

Research Article

Predictive Maintenance Model Based on Multisensor Data Fusion of Hybrid Fuzzy Rough Set Theory Feature Selection and Stacked Ensemble for Fault Classification

Albert Buabeng ¹, Anthony Simons ², Nana Kena Frempong ³,
 and Yao Yevenyo Ziggah ⁴

¹Faculty of Engineering, Mathematical Sciences Department, University of Mines and Technology, Tarkwa, Ghana

²Faculty of Engineering, Mechanical Engineering Department, University of Mines and Technology, Tarkwa, Ghana

³College of Sciences, Statistics and Actuarial Science Department, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

⁴Faculty of Geosciences and Environmental Studies, Geomatic Engineering Department, University of Mines and Technology, Tarkwa, Ghana

Correspondence should be addressed to Albert Buabeng; abuabeng@umat.edu.gh

Received 11 March 2022; Accepted 9 June 2022; Published 29 June 2022

Academic Editor: Muazzam Maqsood

Copyright © 2022 Albert Buabeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rising demand for integrated and autonomous systems in the field of engineering, efficient frameworks for instant detection of performance anomalies are imperative for improved productivity and cost-effectiveness. This study proposes a systematic predictive maintenance framework based on the hybrid multisensor fusion technique of fuzzy rough set feature selection and stacked ensemble for the efficient classification of fault conditions characterised by uncertainties. First, a feature vector of time-domain features was extracted from 17 multiple sensor signals. Then, a comparative study of six different Fuzzy Rough Set Feature Selection (FRFS) methods was employed to select the various combinations of optimal feature subsets for various faults classification tasks. The determined optimal feature subsets then served as inputs for training the stacked ensemble (ESB_(STK)). In the ESB_(STK), Support Vector Machine (SVM), Multilayer Perceptron (MLP), *k*-Nearest Neighbour (*k*-NN), C4.5 Decision Tree (C4.5 DT), Logistic Regression (LR), and Linear Discriminant Analysis (LDA) served as the base classifiers while the LR was selected to be the metaclassifier. The proposed hybrid framework (FRFS-ESB_(STK)) improved the classification accuracy with the selected combinations of optimal feature subset size while reducing the computational cost, overfitting, training runtime, and uncertainty in modelling. Overall analyses showed that the FRFS-ESB_(STK) proved to be generalisable and versatile in the classification of all conditions of four monitored hydraulic components (i.e., cooler, valve, accumulator, and internal pump leakage) when compared with the six base classifiers (standalone) and three existing ensemble classifiers (Stochastic Gradient Boosting (SGB), Ada Boost (ADB), and Bagging (BAG)). The proposed FRFS-ESB_(STK) showed an average improvement of 11.28% and 0.88% test accuracies when classifying accumulator and pump conditions, respectively, while 100% classification rates were obtained for both cooler and valve.

1. Introduction

In the field of engineering, one of the biggest challenges faced with the rising demand for integrated and autonomous systems is the imperfect aspects of the raw sensor data recorded and how they are being processed for predictive-

based maintenance frameworks. These imperfections, ranging from uncertainties caused by inherent noises in sensor measurements to inconsistency in data recorded, could be attributed to diverse operational conditions under which these complicated systems embedded with numerous and varying sensor outputs operate [1]. As a result,

frameworks developed from such imperfections possess tractability challenges and have the tendency of producing misleading information about the state of the system, which could subsequently lead to wrong decisions [2].

In the literature, significant interest has been dedicated to the development of methodologies capable of addressing such imperfections in data. Notable among such methodologies is the Multisensor Data Fusion (MDF) concept of integrating data from multiple sources (taking into account their diversities) into a representation for improving human and automated decision-making frameworks [3–6]. Resounding among the varieties of MDF techniques designed in the literature are the Fuzzy Set Theory (FST) [7] and the Rough Set Theory (RST) [8]. The FST efficiently deals with imprecision by introducing the concept of partial set membership which facilitates reasoning in an imprecise manner (rather than crisp) [7]. This property makes the FST particular useful in fusing data from both homogeneous sources and highly conflicting or ill-defined data [9]. However, the FST requires prior knowledge of membership functions for various fuzzy sets. This is a significant drawback in FST as any decision made by the user may possibly be faulty or based on their subjective judgement.

