

## Article

# Evaluating Deep Learning Networks Versus Hybrid Network for Smart Monitoring of Hydropower Plants

Fateme Hajimohammadali, Emanuele Crisostomi , Mauro Tucci  and Nunzia Fontana \* 

Department of Energy, Systems, Territory and Constructions Engineering, University of Pisa, 56122 Pisa, Italy; fatemeh.hajimohammadali@phd.unipi.it (F.H.); emanuele.crisostomi@unipi.it (E.C.); mauro.tucci@unipi.it (M.T.)  
\* Correspondence: nunzia.fontana@unipi.it

**Abstract:** One of the main goals of the International Energy Agency (IEA) is to manage and utilize clean energy to achieve net zero emissions by 2050. Hydropower plants can significantly contribute to this goal as they are vital components of the global energy infrastructure, providing a clean, safe, and sustainable power source. Accordingly, there is great interest in developing methods to prevent errors and anomalies and ensure full operational availability. With modern hydropower plants equipped with sensors that capture extensive data, machine learning algorithms utilizing these data to detect and predict anomalies have gained research attention. This paper demonstrates that deep learning algorithms are particularly powerful in predicting time series. Three well-known deep learning networks are examined and compared to previous approaches, followed by the introduction of a new, innovative hybrid network. Using real-world data from two hydropower plants, the hybrid model outperforms individual deep learning models by achieving more accurate fault detection, reducing false positives, offering early fault prediction, and identifying faults several weeks before occurrence. These results showcase the hybrid network's potential to enhance maintenance planning, reduce downtime, and improve operational efficiency in energy systems.

**Keywords:** hydropower plants; process control; predictive maintenance; fault detection; hybrid deep learning models; time series forecasting



**Citation:** Hajimohammadali, F.; Crisostomi, E.; Tucci, M.; Fontana, N. Evaluating Deep Learning Networks Versus Hybrid Network for Smart Monitoring of HydroPower Plants. *Energies* **2024**, *17*, 5670. <https://doi.org/10.3390/en17225670>

Academic Editor: Jesús Polo

Received: 1 October 2024

Revised: 1 November 2024

Accepted: 8 November 2024

Published: 13 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Motivation

Many factors motivate the desire for enhanced monitoring and abnormal detection in hydropower facilities. First and foremost, hydropower plants are essential parts of the world's energy infrastructure because they provide a reliable source of electricity with a long lifespan and cost-effective maintenance [1]. In addition, hydropower plants are typically known for their relatively low operation and maintenance costs, which usually amount to approximately 2.5% of the overall costs of the plant [2]. Despite such advantages, these facilities constantly struggle with issues of operational reliability and productivity due to anomalies and malfunctions. To improve the efficiency and performance of hydropower plants, it is thus crucial to employ algorithms that can predict possible breakdowns and notify users before serious damage and shutdowns occur [3].

The integration of sensors for system monitoring, operational control, predictive maintenance, and security has become increasingly crucial in the last ten years [4,5]. Artificial intelligence (AI) has significantly impacted the evolution of smart hydropower plants, offering benefits such as accurate energy generation modeling, predictive maintenance, and fault detection; it is also crucial for reliability modeling [6,7]. Motivated by the several methodologies that have been designed to analyze the operation of hydropower plants, this manuscript provides a comparison of some of the most recent algorithms on a specific case study involving real data measured from two hydropower plants, which had already

been investigated in [2]. It is noted in this manuscript that some of the most recent methodologies can improve the results of [2] in terms of reducing the number of false positives and enhancing the ability to predict the occurrence of faults in advance.

### 1.2. State of the Art

Many anomaly detection techniques have been researched, deployed, and applied in a variety of domains.

Roughly speaking, fault detection and predictive maintenance have been classified into two main groups: Model-based methods and data-driven methods. Various techniques fall into one of these two main categories.

- **Model-Based Methods:**

These algorithms rely on creating a mathematical model of the system being monitored. The model represents the nominal behavior of the system, and deviations from this behavior may indicate the presence of faults. Model-based methods include techniques such as observer-based methods, state estimation methods, and parameter estimation.

- **Data-Driven Methods:**

These algorithms analyze the data collected from sensors or other monitoring devices and try to identify a nominal behavior of the system. They use statistical analysis, machine learning, and pattern recognition techniques to detect anomalies or deviations from the nominal behavior. Data-driven methods are often preferred when the underlying system is complex or poorly understood as they can adapt to various operating conditions, embed any kind of non-linearity behavior, and detect faults without prior knowledge of the system's dynamics. They can apply to inherently high-dimensional datasets collected from a multitude of sensors over a specified time frame. Therefore, they encompass diverse parameters with varying scales and feature extraction is extremely important when working with these systems [8]. Historically, feature extraction in data science literature has relied on statistical methods or machine learning techniques like Principal Component Analysis (PCA). With the advent of deep learning algorithms, feature extraction has transformed, enabling the learning of feature representations directly from the data through neural networks. This helps improve the efficiency of network anomaly detection [9].

Data-driven methods can be further classified into three main subgroups:

1. **Statistical Analysis:**

Statistical process control (SPC), control charts, and hypothesis testing are examples of statistical methods used for fault detection. The main basis of these methods is monitoring the statistical properties of the data and identifying deviations from dominant behavior, but it should be noticed that traditional methods like vibration analysis have limitations in terms of cost and complexity.

2. **Machine Learning and Deep Learning Techniques:**

Various machine learning algorithms, such as support vector machines (SVM), neural networks (NN), decision trees, random forests, k-nearest neighbors (k-NN), and ensemble methods like AdaBoost and gradient boosting, are employed for fault detection by learning patterns and anomalies from historical data. These networks are great at finding and exploring hidden features in the attribute representation space [10]. Among traditional methods, the support vector machine (SVM) technique for condition monitoring and fault diagnostics (CMFD) has been noteworthy [11]. The authors in [12] have been pioneers in the intelligent implementation of energy production networks, studying neural networks for unsupervised learning. A self-organizing map was applied in Italian hydropower plants, leading to the proposal of a new key performance indicator (KPI). Also benefitting from sensor data, an expert system was established for online temperature monitoring systems in [13] and focuses on the predictive maintenance of hydropower plants using a multi-agent system (MAS) and artificial neural network (ANN). The result shows that this network is successful in monitoring, identifying, and diagnosing dynamic performance online.

The authors in [14] detect early faults by analyzing temperature rise. Two types of statistical methods and machine learning are used for data processing and the results show that the feed-forward neural network (FFNN) method works better than Hotelling's multivariate control chart in detecting fault samples.

