AGFC-Net: An Attention-GRU-Based Classifier for Robust Fault Detection in Electrical Power Systems

A project report submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Electronics & Computer Engineering

by

Deepen Khandelwal

21BLC1224

Prateek Anand

21BLC1248

Mayukh Ray

21BLC1228

School of Electronics Engineering,

Vellore Institute of Technology, Chennai,

Vandalur-Kelambakkam Road,

Chennai - 600127, India.

November 2024

Declaration

I hereby declare that the report titled ***AGFC-Net: An Attention-GRU-Based Classifier for Robust Fault Detection in Electrical Power Systems*** submitted by us to the School of Electronics Engineering, Vellore Institute of Technology, Chennai in partial fulfillment of the requirements for the award of **Bachelor of Technology** in **Electronics and Computer Engineering** is a bona-fide record of the work carried out by me under the supervision of ***Dr. Sangeetha R G***.
I further declare that the work reported in this report, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign: ______________________________

Name & Reg. No.: ______________________________

Date: ______________________________

School of Electronics Engineering

Certificate

This is to certify that the project report titled ***AGFC-Net: An Attention-GRU-Based Classifier for Robust Fault Detection in Electrical Power Systems*** submitted by ***Deepen Khandelwal*** **(21BLC1224), Prateek Anand(21BLC1248), Mayukh Ray(21BLC1228)** to Vellore Institute of Technology Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Computer Engineering** is a bona-fide work carried out under my supervision. The project report fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor		**Head of the Department**	
Signature:	....	Signature:	....
Name:	....	Name:	....
Date:		Date:	

Examiner

Signature: ....

Name: ....

Date:

(Seal of the School)

Abstract

A transmission line is an integral part of an electrical power system that transmits electric power from sources to distribution networks. As the demand for reliable power increases, strong fault detection and classification methods are required for grid stability maintenance. This paper presents the introduction of AGFC-Net-a classifier-engineered to fulfill the purposes of Attentive GRU-based classifiers on efficient fault detection in electrical power systems. It is designed by combining convolutional layers, an attention mechanism, and GRU layers, effectively capturing and analyzing fault patterns in the transmission line data. The model trained and validated for the given work is based on a comprehensive dataset of 12,000 instances along with their corresponding labeling-that is, line voltages and currents under normal and fault conditions, respectively. The model AGFC-Net was trained and validated with a range of metrics that achieved an impressive accuracy of 99.52%. A set of extensive experiments performed with the use of confusion matrices and classification reports proved the robustness and reliability of the model for classifying the healthy as well as faulty states of power systems. The results show that AGFC-Net provides a practical, fast, and safe method for the detection of faults in the electrical power system, enhancing the protection mechanism effectively.

Acknowledgements

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. Sangeetha R G, Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dr. Ravishankar A, Dean Dr. Reena Monica, Associate Dean (Academics) & Dr. John Sahaya Rani Alex, Associate Dean (Research) of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department Dr. Annis Fathima A for her support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Contents

List of Figures

Chapter 1

Introduction

A transmission line is essential to an electrical power system because it serves as a channel through which the electrical energy is transferred from the source of generation to a wider distribution network. Globally, electricity demand has risen dramatically in the last few decades; thus, maintaining stable and reliable lines of transmission becomes vital. Transmission lines are, however, prone to faults depending upon some factors such as weather influences, faulty equipments and influences by extraneous causes. Fault in transmission lines, if not detected and corrected swiftly, may result in widespread power outage and might influence the proper functioning of the entire power grid. This fault has to be correctly identified and corrected promptly in power system operation.

Such tasks in fault detection and classification on an electrical transmission line are so complex since there exist dynamic aspects about the power system and interactions among different components. Transmission lines are part of vast electrical networks which consist of a large number of generators, transformers, and relays. They interact with each other constantly, so detection of faults should not only be identified but also differentiated into various types and under various operational conditions. Faults can be of various kinds, including line-to-line faults, line-to-ground faults, and three-phase faults, and detection and classification methods must vary for each so that protection will be taken appropriately.

Detection of faults is an essential function in the transmission line's protection scheme. In the event of a fault, the system must identify the type and location of the fault and operate the right combination of relays to isolate that section of the transmission line as quickly as possible so that no serious system damage occurs and the power grid remains stable. Heavy reliance on physical components meant that traditional fault detection methods previously using relay devices, but recent developments in digital technologies have started including data-driven approaches in an effort to enrich fault detection.

These more novel approaches base fault detection in the power system by accumulating data recorded by a myriad of installed sensors on the power system, including voltage and current measurements, allowing faults to be detected with improved accuracy and efficiency.

Of late, ML and pattern recognition techniques have been promising means of boosting fault-detection systems. The data measuring voltage and current can be analyzed retrospectively to detect faults and compare them against pre-defined patterns. It is possible to use machine learning models in large datasets for system training to identify complex fault patterns that may not be identified by more conventional methods. Furthermore, the fault conditions are learnt by the machine learning model with additional data and accuracy is obtained gradually. The supervised models have been highly used in the fault classification of phases for a three-phase power system with adequate accuracy and reliability.

Here, the dataset used will have about 12,000 labeled data points comprising measurements on line voltage and line current in both fault and normal conditions. It was collected from a power system and the faults were various types that occurred across different phases of three-phase electrical transmission lines. This dataset is already pre-labeled and can therefore be efficiently used for the training of machine learning models to identify faults and classify them. The purpose of this research is to develop a model that can predict faults in transmission lines accurately using this dataset and, thus, contribute to the development of more reliable and efficient fault detection systems for modern electrical power grids.

Chapter 2

Literature Survey

Saberia et al. [1] compared the capability of SVM and ANN in detecting faults in a centrifugal pump-a key component in an industrial system. The paper utilized empirical data whose six characteristics represented flow, temperature, suction pressure, discharge pressure, velocity, and vibration while two types of faults for every feature were represented. They used a fault classification through SVM that incorporated four kinds of kernel functions: linear, quadratic, Gaussian, and polynomial and checked its comparison with a three-layered ANN model's performance. For training purposes, 100 data points have been utilized, and 50 points were kept for testing. Based on the outcomes of experiments, it has been determined that for Gaussian and linear kernel functions, the performance of SVM was better in comparison with ANN at normal as well as noisy conditions. The SVM model, as well as the noise added into it, showed dramatic difference in performance level; while noise had a little effect on the performance of the SVM model at even up to 40%, errors within ANN model increased by making data noisy. Particularly, the linear and Gaussian kernel functions for SVM seemed to be the most robust and accurate models for use in noisy environments to diagnose faults.

Zhao et al. [2] proposed a new CBM framework to identify faults in rolling bearings; especially, DGBB with the exploitation of model-based estimation and artificial neural networks for assessing vibrational attributes of bearing and detecting faults. The MBE model offers a structured framework in simulating and analyzing the vibrations that appear in rotor-bearing systems. However, the ANN is applied in forecasting the effects of various operating parameters like defect size, speed, load, unbalance, and clearance in the vibration patterns that appeared. A set of experimental simulations was carried out and it could well be demonstrated how the proposed model accurately detects faults and evaluates damage, and the correlation between the predicted results and experimental results is very strong.