These limitations resulted in the proposal of the RST which addresses the imprecision, vagueness, and uncertainty in data analysis by exploiting its internal structure (granularity) [8]. By utilising the granularity concept, the hidden knowledge in the pool of data could be identified and expressed as decision rules. In addition, RST has the ability to identify the most informative subset within a pool of attributes (feature reduction or selection) and more importantly does not require any prior knowledge about the data distribution or membership functions [9–12]. These concepts of RST, especially feature reduction, are perhaps the major characteristic that makes it unique from the existing traditional fusion techniques. In that, in addressing the imperfections in a given dataset, the rough set-based feature selection focuses on removing redundant or non-informative features. This leads to a desirable reduction in computational cost, overfitting, training runtime, and uncertainty in modelling whiles improving accuracy [13]. Hence, the usefulness and versatility of RST have been demonstrated successfully in various disciplines including Predictive Maintenance (PdM) [14–17].

However, as frequently observed, data recorded from monitoring the condition of systems for the purpose of developing PdM frameworks are generally real-valued and are characterised by noise. As a result, traditional RST originally developed for crisp or discrete data are unable to process such data [8, 18, 19]. Hence, there is the need for fusion techniques capable of modelling data imperfections, possess the unique characteristic of feature reduction for both crisp and real-valued datasets, and above all, do not require prior knowledge or input to be supplied by the user. These deficiencies could be addressed by coupling and exploiting the unique strengths of fuzzy and rough sets theories [20]. Thus, the hybrid Fuzzy Rough Set Theory (FRST) technique combines the distinct concept of fuzziness and indiscernibility for addressing imperfections in both

discrete and real-valued data without the need for user-supplied input. This ensures that the most informative knowledge from the pool of sensor data (i.e., selected features) is fed as inputs in the development of an intelligent and autonomous predictive maintenance-based framework.

In the context of developing intelligent frameworks for PdM purposes, the utilisation of supervised machine learning algorithms such as the Artificial Neural Networks (ANNs) [21–24], Support Vector Machine (SVM) [25–28], Linear Discriminant Analysis (LDA) [29–31], and the Bayes classifiers [32–35] have been well studied. Although these algorithms have yielded to some extent satisfactory results, they are prone to local optima entrapment when their required hyperparameters are not appropriately fine-tuned. This subsequently affects their performance to discriminate between various classes or label outputs. Also, a considerable number of resources such as time are being invested in the selection of a specific learner for a given task. As a remedy, several researchers have employed ensemble learning which combines different sets of hypotheses from multiple learners with the aim of improving the performance of systems [36–39]. It is important to note that the superiority of ensembles as a suitable alternative over standalone learners has been well established [36, 40, 41]. For these reasons, the stacked ensemble learning which considers heterogenous base learners is proposed in this study for the development of the intelligent PdM framework.

1.1. Contributions of the Study. The contributions of this study are to

- (a) propose a systematic PdM framework for the efficient classification of fault conditions characterised by uncertainties based on the hybrid multisensor Fuzzy Rough Set Feature Selection (FRFS) and the stacked ensemble and
- (b) evaluate and compare the performance of the proposed framework approach with some well-established classifiers (standalone) and ensembles.

The efficiency of the proposed PdM framework was validated on a benchmark dataset of multisensors obtained from the University of California, Irvine (UCI) Machine Learning (ML) repository [29]. The proposed hybrid framework advances the field of PdM by effectively and efficiently fusing recorded data from multiple and varying sensors, thus having enormous potential for enhancing the reliability of autonomous systems during an unexpected sensor failure, enhancing robustness and confidence in estimations, and expanding the sensitivity and specificity ability of systems to efficiently discriminate between class outputs.

The remaining sections are organised as follows: Section 2 briefly describes the experimental hydraulic dataset with multisensors. Section 3 discusses the methodology of the proposed hybrid technique, feature extraction, the FRFS, and stacked ensemble. The results and discussion are presented in Section 4. Section 5 concludes the findings.

