A Machine Learning–Based Approach for Fault Detection in Power Systems

Aakanksha Upadhyay

PhD Scholar
Electrical Engineering
Mahakaushal University

Abstract:
Detection of faults is a fundamental need to reliable and safe operation of modern electric power systems, where protective functions are required to properly discriminate between internal faults and non-fault disturbances within very strict time limits. Traditional protection techniques are still useful in most environments, but their operation may be challenged by varying operating conditions, enhanced penetration of power-electronics interfaced resources, measurement uncertainty and the increasing variety of transient events that are similar to faults. This paper is inspired by these issues and creates a machine learning (ML)-based system to detect faults quickly and classify them into different types using synchronized electrical signals. The proposed methodology is based on the existing principles of protection and the recent developments in the field of data-driven learning, where fault analysis is developed as a supervised learning problem with short sliding windows of multi-channel voltage and current measurements. It introduces a deep neural architecture that trains on minimally processed measurements and thus eliminates the need for manually engineered features, including low detection latency, robustness to noise, fault resistance variation and operating-point variation. The methodology is placed in comparison with classical wavelet- and feature-engineering techniques, contemporary deep-learning techniques of waveform-based fault analysis, and PMU-based wide-area fault analytics. The paper also defines an evaluation plan that focuses on realistic generalization by partitioning the dataset into scenarios, multi-metric measures of dependability and security, and interpretability of the results based on explainable AI methods. The resulting paper offers a strict and repeatable roadmap of ML-based fault detection applicable both to the transmission and distribution settings, and it includes clear instructions on how data should be generated, how the model should be trained, and how it should be validated to be used in practice.

Keywords: fault detection; power system protection; machine learning; deep learning; synchrophasors; convolutional neural networks; explainable AI.

INTRODUCTION

Fault detection in the electrical power transmission systems is an important factor in keeping the grid stable and safe. Line Transmission lines are prone to short-circuit faults (line-to-ground or line-to-line), open-circuit faults, (conductor break) that are caused by high voltage transmission lines. These faults may be the lightning strikes, equipment failures, or external damages, which cause the power outage and equipment damage in case they will not be isolated in time. Conventional protection systems are based on the relays with fixed rules and thresholds that allow identifying abnormal currents or voltages. Although useful, the traditional approaches are not as fast and flexible - particularly as the grid of the modern world gets more complicated due to distributed generation and load variability. Machine learning (ML) has become a promising solution in the past few years to improve the accuracy and speed of fault detection, as it learns patterns using the data, instead of employing some fixed rules. The system of fault detection using machine learning is able to automatically detect a signature of various types of faults in real time by reducing the number of human errors and enhancing the response. The paper provides a thorough research survey and conceptual framework of the ML-based technique of the transmission line fault detection, elaborating on the different methods of ML (between shallow classifiers such as Support Vector Machines and deep learning models) and outlining a hybrid approach to combine their advantages. The aim is to describe how a data-driven model of fault detection can be more accurate, faster and flexible than the traditional ones, eventually improving the reliability of power transmission networks.

BACKGROUND: TRANSMISSION SYSTEMS FAULTS AND CONVENTIONAL DETECTION

The term faults is used in a power transmission system, where the most common abnormal electrical conditions are short-circuit between phases, or short-circuit between phases and ground, and open-circuit conditions in conductors. Shunts faults (also known as short-circuit faults) are single line-to-ground faults, line-to-line faults, double line-to-ground faults and three-phase faults. They are usually high current events which may result in voltage drop and equipment overload. Open-circuit (series) faults are those in which a line is broken or a connection is lost leading to the excessive voltage rises and the electrical pathway disruption. Faults may cascade to grid stability and as such, they must be cleared (by tripping circuit breakers) within a fraction of a second, otherwise the equipment may be damaged, or a large number of people may have outages.

The classical fault detection is based on the use of protective relays and algorithms in relays. An example is the overcurrent relays which detect whether there is more than a predetermined current in a circuit, distance relays which compute the impedance between the fault (current and voltage) to identify and locate short-circuit, and differential relays which compare currents on two sides of a line to identify a discrepancy. Such techniques involve deterministic thresholds and logic settings that are selected by engineers. Although extremely accurate at the conditions to which they are tuned, they will perform poorly when conditions in the system change (eg changing generation mix or fault impedance) or when there is noise in the measurements. Calibration of the right thresholds has the trade-offs of sensitivity and selectivity and mis-calibration can result in false trips or false detections. Furthermore, conventional relays generally evaluate a single cycle (or a small set of cycles) of a waveform data, and the high rate faults can develop before they are detected.