Deep learning techniques have shown superior results compared to classical machine learning methods, especially when the data volume increases. AI algorithms such as autoencoders (AEs), convolutional neural networks (CNN), and Long Short-Term Memory (LSTM) have demonstrated remarkable ability in discerning complex patterns and relationships within extensive datasets. Anomaly detection systems based on deep learning algorithms are increasingly popular and widely applied in both academic and industrial environments. The nature of the collected data plays an important role in the selection of the neural network. Autoencoder networks are a semisupervised learning model proven effective for fault detection [9,15]. The next relevant project employed a long short-term memory (LSTM) neural network for anomaly detection of variables like bearing temperatures and vibration in a 56 MW pumped storage hydroelectric power station in Norway. The LSTM network was utilized to predict the temperature one hour in advance [16]. The paper [17] explores the use of IoT technology to enhance predictive maintenance within the framework of Industry 4.0. By integrating IoT sensors with machine learning, this study demonstrates how real-time data analysis can predict potential equipment failures, thereby reducing downtime and associated costs across various industries. The study utilizes multiple case examples, including applications in aviation, manufacturing, and energy sectors, to illustrate the advantages of IoT-driven predictive maintenance. These examples showcase how predictive maintenance optimizes performance, minimizes unnecessary repairs, and extends equipment lifespan, thus offering competitive advantages in efficiency and cost-effectiveness for industries. These innovations reduce the workload associated with traditional alarm handling and enhance predictive maintenance capabilities by analyzing key trends in equipment parameters. Ref. [18] investigates the use of anomaly detection and explainability algorithms for improving predictive maintenance in hydroelectric power plants (HPPs). The study compares various machine learning models, with the autoencoder model emerging as the most effective for identifying anomalies within operational data. Additionally, the authors employ SHapley Additive exPlanations (SHAP) to provide insights into root causes, enabling experts to pinpoint which features contribute to anomalies. This research demonstrates the potential of machine learning in enhancing operational reliability and reducing costs, making HPPs more attractive to investors and advancing the transition to renewable energy sources. Recently, [19] presents an advanced monitoring system for hydropower stations, focusing on intelligent fault detection and alarm management. This study proposes an intelligent SCADA (Supervisory Control and Data Acquisition) system that integrates real-time monitoring, predictive alarms, and fault diagnosis to streamline data processing and improve decision-making for maintenance. By implementing an equipment alarm model and utilizing machine learning techniques for anomaly detection, the system can rapidly locate and diagnose faults.

### 3. Pattern Recognition Methods:

These methods use pattern recognition techniques such as clustering, Principal Component Analysis, Independent Component Analysis (ICA), and Self-Organizing Maps (SOM) to identify abnormal patterns or clusters in the data indicative of faults [20,21]. Most of the research of recent years uses hybrid networks that use two or more methods in combination. The authors in [22] proposed two approaches for predictive maintenance in the Peña Blanca hydroelectric power plant. The first approach used logistic regression for classifying various types of failures, while the second approach combined recurrent LSTM networks with an autoencoder. Ref. [23] presents a hybrid model that combines the strengths of ARIMA and Bi-LSTM models, increasing the accuracy and robustness of the forecast. The LSTM model was optimal for detecting

high temperatures in generator bearings. A condition monitoring method based on LSTM algorithms has been introduced in [24]. This method establishes correlations between prior known information and current environmental data. In some contexts, the location or type of fault is crucial for subsequent actions. So fault location or diagnosis is a pivotal consideration in this case. Ref. [25] proposes an unsupervised anomaly detection method based on variational modal decomposition (VMD) and a deep autoencoder. The autoencoder based on a convolutional neural network is used to complete unsupervised learning, and the reconstruction residual is used for anomaly detection. Refs. [15,26] merge Principal Component Analysis (PCA) with clustering-based autoencoders (CAE). This strategy enhances CAE's capacity to detect latent representations of normal data.

Despite all the advantages outlined for data-driven methods, the implementation of these systems usually faces many challenges, including the large volume of data and the emergence of database security concerns [5,27]. The next point is the asymmetry in the dataset. Detecting specific anomalous behaviors is challenging due to the rarity, heterogeneity, and low frequency of defective data compared to nominal data [28]. One widely deployed solution is to use deep neural networks trained using only healthy data in a one-class training manner [29].

### 1.3. Contribution and Organization

The main contributions of this work may be summarized as follows:

1. Several methodologies, as outlined in the previous section, have been used for the application of smart monitoring and fault detection, also in the field of hydropower plants. It is usually hard to compare different methodologies on the same problem since each one of them may be more convenient under a different perspective: e.g., the ability to handle massive amounts of data; the ability to distinguish faulty operations more clearly; the ability to recognize incipient faults several days before the actual occurrence is noticed on the field; or the ability to avoid false positives, which may cause unnecessary maintenance actions of parts of hydropower plants. By adopting real data collected from two hydropower plants, this study provides a thorough comparison between classic machine learning algorithms and more recent techniques, most notably those in the class of deep learning algorithms.
2. A methodology is introduced, specifically a hybrid deep learning model that, to the best of the authors' knowledge, has not been used for this particular application. In particular, hybrid deep learning networks combine different architectures with the ultimate goal of leveraging the combined strengths of CNNs, LSTMs, and autoencoders. Indeed, this methodology appears to be slightly superior to the single ones in our case study.

Our paper is organized as follows: the next section describes the particular case study of interest, providing details of the measured quantities that are of interest for the analysis of hydropower plants. Section 3 describes the methodologies adopted in the comparison, while Section 4 is dedicated to explaining how the hybrid network was tailored for the specific application of interest. Section 5 compares the outcomes of the algorithms and extensively discusses the benefits and disadvantages of individual methodologies. Finally, the manuscript concludes by summarizing the findings and outlining potential future research directions on this topic.

## 2. Case Study

Two hydropower plants, referred to as Plant A and Plant B, with respective production capacities of 215 MW and 1000 MW, are utilized in the dataset. Both plants are located in Italy and use Francis turbines. The key difference between these two plants is that Plant A is a storage plant, while Plant B is a pumped storage plant. Additional explanations for a more comprehensive understanding are provided below.

### 2.1. Hydropower Plant

Plant A is in northern Italy and comprises four generation units powered by vertical-axis Francis turbines. Each unit has a 60 MVA capacity. The machinery room is located 500 m inside a mountain. The plant was commissioned in 1951 and is part of a significant hydraulic system. Plant B, as mentioned, is a pumped storage power plant, and it is equipped with two reservoirs. The upper reservoir is located in the province of Isernia -in Italy—at an elevation of 643 m and was created by an earthen dam. Water is released from the upper reservoir to the power plant, generating electricity as it flows from the upper dam to the lower dam. Four reversible Francis turbine pump generators with a 250 MW capacity make up Plant B. This power plant has been operational since 1994. Due to its high pumped storage capacity—approximately 60 GW in recent years—this power plant is strategically significant.

### 2.2. Dataset

For structures like these hydropower plants with numerous Degrees Of Freedom (DOFs), a large number of sensors are needed to accurately estimate the structural properties for health monitoring [30]. To gather this dataset, various vibrations, temperatures, and pressure sensors have been installed in different components, including the water intake, penstocks, turbines, generators, and HV transformers. The final dataset comprises 630 analog signals for Plant A and 60 analog signals for Plant B. The signals were collected from different parts of hydropower plants. Due to the dependence of physical parameters on each other, for a comprehensive evaluation, the key components of a power plant should be monitored, which are shown in Table 1 of these parameters. Datasets were collected from sensors located in these sectors over a period from 1 September 2017 to 17 April 2019. To train the network, 30 percent of the dataset was used as training data, while the remaining samples were saved for testing. After preprocessing, which included the removal of frozen and outliers, 855,241 samples remained. To reduce the computational load, a windowing algorithm was used to determine the final number of samples. For example, using a window size of 5, the training dataset was reduced to 37,251 samples. The training time for each network varied depending on the complexity of the model, and the combined network took approximately 18 min.

**Table 1.** List of main components analyzed for the hydro plants.

Component Name	Measured Signals
Generation Units	Vibrations
HV Transformer	Temperatures, Gasses levels
Turbine	Pressures, Flows, Temperatures
Oleo-dynamic system	Pressures, Temperatures
Supports	Temperatures
Alternator	Temperatures

In a prior study, Ref. [2], the dataset was employed to evaluate the efficacy of a proposed self-organizing map (SOM) technique and to introduce a key performance indicator (KPI) with respect to conventional statistic process control methodologies. The outlined approach has then been successfully implemented and is currently operational. During data acquisition, some data may be missed, frozen, or mixed with noise, all of which can result in significant errors during the machine learning training stage [31]. Therefore, a pre-processing stage is necessary, involving activities such as labeling data normalization, outlier removal, and missing data treatment [32].