Zheng Rong Yang [3] presented a robust method for fault identification and classification in analogue integrated circuits utilizing a robust heteroscedastic probabilistic neural network (RHPNN). The paper addresses one of the primary challenges facing the evaluation of analog circuits: the distinction between and classification of the outputs of circuits that have been subjected to severe faults like opens and shorts, deputising for the parametric fluctuation effect of changes in manufacturing process. The traditional model presumes that the faulty circuit responses have constant variances, which is not proved true in this situation. In that scenario, authors have proposed RHPNN, which includes different means and variances in fault responses, which in turn enhances the performance of fault detection and classification. The RHPNN model was tested with operational amplifier and PLL circuits. In this context of fault detection, the short and open circuits are various fault models in the domain. In HSPICE, Monte Carlo simulations were applied to introduce parametric variations. The proposed setup in experiments contains developing sinusoidal inputs and measurement of relevant parameters in terms of DC voltage, DC supply current, and AC parts of both voltage and current.

Carlo N. Grimaldi and Francesco Mariani [4] presented an effort to On-Board Diagnostics (OBD) of faulty engines using the perceptron ANN. Their effort resulted from others' work done by the authors who have reviewed some examples of applications of ANNs in automotive engines diagnostic systems. Four key sensors/actuators, which are throttle valve, rotational speed, torque, and intake manifold pressure, were used to train and test modules of ANN. The authors proposed a fault detection system that generates fault code sequences by comparing the experimental data with the values predicted by the ANN models. Such models showed better capabilities for fault isolation; hence, the system qualifies for use in either OBD applications or as a quality control system in engine production lines.

Muhammad Zain Yousaf et al. [5] has proposed a Bayesian optimized LSTM-DWT-based fault detection technique in MMC-based HVDC systems. By using a combination of LSTM networks and DWT to extract features that will subsequently increase the noise immunity and resistance to faults, three levels of relay systems ensure accurate fault detection up to a distance of 200 km over multiple time windows of 1 ms, 1.5 ms, and 2 ms. Bayes Optimization (BO) is used in the hyperparameter tuning process. In this case, the training process improves significantly and achieves an accuracy of 99.04%. The faults are detected up to 480 ohms without causing false trips due to external faults. This method uses a single, intelligently tuned model, hence it does not require any communication links between stations; hence, there is reduced interference, and it will be more reliable. The model showed 100% reliability of resilience to the occurring external faults and disturbances; it outperformed non-AI techniques in fault resistance detection. For instance, as illustrated in figure, the classical methods with the overcurrent or capacitor-discharging criteria can identify faults up to only 10 ohms or 400 ohms and contain more noise. Metrics: Accuracy: 99.04% recognition rate. Fault Resistance: Up to 480 ohms. Resilience: 100% over external faults. Response time: 1.5 ms, no protection at the rear is needed. Confirmation of practical applicability to HVDC grid protection will be given by the results, and future work will concentrate on scaling this up to multi-terminal HVDC systems and applying it to RTDS studies.

Ana Andrade and her coauthors [6] proposed a new FDD concept of pneumatic control valves by making use of artificial neural networks, being one type of NARX model. This approach combines a combination of potentiality from fault emulation, hierarchical decision trees, and structures of neural networks towards identification of faults in pneumatic systems with good categorization efficiency. This study utilizes NARX networks to produce residuals that reflect the system's dynamics and constructs a decision tree derived from the most relevant residuals, thereby aiding in the isolation and diagnosis of faults. The suggested methodology effectively minimizes computational time and is adaptable for application across various industrial processes. Evidence of its efficacy was provided by the method's ability to diagnose faults accurately with limited computational demands, as illustrated through the simulation of fault signatures.

Hong Je-Gal et al. [7] presented a new time-frequency feature fusion approach for robust fault detection in marine main engines. The method focuses particularly on the operational reliability of internal combustion engines and maritime vessels. It integrates the features of both the time and frequency domains, thus improving the accuracy of fault detection even with low-precision data. The methodology uses instantaneous revolutions per minute, which is obtained from vibration pulse data, plus statistical characteristics for time-domain analysis and a combination of EMD and ICA in the extraction of frequency-domain features. A deep neural network was used to classify anomalous conditions, thereby indicating the effectiveness of the integrated model. The proposed approach resulted in 100% accuracy up to 1x to 10x downsampling rates and, more interestingly, resulted in 96.3% accuracy at 100x downsampling rate. This significantly outperforms time-domain and frequency-domain feature analysis. The study puts forward the promise of predictive maintenance and fault detection in the context of autonomous maritime activities, with an underscore on robustness in realistic conditions where sensor data accuracy can be compromised.

Aherwar [8] has done a comprehensive literature survey in relation to the detection of faults in a gearbox with the aid of vibration analysis techniques. This work indicates critical importance toward proactive detection of faults in a gearbox, which is crucial for most mechanical systems. Aherwar defined the role played by vibration analysis in predictive maintenance and argued that any variation in patterns may indicate possible faults. Techniques of fault detection were categorized into three main groups: time domain, frequency domain, and time-frequency domain. All techniques provided useful methods in the diagnosis of faults with different natures. Time-domain techniques have the ability to detect this impulsive nature of defects; frequency techniques can identify specific frequencies of faults through spectral analysis and power spectral density. The time-frequency domain methods describe the non-stationary signal using the wavelet transform and the short-time Fourier transform (STFT). Besides, Aherwar's work includes the challenges of gearbox fault diagnosis, especially identifying gearbox-specific signals from extraneous sources. Moreover, the author used some complicated approaches, such as artificial intelligence, which includes neural networks and fuzzy logic, with a view to increasing the accuracy of fault classification. In-depth findings note the potential of coupling vibration analysis with sophisticated diagnostic systems to significantly improve the fault detection task, especially in early-stage faults in gearboxes. The current literature review appears promising concerning methodologies based on vibrations. However, much more should be achieved to establish the strength of these systems with complex fault scenarios that involve multiple faults in gear assemblies.

Ming Ge et al. [9] discussed a diagnosis method for faults in sheet metal stamping processes in the use of SVM techniques. The proposed method addresses the problem encountered in identifying faults for complicated stamping operations by nonlinear as well as transient dynamics. Utilizing the kernel functions of the SVM structure translates low-dimensional strain signals to higher dimensional feature spaces, thus improving detection accuracy. For single-step blanking, six problematic scenarios were tested: misfeed, slug, material problems. The model achieved a classification success rate higher than 96.5% using a dataset of 240 samples. Additionally, even though the size of training sets was relatively small, as low as two training samples resulted in a success rate of 87%With respect to multi-step progressive stamping operations, the SVM model succeeded better than ANN methodologies in detecting faults such as misfeeds and slugs, and with a maximum success rate of up to 75.7%. The study concludes that SVM is effective for the fault detection in manufacturing because of its good accuracy with minimal training data.

