in a reduced dimensional space created by PCA. In particular, we distinguish the PCA values before the flow stop and the PCA values after the flow stop. Using a machine learning method, we can predict the dynamics after the flow is stopped with reasonable accuracy. However, further steps will allow us to have a more robust model. The following steps will include training the model on the principal components before the stop of repeated experiments of the same material and same concentration and predicting the behavior after the flow is stopped of an experiment that the model did not train on (but maintaining the same material and same concentration). Further, we could train on the same sample but different concentrations and analyze the model's predictions on an unseen concentration of the same material. Finally, the ultimate goal would involve predicting the Brownian dynamics after the flow stop of any CNF material and concentration. Therefore, the preliminary work presented in this paper allows us to identify the type of models that can be used to achieve this goal and proves the feasibility of the predictions. If we can reach the final step, we could run a monitoring system at the production line without performing the actual stop in the experiment.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

We could, therefore, achieve an artificial stopping that will not need to stop the flow and will need extra manual work, which will benefit the industry.

## References


1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks **5**(2), 157–166 (1994)
2. Berradi, Z., Lazaar, M.: Integration of principal component analysis and recurrent neural network to forecast the stock price of casablanca stock exchange. Procedia computer science **148**, 55–61 (2019)
3. Britz, D., Goldie, A., Luong, M.T., Le, Q.: Massive exploration of neural machine translation architectures. arXiv preprint arXiv:1703.03906 (2017)
4. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
6. Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., Graves, A.: Associative long short-term memory. In: International conference on machine learning. pp. 1986–1994. PMLR (2016)
7. Elman, J.L.: Finding structure in time. Cognitive science **14**(2), 179–211 (1990)
8. Fang, Q., Zhong, Y., Xie, C., Zhang, H., Li, S.: Research on pca-lstm-based short-term load forecasting method. In: IOP Conference Series: Earth and Environmental Science. vol. 495, p. 012015. IOP Publishing (2020)
9. Gensler, A., Henze, J., Sick, B., Raabe, N.: Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC). pp. 002858–002865. IEEE (2016)
10. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional lstm networks for improved phoneme classification and recognition. In: International conference on artificial neural networks. pp. 799–804. Springer (2005)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
12. Jolliffe, I.T., Cadima, J.: Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences **374**(2065), 20150202 (2016)
13. Jordan, M.: Attractor dynamics and parallelism in a connectionist sequential machine. In: Eighth Annual Conference of the Cognitive Science Society, 1986. pp. 513–546 (1986)
14. Kalchbrenner, N., Danihelka, I., Graves, A.: Grid long short-term memory. arXiv preprint arXiv:1507.01526 (2015)
15. Kim, T.Y., Cho, S.B.: Predicting residential energy consumption using cnn-lstm neural networks. Energy **182**, 72–81 (2019)
16. Li, T., Chen, C., Brozena, A.H., Zhu, J., Xu, L., Driemeier, C., Dai, J., Rojas, O.J., Isogai, A., Wågberg, L., et al.: Developing fibrillated cellulose as a sustainable technological material. Nature **590**(7844), 47–56 (2021)
17. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021)


ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15

18. Lindemann, B., Müller, T., Vietz, H., Jazdi, N., Weyrich, M.: A survey on long short-term memory networks for time series prediction. Procedia CIRP **99**, 650–655 (2021)
19. Medsker, L.R., Jain, L.: Recurrent neural networks. Design and Applications **5**(64-67), 2 (2001)
20. Mittal, N., Ansari, F., Gowda. V, K., Brouzet, C., Chen, P., Larsson, P.T., Roth, S.V., Lundell, F., Wagberg, L., Kotov, N.A., et al.: Multiscale control of nanocellulose assembly: transferring remarkable nanoscale fibril mechanics to macroscale fibers. ACS nano **12**(7), 6378–6388 (2018)
21. Rosén, T., Hsiao, B.S., Söderberg, L.D.: Elucidating the opportunities and challenges for nanocellulose spinning. Advanced Materials **33**(28), 2001238 (2021)
22. Rosén, T., Mittal, N., Roth, S.V., Zhang, P., Lundell, F., Söderberg, L.D.: Flow fields control nanostructural organization in semiflexible networks. Soft Matter **16**(23), 5439–5449 (2020)
23. Song, X., Liu, Y., Xue, L., Wang, J., Zhang, J., Wang, J., Jiang, L., Cheng, Z.: Time-series well performance prediction based on long short-term memory (lstm) neural network model. Journal of Petroleum Science and Engineering **186**, 106682 (2020)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems **27** (2014)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
26. Veličković, P., Karazija, L., Lane, N.D., Bhattacharya, S., Liberis, E., Liò, P., Chieh, A., Bellahsen, O., Vegreville, M.: Cross-modal recurrent models for weight objective prediction from multimodal time-series data. In: Proceedings of the 12th EAI international conference on pervasive computing technologies for healthcare. pp. 178–186 (2018)
27. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: international conference on machine learning. pp. 3560–3569. PMLR (2017)
28. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in time series: A survey. arXiv preprint arXiv:2202.07125 (2022)
29. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural computation **1**(2), 270–280 (1989)
30. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems **2**(1-3), 37–52 (1987)
31. Xie, W., Zhu, Y., Cao, W., Pan, J., Wu, B., Liu, S., Ji, Y.: Pca-lstm anomaly detection and prediction method based on time series power data. In: 2022 China Automation Congress (CAC). pp. 5537–5542. IEEE (2022)
32. Xue, H., Huynh, D.Q., Reynolds, M.: Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1186–1194. IEEE (2018)
33. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
34. Zheng, X., Xiong, N.: Stock price prediction based on pca-lstm model. In: Proceedings of the 2022 5th International Conference on Mathematics and Statistics. pp. 79–83 (2022)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_15