Series DC Arc Fault Detection Based on Ensemble Machine Learning

Vu Le, *Student Member, IEEE*, Xiu Yao, *Member, IEEE*, Chad Miller, and Bang-Hung Tsao, *Senior Member, IEEE*

***Abstract*—Series dc arc fault creates a fire hazard and negative impacts on the distribution bus if not detected and isolated quickly. However, the detection of a series arc fault is challenging due to the low fault current, lack of zero-crossing, and the erratic behavior of arc discharge based on different power electronic loads and controllers in modern power applications. This article presents a practical and versatile series dc arc fault detection method based on ensemble machine learning (EML) algorithms. A buck converter constant power load (CPL) and a boost converter CPL are designed and built to study the different arc fault behaviors and generate training data for the machine learning algorithms. A set of time domain features is extracted from the experimental data and analyzed using the feature importance attribute. An adaptive normalization function then processed the features to mitigate false positive classification caused by load changes. A two-step algorithm is proposed to recognize the arc fault in different load types. Various EML algorithms and the associated hyperparameters are benchmarked to select the most accurate hyperparameters for a detection algorithm for low-cost hardware implementation. Finally, the detection algorithm's effectiveness is verified with CPL testbed.**

***Index Terms*—Arc fault detection, constant power load (CPL), dc arc fault, ensemble machine learning (EML).**

I. Introduction

THE increasing of electronic loads in a modern electrical system using local dc distribution, such as electric vehicles, ships, and more electric aircraft (MEA) [1]–[3], have led to the growing use of converters/inverters and electrical drives. Subsequently, these complex electrical systems create challenges to maintain the system's performance and safety, one of which is series dc arc faults. The problem is not only limited to vehicle-based systems; the growth of photovoltaic (PV) systems that are installed on the residential houses and industries also poses a potential to develop arc faults [4]. The temperature at the center of the arc can range from 5000 to 15 000 °C [5]. If not detected and interrupted in time, dc arc faults could lead to fire hazards and potentially damage the main bus and the neighboring loads.

Unlike arc faults in an ac system, dc arc faults lack zero-crossing in the current, which inherently is more challenging to detect and extinguish. The electromagnetic interference noises from converter/inverter and electrical drive switching in dc systems create even more complications in arc fault detection. Many studies have explored the statistical characteristics of dc arc faults to design detection algorithms [6]–[8], such as mean, variance, max–min difference, and entropy. In [6], a current entropy is calculated, which is used as a signature to differentiate between an arc fault state and no arc. The entropy calculation can reveal the degree of disorder and signal intrinsic behavior. These methods require less sampling frequency and produce short detection latency time, but their performance is easily influenced by the noise level of the surrounding environment.

Extensive studies of dc arc fault signatures use different frequency bands from wavelet transformation [7] and [9]. This mathematical transformation can decompose the original time-domain signal to multiple time-domain signals that contain different frequency bands, which essentially were used to isolate high-frequency noises. The current amplitude differences and rms values of target frequency bands were found to be effective arc fault signatures in [9]. The detection algorithm was implemented using a digital signal processing board in [10]. A relative magnitude-based arc fault detection algorithm incorporated with adaptive threshold values is proposed in [11]. The method requires a sampling frequency of 250 kS/s and a window length of 2 ms. This method is quite capable of detecting arc with very low latency of less than 16 ms. However, the load current magnitude can be easily influenced by the large disturbance of electronic components. The advantages of these pattern recognition algorithms that use "threshold" are fast and straightforward to implement. However, the threshold value is a major limitation for most systems, especially in an electrical system with a large number of power electronic loads that generate different electromagnetic noise magnitudes. The arc fault behaviors also depend on the system configuration, which makes it more challenging to decide the threshold values. Therefore, a more robust detection algorithm is still needed in an electrical system that has more than one type in multiple operating loads.

Manuscript received April 30, 2019; revised August 27, 2019 and December 4, 2019; accepted January 9, 2020. Date of publication January 27, 2020; date of current version April 22, 2020. This work was supported by the Air Force Research Laboratory (AFRL) Intelligent Power System (IPS) program, under Contract FA8650-12-D-2224 (TO0002). Distribution A: Approved for public release. Recommended for publication by Associate Editor Prof. W. Cao *(Corresponding author: Vu Le.)*

Vu Le and Xiu Yao are with the Department of Electrical Engineering, University at Buffalo - The State University of New York, Buffalo, NY 14260-1660 USA (e-mail: vule@buffalo.edu; xiuyao@buffalo.edu).

Chad Miller is with the Electrical Engineering, Wright State University, Dayton, OH 45435 USA (e-mail: Chad.N.Miller@gmail.com).

Bang-Hung Tsao is with the University of Dayton, Dayton, OH 45469 USA (e-mail: btsao1@udayton.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPEL.2020.2969561

As an alternative, many studies have explored the use of machine learning (ML) as a possible tool for an intelligent fault detection algorithm. Besides effective pattern recognition capability, ML can be robust and adaptive in systems with versatile operating conditions. For example, benchmarks of various classification machine learning algorithms were analyzed as a disturbance and cyber-attack discriminator [12]. There were multiple fault scenarios, including short-circuit fault, line maintenance, and three cyber-attack scenarios. A supervised ML called support vector machine (SVM) was utilized for a fault detection technique in an ac system using features from 3-phases current and voltage in [13]. This detection algorithm focuses on a multiphase fault, single-phase fault, line-to-line, and line-to-ground fault. In [14], a fault detection method uses a supervised learning algorithm named DecisionTree (DT) with the utilization of time domain reflectometry as a data acquisition device for PV systems. A semisupervised learning algorithm using graph-based was used to detect and classify line-line and open-circuit faults from the nominal condition in PV arrays with high accuracy [15]. These studies provide promising evidence that ML algorithms can discriminate between nominal and multiple fault scenarios. However, the main focus was only on ac systems and parallel faults. Series arc fault detection in the dc power system was studied using hidden Markov model in [16]. SVM was implemented to detect series dc arc fault in PV systems [17] and [18]. However, these methods were also implemented wavelet transformation for signal feature extraction that required high sampling frequency, which induces high computation power. In [19], an artificial neural network was developed for series dc arc fault detection in spacecraft. This method incorporated fast Fourier transformation to extract arc fault signatures and resulted in an unsatisfactory in the amount of error.

This article proposes an advanced technique of ML called ensemble machine learning (EML) as a series dc arc fault detection algorithm. The difference between EML and the other conventional types of supervised ML algorithm is the use of multiple classifier outputs to form a single decision. This process eliminates extreme cases to reduce bias and variance in the classification process, which can improve the predictive performance [20]. As a result, EML does not suffer from a major drawback called overfitting compared to conventional algorithms. In practice, overfitting is a significant source of error in real-time detection. EML provides the robustness and adaptability in the broad spectrum of real-world applications. Besides electrical fault detection, an ensemble learning using incremental SVM as a base-learner was used for fault diagnosis of the roller bearings [21]. An EML algorithm called RandomForest (RF) was implemented to detect multiple faults in wireless sensor networks [22]. The algorithm demonstrates the superior fault classification and the versatility in all six fault scenarios: offset, gain, stuck-at, out-of-bounds, spike, and data-loss faults, compared to SVM and neural network algorithms. Another example, an RF algorithm was implemented as an industrial process fault detection [23]. Feature importance was used to identify the useful features to enhance the algorithm compared to less reliable features. EML performance seems to be more robust and reliable compared to the conventional ML algorithm, where multiple faults are presented in the system.

Although many approaches using ML for series dc arc fault have been explored, these ML algorithms were selected arbitrarily along with the default settings. Without ML algorithm customization, it becomes challenging for the detection algorithm in more complex power electronic systems. Besides, most developed methods were tested with a simulated arc fault detection environment. Provided that, the classification capability of ML algorithms required further investigation. Specifically, the sensitivity of each algorithm can be examined by using built-in attributes known as hyperparameters for more accurate and reliable arc fault detection. Customized hyperparameters allow the algorithm to adapt to different applications. An experimental arc fault detecting verification using complex power electronic systems in real-time is essential to determine the method's reliability in practice.

Based on the experimental setup, the arc fault is initiated using two copper rods control by a stepper motor. The series dc arc fault behaviors under different current levels, motor speed, and load resistances were studied and analyzed for detection. This article is aimed to provide an accurate and versatile arc fault detection using an EML algorithm. The sensitivity of the EML algorithm performance based on hyperparameters, such as the number of base learners, features, and training data size, were evaluated. This process uses an exhaustive search method that systematically selects the appropriate hyperparameters for arc fault detection. The EML algorithm's performance with selected hyperparameters was further benchmarked, including accuracy, precision, and recall scores as performance metrics. An adaptive normalization function was implemented to prevent false positive classification during nominal load changes. A two-step algorithm was proposed to recognize the arc fault in different load types. The experimental setup, fault generator unit, and the power electronics load-emulated real application scenarios, which is verified by testing the arc fault detection algorithm using a low-cost microcontroller.