Fault detection using Machine Learning provides an alternative or a supplement to these classical approaches to this problem. As opposed to predetermined thresholds, ML algorithms can be trained on past or simulated fault history to notice subtle trends of current and voltage signals which can show a fault. They could possibly identify transient features in order to faster identify faults (e.g., half or quarter of a cycle), and can be retrained to suit new circumstances. ML techniques are also promising in complex conditions such as the evolving faults, high impedance faults or in systems where the traditional protection will not perform as well as the inverter-based resources. We will discuss some of the ML methods used in the detection of transmission line faults in the following sections, and then outline a conceptual detection framework using ML.

MACHINE LEARNING SOLUTIONS TO FAULT DETECTION

Machine learning refers to a collection of data-learning algorithms. In transmission line fault detection, ML models are conditioned on the feature of electrical measurements (currents, voltages, etc.) at varied operating conditions (normally and at different fault scenarios). A prediction is then made, whether there is a fault or not, with the type or location of the fault, usually depending on new measurements. In a broad sense, ML methods may be classified into either shallow (e.g. decision trees, support-vector machines) or deep (neural network architecture) learning methods. Also, scholars have investigated ensemble models and hybrid method to improve performance. The following Table 1 gives an overview of frequently used ML methods to detect faults, which have been found in surveys in recent years:

Decision Trees and Random Forests: These algorithms are able to build branching classifiers on the basis of such features as magnitudes of currents/voltages or frequency components. They are quick to implement and are able to deal with nonlinear decision boundaries. RFs are demonstrated to be robust due to the averaging of various decision results made by multiple trees (RFs). In particular, the accuracy of the Random Forest models in the classification of the transmission line faults was about 97-99% in certain studies. They have the benefit of being easily interpretable (traceable logic) and reasonably cheap to compute in real time.

Support Vector Machines (SVM): SVM is a very strong classifier which is used to establish an optimal hyperplane to classify the classes (fault vs. no-fault, or other types of faults) with the highest margin. SVMs that have kernel functions are capable of capturing complicated relations on the data. Fault detectors based on SVM have shown a very high detection rate as well as a high detection speed. It is worth noting that an SVM model was trained that only requires 1/4th of data ([?]4-5 ms at 60 Hz) on each side of a line to predict and classify faults with a result of 99.89 percent accuracy despite noisy measurements. Here the ability of SVM to be fast, with only a small amount of data through its rich feature space (in that instance, 144 features with time and frequency domain parameters) is emphasized. Comparisons have revealed that SVM usually performs better than other classifiers in accuracy in fault diagnosis when well tuned.

k-Nearest Neighbors (KNN): KNN is an instance based classifier that classifies an unknown sample as to the majority class of its nearest k neighbors in the feature space. Fault detection has been performed by matching features of incoming signals to a database of normal and faulty patterns. On large data, KNN has been shown to be computationally intensive. Research has also indicated that KNN can also be very accurate (e.g., 96-99 of the faults in certain multi-class fault scenarios) when sufficient representative data is provided. Nonetheless, KNN demands the use of efficient data management in order to satisfy real time speed demands in large environments.

Naive Bayes and Logistic Regression: Probability models such as the Naive Bayesian classifier have also been used. They apply statistical probabilities of the values of features in the cases of fault or normal condition. Such techniques are not widely used in contemporary fault studies but can be used as a reference point since they are simple. Quick binary fault detection may also be done with logistic regression, a linear classifier giving a probability of fault although typically not as strong as non-linear techniques of complex signals.

Artificial Neural Networks (ANNs): The history of neural networks in power system fault detection is very long. In a trained multilayer perceptron, it is possible to learn to map inputs (e.g. sampled voltages and currents) to outputs (faulted or healthy, fault type, etc.) by being trained on example cases. Earlier works employed ANN when features (e.g. symmetrical components or wavelet coefficients) were hand-crafted. In the modern approach the raw or less processed waveform samples are usually fed directly to an ANN. As an example, a 3-layer feed-forward neural network was designed in one of the studies that used 56 inputs (samples of three-phase currents and voltages and zero-sequence components in a 3 ms sliding window) and predicted the presence of a fault. The ANN fault detector was able to differentiate between normal and fault conditions in the 420 kV, 150 km transmission line model between normal and different types of fault using a properly selected architecture (e.g., 16 hidden neurons and sigmoid activation). ANNs have strong ability to capture non-linear relationships and are capable of generalizing, though do not need large amounts of training data and are often called a black box, and therefore their decisions are less interpretable.

