Detection and Classification of Short Circuits in Power System Transmission Lines using RNN

1st Džana Čankušić *Department of Automation and Electronics*
Faculty of Electrical Engineering
Sarajevo, Bosnia and Herzegovina
dcankusic1@etf.unsa.ba
2nd Harun Špago *Department of Automation and Electronics*
Faculty of Electrical Engineering Sarajevo
Sarajevo, Bosnia and Herzegovina
hspago1@etf.unsa.ba

***Abstract*—This paper addresses the problem of detecting and classifying short circuits in power system transmission lines using deep learning methods. A neural network based on GRU (*Gated Recurrent Unit*) and LSTM (*Long Short-Term Memory*) architectures was implemented to improve the accuracy and speed of fault detection. The input data consisted of voltages and currents in a three-phase system, while the network outputs represented seven different types of faults.**

The model was trained using ten epochs and a GPU, with different window sizes applied.

The implemented model enables *real-time* fault detection, contributing to the reliability and security of power systems. These results demonstrate that neural networks can significantly enhance the efficiency and speed of fault identification compared to traditional protection methods.

***Index Terms*—short circuit, power system, neural networks, GRU, LSTM, sliding window, fault classification, *real-time* detection**

I Introduction

In the era of Industry 4.0, the demand for electrical energy is constantly increasing, leading to a rise in the number of production units connected through a complex power system (PS). This system consists of three main components: generation, transmission, and distribution. The stable and reliable operation of the PS is crucial to minimizing impacts on industry, transportation, and households. However, system faults can cause major supply interruptions and significant financial losses, making fast fault detection and isolation essential.

Short circuits are one of the most common causes of power outages, occurring due to weather conditions, equipment failures, human factors, or contact with animals. These faults destabilize the system and lead to large energy losses. Traditionally, system protection is carried out using relays and circuit breakers, but their response time can be slow. Therefore, machine learning and deep learning methods have recently been applied for faster and more accurate fault detection and classification.

Machine learning has emerged as a powerful and nowadays popular tool for automatically extracting complex patterns from data, offering the potential to significantly enhance the accuracy and efficiency of fault detection in transmission lines. In recent years, there has been a notable surge in research focused on leveraging machine learning algorithms to address challenges in this domain [1]. Various approaches, such as artificial neural networks (ANNs), support vector machines (SVMs), decision trees, and ensemble methods, have demonstrated success in analyzing electrical signals, identifying fault patterns, and even predicting potential failures before they occur [6] [7]. Additionally, advancements in deep learning, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have further expanded the capabilities of fault detection systems by enabling the processing of high-dimensional data, such as time-series signals and images from drones or sensors [8]. These technologies, combined with the growing availability of real-time data from smart grids and IoT-enabled devices, are paving the way for more robust, adaptive, and proactive fault detection systems in modern power networks. San Kim in [2] proposed an interesting method based on smart system inspection (SIS) that are mounted on an UAV. The fault detection is based on image processing. In order to record the discharges on the lines they are using ultraviolet cameras. Nguyen in [3] gives an extensive report of various ML models used for fault classification and localization. They found the best result for fault classification are being generated when using XGBoost, with an accuracy of 99.82%. In [4] a extreme learning machine (ELM) algorithm [10] is being used as an classiffier. Architecturally, it consists of a single hidden layer where the hidden layer weight metric is generated at random. The output weight metric is developed by implementing the matrix pseudoinverse process. Since the models is simple ELM has gained popularity. Apart from ML method used for detecting faults also traditional ways are popular and accurate. In [5] a method to identify permanent single-pole grounding faults in VSC-HVDC overhead lines using fault line voltages is proposed. It analyzes the hybrid DCCB's operation and capacitor discharge, compares transient voltage behaviors.