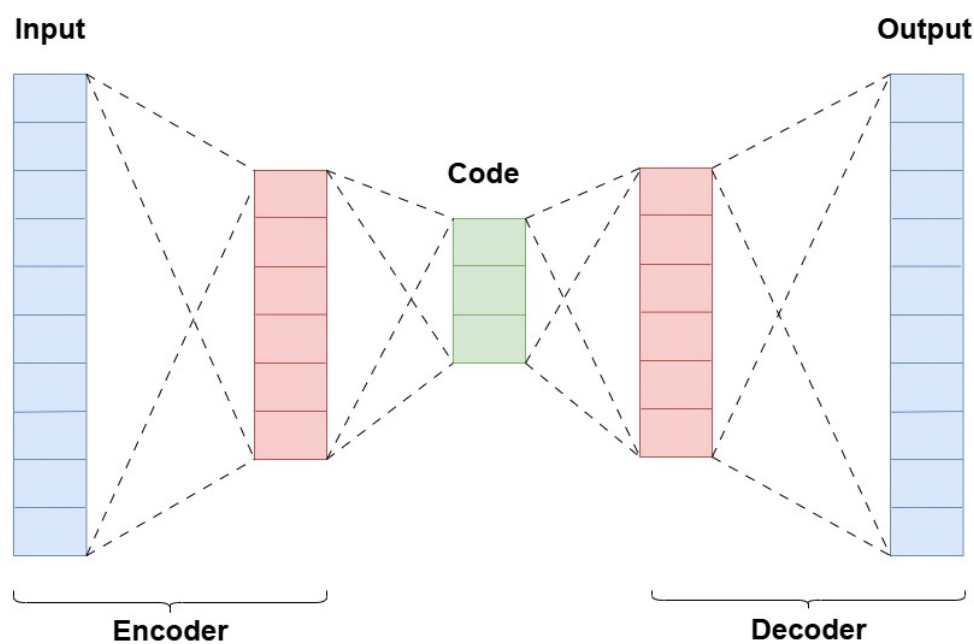
## 3. Methodology

In this research, three distinct approaches, AE, 1D CNN, and LSTM, are investigated, both separately and in an integrated model. The extracted features have clear physical or statistical significance. As the most critical step in fault diagnosis, feature extraction serves

as the foundation for detecting faults and identifying fault types, which directly impacts the accuracy of diagnostic results. Since this step is performed manually in traditional machine learning, these methods tend to be computationally expensive, limiting their application in real-time fault diagnosis systems [33]. First, the basic structure of the considered methods was introduced, followed by an explanation of how they can be used as tools in the field of anomaly detection.

### 3.1. Autoencoder

The first approach proposed uses an autoencoder, which is a semi-supervised learning model based on fully connected feed-forward networks. AEs are commonly used for dimensionality reduction or feature extraction. Their main purpose is to compress representations and preserve critical information for reconstructing input data. A basic autoencoder uses the same loss function output unit as traditional feedforward networks. The main difference between the autoencoder method and other neural networks is related to the structure of the network, which consists of two symmetrical parts: encoder and decoder [34]. The encoder receives the input data and transmits it to the hidden layer, where feature extraction occurs, effectively reducing the dimensionality of the dataset. Then, the decoder reconstructs the original input from the hidden layer. As Figure 1 depicts, autoencoders determine the key features of the data based on reconstruction loss error [35].



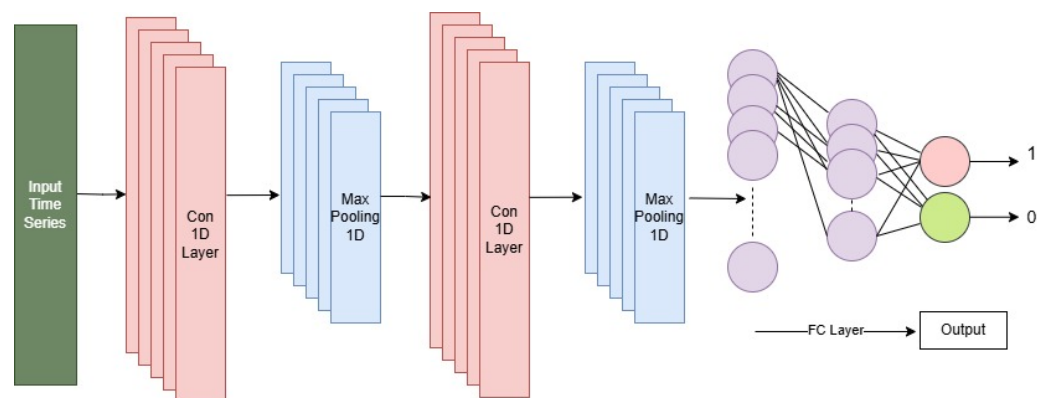
**Figure 1.** An overview of the structure of the autoencoder network structure.

To address the challenge of fault detection as a classification problem, one of the difficulties is dealing with an unbalanced dataset. In a classification problem, the number of samples from different classes must be the same so that the network can train on all classes effectively. However, in fault detection, anomalies occur much less frequently than healthy data. Although there are various ways to address this problem with the aim of balancing different classes, one very practical approach is to train a network with normal data. Once the network converges, test data, including both healthy and abnormal classes, is applied. If the network is functioning correctly, it should show higher reconstruction errors for faulty data compared to healthy data.

### 3.2. 1D CNN

One-dimensional convolutional neural networks (1D CNNs) have emerged as a powerful tool for fault detection in time series forecasting due to their ability to automatically

extract relevant features from sequential data [36]. Convolutional neural networks (CNN) are a well-known type of deep network designed for two-dimensional tasks, especially image processing. One-dimensional CNNs are otherwise designed to process sequential data by applying convolution operations in one dimension, and this makes them especially effective in capturing local patterns and temporal dependencies in short time windows. The main nature of convolutional networks is the convolutional layer, which is responsible for detecting local patterns in the input sequence. This process involves sliding a series of filters or kernels over the input, performing elemental multiplications, and summing the results to produce feature maps. The model is trained to understand the non-linear behavior of the dataset by utilizing a non-linear activation function, such as Relu (Rectified Linear Unit). Pooling layers reduce the dimensionality of feature maps and provide invariance to small shifts in the input. For instance, the pooling layer selects the maximum value within a pooling window, effectively summarizing the presence of a feature. Following multiple convolutional and pooling layers, the feature maps are flattened and passed through fully connected layers. These layers combine the extracted features to make predictions. For fault detection tasks, the output layer typically uses a sigmoid activation function for binary classification or a softmax activation function for multi-class classification (Figure 2).