II. Data Acquisition and Analysis

There are two types of arc fault according to the location: series, and parallel connecting to the loads. DC series arc faults are commonly initiated because of chemical, electrical, or mechanical deterioration of wires or interconnections. It is considered to be more dangerous due to low fault current, which makes it harder for the circuit breaker to respond on time. This circumstance can reduce the safety and performance of the distribution systems if not detected and isolated in time. Hence, this article focuses only on series arc fault detection.

A. Experimental Setup

The testbed is established using a 10 kW Keysight dc power supply, Opal-RT OP4510, arc generator, and power electronic circuits that can be configured as a buck converter or a boost converter, as shown in Fig. 1. The buck converter and the boost converter, along with resistors, are controlled to form a constant power load (CPL) in this article. The Opal-RT OP4510 is a

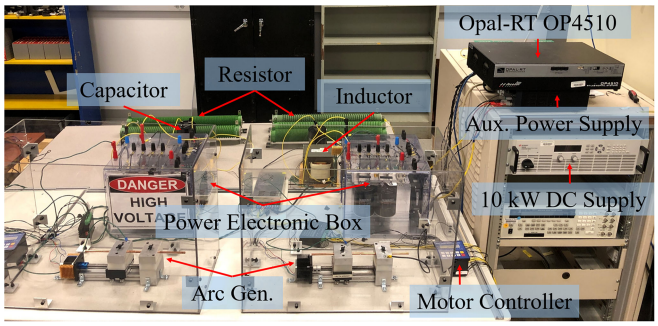

Fig. 1. Experimental setup.

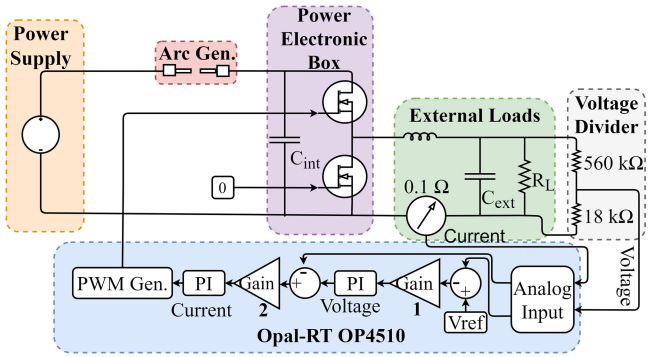

Fig. 2. One-line diagram of the buck converter CPL.

TABLE I
EXPERIMENTAL TEST PARAMETERS

Input Current	Buck Converter Load	Boost Converter Load	Motor Speed	Rods Gap
5 A 10 A 15 A 20 A	29.6 Ω 14.8 Ω 9.87 Ω 7.4 Ω	90.7 Ω 45.4 Ω 30 Ω 22.7 Ω	3 mm/s 6 mm/s 9 mm/s	1.5 mm

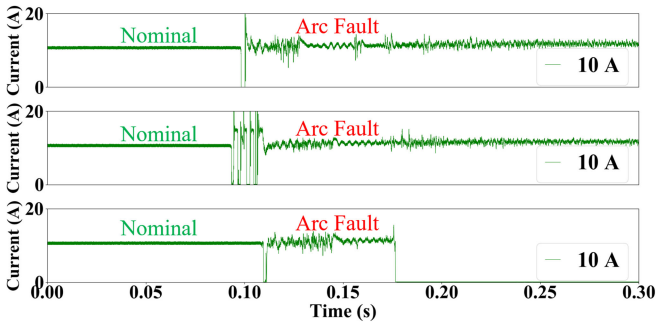

Fig. 3. Buck converter current signals at 10 A.

real-time hardware-in-the-loop simulator platform, but it was implemented solely as a closed-loop controller to control the switching frequency and duty cycle of the CPL output. DC source voltage of 270 V is selected to emulate a dc distribution system of more electric aircraft [24]–[26]. A lower dc voltage bus 28 V_{dc} is also used for lower power and avionic loads via step-down dc/dc converter on small and large aircrafts [27]. Whereas another dc/dc converter can be used to step-up the voltage from 115 V_{ac} to 270 V_{dc} after an ac/dc inverter [28]. The step-ups and step-downs in dc voltage can be realized using buck converter and boost converter.

The experiment parameters are outlined in Table I. The current ranging from 5 to 20 A is tested to emulate the operational load range of certain electromechanical actuators [29]. The arc faults are created by separating two copper rods that were initially in contact with a stepper motor, and therefore, separation speed is also taken into consideration. The faster speed can create a large current mutation that does damage to the wire insulation [30]. Moreover, different speed tests emulate wires in real-world scenarios based on material, stress, and breakage force. Three different speeds were tested in this study ranged from 3 to 9 mm/s. At least five tests were conducted for each testing condition, and all the tests were conducted at room temperature and normal pressure.

The power electronic box consists of six insulated-gate bipolar transistor (IGBT) modules, SKM50GB123D, with freewheeling power diodes connected in series. There are also four built-in capacitors connected in series and parallel with a voltage rating of 750 V_{dc} and a current rating of 30 A. An external capacitor with a value of 600 μF with a current rating of 30 A to minimize the ripple output current. The load resistances are configured differently for different current levels, as opposed to changing the duty ratio to achieve different current levels. This step prevents the individual load resistor from exceeding the rated power of 2.5 kW. All the component values in both CPLs are summarized in Table II.

Fig. 2 illustrates the buck converter type CPL with a closed-loop controller. The voltage and current were being used with two proportional-intergal (PI) controllers to control the duty cycle of the IGBT. A voltage divider with two resistors of 560 and 18 kΩ was used to extract the load voltage for the first controller loop. As a result, the sensor gain of the voltage loop is approximately 32. A 0.1 Ω current sensing resistor with a current rating of 30 A was used to measure the inductor current for the second loop. Therefore, the sensor gain of the current loop is 10, as shown in Table III. The current and voltage signals are connected to the analog ports of the Opal-RT OP4510, where PI controllers are implemented using MATLAB Simulink. Then, the controller gains and PI values were tuned to output an appropriate pulsewidth modulation (PWM) waveform from the controller digital ports that are connected to the IGBTs. The boost converter CPL has a similar closed-loop setup with the current sensor measures the inductor current at the input side.

B. Arc Current Characteristics at Various Conditions

The Tektronix Mixed Signal Oscilloscope MSO56 was used to record all the voltage and current data with a sampling rate of 500 kS/s. The arc current was measured using the TCP0030 A current probe with a current rating of 30 A placed in between the power supply and the arc generator. The differential voltage probe $TMDP0200$ was used to measure voltages. The recording length was 4 s.

TABLE II
EXPERIMENTAL SETUP PARAMETERS

Label	V_{in}(V)	V_{out}(V)	I_{in}(A)	C_{int}(μF)	L_{ext}(mH)	C_{ext}(μF)	$R_L(\Omega)$	I_{out}(A)
Buck Converter	270	200	5, 10, 15, 20	1100	10	600	30.2, 14.7, 9.9, 7.3	6.7, 13.5, 20.3, 27
Boost Converter	270	330	5, 10, 15, 20	1100	10	600	90.75, 45.2, 30.2, 22	4.1, 8.2, 12.3, 16.4

TABLE III
BUCK CONVERTER AND BOOST CONVERTER PI CONTROLLER PARAMETERS

		Sensor Gain		Controller Gain		Voltage PI		Current PI	
Converter	V_{ref}	*Voltage*	*Current*	$Gain_1$	$Gain_2$	*P Value*	*I Value*	*P Value*	*I Value*
Buck Converter	200	32	10	1	1/20	1	35	0.1	40
Boost Converter	330	32	10	1/16	1/15	1	35	0.7	0.5

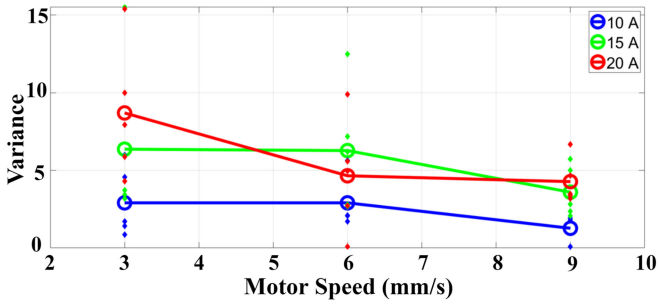

Fig. 4. First 10 ms of arc fault current variance in buck converter.

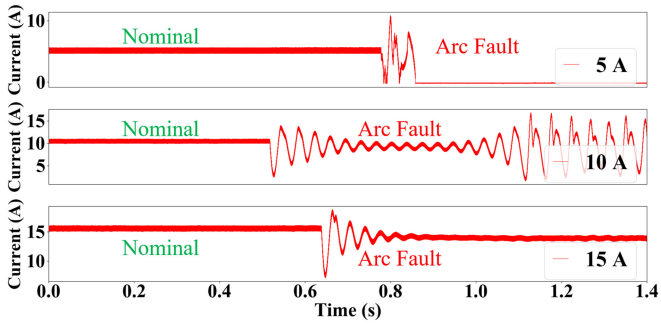

Fig. 5. Boost converter current signals at 5, 10, and 15 A.