TABLE 1: Details of hydraulic components.

| Monitored condition | Unit | Remark | Abbreviation | Class output | Cases |
|-----------------------|------|-----------------------------|--------------|--------------|-------|
| Internal pump leakage | — | No leakage | NL | 0 | 1221 |
| | | Weak leakage | WL | 1 | 492 |
| | | Severe leakage | SIRP | 2 | 492 |
| Cooler condition | % | Close to total failure | CTFc | 3 | 732 |
| | | Reduced efficiency | RE | 20 | 732 |
| | | Full efficiency | FE | 100 | 741 |
| Valve condition | % | Close to total failure | CTFv | 73 | 360 |
| | | Severe lag | SeL | 80 | 360 |
| | | Small lag | SmL | 90 | 360 |
| | | Optimal Switching behaviour | OSB | 100 | 1125 |
| Accumulator condition | Bar | Close to total failure | CTFa | 90 | 808 |
| | | Severely reduced pressure | SeRP | 100 | 399 |
| | | Slightly reduce pressure | SIRP | 115 | 399 |
| | | Optimal pressure | OP | 130 | 599 |

2. Hydraulic System Dataset

The hydraulic system dataset utilised in this paper can be accessed publicly from the ML repository at UCI. The data comprised 2205 instances and 43680 features which were collected from 17 process sensors with varied sample rates. The 17 process sensors were made up of six pressure sensors (PS), four temperature sensors (TS), two volume flow sensors (FS), and five separate sensors for measuring motor power (EPS), vibration (VS), cooling efficiency (CE), cooling power (CP), and system efficiency (SE). With this setup, several typical system failures such as pressure leakage in the accumulator, reduced cooling efficiency, internal pump leakage, or delayed valve switching are simulated to depict variations in fault conditions of four major hydraulic components, namely, the accumulator, cooler, internal pump leakage, and valve. The details of the four major hydraulic components are shown in Table 1. As seen in Table 1, each monitored condition has multiple classes representing the various degradation states of the hydraulic system. Also, a load cycle with a 60-second duration is repeated 2205 times with the distribution of instances as indicated by the cases.

Figures 1–3 show the plot of the 17 process sensors during some selected load cycles. Each subplot represents a specific sensor (abbreviation) followed by a brief description of the degradation states of the four monitored hydraulic components (cooler, valve, pump and accumulator). For instance, the first signal in Figure 1, Cooler Efficiency (CE), is shown for a load cycle with Close to Total Failure (CTFc) in the cooler, Optimal Switching Behaviour (OSB) in the valve, No Leakage (NL) in the pump, and Optimal Pressure (OP) in the accumulator pressure. A similar description is extended to the remaining signals.

3. Methods

The proposed hybrid technique is illustrated in Figure 4 as a schematic starting from feature extraction through to the validation phase. The subsequent sections present the details of the various phases.

3.1. Feature Extraction and Reduction. One major challenge that needs to be addressed in order to develop an efficient PdM framework is the relatively high dimension of features (raw sensor data of 43680 features from the 17 process sensors). This is because, classical ML algorithms experience challenges with tractability, scalability, and high time complexity, which negatively impact classification performance [42, 43]. That is, these classical ML algorithms are unable to detect features that contain the most relevant fault information needed for efficient fault classification [44].

Based on the aforementioned reasons, this paper adopts a well-established strategy by extracting some statistics-based features from the raw 43680 sensor data that describe the hydraulic dataset's characteristic attributes. Hence, different statistical time-domain features were extracted based on the variance, median, mean, standard deviation, kurtosis, skewness, and position of maximum values from various time intervals partitions. Details of the time-domain feature extraction can be found in the prior works of Buabeng et al. [45, 46]. The feature extraction drastically reduced the dimension of the hydraulic dataset to 1806 features, thus to some extent reducing the computational cost.