Deep Learning (CNN, LSTM, etc.): Deep learning neural networks apply feature representations to data, which is frequently better than manual feature-based models. Convolutional Neural Networks (CNNs) are capable of finding spatial or temporal features, and have been applied to fault identify current or voltage signals as images or sequences. As an illustration, a 2D-CNN may be fed with time-frequency images (including scalograms of wavelet transforms) of power signals. Altaie et al. (2023) transformed three-phase current signals into images of scalograms of wavelets and made a CNN to classify faults, and the researchers found high reliability in detecting fault events on a simulated network. Similarly, 1-D CNNs may be used on the sequences of waveforms to identify disruptions. Recurrent Neural Networks such as Long Short-Term Memory (LSTM) networks are sequential data-oriented models that have been used to characterize the time dynamics of fault transients. The LSTMs are capable of learning time dependence of currents/voltages at fault onset and have been shown to be able to identify fault type with accuracy of over 99 percent in AC systems and DC systems as well. The recent research tends to integrate CNN and LSTM into hybrid models to utilize both spatial feature and temporal sequence learning. As an example, a recent method that feeds raw three-phase voltage and current measurements to a CNN-LSTM cascade (CNN to extract local features, LSTM to learn time dependence) has been found to achieve 99.98 per cent accuracy on fault type classification and analogous high accuracy on where the faulted portion of the line is on an IEEE 14-bus system. Deep models are more data intensive, more computationally intensive, but have been shown to perform exceptionally: even multi-class fault classification tasks such as those based on temporal convolutional networks (TCN), CNNs, and the transformer have achieved accuracy of 95-99.9% with architectures such as TCNs.

Adaptive Neuro-Fuzzy Inference System (ANFIS): This is a type of model of neural network learning combined with fuzzy logic reasoning. ANFIS has been used to detect and localize faults by developing fuzzy membership functions that are trained on data in a neural network. ANFIS makes an attractive proposal since it receives an interpretable output (if-then fuzzy rules) and yet still learns on the basis of examples. It has been demonstrated that ANFIS can efficiently detect faults and type in transmission systems, and that it can sometimes be used together with other methods (e.g., the wavelet analysis technique to extract features). As an example, an ANFIS neural network was successfully implemented to classify faults in a smart grid, where the neural network was used to perform the learning and fuzzy logic to provide interpretation to the classification of fault type and fault location.

Ensemble and Hybrid Models: Ensemble models involve the use of multiple models in order to enhance robustness, rather than use a single algorithm. One is RF-LSTM-KNN ensemble of Anwar et al. (2025), in which Random Forest, LSTM, and a tuned KNN are used (in a voting or stacking scheme) to combine the merits of each approach. This ensemble had 99.96% accuracy on a multi-label fault dataset, which was the best possible accuracy compared to each of the individual models (which were approximately 96-98% each). Ensemble models are more likely to generalize and cope with a wide range of scenarios of fault by reducing the vulnerability of individual classifiers. Other forms of hybrid strategies involve combining ML and signal processing: e.g. Discrete Wavelet Transform (DWT) which is used to pre-process signals and subsequently a ML classifier. In a study where DWT was coupled with an LSTM to detect DC transmission line protection, a total of 99.04 percent accuracy was obtained and this is due to the fact that wavelet could capture high-frequency transients and the LSTM had the benefit of learning sequentially. Deep/shallow-hybrid and inclusion of optimization methods to tune (such as Particle Swarm Optimization or hyperparameter Bayesian optimization) have also been found to be able to improve performance.

Table 1.*Summary of Machine Learning Techniques for Transmission Fault Detection.*

Category	Examples	Key Features & Performance
Shallow ML	Decision Tree, Random Forest, SVM, KNN, Naïve Bayes	Fast execution, needs feature engineering; RF/KNN ~96–99% accuracy in studies; SVM 99%+ with short data window.
Deep Learning	ANN (MLP), CNN, LSTM, TCN, Transformer models	Automatic feature extraction; requires large data; CNN/LSTM hybrids >99% accuracy for fault ID; TCN ~99.9% vs LSTM ~92%.
Hybrid/Ensemble	RF+LSTM+KNN Voting, DWT+ML (Wavelet+ANN or LSTM), ANFIS, etc.	Combines strengths for robustness; Ensemble reached 99.96%; Wavelet+ANN improved noise immunity.
Unsupervised	One-class SVM, K-means clustering, PCA-based anomaly detection	Detects novel faults without labels; one-class SVM had ~79–80% accuracy on anomaly detection dataset; useful when labeled data is scarce.