1) Constant Power Load 1, Buck Converter: Fig. 3 presents several nominal and arc fault operation of arc current signals of a buck converter at 10 A input current. As the copper rods pulled apart, the gap creates a momentary open circuit before arc discharge can be formed. It can be seen that the initial arc fault behavior of the buck converter can be different at the same operational current.

The large amplitude spikes during the initial transient in the current are known as an electrical spark. These electrical sparks could be more or less visible and audible at each test within the first 10 ms of the arc fault initiation. Fig. 4 illustrates the relationship within five tests of a particular arc fault generator's motor speed and first 10 ms of a particular current level variance. It can be seen that as the motor speed increased from 3 to 6 mm/s, the average variance of current is large for 10 and 15 A, respectively, and the trend remained the same. This indicates that sparks are more likely to occur within slower motor speeds and lower current levels. However, when the motor speed increased to 9 mm/s, the average variance decreased, which indicates that an electrical spark is less likely to occur. At 20 A, there is a significant decrease from 3 to 6 mm/s, which implies that higher current levels have a large effect on spark formation. At the higher current level, a higher density of electrons in the electrode is able to achieve sufficient energy to overcome the potential barrier and speed up the gap ionization process, which creates sustained arc immediately after the copper rods pulled apart. The average current variance at 9 mm/s remained similar to the average of 6 mm/s.

The electrical sparks produce abnormal current amplitudes, which could be a potential precursor for arc fault detection [31]. However, these sparks can be varied in the initiation, which challenging to determined the appropriate number for fixed threshold arc fault detection methods.

2) Constant Power Load 2, Boost Converter: Fig. 5 shows several current behaviors of the boost converter during an arc fault. Since the input inductor prevents the abrupt change in current, the fault initiation produces a much slower current change compared to the buck converter. Fig. 5 also shows that the current level can have a significant impact on fault response as well. At 5 A, the initial spark caused by the gap breakdown is likely to extinguish quickly without turning into a sustained arc. It is important to note that with a purely resistive load, a current as low as 3 A is sufficient to generate a sustained arc [10]. Having a CPL with passive components mitigates the arc forming capability of a gap, which occurred with buck converter-based CPL as well. At 10 A, the arc voltage induced an input voltage drop for the CPL, which leads the CPL controller to oscillation. The previous theoretical analysis and simulation study has shown that parallel fault can lead to controller instability in dc distribution system [32]. This is the first time that CPL instability is observed experimentally with a series dc arc fault. However, at a higher current, 15 A, the same fault is not sufficient to disturb the stability of the controller, where the sustained arc fault current stabilized quickly after the initial transition.

These test results further demonstrate the critical need for more intelligent fault detection techniques for modern dc power systems. With a power system highly penetrated with power converters and electronic loads, fault response is highly subjective to the load type, current level, and controller performance, which is inherently different from that in a traditional power system

TABLE IV
EXPERIMENTAL DATASETS FOR ML SIMULATION

Label	Testing Datasets	Training Datasets	Total Datasets	# of Trials
Buck Converter	10000	$10-3000$	19106	50
Boost Converter	10000	$10-3000$	36420	50

dominated by linear components. The different behaviors of both CPLs with either the same or different current levels show that the conventional "threshold" method will likely fail in a system with various types of loads and operating conditions.

C. Arc Current Data Pool for Machine Learning

A large amount of buck converter and boost converter data was obtained, including nominal and arc fault conditions. The data was recorded at 500 kS/s; it was then resampled to 80 kS/s by using a MATLAB function to match with the current sensor that is part of the detector hardware. The resampling process involves both operations of data interpolation and decimation to achieve the desired sampling frequency. One of the crucial benefits of resampling is the prevention of aliases by applying a low-pass filter. In practice, aliasing creates distortion and inaccurate data of the final signal. The recorded data was further divided into smaller datasets of approximately 2 ms to train and test ML algorithms.

Table IV illustrates the simulation setup on training and testing data for ML algorithms. A random function was utilized to prevent bias during the datasets selection process. For instance, 10 000 different testing datasets were randomly selected at every trial, which is equivalent to 52.34% of the total number of datasets for buck converter and 27.46% of the total number for the boost converter. The same process was also applied for training datasets selection. All testing and training datasets were split equally between nominal and arc conditions. To further reduce the bias in datasets selection, the training and testing process was simulated 50 times.

III. MACHINE LEARNING

A. Selected Features

Before formulating a reliable machine learning (ML) algorithm, feature engineering is a crucial step in the ML algorithm pipeline. A feature is an individual measurable property of the data and an essential part of the machine learning implementation. A collection of features can describe the raw data, but not completely represent the data itself. Therefore, having informative and independent features can produce a more effective ML algorithm. Nevertheless, having too many features could degrade the classification, also known as overfitting. There are many techniques to extract useful features from raw data such as fast Fourier transform [19], short Fourier transformation [33], and wavelet transformation [10], [16]. However, the drawbacks are a high sampling frequency, which requires extensive computational power. In practice, these factors could affect accuracy and delay time when considering real-time arc fault detection. In contrast, time-domain features can be performed with a lower sampling frequency, which reduces the computation time significantly. Five time-domain features were utilized for the final detection algorithm: average, median, variance, rms, and the difference between the maximum and minimum value. These five features were analyzed rigorously to prevent having features that overlapped between nominal and arc fault in the feature space.

B. Ensemble Machine Learning

ML algorithm can be categorized into three types: supervised, semisupervised, and unsupervised learning. Supervised ML is one of the most prominent branches that use for classification. The method requires prior knowledge to perform a particular classification on new data. The teaching function uses training errors to gain classification ability. This closed-loop feedback can effectively increase ML classification accuracy. As a result, only supervised ML algorithms were used as the base-learner for EML in this article. Built on that, EML can further improve the performance by increasing the adaptability and robustness while reducing the common vulnerabilities such as overfitting.

For the past decades, EML has received much attention within the machine learning community. EML algorithms have been proven to be very capable and highly versatile in a broad spectrum of real-world applications, such as object detection and tracking, image recognition, and data mining [34]–[36]. EML algorithm uses the combination of multiple classifiers to reduce the classification bias and variance effectively. As computational power advances, different ways of combining the base-learner outputs are exploited to produce a robust and quick decision. There are three popular approaches to combine classifiers: bootstrap aggregating (bagging), boosting, and stacking. Each of the techniques has advantages and disadvantages in specific applications. For example, the bagging technique creates several classifiers with fixed bias, then combines their outputs by averaging, to reduce variance in classification. This technique would be a well fit for applications where features of the data have high variance and low bias; whereas the boosting technique is more reliable when the features of the data have low variance and high bias.

1) Bootstrap Aggregating (Bagging): Bagging is one of the simplest yet effective ensemble algorithms. Bagging technique exploits the independence of the parallel generated base-learners. This method combines outputs from the desired number of models with the same algorithm and then performs averaging to determine the final decision. The combination approach allows variance reduction in the final predictive performance. Fig. 6 shows the general procedure and structure of the bagging technique where subset (n) is a set of randomly sampled datasets from the original training data pool. Model (n) is the independent base model, where linear classifiers such as DT, linear SVM, and a single layer of perceptrons can be good candidates. Each model is trained with a subset from the training datasets pool, where each sample dataset in that subset can be selected more than once during the training process. This data selection process is

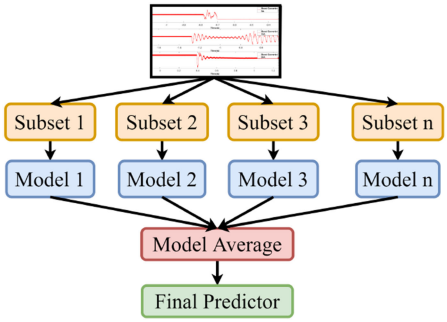

Fig. 6. Bagging ensemble technique flow diagram.

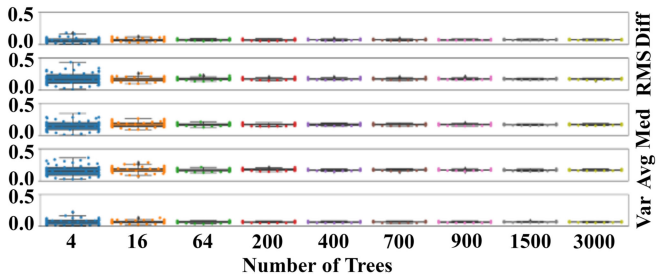

Fig. 7. RF features importance using 800 datasets.

Fig. 8. RF algorithm training time.

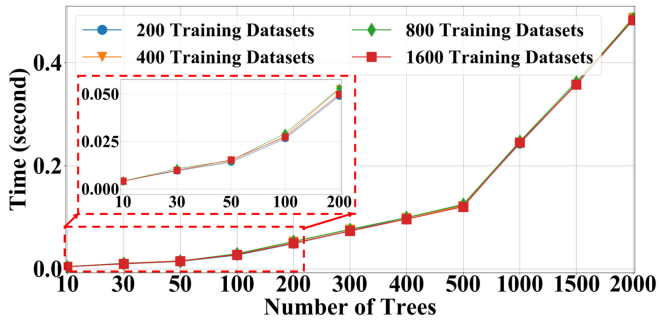

Fig. 9. RF algorithm predicting time.

also known as sampling with replacement. The final step is to aggregate all the predictions by averaging.