**Figure 2.** An overview of the structure of 1D convolutional neural network.

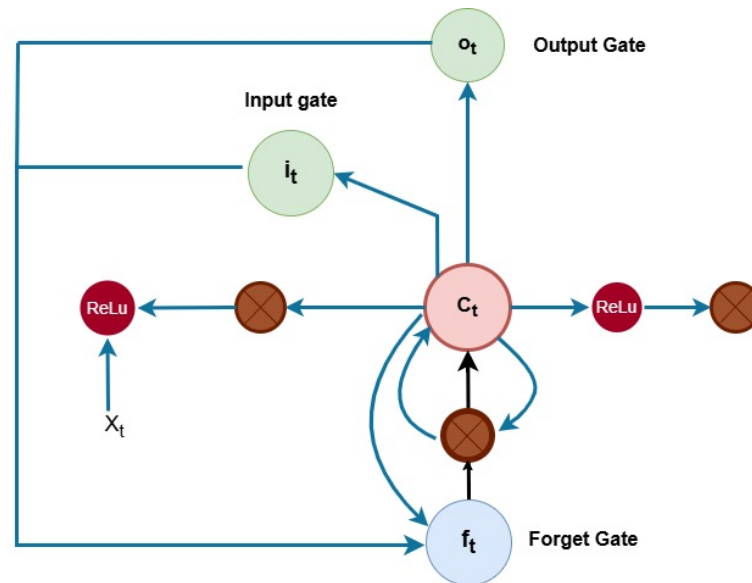
### 3.3. Multi-Input LSTM

Recurrent Neural Networks (RNN) are another popular method that can detect long-term time dependencies and find them in the dataset. These networks avoid the need for a predetermined time window and can accurately model complex multivariate sequences. This feature makes them suitable for finding global temporal patterns in the whole dataset. The LSTM type is a very interesting kind of recurrent network because of its particular structure, which has solved the issue of vanishing and exploding gradients in RNNs. Moreover, LSTMs commonly use a Rectified Linear Unit (Relu) instead of the classic Tanh function. Using Relu activation functions has several advantages; for instance, it encourages network sparsity, which in turn lessens the interdependence of parameters, mitigating overfitting issues [37]. Figure 3 shows the main processing unit of the LSTM cell.

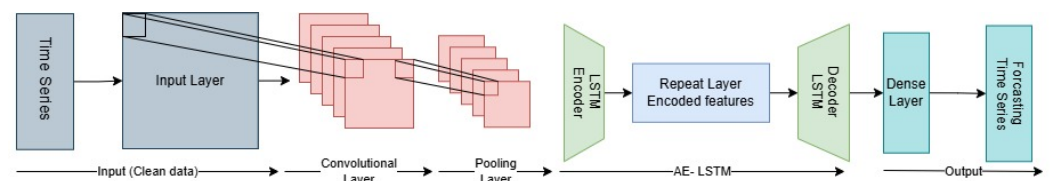
### 3.4. Hybrid Network: 1D CNN LSTM AE

It is clear from the concepts provided in the previous section that each network has specific advantages in data processing due to its distinctive architecture. Hybrid networks can be developed by integrating various networks to combine the benefits of each, gaining the advantages of all at once. The hybrid approach investigated in this research combines the strengths of CNN, autoencoder, and LSTM. It is a powerful and flexible architecture that leverages the strengths of each component to detect spatial and temporal dependencies. Convolutional filters extract local patterns from windows applied to the dataset. The LSTM encoder–decoder is configured to read the input sequences, compress data using an encoder, and decode data, which retain the original structure, using a decoder. With

such a recurrent network, finding global behavior patterns in the dataset would be possible. Figure 4 illustrates the proposed CNN-LSTM-AE architecture.



**Figure 3.** An overview of the structure of an LSTM cell.



**Figure 4.** An overview on the structure of Hybrid 1D CNN LSTM AE model.

## 4. Proposed Network Structure

### 4.1. Tuning Hyperparameters and Optimized Structure

Hyperparameters should be set before training a deep learning network. Their optimal combination is crucial for effective model performance. This process, known as hyperparameter tuning, involves using algorithms to search for the best values of epochs, batch size, learning rate, and more, thereby improving a neural network's accuracy and efficiency. One critical hyperparameter is the number of hidden layers. Increasing the number of hidden layers allows the model to identify more complex patterns within the data, although it may lead to overfitting and require more computational power. Similarly, the number of neurons per hidden layer should be adjusted based on the complexity of the data. Increasing the number of neurons generally helps to better fit the data until no further performance improvement is observed. However, overfitting is a risk here as well. Another significant hyperparameter is the learning rate, which dictates the step size at which the model weights are updated during training. A lower learning rate can ensure a more accurate model by allowing a slower convergence, whereas a higher learning rate might accelerate convergence at the cost of accuracy. The batch size also plays a pivotal role in the training process. A larger batch size might improve generalization by averaging over more samples, thereby reducing the variance in model weights. Choosing the right number of neurons in the hidden layer is crucial to balance model performance and complexity. To help with this, Ref. [38] offers an equation that estimates the minimum and maximum number of neurons. This equation takes into account the number of inputs, outputs, and data samples, providing boundaries that help avoid issues like underfitting or overfitting. In this study, In the structure of each network, the number of neurons has been checked using this equation to prevent excessive complexity and pre-fitting. However, too large a batch size can hinder convergence, slow down the learning process, and lead to



memory issues due to the increased need to store gradients and intermediate activations. It is essential to balance these factors to optimize learning and hardware efficiency. As described in detail in the following, in our experiments, we tune some of the parameters using a grid search approach based on k-fold cross-validation, while other parameters are selected by comparing different settings with a validation set. The k-fold cross-validation method further strengthened the tuning process by partitioning the dataset into k subsets, iteratively training the model on each subset while validating the remaining data. This approach provided a more comprehensive evaluation of model performance across various data splits, ensuring that the selected hyperparameters generalized well across unseen data. The value of  $k = 5$  is chosen for this research and Table 2 presents the tuning and optimization results for the AE network, while similar optimization was conducted for other grid models, with final results documented for efficient reporting.

**Table 2.** Hyperparameter tuning results for AE network.

Validation Loss	latentNeuron	Learning Rate	Batch Size	Best
0.0304704045	6	$5.819310969 \times 10^{-5}$	16	
0.0062245624	27	$3.922913775 \times 10^{-5}$	32	
0.0027203022	5	$1.509640191 \times 10^{-5}$	8	✓
0.0039545825	24	$6.789552156 \times 10^{-5}$	8	
0.0306758715	12	$3.963104132 \times 10^{-5}$	16	
0.0042250373	46	$6.215608685 \times 10^{-4}$	8	
0.0070399883	19	$1.353344494 \times 10^{-3}$	16	
0.0042112886	40	$1.196641788 \times 10^{-3}$	8	
0.0183390009	7	$2.424698715 \times 10^{-3}$	32	
0.0072042440	49	$2.951888870 \times 10^{-3}$	16	

✓ denotes the best performing tuple of hyperparameters.

#### 4.2. Structure of Implemented AE Network

The dataset is initially pre-processed into a time series format, and a critical step in the pre-processing involves handling outliers and sampling the data using a windowing algorithm to ensure the quality and consistency of the input signals. This preprocessing reduces noise and stabilizes the data, which is essential for effective AE network training. The first network examined is a deep autoencoder network. Handling outliers and employing a windowing algorithm (moving averages) for analyzing the data are crucial preprocessing steps to ensure the accuracy and consistency of the input signals. An iterative process is employed to arrive at the optimal network structure, in which different AE settings are tested. In particular, the number of layers and neurons is adjusted, and after each change, the network is trained and evaluated. Finally, the optimal structure is selected based on the error diagram and the resulting efficiency. This systematic approach ensured that the final model configuration strikes a balance between complexity and performance. Training the AE involves using the Adam optimizer with a mean squared error (MSE) loss function for over 50 epochs and with a batch size of 8. The training data are split, with a portion reserved for validation (10 percent of training share) to monitor the network's performance and prevent overfitting. The autoencoder network used in this study consists of several layers. The architecture includes an input layer, followed by three encoding layers with progressively decreasing numbers of neurons. The first encoding layer reduces the dimensionality of neurons with ReLu activation, followed by a second layer with half the neurons and a third with one-fourth of the neurons. The decoding part mirrors the encoding layers, symmetrically increasing the number of neurons back to the original latent size. The final output layer reconstructs the input using a sigmoid activation function. The model is trained for 40 epochs and minimizes the mean squared error (MSE) between the input and the reconstruction. This structure enables the network to learn compressed representations of the data and detect anomalies based on the reconstruction error.