Various neural network algorithms, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and artificial neural networks (ANNs), have demonstrated high accuracy in fault recognition. Other methods, such as support vector machines (SVMs) and decision trees (DTs), are also used. Although effective, many of these methods are computationally demanding and complex to implement. This paper explores short circuit detection and classification in PS with the aim of improving detection speed and accuracy, the goal is to produce a real-time predictor that is able to receive needed inputs for classification in this case the system variables that are needed are three phase voltages and three line currents. The analysis is going to be based around recurrent neural networks (RNNs). We decided to analyise the behaviour of LST (Long Short-Term Memory) networks and GRU (Gated Reccurent Unit) networks. Also this paper introduces a slidig window. Simple said that the input of the neural network will not be a vector only containing the current voltage and current values, it is also going to need $WINDOW\_SIZE - 1$ past inputs i order to get a better and more precise prediction. This window will slide as new samples come in so that in each moment we have the current sample and $WINDOW\_SIZE - 1$ past samples. At the end of the paper we will present the value of the detection delay. This is the time that passes from the moment the fault occurs to the moment the neural network signal the fault. The goal is to have this time as much smaller as possible, but this time is also very dependent on the sampling rate of the system itself.

A. Dataset

For training the neural network, an appropriate dataset needs to be generated. The input to the neural network will consist of phase voltages and line currents, namely voltages V_a, V_b and V_c and currents I_a, I_b and I_c. The data will be generated based on the mathematical model of the transmission line where the fault occurred [9]. Figure 1 shows the transmission line model under consideration.

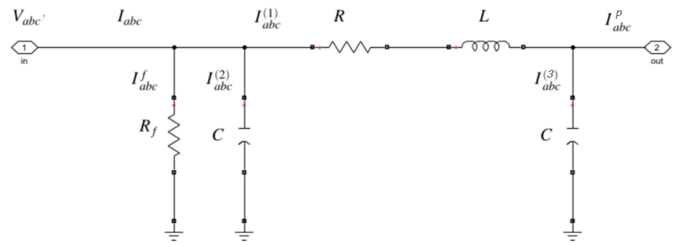

Figure 1: Transmission line model

Unknown parameters in the figure are:

- V_{abc} - phase voltages, one of the network inputs,
- I_{abc} - line currents, one of the network inputs,
- I^{f}_{abc} - fault current,
- I^{p}_{abc} - load current,
- R - line resistance,
- L - line inductance,
- C - shunt capacitance,
- R_f - fault resistance.

Using *Kirchhoff's* laws, a state-space representation of the system can be obtained, which will be used for data generation.

Applying *Kirchhoff's* current law, the following equations are obtained:

$$I_{abc} = I^{1}_{abc} + I^{2}_{abc} + I^{f}_{abc} \tag{1}$$

$$I^{1}_{abc} = I^{3}_{abc} + I^{p}_{abc} \tag{2}$$

Equations derived using *Kirchhoff's* voltage law are:

$$I^{2}_{abc} = C \cdot \frac{dV_{abc}}{dt} \tag{3}$$

$$I^{f}_{abc} = \frac{V_{abc}}{R_f} \tag{4}$$

$$V_{abc} = R \cdot I^{1}_{abc} + L \cdot \frac{dI^{1}_{abc}}{dt} + \frac{1}{C} \cdot \int_{\infty}^{t} I^{3}_{abc}\, dt \tag{5}$$

Substituting equation (2) into (5) yields:

$$V_{abc} = R \cdot I^{1}_{abc} + L \cdot \frac{dI^{1}_{abc}}{dt} + \frac{1}{C} \cdot \int_{\infty}^{t} (I^{1}_{abc} - I^{p}_{abc})\, dt \tag{6}$$

Differentiating the previous equation, we obtain:

$$\frac{dV_{abc}}{dt} = R \cdot \frac{dI^{1}_{abc}}{dt} + L \cdot \frac{d^2 I^{1}_{abc}}{dt^2} + \frac{1}{C} \cdot I^{1}_{abc} - \frac{1}{C} \cdot I^{p}_{abc} \tag{7}$$

II Used Algorithm

For the purpose of detecting and classifying short circuits in the power system, a recurrent neural network was used, which will be discussed in more detail below. The data is organized into two CSV files:

- inputs.csv - contains 6 input features (three phase voltages and three line currents),
- outputs.csv - contains 1 output value representing the type of detected short circuit.

The goal is to train the model so that it can analyze data in real-time and classify fault types. If the neural network is to be suitable for *real-time* application, instead of processing all data at once, a sliding window approach is used. This means that at any given moment, the model uses the current input sample and the previous N samples, allowing for the analysis of the temporal dynamics of the signal. The size of N can be adjusted to optimize model accuracy.

A. Signal Preprocessing

Before the data is used for training the neural network, proper preprocessing must be performed to ensure model efficiency and improve classification accuracy. The main goal of preprocessing is to adapt the data to be suitable for the neural network, reduce noise effects, and enable faster model convergence.