However, due to the influence of insignificant and redundant features, training a classifier with the entire 1806 features can negatively impact the classifier's ability to effectively and efficiently discriminate between class output [47]. This will in effect increase the cost of computation, overfitting, training runtime, and the model's uncertainty. For these reasons, the most optimal feature subsets that capture relevant fault characteristics were identified using the FRFS. The FRFS technique, as opposed to the correlation-based feature selection technique, was adopted by prior works such as Helwig et al. [29]. The advantage of the FRFS is that it does not require prior knowledge or the user to subjectively specify the number of features to select as wrong judgement may influence the efficacy of the model being developed.

3.2. Handling Data Imperfection with Multisensor Data Fusion (MDF). The MDF technique is a multidisciplinary

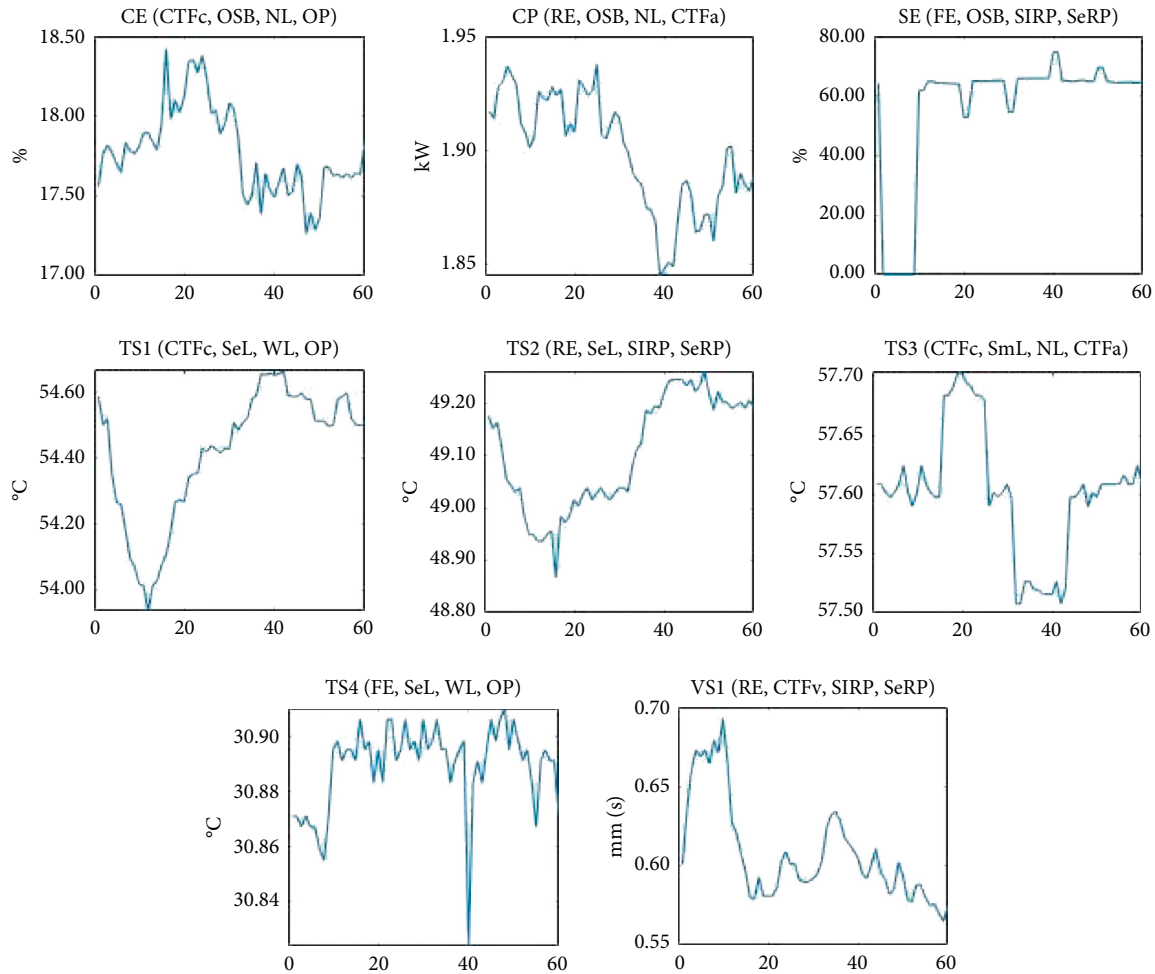


FIGURE 1: Sensors with 1 Hz sampling rate.