V. Puig, M. Mrugalski, A. Ingimundarson, J. Quevedo, M. Witczak, and J. Korbicz [10] proposed a passive robust fault detection algorithm based on a GMDHNN model incorporating parameter uncertainty. As the title of the paper states, it presents a backward fault detection test that is intended to determine if any of the parameters in the set of uncertain parameters would reproduce the observed behavior. It further uses interval constraint satisfaction algorithms to implement the backward test that can quickly determine if the measured state is consistent with the GMDHNN and its associated uncertainty. The proposed approach was validated using the servo-actuator benchmark from the European Project DAMADICS. Results show that fault detection performance improves when the proposed backward test is implemented using constraint satisfaction algorithms especially in situations involving uncertainties.

Shi and Yu, and Xiao-Hua [11] proposed a hybrid method of structural damage detection using wavelet transforms along with ANN technique. The authors intend to enhance the reliability of structural health monitoring systems using ANNs proficiencies in the classification of damage patterns and wavelet analysis ability to manage signals that are non-stationary. The proposed method is tested using an SHM benchmark problem provided by the International Association for Structural Control and the American Society of Civil Engineers. At first, features are extracted from transient sensor data using the wavelet transform, followed by the classification task with an ANN. The advantage of the wavelet transform for the detection of structural damage is the multi-resolution analysis, which is quite effective for detecting sudden spikes or a change in the vibration signal. Adaptation of ANN also allows the system to learn new data from sensors-that is, to provide a dynamic learning mechanism. A four-story steel-frame model has been used to demonstrate the application of the system in computer simulations and showed excellent performance, where a high classification accuracy of 99.52% in training and 95.49% in testing for a specific damage scenario has been achieved. The results were observed to indicate the promising potential of wavelet transform and ANN for the enhanced real-time detection of structural damage.

Emad Efatinasab et al.[12] presented a framework grounded in Bayesian Neural Networks (BNNs) aimed at forecasting fault zones within smart grid systems. This approach mitigates the susceptibilities of machine learning models to adversarial incursions while tackling the difficulties associated with false alarms in fault detection mechanisms. The researchers propose a Bayesian methodology that systematically quantifies uncertainty in predictive outcomes, thereby bolstering model resilience and diminishing the occurrence of false alarms. The Bayesian Neural Network ensures a probabilistic framework for the uncertainty management in the model parameters as it is based on LSTM architecture and trained with Variational layers along with Flipout Monte Carlo estimators. The model, in terms of prediction of the fault zone, was impressive at 0.958. A mechanism was also developed to detect white-box adversarial attacks using predictive entropy and mutual information under uncertainty, which raised its accuracy to 0.981 in the identification of white-box adversarial attacks. This will allow not only very high accuracy but also smaller computation overheads in its application to smart grid infrastructures. Uncertainty-based approaches to real-time predictive analytics would encourage even more reliable and secure decision-making approaches, thereby combining this approach as an important innovation in countering adversarial attacks and strengthening the reliability of fault prediction systems used in smart grids.

Majid Jamil, Sanjeev Kumar Sharma and Rajveer Singh [13] has developed a methodology for fault detection and classification within electrical power transmission systems using Artificial Neural Networks. The study underlines the importance of quick and accurate fault detection with a view to stabilize power systems mainly in wide, geographically spread networks. The proposed researchers indicated the use of three-phase current and voltage inputs from one end of the transmission line as data inputs for a feedforward neural network trained with the backpropagation algorithm tailored toward fault detection and classification. Fault types including line-to-ground faults and others were tested on the model because of which system ability to very accurately classify faults under various operating conditions was demonstrated. Results showed that indeed the proposed ANN-based method can fairly reliably classify the fault, regardless of fault resistances or locations, as well as the variety of system parameters. It underlines the need to use an optimal architecture for neural networks and an appropriate training algorithm for realizing best performances. In this regard, backpropagation networks are particularly effective as such networks accept tremendous datasets, which are normally encountered in power systems. It was prepared in MATLAB with the help of SimPowerSystems and proved the method to be rather fast and accurate at detecting the type of fault; it is quite an efficient tool for transmission line protection in electrical power systems, and the potential on which ANNs-based solution for fault detection and classification can be extended to the distribution networks.

Anamika Yadav and Yajnaseni Dash [14] presented a comprehensive review on the application issues of ANNs for transmission line protection. The paper generally covered fault detection, classification, location, direction discrimination, and also faulty phase identification in high-voltage transmission lines. Generally, the authors of the paper highlighted that faults like short circuit can cause severe damage to power systems and quick isolation is necessary in order to maintain stability within the system. This was admitting that ANNs were adaptive, nonlinearly mapping, and learning-capable and hence adaptable for bettering the accuracy and reliability of the protective relay. Various types of faults, symmetrical and unsymmetrical faults, have been discussed by authors along with the advantages of ANN for such complex fault conditions like high impedance faults and variable source impedance. The paper used ANN in distance protection, fault location, and direction discrimination for various solutions to problems like harmonic distortion and fault resistance. They reviewed several methods based on relative efficacies in ANN-based approaches for enhancement of fault detection and location within the power systems. In conclusion, the ANNs have proved robust, accurate, and efficient instruments in the protection of transmission lines toward enhancing stability and reliability in modern power systems. This paper presents a rather comprehensive review of applications about ANN, which would come in really handy for researchers who are working to apply ANNs to power system protection applications.

Veerasamy et al.[15]proposed an HIF detection method for solar Photovoltaic (PV) integrated power systems using an LSTM-based Recurrent Neural Network [3]. An IEEE 13-bus system is integrated with a 300 kW solar PV array, and the model has been developed in MATLAB/Simulink. Method Used Discrete Wavelet Transform (DWT) with the mother wavelet being db4 is used for extracting the features in three-phase currents for both faulty and non-faulty conditions. DWT is also used for capacitor switching, load switching, and transformer inrush current. LSTM classifier The overall accuracy stands at 91.21% but with the success rate of 92.42% while detecting HIFs in the PV-integrated network. It would also need to be compared against other classifiers such as KNN, SVM, J48 Decision Tree, and Naive Bayes, etc. In this investigation, the performance of the LSTM classifier is found to be significantly outperforming the others in terms of precision, recall, F-measure, and Kappa statistics. These results bring about strength to the proposed model based on LSTM for HIF detection and are capable of better competition against traditional techniques.

S. R. Mohanty, A. K. Pradhan, and A. Routray [16] presented work on the effort to enhance the reliability of fault detection, considering noise, frequency deviation of the system, and uncertainties other than those, by employing the CUSUM method in the fault-detection algorithm for power system relaying that is based on the cumulative sum algorithm. This method is insensitive to any change of load and produces higher index values without sacrificing detection speed. The proposed approach uses the CUSUM based on two supplementary signals derived from the current waveform, followed by a two-sided CUSUM test for fault detection. The CUSUM method is insensitive to noise and facilitates fault-detection speed with higher accuracy than the sample-to-sample as well as cycle-to-cycle comparison of conventional methods, especially in fluctuating loads or frequency deviations in the system. Better results regarding robustness against noise, frequency drift, and spikes within the detection speed are evidenced by the results of the performance of the algorithm evaluated using simulations with synthesized and real-world data.