#### 4.3. Structure of Implemented 1D CNN Network

The next network is a one-dimensional convolutional structure. Also, in this case, the optimal structure for this network is obtained by validating a number of different settings of the hyper-parameters. The model consists of a Conv1D layer with 64 filters and a kernel size of 2, followed by a Max Pooling1D layer, a Flatten layer, and two Dense layers, the latter with 50 neurons. This structure, trained over 50 epochs with a batch size of 100, effectively captures and processes temporal patterns in the data. Preprocessing includes outlier handling and data scaling using MinMaxScaler. Then, the sequence is split into training and test sets for supervised learning. Performance evaluation on test data shows strong predictive capabilities of the model as evidenced by metrics such as MSE, MAE, and RMSE. In the following, the effectiveness of the network will be discussed by displaying the obtained results.

#### 4.4. Structure of Implemented LSTM Network

For LSTM-based networks, the optimal configuration includes not just the right number of layers and neurons but also an effective dropout rate to prevent overfitting by randomly deactivating neurons during training. This improves generalization across different data presentations. Moreover, a fixed learning rate of 0.001 is used to balance the speed and accuracy of convergence. The model has a sequential structure with LSTM layers that are skilled at capturing temporal dependencies in the data. Preprocessing consists of processing and scaling using MinMaxScaler, followed by sequence segmentation for supervised learning. The LSTM network, trained over 50 epochs with a batch size of 100, shows strong predictive capabilities. Evaluation criteria such as MSE, MAE, and RMSE highlight the performance of the model. Visualization of predicted versus actual signals and error distributions confirms the effectiveness of the network and emphasizes its application in predictive maintenance and fault diagnosis in industrial settings.

#### 4.5. Structure of Implemented Hybrid Network

According to the literature, most deep learning systems consist of a single architecture. When two or more deep learning architectures are combined, the result is called a multi-modal hybrid deep learning model. In this study, a hybrid deep neural network is developed that includes a convolutional autoencoder LSTM network.

### 5. Results

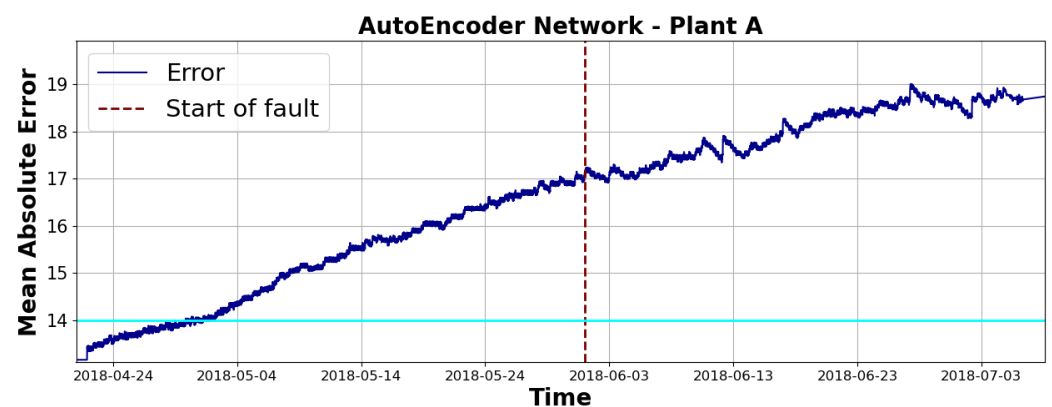
All model training and evaluations were conducted on a personal device equipped with an 11th Gen Intel Core i7-1195G7 processor with a base frequency of 2.90 GHz, capable of reaching up to 2.92 GHz. The system has 16 GB of installed RAM, with 15.7 GB usable, running on a 64-bit operating system. No dedicated GPU was available, meaning all processing was carried out on the CPU. While this configuration is suitable for basic model training and evaluation tasks, it imposes some limitations on training speed and efficiency for larger models. Training times varied across different configurations, with the longest total training duration being approximately 60 min and the shortest being 5 min, depending on the hyperparameter configuration and model complexity. Although training was performed on a personal device, approximate energy costs could be calculated if required for large-scale deployments.

After the pre-processing stage, the data are divided into train and test parts. For this preliminary analysis, the same strategy outlined in [2] is followed, as this reference will also serve as a term of comparison. The threshold for determining possible faults is positioned at three times the average value of the MAE, as computed in the training set. In the previous work based on SOM-based KPI [2], two faults were specifically investigated, and the same ones for comparison purposes are considered. Table 3 lists these two faults detected and reported by the operator. The first fault is related to Plant A. This fault began on June 1st. The proposed method in the previous paper successfully detected the fault on time. In this study, the performance of each of the four networks is evaluated individually.

Figures 5–8, presents the corresponding results. To make detection easier, a specific time window from 22 April 2018 to the end of 10 July 2018 has been considered. As mentioned earlier, since the networks have been trained with healthy data, the reconstruction error of the network should increase when the fault occurs. Thus, a sequence of points placed above the threshold is a classic symptom of the occurrence of a fault. Figure 5 shows the outcome of the autoencoder network.

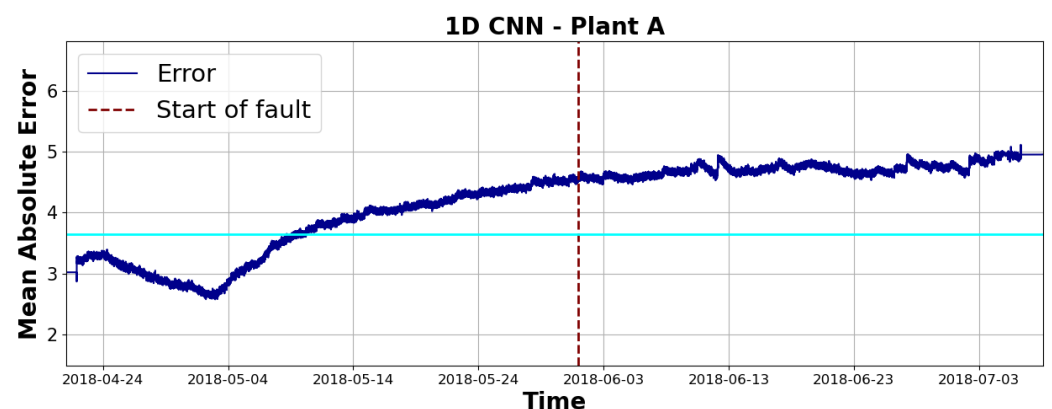
**Table 3.** List of anomalous behaviors on the two hydropower plants.