DT has been studied to be one of the most suitable base learners than other linear learners for more extended bagging techniques, such as RF [37] and ExtraTrees (ET) [38] algorithms. Both methods' objective is to decorrelate further any relationship between all the models, which improves the final decision. RF algorithm integrates an additional function that randomly selects a smaller subset of training data from the training data pool to reduce bias, then determines the number of splits needed. Meanwhile, ET selects a random number of splits for calculated features from all training data pool.

To optimize RF and ET algorithm for better performance, hyperparameters such as number of models, number of splits, number of leaf-nodes, training data size, and number of features need to be further tuned. One of the unique attributes that RF and ET provide is the relative importance of the calculated features. It is critical to know which features are the most useful to represent the data and potentially determine other hyperparameters. Fig. 7 illustrates the relative importance of the five features from the RF algorithm using 800 training datasets.

It should be noted that the sum of all features importance is one. The information gain is used at each split of the DT to determine the relative importance between different features. Once the feature importance of a single tree (model) is computed, the same process is applied to all models, and all outputs are then averaged. As can be seen from Fig. 7, variance (var) has the lowest importance value, which is considered to be less effective than other features. The number of trees can mitigate the effect of extreme cases, which further reduces the variance of each feature importance. For example, the box of each feature gets smaller and more refined as the number of trees increases. This indicates that this process can alleviate a problem such as overfitting.

Fig. 8 illustrates the RF algorithm training time ranging from milliseconds up to 13 seconds as the number of trees increases. The number of training datasets can also affect training time proportionally. The training time should be carefully considered when adaptive training is implemented. All ML algorithms are trained offline in this article.

The predicting time has a very similar trend as the training time. The time to predict every 2 ms window can be done in ms time frame, as shown in Fig. 9. However, it is critical to reduce the predicting time as much as possible and still output an accurate result. The predicting time increases significantly with a higher number of trees. For example, it takes approximately 4 ms to predict for a model with 10 trees, 30 ms for a model with 100 trees, and 50 ms for a model with 200 trees.

2) Boosting: In contrast, the boosting technique requires the dependence of the generated sequentially weak base-learners. A weak base-learner is an algorithm that uses the least additional parameters that are just enough to perform classification better than chances. For example, DT could be one of the weak base-leaner by using only one tree (if-else statement); it is also called stump DT. Whereas a strong learner uses optimized hyperparameters that can yield classifiers with small error.

Consequently, the weak base-learners (model) in the boosting technique are always going to learn something from the training datasets, but it not always accurate. The error of the previous model can be used to correct the next model's classification ability. A calculated weight is also added to each model based on the performance error. The final step is to fit all the sequence

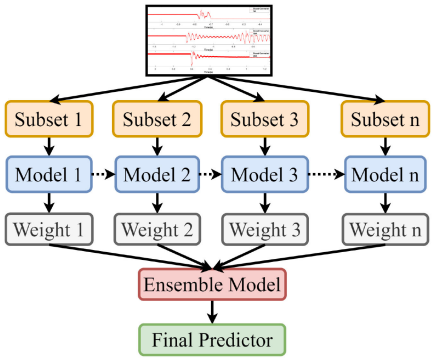

Fig. 10. Boosting ensemble technique flow diagram.

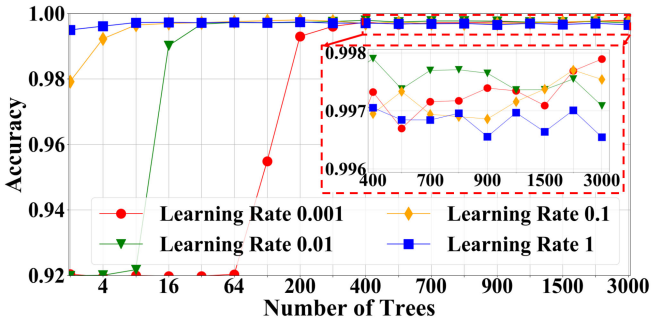

Fig. 11. Adaboost accuracy based on learning rate.

weak base-learners and its weight to form a strong learner, as shown in Fig. 10.

One of the most widely used forms of the boosting technique is Adaptive Boosting (AdaBoost). Similar to the bagging technique, there are also hyperparameters in the boosting technique, such as the number of models, splits, and features that can be tuned to a specific application. Since the weak base-learners train sequentially, there is an additional hyperparameter that dictates how fast the algorithm corrects the error at each model (base-learner) known as the learning rate; it ranges from 0 to 1. Generally, the lower learning rate requires more trees to complete the final model's classification ability, as shown in Fig. 11. At the learning rate of 1, the Adaboost algorithm does not require as many trees as the learning rate of 0.001 to output a high accuracy performance. However, as the number of trees increases to 3000, the learning rate of 0.001 outputs a slightly higher accuracy score.

Another version of the boosting technique is called gradient boosting (GB). This method uses optimized arbitrary differentiable loss functions to improve the data generalization. Both boosting algorithms can either be used for regression and classification. Each model in the boosting technique is trained with a random subset of training data with replacement. The comparison between Adaboost and GB algorithms is presented in the later part of this section.

3) Stacking: Lastly, stacking is a technique that uses the combination of different strong base-learners information to generate a new model via a meta-classifier. Unlike the bagging and boosting techniques that use a single base-leaner algorithm

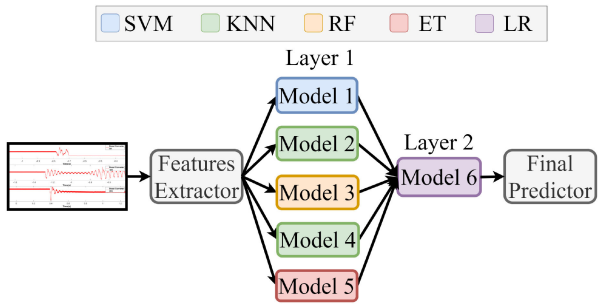

Fig. 12. Stacking ensemble flow diagram.

throughout the training process. Where the training pool can be small, bagging and boosting algorithms might output a model with the same set of hypotheses that are having the same accuracy on selected training data. This can be problematic when predicting new unknown data. As the computational power becomes more available, the stacking ensemble can find the best prediction boundary instead of gets stuck in the wrong local minimum [39]. The stacking technique uses various types of the ML algorithm as base-learner. For instance, a supervised learning SVM is model 1; KNN is model 2. Even the ensemble ML algorithm could be a part of the stacking technique itself, such as RF (Model 3) and ET algorithm (Model 5), as shown in Fig. 12. In addition, a second layer is mandatory for the stacking technique called meta-learner (Model 6); it is typically a linear algorithm.

The stacking ensemble train and predict functions are similar to the bagging technique. The final prediction is the average (majority voting) from all chosen models. However, the stacking technique can incorporate appropriate different base learners where it performs best, and discard any base learners where it performs poorly. As a result, it produces a more generalized classification, which creates less bias. For this reason, the stacking technique is most effective when the base learners are significantly different.

Furthermore, the stacking technique can have as many layers to optimize classification capability. This technique resembles the similarity to a neural network algorithm. However, a neural network algorithm is sensitive to the initial training datasets, which produces a high variance final predictor. It trains the algorithm as a whole via a stochastic algorithm. Whereas the stacking technique can be more flexible, it can train the individual model (node) then combine the predictions from all the models.

C. Model Evaluation and Performance

To examine the effectiveness and reliability of all ensemble techniques for arc fault detection, each algorithm has to be well-tuned to optimize series dc arc fault detection capability. In addition, tradeoffs need to be made to minimize the limitations, including computational time, amount of required labeled data, the complexity of the models, and maximize the classification performance of the models, which can be correlated to the ML model's hyperparameters. Gridsearch is one of the methods that can be used to find the optimal hyperparameters of a model by exhaustively sweep through all the hyperparameter combinations,

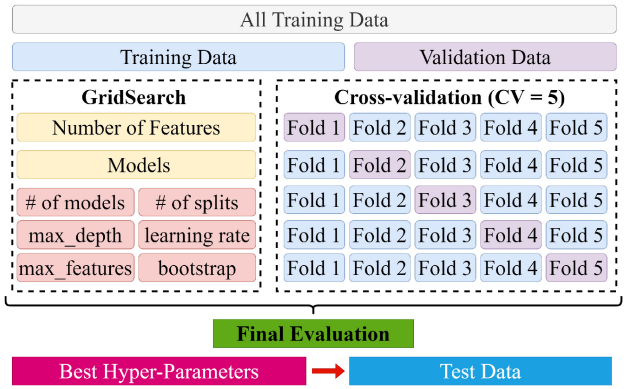

Fig. 13. Gridsearch with cross-validation.