Note: The performance figures are indicative of specific research scenarios (often simulation-based) and may vary in real deployments.

In literature, it is clear that there is no one single approach that can be universally best, therefore, there is a tendency towards hybrid solutions and comprehensive comparative research. Also, the type of features (time-domain samples or transformed features such as frequency components or wavelet coefficients) may play an important role in the success of a particular ML approach.

ML-BASED FAULT DETECTION FRAMEWORK PROPOSAL

After considering the scenario of ML techniques, we present a conceptual model of fault detection of transmission systems by machine learning. The framework consists of data acquisition, optional feature engineering, an ML model (or ensemble), and a fault classification and identification decision logic. Figure 1 shows the general process flow of the proposed solution, including input data, through the final fault decision.

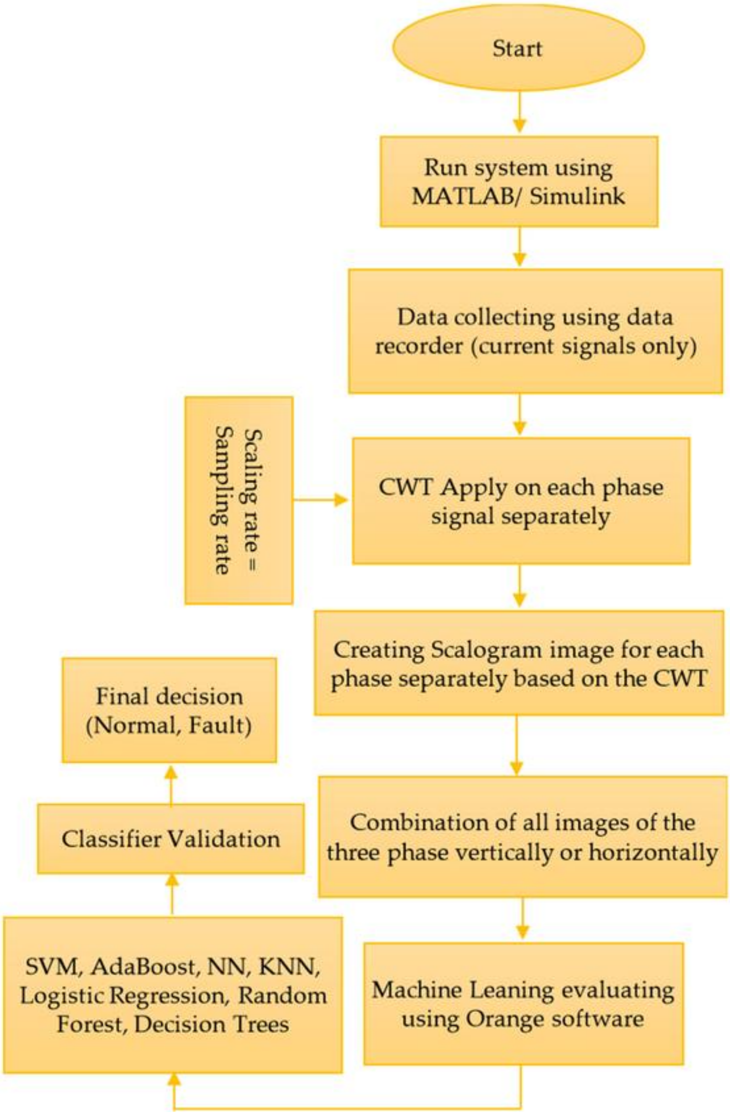

Figure 1: Example process flow of a machine learning-based fault detection system of a transmission line.

The system starts with the data acquisition (i.e., recording three-phase currents and voltages through sensors or simulation), and optional signal processing (e.g., use of a wavelet transform to obtain time-frequency information about the line currents). The central processing unit is an ML-based classifier (that may be a trained SVM or neural network, or a group of models) which takes the features or raw data and decides whether a fault exists and, potentially, the type of fault. An output of the classifier is then validated and a decision logic is provided and then an alarm or trip signal is provided should a fault be confirmed. This method uses the data of past or simulated faults to train the model, which can automatically identify them with no specification of threshold values.