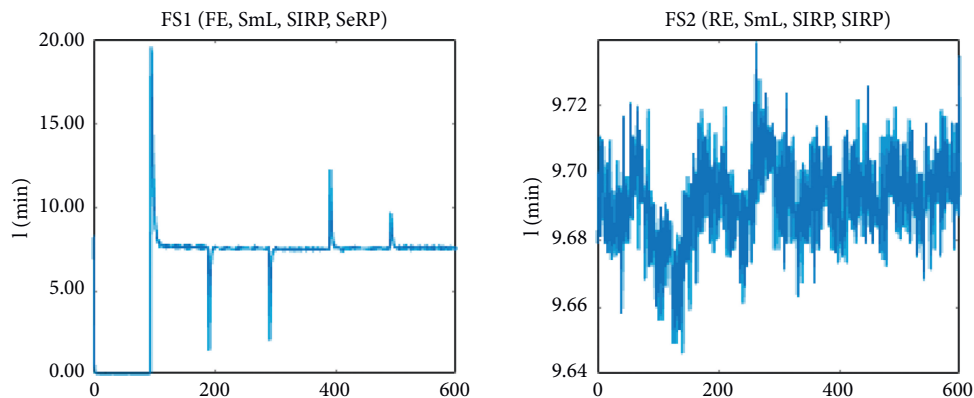


FIGURE 2: Sensors with 10 Hz sampling rate.

research discipline utilising ideas from signal processing, information theory, statistical estimation and inference, and artificial intelligence [2]. Its core concept of integrating data from multiple sources (taking into account their diversities) into a representation for improving human and automated decision-making frameworks was first introduced in the 1960s, however, was implemented in the field of robotics and the military in the 1970s [4, 6, 48]. Today, MDF techniques

are widely used for preprocessing and handling data imperfections and are known for their enormous potential in numerous fields of autonomous systems such as pattern recognition, system monitoring, fault identification, intrusion and malware detection, and health care, among others [49–52]. This has been realised through the dynamic integration of the comprehensive and varied pool of knowledge arriving from the multiple sensors.