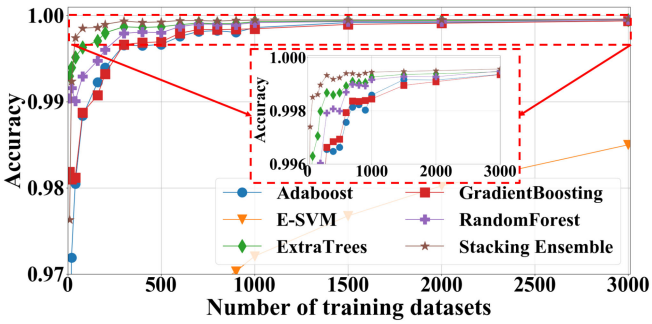

Fig. 14. Accuracy scores.

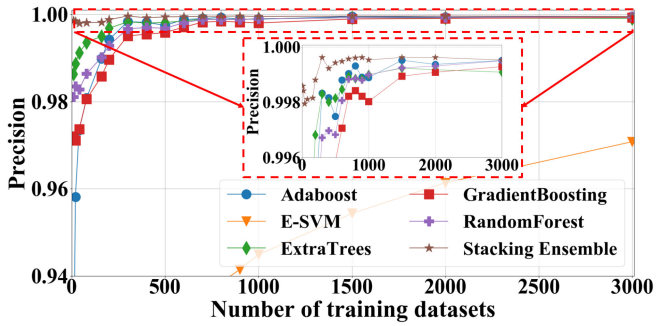

Fig. 15. Precision scores.

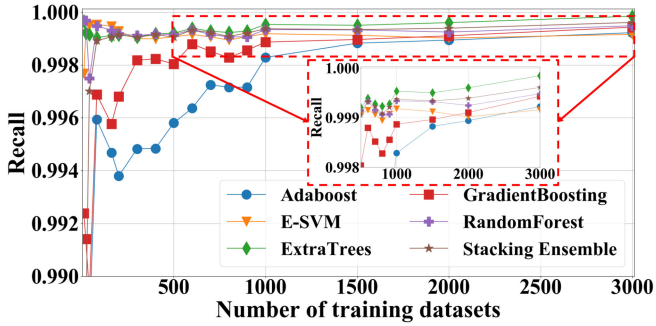

Fig. 16. Recall scores.

as shown in Fig. 13. In addition, cross-validation is following to ensure the test data does not leak over to train data, which prevents overfitting. Once the optimal hyperparameters found, it can be input to a corresponding ML model to evaluate the classification performance further.

Different metric scores, accuracy, precision, and recall, are presented to evaluate and analyze all EML algorithms. These scores range from zero to one. The ideal algorithm should perform with all three scores as close to one as possible. A number of different training datasets ranging from 20 to 3000 were tested with 10 000 datasets. The scores were calculated based on the actual and predicted results.

The accuracy score is defined as the percentage of correct classification decisions that were made. Fig. 14 presents the accuracy score of all the ML algorithms from three methods bagging, boosting, and stacking. It can be seen that the bagging technique that uses SVM as a base learner (E-SVM) has the lowest accuracy score. The E-SVM only achieved 98.5% accuracy at 3000 training datasets. Whereas all the ensemble techniques with the DT algorithm as base-learner outperformed the E-SVM even with the lower number of training datasets. The ET and the RF algorithms achieved up to 99.9% with 1000 training datasets. The Adaboost and the GB from the boosting technique also achieved a high accuracy score of approximately 99.8% at 1000 training datasets. Lastly, the stacking ensemble (SE) technique formed by three different algorithms, K-Neighbor (KNN), RF, Gaussian Naive Bayes (GNB), and Logic Regression as a meta-classifier achieved the highest accuracy score, approximately 99.95%.

While the accuracy score provides the overall ML algorithm performance, precision and recall scores provide a further understanding of how algorithms produce errors. Ideally, the higher the score in precision and recall, the better. Mathematically, a precision score is the number of true positives divided by the number of true positives plus false positives. False positives are the cases that were identified as positive but are actually negative. Fig. 15 illustrates the precision scores of all three ensemble techniques in both converters for further comparison. This provides a sense of false positives performance when predicting arc fault. The SE algorithm has the most reliable performance over other ML algorithms.

Mathematically, a recall score is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are the samples that were identified positive by the classifier that are actually positive (correct); whereas false negatives are the samples that identified as negatives that actually positive (incorrect). Fig. 16 illustrates the recall score of all three ensemble techniques. It can be seen as the number of training datasets increases beyond 1000, ET has the highest recall score. However, the stacking technique performed comparably to ET across all training datasets. This indicates the superior performance of correctly identified true positives of most ML algorithms besides the Adaboost algorithm with a slightly lower performance.

Even though the performance from most EML algorithms does not vary much from each other in all three scores, the SE algorithm outputs the highest accuracy score from 20 to 3000 training datasets. The precision and recall scores from the SE algorithm are also comparable to other algorithms. As a result,

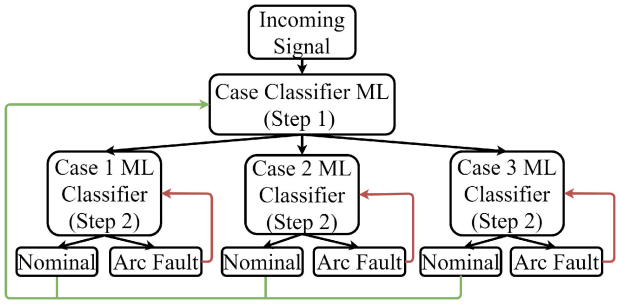

Fig. 17. Two-step detection flow diagram.

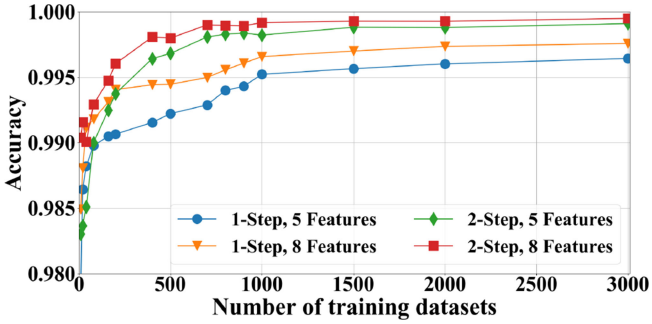

Fig. 18. One-step and two-step detection accuracy.

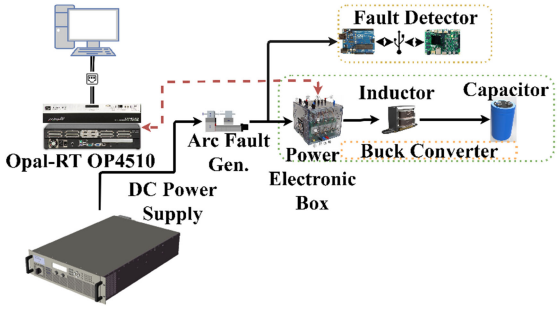

Fig. 19. Experimental setup with fault detection unit diagram.

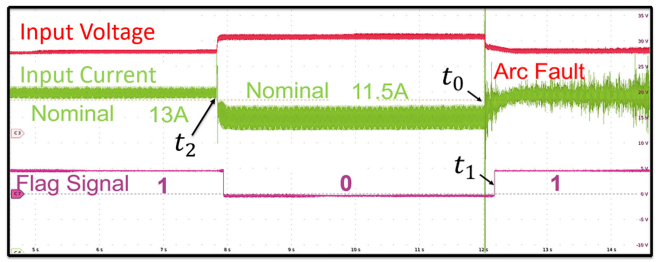

Fig. 20. Arc fault detection output using ML algorithm.

SE provides a sense of better classification capability over other algorithms.

D. Two-Step Detection Algorithm Structure

One of the critical functions in this article is a two-step detection algorithm, which enables the detection algorithm to detect arc fault in different load types. Fig. 17 illustrates the flow diagram of the arc fault detection in different load types sequentially.

This function can reduce the confusion of arc fault in different load types; it is divided into two steps. The first step classifies the load type based on the nominal operation of the CPLs, while the second step identifies the nominal and arc fault conditions. The continuous load type detection (step 1) is only triggered when the load is operating in a nominal condition. Otherwise, the load type detection is paused, and the arc fault detection continues with the last detected load type. Two types of power electronic loads are presented—a buck converter and a boost converter. However, the function is expected to work with other load types as well.

Fig. 18 illustrates the accuracy of two different detection algorithm structures, one-step and two-step using the RF algorithm. In addition, two different sets of features that including five and eight features were used. It can be seen that more useful features are essential to perform better in detection as eight features trained with SE algorithm has higher accuracy score in both one-step and two-step. More importantly, a more advanced detection structure is required for better performance. The accuracy scores from all the algorithms are increased with a noticeable trend, as the training datasets increased from 10 to 1000. However, it is stagnated beyond 1000 training datasets. The two-step algorithm with either five or eight features sets outperformed the one-step algorithm. This indicates that after an optimal number of training datasets, the accuracy score does not improve significantly unless an advanced detection structure is used.