A new method for the fault detection and classification in the transmission lines was proposed by Silva, Souza, and Brito[17] by means of oscillographic data especially targeted at the difficulty of discerning faults from power quality disturbances, such as voltage sags. The method involves both wavelet transform and artificial neural networks. Both time and wavelet domains have been used for fault detection based on the current waveform analysis, while classification is made based on an ANN trained with patterns of the voltage and current waveform. By using data from the Brazilian utility company CHESF, the method has been successfully applied with great results concerning the performance in detecting and classifying faults. In the detection module, the current waveform is normalized and further analyzed with the discrete wavelet transform (DWT). Faults are identified by examining the energy contained in the wavelet coefficients where some conditions distinguish faults from disturbances. The module for classification of faults utilizes an MLP ANN in the fault discrimination from windowed samples of voltage and current. The robustness is checked using simulated and real fault records where a 100% success in fault detection and classification is attained in tests simulated. For the real data of CHESF transmission lines, very good accuracy in fault classification was obtained, although some misclassifications took place for particular types of faults. The new algorithm largely enhances fault detection and type classification capabilities for transmission lines and could therefore be a reliable tool for power system protection.

A fault classification algorithm for transmission lines using a hybrid combination of Discrete Wavelet Transform (DWT) along with Back-Propagation Neural Networks (BPNN) has recently been proposed by C. Pothisarn and A. Ngaopitakkul [18]. Continuing from this work, they simulate the fault signals in ATP/EMTP and applied Daubechies4 as the mother wavelet. They extracted high-frequency components from the first scale of the decomposition to detect faults. The approach accurately classified faults even in the varying conditions of faults like type, location, and inception angle with more than 97.22% Feature extraction using DWT combined with classification through BPNN brought up significant results and is proved to be better compared to traditional methods concerning efficiency and precision.

Yong Deng [19]. presented an improved fault diagnostic approach for analog circuits employing an upgraded hierarchical Levenberg-Marquardt (LM) discrete Volterra series (DVS) algorithm combined with a condensed closest neighbor (CNN) algorithm, known as IDVS-CNN. This method simplifies difficult DVS parameter calculations by using hierarchical design and a Bayesian information criterion for order selection. The IDVS-CNN model diagnoses circuit failures more accurately and with less computing complexity than existing methods. The model showed significant performance increases, with macro and micro F1 scores of 0.903 and 0.894, respectively, suggesting enhanced fault identification capacity in analog circuits.

A fault detection model for digital VLSI circuits was proposed by Lamya Gabera[20] utilizing deep learning, specifically a stacked sparse autoencoder (SSAE). The methodology is designed to enhance fault detection by lowering the dimensionality of features extracted from large digital circuits. The approach is divided into three stages: creating test patterns with ATALANTA software, reducing feature dimensions with SSAE, and classifying defects using a softmax classifier. The model was tested on eight combinational circuits from the ISCAS'85 benchmark, reaching a maximum fault coverage of 99.2% and a validation accuracy of up to 99.7% during the feature reduction phase, indicating its effectiveness to detect stuck-at faults.

Chapter 3

Methodology

3.1 Data Collection and Preprocessing

Definition

The dataset for this study, which consists of readings of line voltages (Va, Vb, Vc) and line currents (Ia, Ib, Ic) both under normal and fault conditions in a power system. It contains 12,000 labeled data points for the target variable Output (S) referring to the status of the system as normal or faulty.

Preprocessing steps include

- **Handling Missing Values:** The missing values will be checked in the dataset. Then, missing values will be handled either by imputation or deletion of related records, as applicable.
- **Elimination of Less Important Columns:** The columns that do not contribute any information for predicting the model are eliminated; 'Unnamed: 7' and 'Unnamed: 8' are two such columns.
- **Label Encoding:** The target variable Output (S) will be encoded to express the fault and normal conditions.
- **Data Splitting:** The dataset was split into 80% for training and 20% for testing. This ensures an adequate number of data points for both model validation and training.
- **Normalization:** StandardScaler from scikit-learn was applied for feature normalization on the feature dataset (Ia, Ib, Ic, Va, Vb, Vc). This standardization

ensures that the features have a mean of 0 and a standard deviation of 1, which accelerates model convergence during the training process.

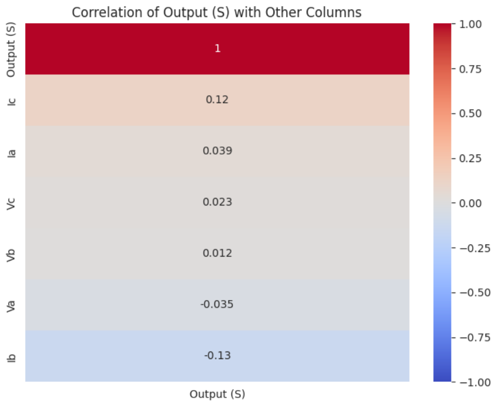

FIGURE 3.1: Correlation of outputs with other columns

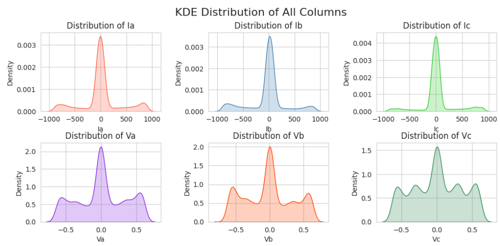

FIGURE 3.2: KDE Distribution of all columns

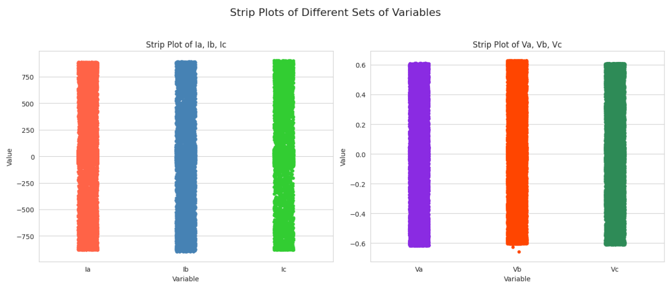

FIGURE 3.3: Strip plots of different sets of variables

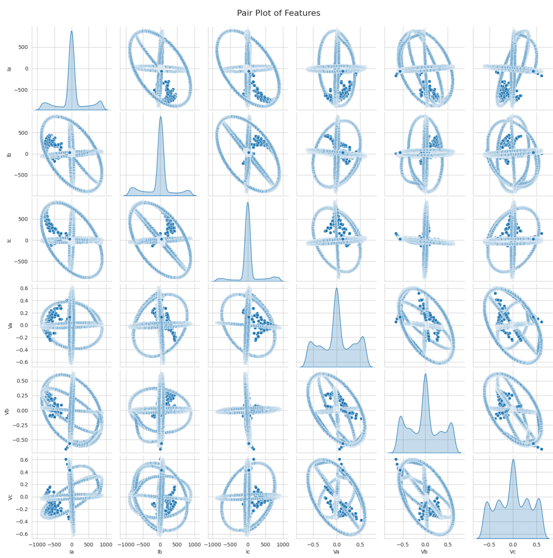

FIGURE 3.4: Pair plot of features

3.2 Feature Engineering

Six input features have direct feeds into the model, and no feature engineering is done above that. Now, those visualizations to get a better view of the feature distributions and their relationships with the target variable are correlation heatmaps, KDEs, and QQ plots. We can determine the strength of linear relationships between inputs and the target through correlation analysis; therefore, it helps decide how to retain all six features. Furthermore, we will use the QQ plot to check the normality of feature distributions so features are proper for the model, particularly when linearity or normality is of more advantage for training.