ACQUISITION AND PREPROCESSING OF DATA

The initial stage is to retrieve quality data of transmission network. This is in practice accomplished by instrument transformers and phasor measurement units (PMUs) or other high-speed sensors, to measure the three-phase currents (Ia, Ib, Ic) and voltages (Va, Vb, Vc) at strategic locations (usually at each end of a transmission line). These signals are sampled in the digital version of relays with rates ranging between a

few kHz and tens of kHz. As an input to ML, a sampled data window of data around the fault inception is used. In simulation analysis, the application of simulation software such as MATLAB/Simulink is common to create fault data in diverse conditions (type of faults, location along line, fault resistivity, fault loading levels) to have a complete training data.
Preprocessing can be the removal of noise and the calculation of derived quantities. The standard preprocessing procedures are:

Normalization: Signal amplitudes in the raw signal may differ greatly, therefore features tend to be normalized. An example is a z -score normalization, which converts a feature X into z = (X -) /4) / - (X-) into a zero-mean feature with unit variance (where is the mean and is the standard deviation). Normalization makes sure that no individual feature (e.g. a phase current magnitude) dominates the learning because of difference in scale.

Feature Extraction: It is not necessary to feed raw time samples directly: one can extract informative features. There are choices of computing RMS or phase angles or symmetrical components (I0, I1, I2) or sliding window fourier or wavelet transforms. Discrete Wavelet Transform (DWT) is particularly well-liked, since faults cause the high frequency transient in the fault, which can be isolated by wavelets. Through the decomposition of every phase current signal into wavelet coefficients at various scales, the temporal patterns representing faults can be seen. It is possible for some methods to transform these coefficients into images (scalograms) and analyse them using CNNs. Others merely feed ANN or SVM with the summary statistics of coefficients as input. The features can significantly decrease the size of the input data and emphasize properties such as the sharp increase in current or the great frequency of content during a fault with the use of feature engineering.

Dimensionality Reduction: When the set of features is very large then methods such as Principal Component Analysis (PCA) may be used to dimensionality reduce the set of features but still preserve the variance. Indicatively, when the features of a fault dataset were multi-feature, PCA was employed to see the most important factors (principal components) that were used to discriminate faults. Eliminating dimensions also prevents over-fitting and reduces computing of the classifier.

Balancing and Augmentation It is common that fault datasets are imbalanced (there are much more normal samples than fault samples, or that certain types of faults are rarer than others). In order to eliminate bias, it is possible to use oversampling techniques, such as SMOTE (Synthetic Minority Over-sampling Technique) to produce artificial faults cases within minority groups. SMOTE uses an equation, like: to interpolate between other available samples in order to generate new plausible data points.

$$X_{new} = X_m + \delta\,(X_n - X_m)$$

where X_m is a minority class sample, X_n its nearest neighbor, and $\delta \in [0,1]$ a random number. This yields a balanced training set, which experiments have shown improves classification accuracy for infrequent fault cases. Data augmentation can also include adding noise or varying fault parameters to make the model more robust.
The choice of the ML model depends on the needs (speed, accuracy) and computational resources. To be able to make a conceptual discussion, we shall take two candidate methods: (a) a Support Vector Machine classifier, and (b) a Deep Neural Network (CNN-LSTM hybrid). We may train and compare them, or even bring them together in an ensemble, but we here show how each would be grown:

SVM training: Feature vectors that have been preprocessed are inputted into an SVM training program. The decision function that the SVM identifies, f(2) = w.x+b is the one that best divides classes in the feature space with maximum margin between fault and no-fault classes. In the case of multi-class fault type classification, such methods as one-vs-rest SVM or multiclass extensions are applied. The hyperparameters (such as the type of kernel - radial basis function is widespread - and the regularization parameter C) are tuned with the help of cross-validation, which may be grid search or evolutionary algorithms. One of the studies has successfully utilized Particle Swarm Optimization to tune the hyperparameters of SVM (and other model) with improved results. It is performed by the training process using a part of the data (e.g., 70-80%), and the rest is set aside as the one to be tested. To assess the quality of the model, we consider the accuracy, precision, recall, and F1-score to make sure that the model not only identifies faults, but also does it with great consistency and with the number of false alarms being minimal. Once trained, A SVM executes very fast (nearly a dot product), and therefore it is appropriate in the real-time application in a relay.

Training a Deep CNN-LSTM: In this model, raw sample sequences (or at least some form of minimal-processing such as normalized waveforms) are used in place of manual features. The model structure may be a sequence of 1-D convolutional layers (to identify local features of the waveform, such as spikes or notches) and an LSTM layer (to detect the dynamics of the patterns over time steps). An example is that a CNN having a variety of filter sizes can serve as an automatic feature extractor of each phase current and voltage. The LSTM then takes the concatenated features of all the stages over time and it learns the dynamics of the fault. The last output layer might be the type of all faults (and a no fault class) to which a softmax activation can be used to give a probability of a class. The training of such a network needs additional data and computational resources; nevertheless, libraries such as TensorFlow/PyTorch can help to train it faster with the help of GPUs. In training, methods such as early stopping and dropout regularization are used to avoid overfitting, as there is not a lot of data of faults. The cross-validation is also applied to help in making sure that the model is well generalized. Another interesting advantage of deep learning is that it can learn complex patterns of waveforms which may otherwise be difficult to encode manually - such as differentiating between an incipient fault and a temporary change of load by the form of high-frequency content.