IV. Experimental Verification

Experiments were conducted to verify the detection algorithm using a SE algorithm. Fig. 1 shows the experimental setup. In addition, the fault detection unit is placed between the arc fault generator and the power electronic box, as shown in Fig. 19. The current sensor bandwidth is 80 kHz. The analog-to-digital conversion capability of the detection unit is also 80 kHz. Hence, the maximum sampling frequency of the detection unit is 80 kHz. The detection window size for real-time arc fault detection is 2 ms.

A. Experiment Results Without Adaptive Normalization

Fig. 20 illustrates the fault detection using the ML algorithm. It should be noted that this detection result was obtained using real-time hardware with a buck converter as CPL. The input voltage (red) operates at 200 V. The input current signal (green) starts at nominal operation; then, the arc is initiated by pulling two conducting copper rods apart at t_0. The arc fault detection algorithm's flag signal (maroon) remained (0) during the CPL nominal operation and outputted a high signal (1) at t_1 as soon as an arc fault was detected. The latency time between arc fault initiation and detection was approximately 80 ms.

In addition, Fig. 20 also shows a current load change at t_2. The input voltage stepped up caused the input current to decrease

from 13 to 11.5 A. It should be noted that the flag signal was high (1) during the CPL operation at 13 A, even though it was operating in nominal condition. This error is known as false positive classification, and it occurred because the ML algorithm was only trained with 11.5 A data. Therefore, as the current stepped down to 11.5 A, the ML detection algorithm was able to recognize it as a nominal operation. Provided that, it is critical to prevent the detection algorithm from misclassified normal load changes from an arc fault. An adaptive normalization is proposed to combat this problem in the following Section.

B. Adaptive Normalization Function

Normalization is a standard prerequisite process for many machine learning algorithms. It allows all features in the feature space to maintain the same scale, which ensures no single feature has an advantage over the other features. With the experimental testbed operates in a wide range of current in real-time, the normalization method must be able to normalize a targeted current signal properly to produce consistent scaled features. However, a fixed normalization method is not capable of normalizing real-time signals as its characteristics always influenced by the different current levels. It is also easily generated inconsistent scaled features by a large spike during load changes and noises. To address this problem, an adaptive normalization is employed to transform a wide range of operating currents to the same scale in real-time. The proposed adaptive normalization algorithm calculates the average of the first 15 consecutive data windows defined as a normalization factor. Next, the new incoming signal windows are divided by the normalization factor to achieve normalization. If there is a change in the nominal current that is larger than a preset sensitivity threshold of 0.3 A, then the algorithm recollects another 15 windows to update the normalization factor. The adaptive normalization function only takes approximately 50 μs based on the internal clock of the microcontroller. The number of the windows and the sensitivity threshold were tested extensively to ensure the final detection algorithm's stability and low computation time. With a small number of windows, the normalization factor can be easily influenced by noises or transients during start-up and load changes. In contrast, a large number of windows requires higher computation power, which prolongs the detection latency time.

Fig. 21 illustrates the result of the proposed adaptive normalization. A nominal load change followed by an arc fault was tested. The raw data of input current is shown in the top plot; whereas the result of the adaptive normalization is shown in the bottom plot. The different input current levels and the arc fault were scaled accordingly. This normalization enables the detection algorithm to work with a wide range of current levels.

C. Stacking Ensemble Algorithm

A real-time arc fault detection unit was used to verify the performance of the SE algorithm that incorporated with the adaptive normalization in a two-step detection structure. It should be noted that the detection results were obtained using real-time

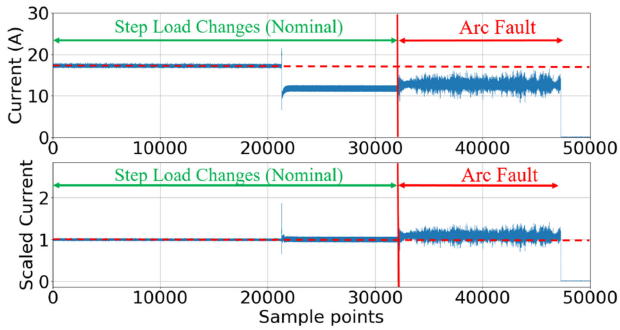

Fig. 21. (Top) Raw data of input current. (Bottom) Normalized input current.

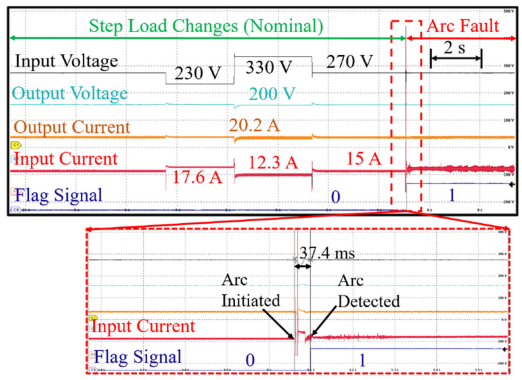

Fig. 22. Detection output using SE algorithm in buck converter with input current of 15 A.

hardware with a buck converter and a boost converter as the CPLs.

The input current signal (red) is started at the nominal operation, and the arc fault is then initiated by pulling the two conducting copper rods apart in the buck converter, as shown in Fig. 22. The flag signal (blue) indicates nominal and arc fault conditions. The detection algorithm was able to ignore the nominal step load changes from 270, 230, and then to 330 V, then back to 270 V as the flag signal remained low (0). The detection algorithm immediately outputted a high signal (1) when the arc fault was initiated. Moreover, the flag signal was kept high as the arc fault was sustained; this action indicated that the detection algorithm was able to not only detect the onset of the arc fault but also to recognize the sustained arc fault until it was extinguished. The delay time between fault initiation and detection is approximately 37.4 ms.

The detection algorithm was also tested with the boost converter testbed, as shown in Fig. 23. The input current (orange) and the flag signal (green) were measured. During a step change in current from 15 to 17.6 A, the detection algorithm flag signal remained low (0). As soon as the arc fault was initiated, the flag signal quickly changed to high (1) within 37 ms. The detection algorithm can also detect arc fault at 10 A with a latency time of approximately 69 ms, as shown in Fig. 24. The results highlighted the ability to recognize accurately multiple arc fault behaviors within the same converter.

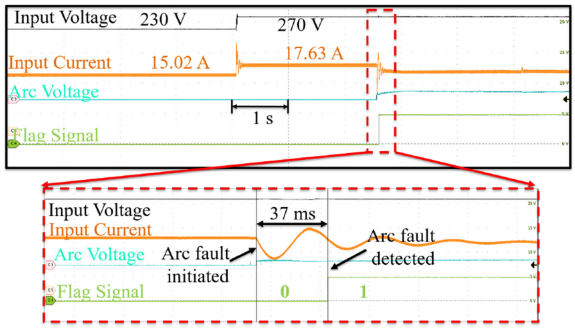

Fig. 23. Detection output using SE algorithm in boost converter with input current of 15 A.

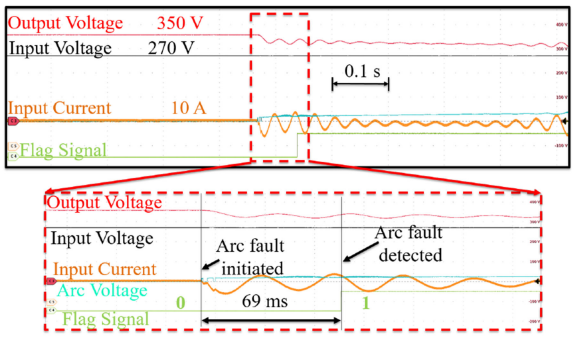

Fig. 24. Detection output using SE algorithm in boost converter with input current of 10 A.

D. DC Start-Up Versus Arc Fault

A dc start-up creates a high overshoot that resembles an initial arc fault transient. This problem can cause false positive classification and mistrigger of the flag signal. To test the reliability of the proposed detection algorithm, a dc start-up test was conducted using a buck converter CPL. The capacitive load of the buck converter was increased from 600 μF to 2.8 mF to emulate a high overshoot start-up. Fig. 25(a) illustrates the completed operation of the CPL: off state, start-up, nominal, and arc fault. The power source was initially off until turned on at time t_0. The voltage started to rise and stabilize at 270 V. Then, the CPL was initiated and started to draw current from the power source at time t_1. With a high capacitive load, the peak of the input current transient start-up reached 8.4 A, as shown in Fig. 25(b). This overshoot resulted in a 3.4 A over the nominal operating current of 5 A.

After approximately 4.5 s nominally operated, an arc fault was initiated at time t_3 and detected at time t_4, as shown in Fig. 25(c). Despite the similarity of the transient between start-up and arc fault, the detection algorithm was able to ignore the transient start-up and recognize the arc fault, as expected. This result essentially verifies the capability of the EML algorithm to differentiate nominal transients from arc faults.

E. Two Parallel CPLs Test

Existing literature on dc arc faults has shown that the high frequency components of the arc current noise can travel to the adjacent branches and cause mistrigger of the detector [10], [40], and [32]. It has been suggested that a system-level master controller is required to detect the faulted branch and prevent mistrigger of the unfaulted branches. While fault localization is beyond the scope of this article, a preliminary experiment has been designed to demonstrate the compatibility of the proposed arc detection algorithm with a potential system-level master controller.