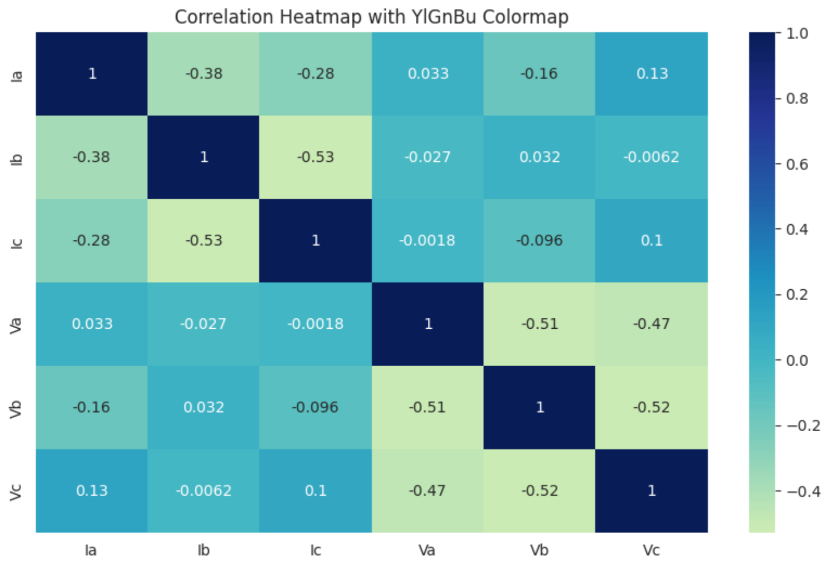

FIGURE 3.5: Correlation heat map

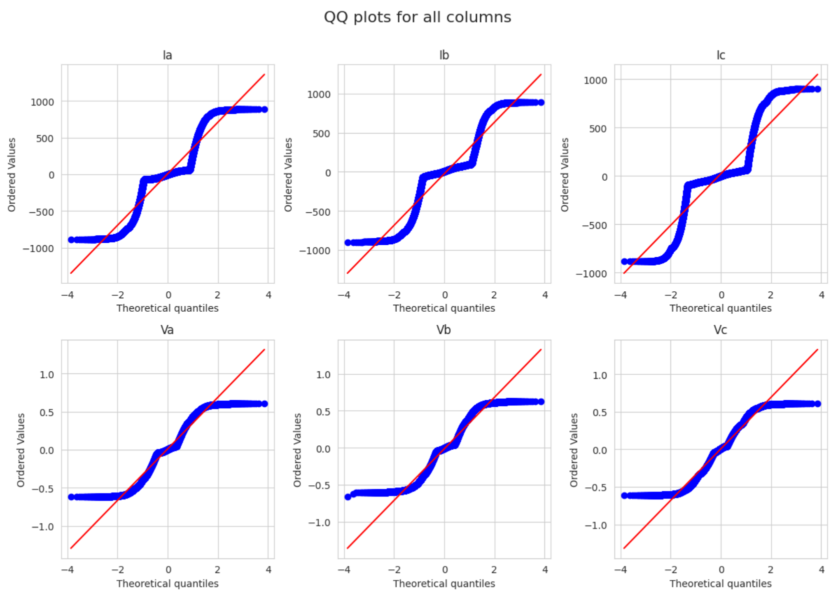

FIGURE 3.6: QQ Plots for all columns

3.3 Model Architecture

The suggested architecture of the proposed model, the AGFC-Net, The basic core consists of:

- **Convolutional Layers**: Two 1D Convolutional layers extract spatial features from sequences of input measurements of current and voltage. These layers have a kernel size of 3 and zero padding to keep the length of the sequence constant.

  The 1D convolution operation is defined as:

  $$\text{Conv1D}(x)_i = \sigma\left(\sum_{k=1}^{K} w_k \cdot x_{i+k-1} + b\right)$$

  where x is the input sequence, w and b are the weights and biases of the convolutional filter, K is the kernel size, and σ is the activation function (LeakyReLU).

- **Inclusion of Attention Mechanism**: An attention layer is incorporated into the model, so that while the model trains, it learns to focus on the most relevant features. This is calculated through a linear transformation followed by a softmax operation to obtain the attention weights.

  The attention mechanism is given by:

  $$\text{Attention}(x) = \text{Softmax}(W_{\text{attn}} \cdot x + b_{\text{attn}})$$

  where W_{attn} and b_{attn} are the weights and biases for the attention layer, and the softmax function is used to normalize the attention weights.

- **GRU Layer**: A GRU layer is applied to capture sequential dependencies in the data, helping the model capture temporal features that may signify faults in the system. The GRU update rules are as follows:

  $$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

  $$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

  $$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t])$$

  $$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

  where z_t and r_t are the update and reset gates, $\tilde{h}_t$ is the candidate hidden state, h_t is the new hidden state, and $\odot$ denotes element-wise multiplication.

- **Fully Connected Layers**: The output from the GRU layer is passed through several fully connected layers with LeakyReLU activation, which reduces the dimension of the output to ultimately predict the fault condition. These layers are given by:

$$\mathrm{FC}_1(y) = \sigma(W_1 \cdot y + b_1)$$

$$\mathrm{FC}_2(y) = \sigma(W_2 \cdot y + b_2)$$

$$\mathrm{Output}(y) = \sigma(W_3 \cdot y + b_3)$$

  where W_i and b_i are the weights and biases for the fully connected layers, and σ is the LeakyReLU activation function for intermediate layers and sigmoid activation function for the output layer.

- **Regularization**: Batch normalization and dropout techniques are applied to prevent overfitting and help the network generalize well. The batch normalization operation is defined as:

$$\mathrm{BN}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

  where μ and σ^2 are the mean and variance of the batch, and ϵ is a small constant. Dropout is applied as:

$$\mathrm{Dropout}(x) = x \cdot \mathrm{mask}$$

  where the mask is a binary vector sampled from a Bernoulli distribution.