Ensemble or Hybrid Training: Alternatively, it is possible to train several models and mix them. As an example, train a random forest, SVM and a neural network. After this make the final decision using a voting scheme (majority vote or weighted voting). Another option is stacked ensemble in which the output of certain models are fed to a meta-classifier. Anwar et al. (2025) used a stacked ensemble design known as RF-LSTM Tuned KNN, with the output of RF and LSTM as the input of a KNN, the hyperparameters of which were optimized. The ensemble was capable of capturing various facets of the fault signatures and hence almost perfect classification in their experiments. Ensembles are robust in that they require greater complexity in ensuring robustness at different conditions, though. In our conceptual model, an ensemble is defined when a single model performance plateaus or we desire our additional confidence (e.g. only say a fault when two models out of three say there is a fault).
During training, it is imperative to be tested on unobservable data. We apply different fault conditions that may not be experienced during training (different angles of fault entry, different noise levels, different system topology in case it is applicable) to ensure that the model is not merely memorizing but actually learning the nature of faults. Noisy performance is of interest, especially, as field measurements will entail noise. Unsupervised training of models such as one-class SVM or autoencoders can also be used to identify anomalies without any marked fault data, although in our case we are interested in supervised learning of faults of known categories.

DETECTION AND DECISION LOGIC IN REAL-TIME

After the validation and training, the ML model is applied into the protective scheme. This might imply the integration of the model into a digital relay or an edge computing device which has accessibility to the sensor data in a real substation. The model will keep on analyzing (or at a limited frequency) the incoming window of data and make a decision.
The grid protection operations should be incorporated with the decision logic. Commonly, when the output of the ML model points to the presence of a fault with a high degree of confidence, a trip signal will be transmitted to open circuit breakers of such line, to isolate the fault. Since false trips are extremely expensive (unnecessary outages), a check or a tie may be added. As an example, a two-criteria system may be employed such as the ML detector and a conventional relay both have to come to an agreement, or the ML output has to continue over a few milliseconds before triggering. Nevertheless, the accuracies quoted are very high (often above 99%), which offers a possibility that the ML model may serve as the main protection, and in particular fault detection (binary fault vs no fault). Fault type (e.g. what phases are

involved) classification may help system operators to be aware and fault location algorithm but are a secondary concern compared to the primary objective of rapid fault clearing.
A ML-based detector is usually capable of making decisions very quickly - the SVM with quarter-cycle data can successfully detect faults in less than 5 ms. There are multiple deep learning algorithms that may just require a single cycle (16.67 ms at 60 Hz) of information in their input, yet will be able to make a decision shortly after the information becomes available, far below the 50-100 ms latency required to make a decision at a very high speed. Computational resources are also required in real-time performance. However, with current DSPs and microprocessors in relays, even neural networks can be implemented in a few milliseconds (particularly when simplified or quantized to be used in an embedded system).
Once a fault has been detected, fault location may also be supported by the system. Most of the ML models are often multi-task: they may produce both the fault type and some distance or location estimate of the fault (a regression, or a localized classification along the line). Indicatively, the CNN-LSTM model mentioned above had independent outputs with 96% accuracy in the identification of the faulty line section of a test grid. Proper fault location assists the crews to react to repair more quickly.
Lastly, the strategy to make decisions: the system must record the occurrence, and possibly the newly stored data may be used to learn continuously (adaptive learning). There are more sophisticated ideas that the retraining or fine-tuning of ML models can be done online when additional data (such as new fault cases) is available. This would enable the detection system to gain better results over time which the traditional relays cannot achieve without manually reconfiguring.

THE BENEFITS OF THE ML-BASED APPROACH

The benefits of the machine learning process of transmission fault detection are as follows:

Enhanced Detection Accuracy: ML models have been shown to be more accurate in separating faults and normal transient in case they are trained on large datasets. This is because they are able to pick up finer patterns indicating a fault and minimises false negatives (missed faults) and false positives (false trips). As an example, ensemble and deep models with more than 99 percent accuracy on simulations suggest high chances of working reliably. Although the system can be in noisy conditions or with different system configurations, ML detectors can still remain highly accurate as they learn the underlying fault physics based on data as opposed to being based on fixed thresholds.