One of the key steps in preprocessing is normalizing the input data. Voltage and current values are divided by the nominal voltage and current values of the system, i.e., the transmission line, to ensure a uniform data scale. This achieves:

- preventing the dominance of certain variables due to their large absolute values,
- improving the stability of neural network training,
- enabling faster and more efficient model convergence,
- reducing the network's sensitivity to extreme values.

This normalization is especially important because neural networks perform better when input data is within a specific numerical range. Additionally, in real systems, different sensors may have different measurement ranges, so it is crucial to bring all values to the same scale.

B. Data Processing

For the neural network to learn effectively and make accurate decisions, the data must go through several processing steps. These steps ensure that the network receives high-quality and consistent input data, which directly impacts model accuracy.

The first step in data processing is loading it from CSV files containing information about voltages, currents, and fault types. The input data comes from the file *inputs.csv* and consists of six numerical features (three phase voltages and three line currents), while the file *outputs.csv* contains integer labels representing the type of short circuit.

The output of the neural network represents a multi-class classification problem, where the model must recognize and classify a given signal into one of eight classes. Each class corresponds to a specific system state:

- **class 0** – normal system state (no fault),
- **class 1** – short circuit of phase L1 to ground,
- **class 2** – short circuit of phase L2 to ground,
- **class 3** – short circuit between phases L1 and L2,
- **class 4** – short circuit of phase L3 to ground,
- **class 5** – short circuit between phases L1 and L3,
- **class 6** – short circuit between phases L2 and L3,
- **class 7** – three-phase short circuit (L1, L2, and L3).

To avoid errors during model training, it is necessary to check the data for missing values. If there are missing values, they can be removed or interpolated to ensure data continuity. The most commonly used methods are deleting rows with missing values or filling them using the mean, median, or linear interpolation.

Since the model is designed for *real-time* data processing, the sliding window method is used. Instead of analyzing each sample separately, the neural network uses a set of $N+1$ consecutive samples to recognize temporal patterns in the data. Thus, the input to the model has the shape:

$$(N+1, 6) \tag{8}$$

where N represents the number of previous samples used for prediction. This method allows the network to utilize the temporal dependence between data points, thereby increasing the accuracy of fault classification.

To ensure good model generalization, the data is divided into training and test sets. It is essential that the data remains temporally consistent to prevent information leakage between sets. A typical data split is:

- training set – 80% of the data, used for model training,
- test set – 20% of the data, used for model evaluation.

During data splitting, a sequential sampling principle is used, where test data is taken from the last portion of the dataset to ensure the model is tested on data it has not seen during training.

C. Neural Network Architecture

Due to the temporal nature of the data, several approaches were considered when designing the neural network architecture:

- recurrent neural network (RNN) – using LSTM or GRU layers to analyze sequential data, allowing the network to recognize temporal dependencies between input samples, which is crucial for detecting faults with a dynamic nature
- convolutional neural network (CNN 1D) – uses convolutional filters to analyze local temporal patterns in the data, which can help identify sudden changes in voltages and currents that signal faults
- *feed-forward* neural network (DNN) – the simplest model that uses fully connected layers, does not capture temporal relationships between consecutive measurements, making it less suitable for this problem.

In this work, GRU (*Gated Recurrent Unit*) and LSTM (*Long Short-Term Memory*) networks, which are types of recurrent neural networks, were used for the analysis of time series and sequential data. LSTM networks have additional memory cells that better retain long-term dependencies in data, while GRU uses a simpler approach with fewer parameters, making it faster and more efficient for training. In this case, the combination of LSTM and GRU allows the model to better detect patterns in voltage and current signals, improving the accuracy of short-circuit classification in the power system.

D. Model Training and Testing

Training and evaluating the model are key steps in ensuring accurate short-circuit classification. The steps include:

1) model training – *batch* data generation was performed to optimize memory consumption when processing large datasets; data was split into training (80%) and test (20%) sets
2) *real-time* inference – after training, the model maintains a *buffer of the last N samples*, where each new sample is added to the buffer, and the oldest one is removed
3) validation and testing – the model was tested on real and simulated data to ensure its robustness (test results were saved in the files *predicted_predictions_gru_model_window_15.npy* and *true_predictions_gru_model_window_15.npy*)

A script, *plot_data.py*, was created for result visualization, allowing the display of model predictions compared to actual values from the *dataset*.