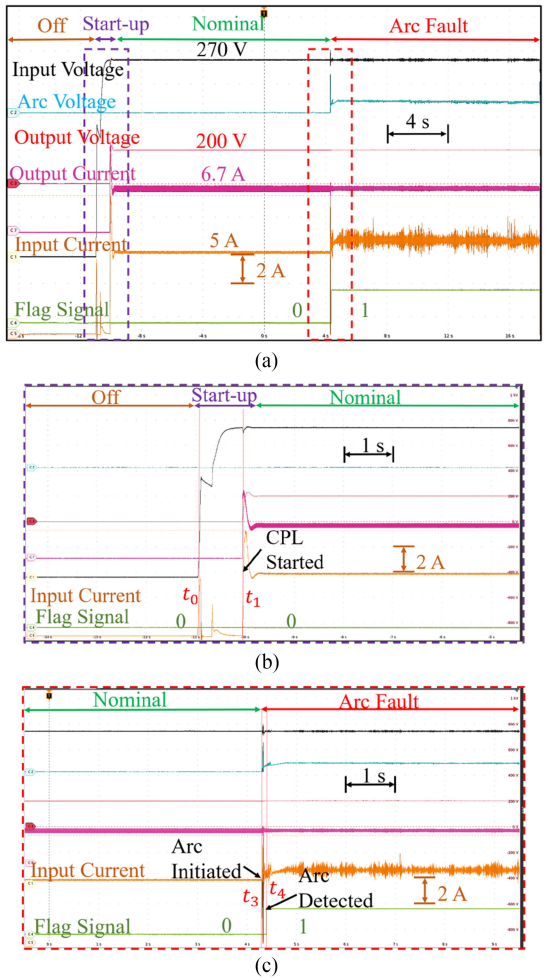

Fig. 25. Start-up and arc fault detection result. (a) Buck converter CPL operation. (b) Start-up detection. (c) Arc fault detection.

Fig. 26 shows the test setup with two buck converters with the same input current of 13.5 A and an output voltage of 200 V. Each branch is equipped with its fault detector unit. A series arc fault is introduced at the input stage of CPL 1. The goal of this set of tests is to verify that the detector at CPL 1 branch can detect arc fault timely while the detector at the unfaulted CPL 2 branch unaffected by the high frequency components of the arc and remained unchanged.

To emulate a simple master controller in the hardware setup, a communication channel is established to exchange flag signals between the two detector units. The detector unit that detects a fault, it then sends a signal to disable the other detector until the fault is extinguished.

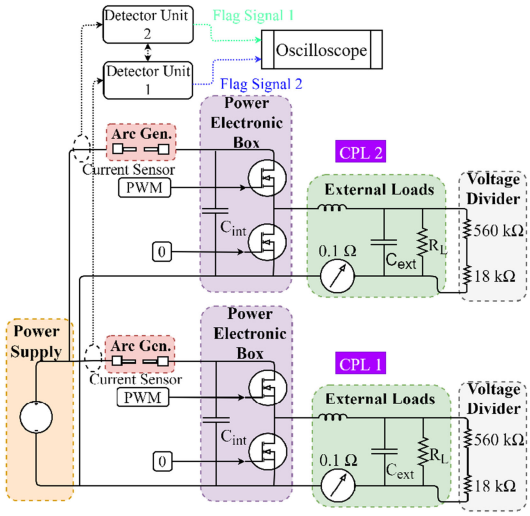

Fig. 26. Two buck converters with fault detector units setup diagram.

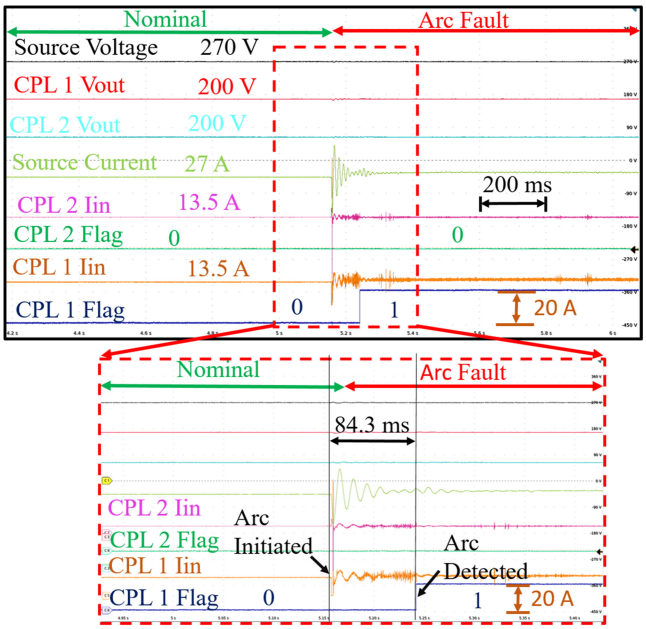

Fig. 27. Two buck converters detection result.

The test result is shown in Fig. 27. An arc fault is initiated in CPL 1, and the detector unit 1 detected the arc fault first, as can be seen from the flag signal (blue). At the same time, the flag signal of the detector unit 2 remained low.

This preliminary test is only to show the compatibility of the proposed detection method with such a master controller. It laid out the foundation for a study of arc fault in a system-level using a master controller that communicates with all the local fault detectors. With the capability of coordinating all the fault detectors, the master controller can localize the arc fault, and isolate it with proper dc circuit breaker or solid-state power controller. It is worth noting that a more sophisticated master detector would be needed to cope with a versatile system-level fault localization problem and reduce mistrigger. In a broader perspective, the proposed EML algorithm can work with a higher-level system, such as ships, MEA, and dc microgrid.

F. Discussion

The detection latency time can be broken down into three parts: pre-processing time, predicting time, and communication time between the analog-digital converter and the microcontroller. Using the microcontroller internal clock, the pre-processing time and predicting time are measured to be around 5.5 and 15.5 ms, respectively. The communication time can take up to tens of milliseconds. As shown from the experimental results, the typical detection latency time is ranged from 37 to 69 ms. The time varies due to the randomness of the arc fault signal. More specifically, if one time window failed to capture the arc current features, it would take one or more time window to detect the fault, and thus prolong the detection time. Therefore, the detection latency time can be affected by the effectiveness of the features, the detection algorithm, as well as the hardware.

The experimental results have shown shorter detection time compared to most of the existing techniques in [41] and [42]. The detection time is also well below the fault interruption time outlined in UL 1699B, leaving sufficient time for fault interrupting devices to act on the detection result.

V. Conclusion

An experimental testbed, including arc generator, buck converter CPL, and boost converter CPL, was designed and built to study the behavior of arc fault in different power electronics loads. A set of effective features was extracted from the experimental data to train the machine learning algorithms. These features were validated to be computationally inexpensive based on the computation time from the microcontroller. An EML's attribute, feature importance, was used to verify the effectiveness of the features. They are further processed using an adaptive normalization to eliminate false positive caused by load changes.

A series dc arc fault detection method based on the EML algorithm was proposed. Compared to conventional threshold methods and other machine learning algorithms, the implementation of the EML algorithm enabled the versatility, reliability of arc fault detection in different load types. Three ensemble techniques, including bagging, boosting, and stacking, were studied. Performance evaluation of various EML algorithms for arc fault detection was established using accuracy, precision, and recall scores. Each EML algorithm sensitivity was further analyzed to optimize arc fault classification capability by exploring the hyperparameters, the training time, and the predicting time. The stacking ensemble algorithm maintains superior performance compared to other EML algorithms.

Finally, the prototype arc fault detector was realized using low-cost hardware. The stacking ensemble algorithm was implemented on the hardware and tested on the experimental arc fault testbed with a range of 37 to 69 ms in detection latency time for both buck converter and boost converter. The detection method was also tested with a high transient start-up and an arc fault in one CPL that connected in parallel with another CPL to verify the capability of EML. The proposed method has achieved remarkable versatility, reliability, and low detection latency time.

ACKNOWLEDGMENT

Distribution A: Approved for public release. Distribution unlimited. Case number: 88ABW-2019-5848.

REFERENCES

[1] A. Emadi, Y. J. Lee, and K. Rajashekara, "Power electronics and motor drives in electric, hybrid electric, and plug-in hybrid electric vehicles," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2237–2245, Jun. 2008.

[2] Y. Ji and J. Bals, "Multi-model based fault detection for the power system of more electric aircraft," in *Proc. IEEE 7th Asian Control Conf.*, 2009, pp. 93–98.

[3] J. Rosero, J. Ortega, E. Aldabas, and L. Romeral, "Moving towards a more electric aircraft," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 3, pp. 3–9, Mar. 2007.

[4] Z. Wang and R. S. Balog, "Arc fault and flash detection in dc photovoltaic arrays using wavelets," in *Proc. IEEE 39th Photovolt. Specialists Conf.,* 2013, pp. 1619–1624.