Case Study	Plant ID	Warning Name	Fault Date
1	A	HV transformer gasses	1 June 2018
2	B	Generator Temperature	1 October 2018



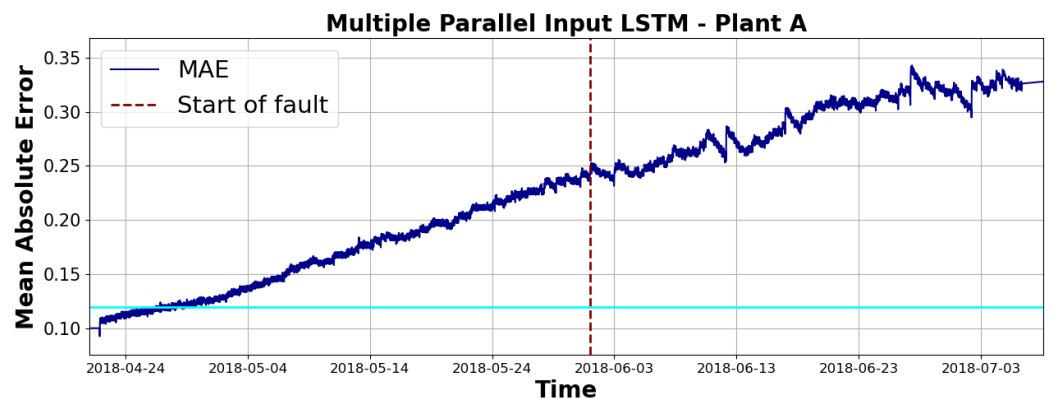
**Figure 5.** MAE related to AE structure related to Plant A. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

The moment of fault occurrence is marked with a vertical red dashed line. As can be seen, from mid-April, the error graph shows an increasing trend, and as the fault approaches, this increase becomes more rapid. In Figure 6, the result related to the convolutional network is examined. Also, in this case, the faulty behavior can be well noticed, already a few days before the operator recognizes the fault.



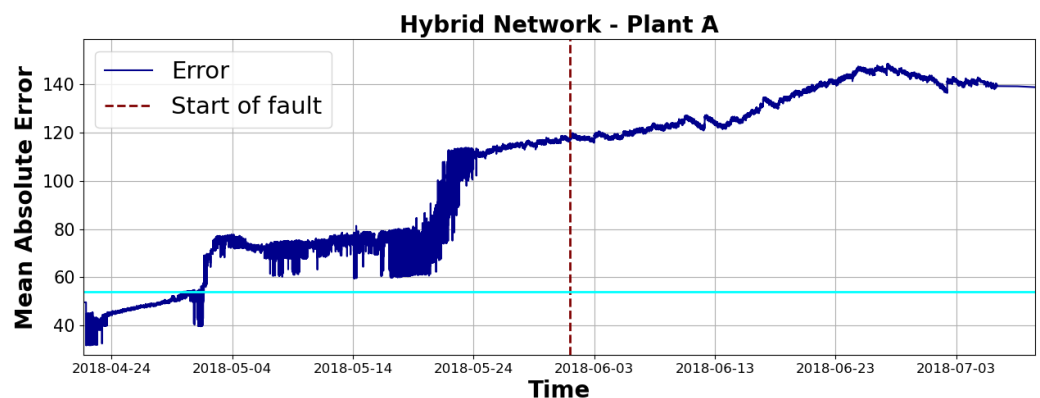
**Figure 6.** MAE related to CNN structure related to Plant A. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

Figure 7 analyzes the performance of LSTM networks. These networks, known for their ability to capture temporal dependencies, successfully identify the time dependencies in this dataset.



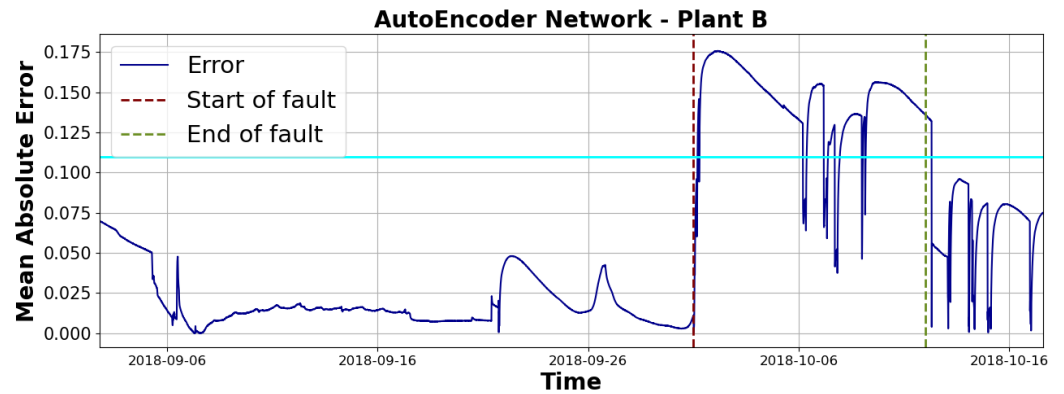
**Figure 7.** MAE related to LSTM structure related to Plant A. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

The network's reconstruction error starts increasing with a mild slope already from the end of April, and the curve of the error stays clearly above the safety threshold for the period of the fault. Accordingly, this type of network appears particularly convenient as not only does it recognize the fault when it occurs, but it also has a predictive power of a few weeks, which can be exploited to try to prevent the occurrence of the fault and possibly the closure of the plant for maintenance purposes. The last network reviewed is the hybrid network; see Figure 8.



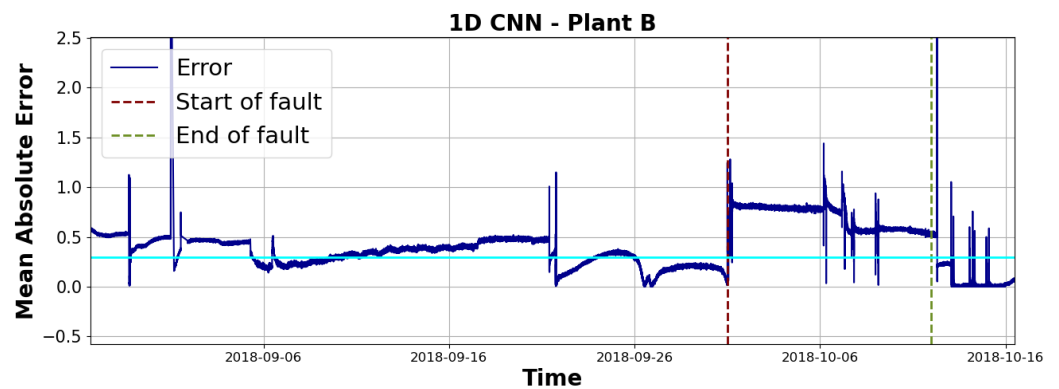
**Figure 8.** MAE related to Hybrid structure related to Plant A. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

Its performance is similar to that of LSTM networks, and the recognition of the fault occurs also around the end of April. Accordingly, autoencoders, CNN, LSTM, and Hybrid networks are all able to recognize the fault, but the last two are more convenient because of their predictive power, which allows them to recognize the incipient fault a few weeks ahead of the operator. From this perspective, their performance is even better than that of the two algorithms that had been proposed in [2], which did not exhibit the same predictive power. The second case study concerns Plant B, during the fault occurrence registered between October 1st and 12th. The beginning and the end of the fault are indicated by two vertical dashed lines. Both methods described in [2] managed to identify the fault, but the  $t^2$ -based algorithm suffered from some false positive alarms. Conversely, the SOM-based strategy was shown to be more robust. The first methodology described in this paper, based on an autoencoder reconstruction error, is shown in the first graph of Figure 9.