E. Model Training Results

For model training, **10 epochs** and a graphics processing unit (GPU) were used to accelerate the processing of large amounts of data. LSTM and GRU neural networks were tested with different window sizes (WINDOW_SIZE) to analyze their performance and accuracy in fault classification. Specifically, window sizes of **5, 10, and 15** were tested. The models were saved in *.h5* format for later evaluation, and the best results were achieved with a window size of **15**, while smaller windows (e.g., size 5) resulted in lower accuracy due to insufficient data for analysis.

III Simulation Results

In this section, the simulation results for detecting and classifying short circuits in the power system using the GRU neural network are analyzed. The graphical representations indicate the actual and predicted outputs of the model, along with potential prediction errors. The simulation results are as follows:

1) LSTM networks:
   - WINDOW_SIZE = 10: where an accuracy of **99.20%** was achieved.
   - WINDOW_SIZE = 15: The model was trained on the same number of samples but with a larger time window, achieving an even better accuracy of **99.85%**.
2) GRU networks:
   - WINDOW_SIZE = 15 with 16 units in the hidden layer: **99.16%**.
   - WINDOW_SIZE = 15 with 8 units: **97.46%** accuracy.
   - WINDOW_SIZE = 15 with 2 units: **95.46%** accuracy.

The analysis shows that increasing the time window size (**WINDOW_SIZE**) contributes to improving accuracy but also increases model complexity and training time. Additionally, a higher number of units in the hidden layer of the GRU network results in better performance, but overly complex models can lead to overfitting. LSTM proved to be more accurate for this type of problem, especially when using a larger time window.

These findings confirm that a combination of LSTM and GRU networks, along with optimization of WINDOW_SIZE and the number of units, can achieve high accuracy in real-time classification of short circuits in the power system.

The displayed graph (Figure 2) shows the prediction of the GRU model with 8 neurons and a window size of 15. The blue line represents actual outputs, while the red dashed line indicates model predictions. It is evident that the model struggles with prediction stability at the beginning of the sequence but aligns with real data in the later part. Errors are present during transitions between different fault classes.

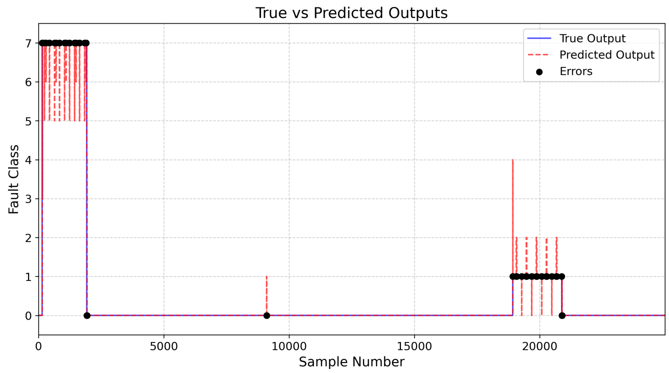

Figure 2: GRU model with 8 neurons and a window size of 15

The following figure 3 analyzes the GRU model with 2 neurons and a window size of 15. The model shows poorer stability compared to the previous one, with visible large fluctuations in detected faults.

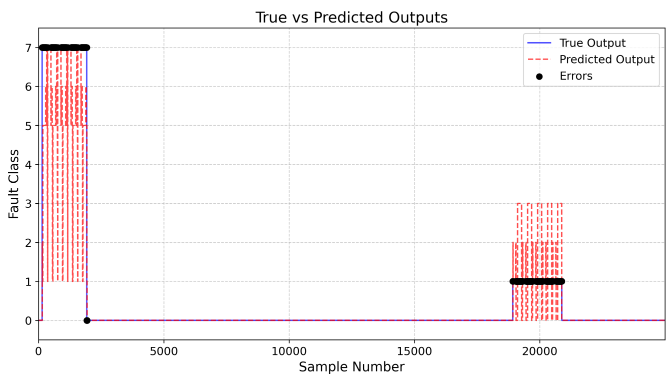

Figure 3: GRU model with 2 neurons and a window size of 15

Figure 4 presents the prediction of the GRU model with 16 neurons and a window size of 16. It is easy to notice that fault detection is much more stable and does not exhibit oscillations as in the previous two models.