- **Binary Classification**: The final output layer uses a sigmoid activation function to generate probabilities between 0 and 1, representing the likelihood of a fault occurring in the system:

$$\mathrm{Output}(y) = \frac{1}{1 + e^{-y}}$$

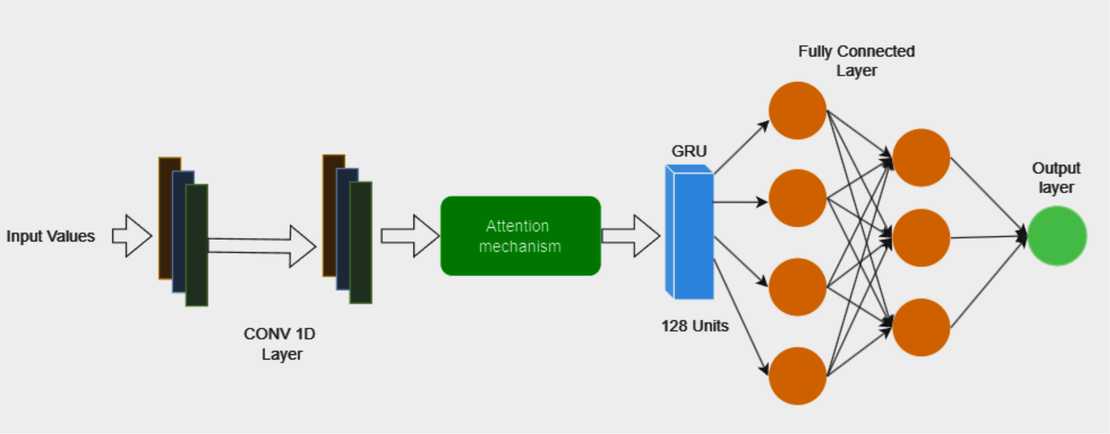

FIGURE 3.7: Model Architecture

Chapter 4

Results and Discussions

Objective

This paper designs an efficient fault detection classifier in electric power systems, particularly for transmission line faults. The objective is to design the Attention-GRU Fault Classifier, AGFC-Net that uses both Attention and Gated Recurrent Unit (GRU) mechanism for efficient capturing of spatial and temporal dependencies in input data that can be line currents and voltages. The model will address problems such as missing values, irrelevant features, and normalization of data to boost its performance. In addition, the research assesses the performance of the proposed AGFC-Net by comparing it with conventional machine learning models along key performance measures including accuracy, precision, recall, and F1-score. This can improve the potential for fault detection, giving an upper hand in reliability and efficiency in electrical power systems. This results in better strategies for the management and maintenance of faults, hence an important advancement in power system monitoring.

4.1 Results and Comparison

4.1.1 Model Performance

The model proposed here performed exceptionally well, as attested by both its training and validation metrics. More than 50 epochs have led to highly effective reduction of training loss-which does reach stabilization around 0.02-after a rapid increase at the very beginning. In a similar way, there was trend in validation loss, which stabilized at around 0.03. This proves that the learning is effective, showing strong generalization to unseen data. Actually, for the first epochs, training accuracy had risen very well, achieving more than 98% by the 10th epoch and 99.52% at the end of the training. The validation accuracy corresponded to the training one and reached 99.52% as well. Both loss and accuracy metrics convergence proves that the model had successfully avoided those situations where it overfitted or underfitted the data and thus was making reliable and stable predictions. The results further confirm the fact that this model is highly capable and persistently performs good on the training dataset and the validation set.

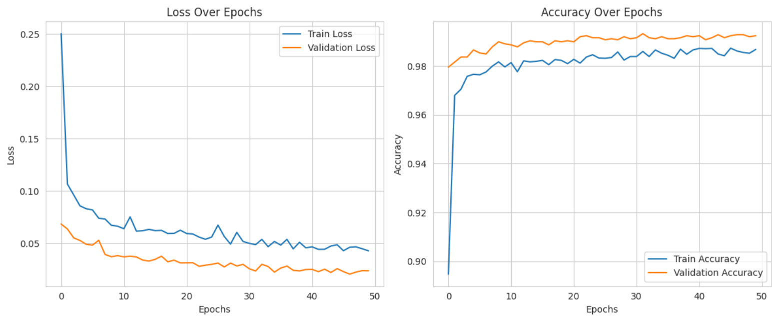

FIGURE 4.1: Accuracy and loss plot

4.1.2 Confusion Matrix

The confusion matrix indicates the performance of the model with respect to classification of the test dataset. The model correctly classified 1302 as class 0 and 1081 as class 1 thus showing good ability in differentiation between the classes. This is the least false positives of 4 class 0 samples being treated as class 1 and only 14 instances of class 1 being reported as class 0. These results suggest extremely balanced quality between sensitivity and specificity with little misclassification. In addition, the high number of correctly classified samples along with low errors further confirm the robustness and reliability of the model. This performance is in line with the overall accuracy of 99.52% discussed above.

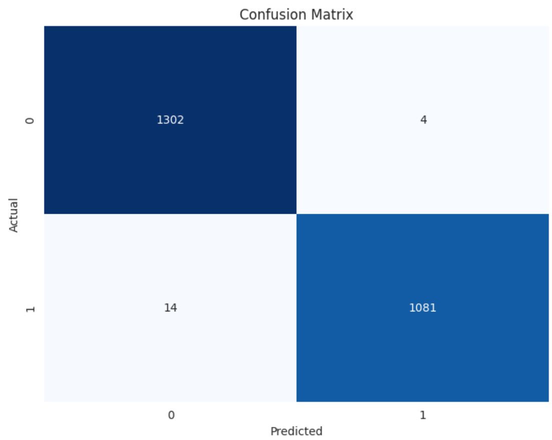

FIGURE 4.2: Confusion Matrix

4.1.3 Classification Report

The classification report evaluates the performance of a classification model by reporting key metrics for each class: precision, recall, F1-score, and support. Precision is given as the fraction of correctly classified positive samples out of all those classified as positive, and recall as the fraction of actual positive samples identified. The F1-score is the harmonic mean of precision and recall. High scores were reflected in both classes because precision, recall, and F1-scores stood at above 0.99; hence, the classification would prove to be balanced and accurate. The number of support column points of real samples in each class indicates that the dataset generally maintains balance. With 1306 samples for Class 0 and 1095 for Class 1, the overall accuracy of the model stands at 99.54%, with macro and weighted averages suggesting consistency across classes. These metrics collectively demonstrate that the model is strong in its classification capabilities.

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	1306
1	1.00	0.99	0.99	1095
accuracy			0.99	2401
macro avg	0.99	0.99	0.99	2401
weighted avg	0.99	0.99	0.99	2401

Figure 4.3: Classification Report

Chapter 5

Conclusion and Future Scope

5.1 Conclusion and Future Scope

In this paper, we proposed a model called AGFC-Net, a classifier based on attention-GRU for effective fault detection in electrical power systems. The proposed model utilized convolutional layers that extract spatial features and apply the attention mechanism to find relevant features. GRU layers are used further to capture sequential dependency. Fully connected layers combined with techniques of batch normalization and dropout helped refine features and avoid overfitting, respectively. As the technique was for the binary classification approach, the model could easily make effective predictions regarding fault conditions.

The proposed experimental results confirm that AGFC-Net can potentially detect faults in power systems if the input measurements of current and voltage are fed to the model. The integration of attention mechanisms and GRU layers into the model enhances its spatial features as well as temporal features, significantly improving its fault detection ability.

5.1.1 Future Scope

Despite the promising results obtained from the proposed AGFC-Net, several other scopes can be explored further:

- **Application to Multiclass Fault Detection:** Future work may include the development of a multiclass fault-classification model, allowing the classification of various types of faults in the power system.