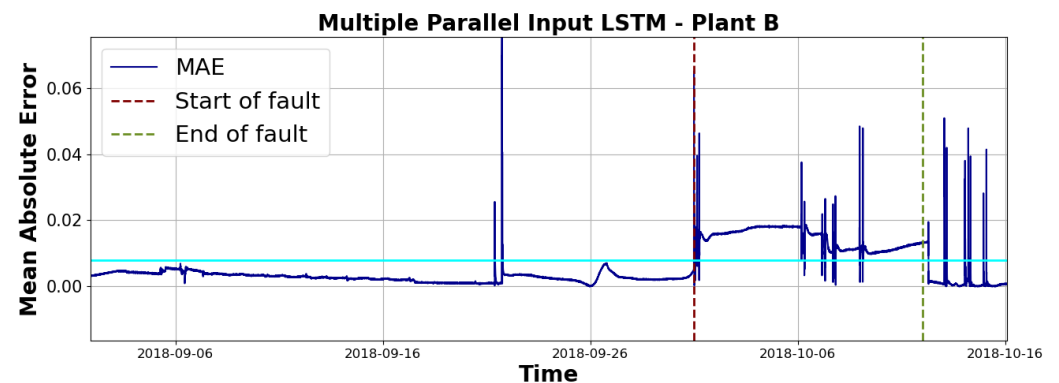
**Figure 9.** MAE related to AE structure related to Plant B. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

While the fault is correctly spotted, some false alarms are triggered in this case as well. In the second result about plant B, Figure 10 similar behavior is observed by the convolutional network as well as some false positive samples are gathered at the beginning of September as well.

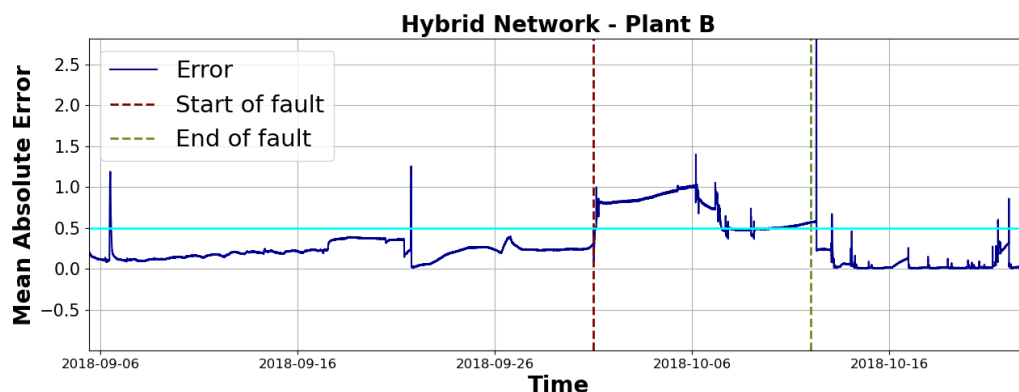


**Figure 10.** MAE related to CNN structure related to Plant B. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

Conversely, better results are obtained if LSTM and hybrid networks are used instead. In particular, in both cases, the fault is signaled to the operator, and no false positives are observed, apart from isolated and single spikes in the error curves (Figures 11 and 12).



**Figure 11.** MAE related to LSTM structure related to Plant B. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.



**Figure 12.** MAE related to Hybrid structure related to Plant B. The cyan line denotes the threshold defined as three times the average value of MAE in the training phase: error values above this line indicate that a fault has been detected.

Accordingly, it is recommended that alarms from LSTM and hybrid networks should be triggered if the signal exceeds the threshold for some consecutive time steps.

Overall, as a final discussion, it should be noticed that LSTM and hybrid networks provide better results with respect to CNN and autoencoders because, while all methodologies correctly identify the occurrence of faults, LSTM and hybrid networks reduce the number of false positives and, at least in the first case study, they are able to predict the occurrence of the fault a few days ahead.

## 6. Conclusions

This study addresses one of the most critical challenges in the hydropower industry: analyzing the behavior of measured time signals and utilizing intelligent algorithms to detect, and possibly predict, the presence of faults, eventually preventing major failures from occurring. With the advent of the fourth industrial revolution and the integration of real-time signals and data, control concepts have shifted from the classic and analog domain to the digital and modern realm. In this context, selecting an appropriate artificial intelligence methodology with a thorough understanding of dataset behaviors is essential for achieving optimal performance based on specific needs.

Specifically, in this paper, we have analyzed three well-known and supposedly efficient types of deep learning networks by describing in detail the structure of each network and discussing their strengths and weaknesses. In addition, a hybrid network that combines the previous three simpler networks has been introduced. Thanks to its hybrid structure, this network can leverage the advantages of all the other three types, with promising performance improvements. The performance of all four strategies has been compared over two real case studies with actual measured data. While all four methodologies correctly signal the occurrence of the faults, LSTM and the hybrid network provide the most promising results as they decrease the number of false positives and, thus, of false alarms; they also exhibit a better predictive potential as, in some cases, they recognize incipient faults several weeks in advance before they are recognized by the plant operators. Such machine learning algorithms also outperform other methodologies that had been tested on the same data in previous scientific papers. A key limitation of this study lies in the sensitivity of deep learning models to hyperparameters, which, if not optimally tuned, can impact performance. Additionally, the hybrid network, while effective, requires significant computational resources, making real-time deployment challenging. Another limitation is the reliance on a specific dataset from two hydropower plants, which may limit the applicability of the results to other energy systems or operational environments. Future work can focus on developing automated hyperparameter tuning techniques, such as Bayesian optimization, for enhancing model performance without manual intervention. Efforts can also be directed at exploring lightweight model architectures or model pruning to reduce computational demands for real-time applications. Expanding the dataset to include data

from diverse plants or other renewable energy systems will improve the model's robustness and adaptability, making it more applicable to a broader range of scenarios.