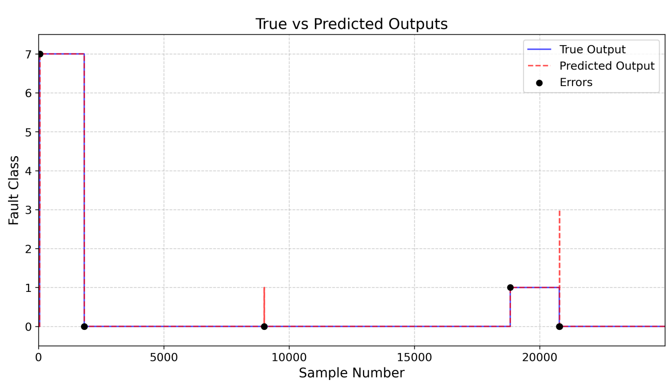

Figure 4: GRU model with 16 neurons and a window size of 15

Figure 5 compares the actual and predicted outputs of the LSTM model for short circuit detection using a window size of 10. Important to note is that this model has two LSTM layers in the background working, so it is pretty computational complex. The model shows high accuracy, and good timing, no missed faults. But errors are noticeable during transitions between different faults, suggesting that the model may struggle with recognizing sudden changes accurately but for protecting the system the model is very reliable. This behavior is very similar to that of the GRU model with 18 neurons and a window size of 16. Despite this, the predictions are generally stable, and the model is capable of *real-time* fault detection with a high degree of reliability.

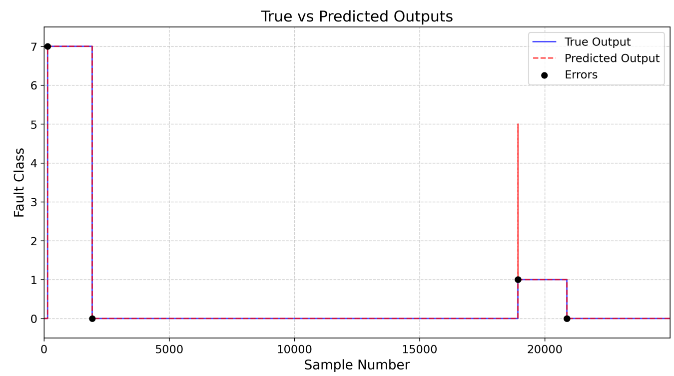

Figure 5: LSTM model with a window size of 10

The graph 6 below presents a comparison of actual and predicted outputs of the LSTM model for short circuit detection using a window size of 15. In this case we are using only one LSTM layer but we are using more previous samples.

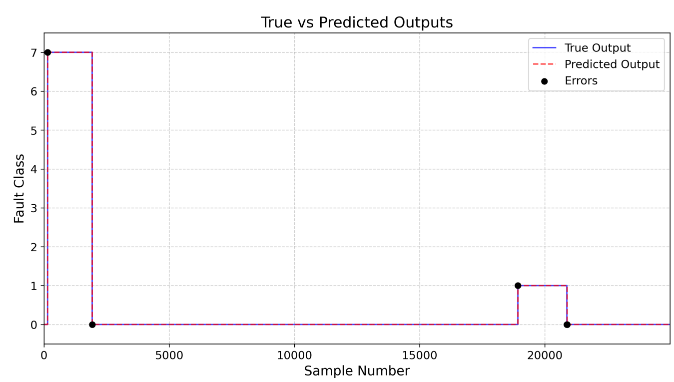

Figure 6: LSTM model with a window size of 15

From Figure 6, it is clear that this model shows the best accuracy in real-time fault detection. It is evident that the model does not exhibit undesirable fluctuations, making fault detection precise, timely, and reliable. The only drawback of this model is its computational complexity due to the large number of parameters. However, in today's era of extremely fast CPUs and GPUs with high levels of parallelism, this model is practically feasible. On figure 7 we have shown the detection deley. On the figure it can be noted that the detection deley is small and is equal less than 20 samples. In practical terms if we would have a sampling rate of 20000 Hz the detection delay would be around 1 ms.