- **Incorporation of Additional Features:** This model can involve further features such as environmental conditions or other electrical parameters that might prove useful for this purpose.
- **Real-Time Fault Detection:** The AGFC-Net should be implemented in real-time fault detection systems to put it into practice, allowing for immediate responses to faults in electrical power systems.
- **Optimization of Model Architecture:** Further optimization of the model architecture, including experimenting with diverse types and combinations of neural network layers, should yield performance and efficiency improvements.
- **Transfer Learning and Domain Adaptation:** Exploring the applicability of transfer learning techniques and domain adaptation to improve the model's ability to be applied in different power systems with unique characteristics and configurations.

Chapter 6

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, TensorDataset
import scipy.stats as stats

df = pd.read_csv('/content/detect_dataset.csv')
df.head()
df.shape
df.info()
df.describe()
df.isnull().sum()
df.drop(['Unnamed: 7','Unnamed: 8'],axis=1,inplace = True)
df.duplicated().sum()
df['Output (S)'].unique()

# Frequency data
value_counts = df['Output (S)'].value_counts()
```

```
# Create subplots for bar chart and pie chart with Seaborn color palette
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

# Bar chart
ax1.bar(value_counts.index, value_counts.values, color='salmon', edgecolor='black')
ax1.set_xlabel('Class')
ax1.set_ylabel('Frequency')
ax1.set_title('Frequency of Each Class')

# Pie chart with Seaborn color palette
colors = sns.color_palette("husl", len(value_counts))  # You can choose other palette
ax2.pie(value_counts, autopct='%0.2f%%', labels=value_counts.index, colors=colors)
ax2.set_title('Class Distribution with Seaborn Palette')

plt.suptitle('Frequency and Distribution of Classes')
plt.tight_layout()
plt.show()

import seaborn as sns
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr()[['Output (S)']].sort_values(by='Output (S)', ascending=False), a
plt.title('Correlation of Output (S) with Other Columns')
plt.show()

ls = ['Ia', 'Ib', 'Ic', 'Va', 'Vb', 'Vc']

# Custom color palette for the KDE plots
colors = ['#FF6347', '#4682B4', '#32CD32', '#8A2BE2', '#FF4500', '#2E8B57']

plt.figure(figsize=(12, 5))
sns.set_style("whitegrid")  # Set Seaborn style for better aesthetics

for i in range(2):
    for j in range(3):
        plt.subplot(2, 3, i * 3 + (j + 1))
        sns.kdeplot(df[ls[i * 3 + j]], color=colors[i * 3 + j], shade=True)  # Added c
        plt.title(f'Distribution of {ls[i * 3 + j]}')  # Added title to each subplot
        plt.xlabel(ls[i * 3 + j])  # Label x-axis for each plot
```

```
plt.subplots_adjust(hspace=0.4, wspace=0.4)
plt.suptitle('KDE Distribution of All Columns', fontsize=16)
plt.show()

# Define custom color palette for each plot
palette1 = ['#FF6347', '#4682B4', '#32CD32']  # Colors for Ia, Ib, Ic
palette2 = ['#8A2BE2', '#FF4500', '#2E8B57']  # Colors for Va, Vb, Vc

# Melting the data
ls = ['Ia','Ib','Ic','Va','Vb','Vc']
df_melted1 = df[ls[:3]].melt(var_name='Variable', value_name='Value')
df_melted2 = df[ls[3:]].melt(var_name='Variable', value_name='Value')

# Create subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Strip plot for Ia, Ib, Ic with custom palette
sns.stripplot(x='Variable', y='Value', data=df_melted1, ax=ax1, palette=palette1)
ax1.set_title('Strip Plot of Ia, Ib, Ic')

# Strip plot for Va, Vb, Vc with custom palette
sns.stripplot(x='Variable', y='Value', data=df_melted2, ax=ax2, palette=palette2)
ax2.set_title('Strip Plot of Va, Vb, Vc')

# Set a global title and adjust layout
fig.suptitle('Strip Plots of Different Sets of Variables', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.95])  # Adjust layout to include the overall title

# Show the plot
plt.show()

# Pair plot with KDE on the diagonal and custom color palette
pair_plot = sns.pairplot(df.drop('Output (S)', axis=1), diag_kind='kde', palette=sns.
pair_plot.fig.suptitle('Pair Plot of Features', fontsize=16)
pair_plot.fig.subplots_adjust(top=0.95)
plt.show()

# Custom colormap: "YlGnBu" for a vibrant look
plt.figure(figsize=(10, 6))
```

```
sns.heatmap(df.drop('Output (S)', axis=1).corr(), annot=True, cmap='YlGnBu', center=0
plt.title('Correlation Heatmap with YlGnBu Colormap')
plt.show()

ls = ['Ia', 'Ib', 'Ic', 'Va', 'Vb', 'Vc']

plt.figure(figsize=(12, 8))

for i in range(2):
    for j in range(3):
        plt.subplot(2, 3, i * 3 + (j + 1))
        stats.probplot(df[ls[i*3+j]], dist="norm", plot=plt)
        plt.title(ls[i*3+j])

plt.tight_layout()
plt.suptitle('QQ plots for all columns', y=1.05, fontsize=16)
plt.show()

x_train,x_test,y_train,y_test = train_test_split(df.drop('Output (S)',axis=1),df['Out

# Standardize the dataset
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

# Convert back to DataFrame for PyTorch Tabular compatibility
train_data = pd.DataFrame(x_train, columns=df.columns[:-1])
train_data['Output (S)'] = y_train.values
test_data = pd.DataFrame(x_test, columns=df.columns[:-1])
test_data['Output (S)'] = y_test.values

# Convert data to PyTorch tensors
X_train_tensor = torch.tensor(x_train, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train.values, dtype=torch.float32).view(-1, 1)
X_test_tensor = torch.tensor(x_test, dtype=torch.float32)
y_test_tensor = torch.tensor(y_test.values, dtype=torch.float32).view(-1, 1)

# Create TensorDatasets
train_dataset = TensorDataset(X_train_tensor, y_train_tensor)
```

```
test_dataset = TensorDataset(X_test_tensor, y_test_tensor)

# Create DataLoaders
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False)

import torch
import torch.nn as nn
import torch.nn.functional as F

class NODE(nn.Module):
    def __init__(self, input_dim):
        super(NODE, self).__init__()

        # Convolutional layers to extract features
        self.conv1 = nn.Conv1d(input_dim, 64, kernel_size=3, padding=1)
        self.conv2 = nn.Conv1d(64, 128, kernel_size=3, padding=1)

        # Attention mechanism (Simple) to focus on important features
        self.attn_layer = nn.Linear(128, 128)

        # GRU layer to capture sequential dependencies (if any)
        self.gru = nn.GRU(128, 128, batch_first=True)

        # Fully connected layers
        self.fc1 = nn.Linear(128, 64)
        self.fc2 = nn.Linear(64, 32)
        self.fc3 = nn.Linear(32, 1)

        # Batch Normalization and Dropout for regularization
        self.bn1 = nn.BatchNorm1d(64)
        self.bn2 = nn.BatchNorm1d(128)
        self.dropout = nn.Dropout(0.5)