[5] G. D. Gregory and G. W. Scott, "The arc-fault circuit interrupter, an emerging product," in *Proc. IEEE Ind. Commercial Power Syst. Tech. Conf. Rec. Papers Presented Annu. Meeting (Cat. No. 98CH36202)*, 1998, pp. 48–55.

[6] N. L. Georgijevic, M. V. Jankovic, S. Srdic, and Z. Radakovic, "The detection of series arc fault in photovoltaic systems based on the arc current entropy," *IEEE Trans. Power Electron.*, vol. 31, no. 8, pp. 5917–5930, Aug. 2016.

[7] Z. Wang, S. McConnell, R. S. Balog, and J. Johnson, "Arc fault signal detection-fourier transformation vs. wavelet decomposition techniques using synthesized data," in *Proc. IEEE 40th Photovolt. Specialist Conf.*, 2014, pp. 3239–3244.

[8] M. Rabla, E. Tisserand, P. Schweitzer, and J. Lezama, "Arc fault analysis and localisation by cross-correlation in 270 V dc," in *Proc. IEEE 59th Holm Conf. Elect. Contacts* , 2013, pp. 1–6.

[9] X. Yao, L. Herrera, Y. Huang, and J. Wang, "The detection of dc arc fault: Experimental study and fault recognition," in *Proc. IEEE 27th Annu. Appl. Power Electron. Conf. Expo.*, 2012, pp. 1720–1727.

[10] X. Yao, L. Herrera, S. Ji, K. Zou, and J. Wang, "Characteristic study and time-domain discrete-wavelet-transform based hybrid detection of series dc arc faults," *IEEE Trans. Power Electron.*, vol. 29, no. 6, pp. 3103–3115, Jun. 2014.

[11] S. Chae, J. Park, and S. Oh, "Series dc arc fault detection algorithm for dc microgrids using relative magnitude comparison," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 4, pp. 1270–1278, Dec. 2016.

[12] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst.*, 2014, pp. 1–8.

[13] N. S. Coleman, C. Schegan, and K. N. Miu, "A study of power distribution system fault classification with machine learning techniques," in *Proc. IEEE North Amer. Power Symp.*, 2015, pp. 1–6.

[14] Y. Zhao, L. Yang, B. Lehman, J.-F. de Palma, J. Mosesian, and R. Lyons, "Decision tree-based fault detection and classification in solar photovoltaic arrays," in *Proc. 27th Annu. IEEE Appl. Power Electron. Conf. Expo.*, 2012, pp. 93–99.

[15] Y. Zhao, R. Ball, J. Mosesian, J.-F. de Palma, and B. Lehman, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *IEEE Trans. Power Electron.*, vol. 30, no. 5, pp. 2848–2858, May 2015.

[16] R. Telford, S. Galloway, B. Stephen, and I. Elders, "Diagnosis of series dc arc faults: A machine learning approach," *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1598–1609, Aug. 2017.

[17] K. Xia, S. He, Y. Tan, Q. Jiang, J. Xu, and W. Yu, "Wavelet packet and support vector machine analysis of series dc arc fault detection in photovoltaic system," *IEEJ Trans. Elect. Electron. Eng.*, vol. 14, no. 2, pp. 192–200, 2019.

[18] Z. Wang and R. S. Balog, "Arc fault and flash detection in photovoltaic systems using wavelet transform and support vector machines," in *Proc. IEEE 43rd Photovolt. Specialists Conf.*, 2016, pp. 3275–3280.

[19] J. A. Momoh and R. Button, "Design and analysis of aerospace dc arcing faults using fast fourier transformation and artificial neural network," in *Proc. IEEE Power Eng. Soc. General Meeting*, 2003, vol. 2, pp. 788–793.

[20] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. Berlin, Germany: Springer, 2012.

[21] X. Zhang, B. Wang, and X. Chen, "Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine," *Knowl.-Based Syst.*, vol. 89, pp. 56–85, 2015.

[22] Z. Noshad *et al.*, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, 2019, Art. no. 1568.

[23] Z. Chai and C. Zhao, "Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification," *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 54–66, Jan. 2020.

[24] K. Emadi and M. Ehsani, "Aircraft power systems: Technology, state of the art, and future trends," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 15, no. 1, pp. 28–32, Jan. 2000.

[25] W. Cao, B. C. Mecrow, G. J. Atkinson, J. W. Bennett, and D. J. Atkinson, "Overview of electric motor technologies used for more electric aircraft (MEA)," *IEEE Trans. Ind. Electron.*, vol. 59, no. 9, pp. 3523–3531, Sep. 2012.

[26] D. Izquierdo, R. Azcona, F. L. Del Cerro, C. Fernandez, and B. Delicado, "Electrical power distribution system (HV270DC), for application in more electric aircraft," in *Proc. 25th Annu. IEEE Appl. Power Electron. Conf. Expo.*, 2010, pp. 1300–1305.

[27] P. Wheeler, "Technology for the more and all electric aircraft of the future," in *Proc. IEEE Int. Conf. Automatica*, 2016, pp. 1–5.

[28] B. Rahrovi and M. Ehsani, "A review of the more electric aircraft power electronics," in *Proc. IEEE Texas Power Energy Conf.*, 2019, pp. 1–6.

[29] S. C. Jensen, G. D. Jenney, and D. Dawson, "Flight test experience with an electromechanical actuator on the F-18 systems research aircraft," in *Proc. 19th Digit. Avionics Syst. Conf.*, 2000, vol. 1, pp. 2E3–1.

[30] L. Yuan, J. Shengchang, W. Jin, Y. Xiu, and Z. Yeye, "Study on characteristics and detection of dc arc fault in power electronics system," in *Proc. IEEE Int. Conf. Condition Monitoring Diagnosis*, 2012, pp. 1043–1046.

[31] L. Yue, V. Le, Z. Yang, and X. Yao, "A novel series arc fault detection method using sparks in dc microgrids with buck converter interface," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2018, pp. 492–496.

[32] L. Herrera, W. Zhang, and J. Wang, "Stability analysis and controller design of dc microgrids with constant power loads," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 881–888, Mar. 2017.

[33] S. Chen and X. Li, "PV series arc fault recognition under different working conditions with joint detection method," in *Proc. IEEE 62nd Holm Conf. Elect. Contacts*, 2016, pp. 25–32.

[34] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1311–1318.

[35] S. K. Zhou, J. Park, B. Georgescu, D. Comaniciu, C. Simopoulos, and J. Otsuki, "Image-based multiclass boosting and echocardiographic view classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1559–1565.

[36] G. Riddick *et al.*, "Predicting in vitro drug sensitivity using random forests," *Bioinf.*, vol. 27, no. 2, pp. 220–224, 2010.

[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[38] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[39] R. Sikora *et al.*, "A modified stacking ensemble machine learning algorithm using genetic algorithms," in *Handbook of Research on Organizational Transformations Through Big Data Analytics*. Hershey, PA, USA: IGI Global, 2015, pp. 43–53.

[40] J. Johnson, C. Oberhauser, M. Montoya, A. Fresquez, S. Gonzalez, and A. Patel, "Crosstalk nuisance trip testing of photovoltaic dc arc-fault detectors," in *Proc. 38th IEEE Photovolt. Specialists Conf.*, 2012, pp. 001383–001387.

[41] Y. Abdullah, B. Hu, Z. Wei, J. Wang, and A. Emrani, "Adaptive detection of dc arc faults based on hurst exponents and current envelope," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2018, pp. 3392–3397.

[42] G. Yunmei, W. Li, W. Zhuoqi, and J. Binfeng, "Wavelet packet analysis applied in detection of low-voltage dc arc fault," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, 2009, pp. 4013–4016.

Vu Le (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering in 2016 and 2018, respectively, from the State University of New York, Buffalo, NY, USA, where he is currently working toward the Ph.D. degree in electrical engineering.

His current research interests include dc series arc fault detection, machine learning, FPGA implementation, and zonal protection.

Xiu Yao (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Ohio State University, Columbus, Ohio, in 2015.

She is currently an Assistant Professor with the Department of Electrical Engineering, University at Buffalo - The State University of New York, Buffalo, NY, USA. Her research interests include dc arc detection, dc microgrid protection and control, and high voltage dc transmission.

Chad Miller received the B.Sc. degree in computer engineering and the M.Sc. degree in electrical engineering from Wright State University, Dayton, OH, USA, in 2012 and 2014, respectively.

He is currently an Engineering Program Manager with the Aerospace Systems Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA. He also leads the Intelligent Power Systems research program, addressing power stability, power quality, zonal protection, fault tolerance, and emerging technologies through integration, advanced communication, and control.

Bang-Hung Tsao (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Arizona State University, Tempe, AZ, USA, in 1986 and 1989, respectively.

He has over thirty years of experience in developing materials for high power semiconductors, capacitors, and batteries. His experience also includes MATLAB and Simulink system modeling and integration, thin film deposition and processing, materials characterization, device to system level design, fabrication, and testing, thermal management of electronic components, finite element analysis (FEA) modeling and simulation.