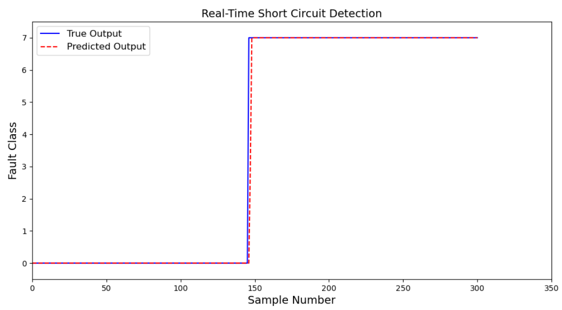

Figure 7: Moment of detection

The simulation results indicate that GRU and LSTM neural networks can efficiently classify short circuits in real time, but the model's accuracy significantly depends on the time window size and the number of neurons (parameters). Models with longer time windows and a higher number of neurons generally show greater accuracy, while errors most commonly occur during fault transition states. Further optimization may involve improving regularization and fine-tuning hyperparameters to enhance model robustness. The detection delay is dependent on the sampling rate and the relation is inverse proportional with increase in the sampling rate we get a smaller detection delay.

IV Conclusion

This paper presents the implementation of deep learning for fault detection and classification in power system transmission lines. Using GRU and LSTM neural networks, an analysis was conducted on how different network architectures and training parameters affect fault classification accuracy.

The results show that a larger time window contributes to higher model accuracy, with LSTM achieving 99.85% accuracy with a window size of 15, while GRU with 16 units achieves 99.16% accuracy. Error analysis revealed that the network most frequently makes mistakes during sudden changes in the system, suggesting the need for further model optimization.

It is concluded that the application of neural networks in *real-time* fault detection can significantly improve the security and stability of power systems. Future work may include hyperparameter optimization, improving model regularization, and testing on real-world data from power networks.

NOTE: The implemented code is available on the GitHub repository, accessible by clicking on https://github.com/Spago123.

References

[1] Shakiba FM, Azizi SM, Zhou M, Abusorrah A. "*Application of machine learning methods in fault detection and classification of power transmission lines: a survey*. Artif Intell Rev 2023, 56:5799–836. https://doi.org/10.1007

[2] Kim, San, Kim Donggeun, Jeong Siheon, Ham Ji-Wan, Lee Jae-Kyung, Oh Ki-Yong. (2020). Fault Diagnosis of Power Transmission Lines Using a UAV-Mounted Smart Inspection System. IEEE Access. PP. 1-1 https://www.researchgate.net/UAV-Mounted$_{smart_{I}nspection_{s}ystem}$

[3] Nguyen Quoc Minh, Nguyen Trong Khiem, Vu Hoai Giang, "*Fault classification and localization in power transmission line based on machine learning and combined CNN-LSTM models*," Energy Reports, volume 6, pages 5610-5622, 2024. URL: https://www.sciencedirect.com/science/article

[4] Nguyen Quoc Minh, Nguyen Trong Khiem, Vu Hoai Giang, "*Fast and Accurate Fault Detection and Classification in Transmission Lines using Extreme Learning Machine*," Advances in Electrical Engineering Electronics and Energy, volume 3, 2023. URL: https://www.sciencedirect.com/science

[5] Botong Li, Hanqing Cui, Bin Li, Weijie Wen, Dongkang Dai, "*A Permanent Fault Identification Method for Single-Pole Grounding Fault of Overhead Transmission Lines in VSC-HVDC Grid Based on Fault Line Voltage*," International Journal of Electrical Power and Energy Systems, volume 117, 2020. URL: https://www.sciencedirect.com/science.

[6] Ağir T. Using machine learning algorithms for classifying transmission line faults. DÜMF Mühendis Derg 2022. https://doi.org.

[7] Kharusi KA, Haffar AE, Mesbah M. Adaptive Machine-Learning-Based Transmission Line Fault Detection and Classification Connected to Inverter-Based Generators. Energies 2023, 16:5775. https://doi.org/10.3390.

[8] Konjicija Samim, Machine learning and inteligent control, lectures 2024/2025 https://c2.etf.unsa.ba/course

[9] Džafić Izudin, Protection and management of electrical power plants, lectures 2021/2022 https://c2.etf.unsa.ba/course

[10] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, "*Extreme learning machine: Theory and applications*," Neurocomputing, volume 70, 2006. URL: https://www.sciencedirect.com/science