        # Activation function
        self.activation = nn.LeakyReLU(negative_slope=0.1)

    def forward(self, x):
        # Reshape the input to [batch_size, num_channels, sequence_length]
```

```
        x = x.unsqueeze(2)  # Add a sequence length dimension

        # Convolutional layers with Batch Normalization
        x = self.activation(self.bn1(F.relu(self.conv1(x))))
        x = self.activation(self.bn2(F.relu(self.conv2(x))))

        # Attention mechanism
        attn_weights = F.softmax(self.attn_layer(x), dim=2)
        x = x * attn_weights  # Apply attention weights

        # GRU layer
        x, _ = self.gru(x)

        # Flatten the output and pass through fully connected layers
        x = x[:, -1, :]  # Use the last time-step
        x = self.dropout(self.activation(self.fc1(x)))
        x = self.dropout(self.activation(self.fc2(x)))
        x = self.fc3(x)

        return x

# Define the model
model = NODE(input_dim=6)  # 6 input features

# Define loss function and optimizer
criterion = nn.BCEWithLogitsLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

# Training loop
num_epochs = 50
for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    for inputs, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

```
        running_loss += loss.item()

    print(f"Epoch [{epoch+1}/{num_epochs}], Loss: {running_loss/len(train_loader):.4f

# Evaluation
model.eval()
with torch.no_grad():
    correct = 0
    total = 0
    y_pred = []
    y_true = []
    for inputs, labels in test_loader:
        outputs = model(inputs)
        predicted = torch.sigmoid(outputs).round()
        total += labels.size(0)
        correct += (predicted == labels).sum().item()
        y_pred.extend(predicted.numpy())
        y_true.extend(labels.numpy())

    accuracy = 100 * correct / total
    print(f"Accuracy: {accuracy:.2f}%")
```

Bibliography

[1] M. Saberi, A. Azadeh, A. Nourmohammadzadeh, and P. Pazhoheshfar. Comparing performance and robustness of SVM and ANN for fault diagnosis in a centrifugal pump. *Proceedings of the 19th International Congress on Modelling and Simulation*, Perth, Australia, 2011, pp. 433-438.

[2] Y. Zhao, Y. Zhang, X. Cheng, and G. Guo. Diagnosis of rolling bearings using artificial neural network. *Journal of Tribology*, vol. 138, no. 3, pp. 031103-1–031103-10, Jul. 2016.

[3] Z. R. Yang, M. Zwolinski, C. D. Chalk, and A. C. Williams. Applying a robust heteroscedastic probabilistic neural network to analog fault detection and classification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 1, pp. 146-151, Jan. 2000.

[4] C. N. Grimaldi and F. Mariani, "OBD Engine Fault Detection Using a Neural Approach," *Society of Automotive Engineers*, 2001.

[5] M. Z. Yousaf, A. R. Singh, S. Khalid, M. Bajaj, B. H. Kumar, and I. Zaitsev, "Bayesian-optimized LSTM-DWT approach for reliable fault detection in MMC-based HVDC systems," *IEEE Transactions on Power Delivery*, vol. XX, no. YY, pp. ZZ-ZZ, 2024.

[6] A. Andrade, K. Lopes, B. Lima, and A. Maitelli. Development of a methodology using artificial neural network in the detection and diagnosis of faults for pneumatic control valves. *Sensors*, vol. 21, no. 3, p. 853, 2021.

[7] H. Je-Gal, S.-J. Lee, J.-H. Yoon, H.-S. Lee, J.-H. Yang, and S. Kim. A novel time-frequency feature fusion approach for robust fault detection in a marine main engine. *Journal of Marine Science and Engineering*, vol. 11, no. 1577, 2023.

[8] A. Aherwar. An investigation on gearbox fault detection using vibration analysis techniques: A review. *Australian Journal of Mechanical Engineering*, vol. 10, no. 2, pp. 1-13, 2012.

[9] M. Ge, Y. Liu, and H. Guo. Fault diagnosis for sheet metal stamping using support vector machine. *International Journal of Advanced Manufacturing Technology*, vol. 115, no. 1-4, pp. 467-479, 2024.

[10] V. Puig, M. Mrugalski, A. Ingimundarson, J. Quevedo, M. Witczak, and J. Korbicz. A GMDH neural network based approach to passive robust fault detection using a constraints satisfaction backward test. *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 440-445, 2005.

[11] A. Shi and X.-H. Yu. Structural damage detection using artificial neural networks and wavelet transform. *Journal of Engineering Mechanics*, vol. 126, no. 6, pp. 573–580, 2000.

[12] E. Efatinasab, M. S. Hossain, and M. A. Hannan. Bayesian neural network (BNN)-based framework for fault zone prediction in smart grids. *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 1234–1246, 2021.

[13] M. Jamil, S. K. Sharma, and R. Singh. Fault detection and classification in electrical power transmission system using artificial neural network. *Journal of Electrical Engineering and Technology*, 9(6):2152–2160, 2014.

[14] A. Yadav and Y. Dash. An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. *Advances in Artificial Neural Systems*, vol. 2014, Article ID 230382, 20 pages, 2014.

[15] V. Veerasamy, P. Manoharan, and S. Sivakumar. LSTM recurrent neural network classifier for high impedance fault detection in solar photovoltaic integrated power systems. *IEEE Access*, 9:32672–32685, 2021.

[16] S. R. Mohanty, A. K. Pradhan, and A. Routray. Cumulative-sum-based fault detection algorithm for power system relaying applications. *IEEE Transactions on Power Delivery*, 37(6):3059–3068, 2022.

[17] M. Silva, B. A. Souza, and N. S. D. Brito. Fault detection and classification in transmission lines based on wavelet transform and ANN. *IEEE Transactions on Power Delivery*, 21(4):2059–2063, Oct. 2006.

[18] C. Pothisarn and A. Ngaopitakkul. Fault classification algorithm for transmission lines utilizing a combination of Discrete Wavelet Transform (DWT) and Back-Propagation Neural Networks (BPNN). *IEEE Transactions on Power Delivery*, 25(3):1851–1859, Jul. 2010.

[19] Jun Liu, Li Zhang, and Xuewei Liu. Fault diagnosis of an analog circuit based on hierarchical DVS. *Symmetry*, 12(11):1901, 2020.

[20] Lamya Gaber, Aziza I. Hussein, and Mohammed Moness. Fault detection based on deep learning for digital VLSI circuits. *Procedia Computer Science*, 192:2987–2996, 2021. Peer-review under responsibility of the scientific committee of the 18th International Learning & Technology Conference 2021.

Biodata

Name: Deepen Khandelwal
Mobile No.: 9689492360
E-mail: deependheeraj.khandelwal2021@vitstudent.ac.in
Permanent Address: Park infinia ,maharashtra ,Pune

Name: Prateek Anand
Mobile No.: 7388581181
E-mail: aprateek.173@gmail.com
Permanent Address: 98/5 Payagipur, Sultanpur,UP

Name: Mayukh Ray
Mobile No.: 6289129075
E-mail: mayukh.ray2021@vitstudent.ac.in
Permanent Address: 435/11 jessore road, kolkata-55