Speed and Sensitivity: ML-based detectors can prove to be very fast. Through their learning of a signal on the high-frequency aspects, they are able to detect a fault within a fraction of a cycle. According to the reports, some ML algorithms can decide in 4-5 ms of fault occurrence. This fast response is essential in high voltage transmission lines in which fault currents are required to be diagnosed in a short time span to prevent the destruction of equipment. In addition, ML models may be better sensitive to incipient faults - it may identify small changes that signal the presence of a fault (such as a partial discharge or high impedance fault) which standard methods would not do until the fault is more serious.

Adaptive and Robust: Adaptability is one of the notable assets. When the system varies (e.g. change in network topology, incorporation of renewable generation which varies fault currents), an ML model can be retrained. It implies that protection schemes are able to support the changing grid without the need to modify hardware - simply re-train the model with updated training data. Also, non-linearities and interactions among a variety of inputs (e.g., faults occurring simultaneously or faults occurring in power swings) can be addressed by ML models such as neural networks compared to more linear threshold-based approaches. It has also been found that ML methods are able to cope with the variable such as different angles of fault inception or different fault resistance by learning to be strong under these conditions.

Multi-task Capabilities: It is possible that a single ML model can do fault detection, classification and even localization. Conventional relays usually require individual functions or algorithms per (such as a distance relay to locate it, overcurrent to determine it). Conversely, a properly developed neural network is capable of producing several pieces of information on the fault. This can be used to streamline the protection scheme and make it consistent (the same data drives all decisions).

Minimization of Human Fault: With the pattern recognition being automated, ML eliminates the need to use protection engineers who are required to forecast all the potential system states and program the relay to match those states. The model adapts to data, even the cases that can be too complicated to obtain an analytical approach. This can minimize the error of humans in the settings and coordination of protection. The ML model will automatically take care of such cases as long as the training data is representative.
One must agree that most of these benefits have been demonstrated in simulation or controlled experiments. They must be validated in the real world since there is a plethora of field conditions they have to be tested in.

CHALLENGES AND CONTEMPLATIONS

Even though it holds promise, ML-based fault detection in transmission systems is associated with challenges that should be overcome to implement it in practice:

Requirement Quality Training Data: ML models can be as good as the data that they are trained on. It may be challenging to obtain a full set of fault events of a transmission system. Malfunctions in actual grids are comparatively uncommon, and every occasion may possess its peculiarities. There are some historical fault recordings of utilities though ML may need thousands of examples. The computations usually are simulated data (modeled by using software tools on standard test systems such as IEEE 9-bus or 118-bus), although models that have been trained on simulations should be thoroughly checked against real data to prevent simulation bias. It is essential to make the training set (different types of faults, locations, system conditions) diverse in order to guarantee the generalization of the model.

Complexity and Interpretability Model: Deep learning models containing millions of parameters are effectively black boxes. Explainability is significant in a protection context, the operators must be sure of why a relay reached a decision. Explainable AI (XAI) research is examining methods of explaining the decision of neural networks. Other models such as decision trees or rule-based systems are less complicated to interpret and may not be as accurate. The balance or some interpretation can be achieved by a balance or the application of hybrid neuro-fuzzy systems (such as ANFIS) with the help of linguistic rules. Also, protection regulatory criteria might demand deterministic conduct, which is more difficult to ensure with an acquired model.

Execution speed: This is an important parameter. Whereas most studies imply the rapid detection, the device application (relay) may contain limited CPU or memory. A large CNN-LSTM model may need techniques such as model compression, neural network quantization, or executing it on specialized hardware (FPGAs or real-time processors), to execute on a few milliseconds. The positive side is that there are a few ML models that are very lightweight (SVM or small neural nets). In the case of larger models, it is necessary to assure that they clear faults within the worst-case time requirements (usually 1-2 cycles transmission systems).

Reliability and Security: An ML-based relay should be very reliable and safe against failure. Issues of concern are the model behavior when the inputs do not follow the training distribution - e.g. extreme noise, sensor faults, or malicious data (cybersecurity aspects). Fail safe mechanisms should be included. Indicatively, in case the ML output is uncertain or the system has poor confidence offered by the model, the system might revert to a traditional protection scheme. Other researchers apply an ensemble in which an anomaly detection model observes the input; in case of something extremely unusual (that is not similar to training data) it is flagged by the system instead of blindly using the ML decision. Testing (Hardware-in-loop testing with actual relay equipment) After testing, field deployment is required before deployment.