**Author Contributions:** Conceptualization, F.H., E.C., M.T. and N.F.; methodology, F.H., E.C., M.T. and N.F.; software, F.H.; validation, F.H., E.C., M.T. and N.F.; formal analysis, F.H., E.C., M.T. and N.F.; investigation, F.H., E.C., M.T. and N.F.; resources, F.H., E.C., M.T. and N.F.; data curation, F.H., E.C., M.T. and N.F.; writing—original draft preparation, F.H., E.C., M.T. and N.F.; writing—review and editing, F.H., E.C., M.T. and N.F.; visualization, F.H., E.C., M.T. and N.F.; supervision, E.C., M.T. and N.F.; project administration, E.C., M.T. and N.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data is unavailable due to privacy restrictions.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Gøtske, E.K.; Victoria, M. Future operation of hydropower in Europe under high renewable penetration and climate change. *iScience* **2021**, *24*, 102999. [[CrossRef](#)] [[PubMed](#)]
- Betti, A.; Crisostomi, E.; Paolinelli, G.; Piazzzi, A.; Ruffini, F.; Tucci, M. Condition monitoring and predictive maintenance methodologies for hydropower plants equipment. *Renew. Energy* **2021**, *171*, 246–253. [[CrossRef](#)]
- Spencer, B., Jr.; Nagarajaiah, S. State of the art of structural control. *J. Struct. Eng.* **2003**, *129*, 845–856. [[CrossRef](#)]
- Li, S.; Pozzi, M. What makes long-term monitoring convenient? A parametric analysis of the value of information in infrastructure maintenance. *Struct. Control Health Monit.* **2019**, *26*, e2329. [[CrossRef](#)]
- Nagarajaiah, S.; Yang, Y. Modeling and harnessing sparse and low-rank data structure: A new paradigm for structural dynamics, identification, damage detection and health monitoring. *Struct. Control Health Monit.* **2017**, *24*, e1851. [[CrossRef](#)]
- Jana, D.; Patil, J.; Herkal, S.; Nagarajaiah, S.; Duenas-Osorio, L. CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. *Mech. Syst. Signal Process.* **2022**, *169*, 108723. [[CrossRef](#)]
- Velasquez, V.; Flores, W. Machine Learning Approach for Predictive Maintenance in Hydroelectric Power Plants. In Proceedings of the 2022 IEEE Biennial Congress of Argentina (ARGENCON), San Juan, Argentina, 7–9 September 2022; pp. 1–6. [[CrossRef](#)]
- Amini, N.; Zhu, Q. Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network. *Neurocomputing* **2022**, *488*, 618–633. [[CrossRef](#)]
- Chalapathy, R.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407.
- Mehrotra, K.; Mohan, C.; Huang, H. *Anomaly Detection*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 21–32.
- Xayyasith, S.; Promwungkwa, A.; Ngamsanroj, K. Application of Machine Learning for Predictive Maintenance Cooling System in Nam Ngum-1 Hydropower Plant. In Proceedings of the 2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, 21–23 November 2018; pp. 1–5.
- Selak, L.; Butala, P.; Sluga, A. Condition monitoring and fault diagnostics for hydropower plants. *Comput. Ind.* **2014**, *65*, 924–936. [[CrossRef](#)]
- Jiang, W. Research on Predictive Maintenance for Hydropower Plant Based on MAS and NN. In Proceedings of the 2008 Third International Conference on Pervasive Computing and Applications, Alexandria, Egypt, 6–8 October 2008; pp. 604–609.
- Jain, S.; Barmada, E.; Crisostomi, E.; Romano, F.; Tavano, F.; Tucci, M. Indirect monitoring and early detection of faults in trains' motors. *IET Electr. Syst. Transp.* **2018**, *8*, 86–94. [[CrossRef](#)]
- Nguyen, V.; Nguyen, V.; Hoang, T.; Shone, N. A Novel Deep Clustering Variational Auto-Encoder for Anomaly-based Network Intrusion Detection. In Proceedings of the 14th International Conference on Knowledge and Systems Engineering (KSE), Nha Trang, Vietnam, 19–21 October 2022; pp. 1–7.
- Buaphan, I.; Premrudeepreechacharn, S. Development of an expert system for fault diagnosis of an 8-MW bulb turbine downstream irrigation hydropower plant. In Proceedings of the 6th International Youth Conference on Energy (IYCE), Budapest, Hungary, 21–24 June 2017; pp. 1–6.
- Sharma, A.; Aslekar, A. IoT-Based Predictive Maintenance in Industry 4.0. In Proceedings of the 2022 IEEE International Interdisciplinary Humanitarian Conference for Sustainability (IIHC-2022), Bengaluru, India, 18–19 November 2022; pp. 143–145. [[CrossRef](#)]
- Fanan, M.; Baron, C.; Carli, R.; Divernois, M.A.; Marongiu, J.C.; Susto, G.A. Anomaly Detection for Hydroelectric Power Plants: A Machine Learning-based Approach. In Proceedings of the 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany, 18–20 July 2023. [[CrossRef](#)]
- Li, Z.; Wang, Y.; Wang, B.; Zhang, H. Research and Design of Intelligent Hydropower SCADA. In Proceedings of the 3rd International Conference on Energy and Power Engineering, Control Engineering (EPECE), Chengdu, China, 23–25 February 2024; pp. 1–5. [[CrossRef](#)]

20. Garbea, R.; Grigoras, G. Clustering-Using Data Mining-based Application to Identify the Hourly Loading Patterns of the Generation Units from the Hydropower Plants. In Proceedings of the 2022 International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 20–22 October 2022; pp. 426–431.
21. Calvo-Bascones, P.; Sanz-Bobi, M.; Welte, T. Anomaly detection method based on the deep knowledge behind behavior patterns in industrial components. Application to a hydropower plant. *Comput. Ind.* **2021**, *125*, 103376. [[CrossRef](#)]
22. Li, M.; Francis, E.; Hinkle, S.; Ajjarapu, A.; Zhang, C. Preconception and prenatal nutrition and neurodevelopmental disorders: A systematic review and meta-analysis. *Nutrients* **2019**, *11*, 1628. [[CrossRef](#)] [[PubMed](#)]
23. Malhan, P.; Mittal, M. A novel ensemble model for long-term forecasting of wind and hydropower generation. *Energy Convers. Manag.* **2022**, *251*, 114983. [[CrossRef](#)]
24. Qian, P.; Tian, X.; Kanfoud, J.; Lee, J.; Gan, T. A novel condition monitoring method of wind turbines based on long short-term memory neural network. *Energies* **2019**, *12*, 3411. [[CrossRef](#)]
25. Wang, H.; Liu, X.; Ma, L.; Zhang, Y. Anomaly detection for hydropower turbine unit based on variational modal decomposition and deep autoencoder. *Energy Rep.* **2021**, *7*, 938–946. [[CrossRef](#)]
26. Qian, J.; Song, Z.; Yao, Y.; Zhu, Z.; Zhang, X. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemom. Intell. Lab. Syst.* **2022**, *231*, 104711. [[CrossRef](#)]
27. Gârbea, R.; Scarlatache, F.; Grigoras, G.; Neagu, B.C. Extracting the Operating Characteristics of Hydropower Plants Using a Clustering-based Efficient Methodology. In Proceedings of the 9th International Conference on Modern Power Systems (MPS), Cluj-Napoca, Romania, 16–17 June 2021; pp. 1–4.
28. Pang, G.; Cao, L.; Aggarwal, C. Deep learning for anomaly detection: Challenges, methods, and opportunities. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual, 8–12 March 2021; pp. 1127–1130.
29. Nguyen, V.; Nguyen, V.; Le-Khac, N.; Cao, V. Clustering-based deep autoencoders for network anomaly detection. In Proceedings of the Future Data and Security Engineering: 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, 25–27 November 2020; Proceedings 7; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 290–303.
30. Nagarajaiah, S.; Erazo, K. Structural monitoring and identification of civil infrastructure in the United States. *Struct. Monit. Maint.* **2016**, *3*, 51. [[CrossRef](#)]
31. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*; Elsevier: Amsterdam, The Netherlands, 2006.
32. Harish, A.; Prince, A.; Jayan, M.V. Fault detection and classification for wide area backup protection of power transmission lines using weighted extreme learning machine. *IEEE Access* **2022**, *10*, 82407–82417. [[CrossRef](#)]
33. Yan, W.; Wang, J.; Lu, S.; Zhou, M.; Peng, X. A Review of Real-Time Fault Diagnosis Methods for Industrial Smart Manufacturing. *Processes* **2023**, *11*, 369. [[CrossRef](#)]
34. Hajimohammadali, F.; Fontana, N.; Tucci, M.; Crisostomi, E. Autoencoder-based Fault Diagnosis for Hydropower Plants. In Proceedings of the 2023 IEEE Belgrade PowerTech, Belgrade, Serbia, 25–29 June 2023; pp. 1–6.
35. Zhu, X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [[CrossRef](#)]
36. Adari, S.K.; Alla, S. *Beginning Anomaly Detection Using Python-Based Deep Learning: Implement Anomaly Detection Applications with Keras and PyTorch*; Apress: New York, NY, USA, 2024; pp. 393–398.
37. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 855–868. [[CrossRef](#)]
38. Koleva, R.; Babunski, D.; Zaev, E.; Tuneski, A.; Trajkovski, L. New Approach in Hydropower Plant Control Based on Neural Networks. *Energ. Ekon. Ekol.* **2022**, *XXIV*, 39–46. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.