Integration with Existing Protection Schemes: Existing power systems have well coordinated protection schemes (between transmission lines, feeders, transformers etc.). The addition of an ML-based element should not disrupt the coordination. An example is that one line ML relay can be super sensitive and therefore my relay may be over-functioning in comparison with other relays on the line. Thus, it may be necessary to set the sensitivity of the ML detector or coordinate its activity (it may be necessary to introduce some deliberate delay), in order to recreate the existing selectivity of zone protections. In the long term, additional ML relays would also have the ability to coordinate with each other in more dynamic manners, which is a research and standardization area.

CONCLUSION

Machine learning is a new promising paradigm of fault detection in power transmission systems. Using previous and simulated fault data, the ML-based strategies are capable of detecting faults accurately and swiftly regardless of complicated grid conditions, which confuse conventional protection strategies. This paper has discussed the potential use of many ML methods, including classical algorithms such as the SVM and decision trees on the one hand, and deep learning-based models, on the other, to locate and identify transmission line faults. We also suggested a conceptual architecture that considers the extraction of features (e.g. the wavelet-based analysis) and a hybrid CNN-LSTM classifier, which can be able to recognize fault near-real time. This solution is expected to deliver better reliability (reduced number of missed faults or missed trips), speed (measured in milliseconds) and flexibility (model updated with new data) to the high-voltage transmission system.

The research survey emphasizes the fact that already some studies have obtained great outcomes (usually over 99% accuracy) in test conditions. Ensemble ML model can be effective in reducing uncertainty as well as enhancing confidence in fault decisions, and hybrid methods uses the advantages of various algorithms in addressing the diversity of fault signature. As power grids are becoming more and more complex, as renewable penetration increases and electronics become more advanced, an AI/ML capability to continuously learn and adapt will become priceless.

Nonetheless, the strategies of transferring these techniques to practice will have to resolve the problems associated with data collection, model testing, and interface with the old systems. The next steps in future work would be to generate large datasets on faults (possibly by collaborating with industries and sharing), construct explainable and certifiable ML models to protect and run pilot experiments in real substations to see large-scale fault detection in practice. Furthermore, it could be of great benefit to the grid resilience when the strategy is expanded to not only identify faults but also help predictive maintenance (based on equipment condition data) also.

Finally, machine learning-based fault detection solution is also highly promising and will presumably be a major contributor to the protection of smart grids in the next generation. It adds the smartness of a data-driven learning process to the pace of digital relays and takes us a step further to a more reliable and intelligent electrical grid that can independently detect and respond to faults with accuracy and speed.

REFERENCES:

1. N. Q. Minh, N. T. Khiem, and V. H. Giang, “Fault classification and localization in power transmission line based on machine learning and combined CNN-LSTM models,” *Energy Reports*, vol. 12, pp. 5610–5622, Dec. 2024, doi: 10.1016/j.egyr.2024.11.061.
2. Altaie, A. S., Majeed, A. A., Abderrahim, M., & Alkhazraji, A. (2023). *Fault Detection on Power Transmission Line Based on Wavelet Transform and Scalogram Image Analysis*. **Energies, 16**(23), 7914. https://doi.org/10.3390/en16237914
3. T. Anwar, C. Mu, M. Z. Yousaf, W. Khan, S. Khalid, A. O. Hourani, and I. Zaitsev, “Robust fault detection and classification in power transmission lines via ensemble machine learning models,” *Scientific Reports*, vol. 15, no. 1, p. 2549, Jan. 2025, doi:10.1038/s41598-025-86554-2.
4. Goni, M. O. F., Nahiduzzaman, M., Anower, M. S., Rahman, M. M., Islam, M. R., Ahsan, M., Haider, J., & Shahjalal, M. (2023). *Fast and accurate fault detection and classification in transmission lines using extreme learning machine*. **e-Prime – Advances in Electrical Engineering, Electronics and Energy, 3**, 100107. https://doi.org/10.1016/j.prime.2023.100107
5. Shakiba, F. M., Azizi, S. M., Zhou, M., & Abusorrah, A. (2023). *Application of machine learning methods in fault detection and classification of power transmission lines: a survey*. **Artificial Intelligence Review, 56**(7), 5799–5836. https://doi.org/10.1007/s10462-022-10296-0

6. Venkata, P., Pandya, V., Vala, K., & Sant, A. V. (2022). *Support vector machine for fast fault detection and classification in modern power systems using quarter cycle data*. **Energy Reports, 8**(Suppl. 2), 92–98. https://doi.org/10.1016/j.egyr.2022.10.279