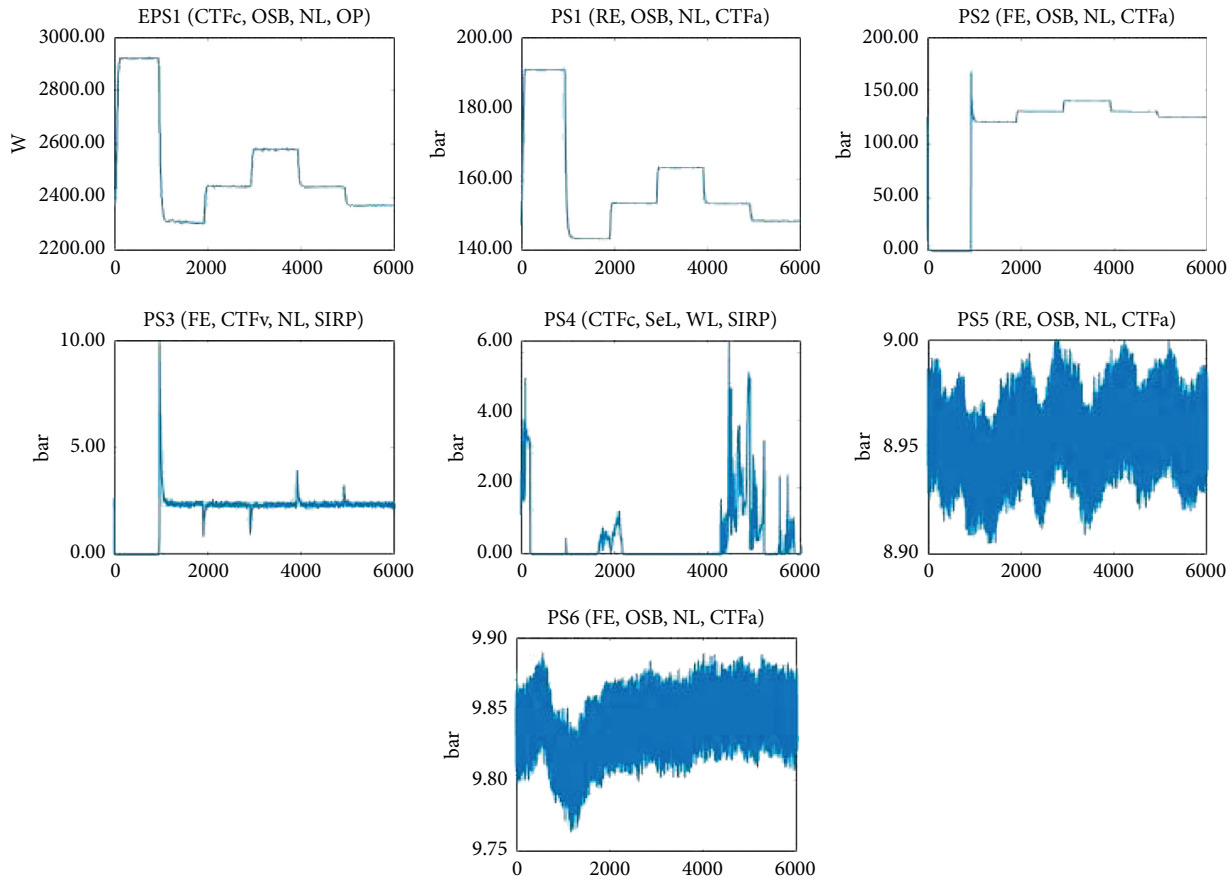


FIGURE 3: Sensors with 100 Hz sampling rate.

In dealing with various aspects of data imperfection, varieties of MDF techniques have been designed. Traditionally, the most dominant are the Probabilistic Fusion (PF) [53], FST [7], Evidence Belief Reasoning (EBR) [54], and the RST [8]. However, these varieties in their implementation differ with varying characteristics, strengths, and limitations. For instance, the PF which is based on the probability distribution or density functions addresses to some extent the uncertainties in data but however is perceived with limitations such as complexity, inconsistency, and precision and requires prior knowledge of the probability distribution [7]. For these reasons, the FST which addresses imprecision, and the EBR which addresses both uncertainty and imprecision were proposed as alternatives [9].

The EBR in addressing uncertainty and imprecision assigns belief and plausibility to possible measurement hypotheses in addition to an appropriate combination rule based on Dempster Shafer's theory in the fusion process [54]. However, EBR is computationally expensive and has the tendency of generating counterintuitive results in the presence of fusing conflicting (heterogenous) data [9]. The FST efficiently deals with imprecision by introducing the concept of partial set membership which facilitates reasoning in an imprecise manner (rather than crisp) [7]. This property makes the FST particularly useful in fusing data from both homogeneous sources (conjunctive fuzzy fusion rules are utilised) as well as highly conflicting data

(disjunctive fuzzy fusion rules are utilised) [9]. Thus, while both the PF and EBR are well suited for handling the uncertainty of data with a well-defined class of objects, the FST is suited for an ill-defined class. However, just like the PF, the FST requires prior knowledge of membership functions for various fuzzy sets. This is a significant drawback in FST as any decision made by the user may possibly be faulty or based on their subjective judgement.

These limitations resulted in the proposal of the RST which has been successful over the years [14, 16, 17, 55]. The RST addresses the imprecision, vagueness, and uncertainty in data analysis by exploiting its internal structure (granularity) [8]. The success of RST is primarily due to its capacity to utilise the most informative features, possesses feature reduction ability, and moreover does not require any user-supplied information [9–12]. However, the traditional RST was proposed for crisp or discrete data, thus, was limited in its application to real-valued data [8, 18, 19]. This led to developing a synergistic approach of combining the FST as a complementary technique to RST since the techniques could both address imprecision and inconsistency. Hence, the hybrid FRST provides a comprehensive framework by combining the distinct concept of fuzziness and indiscernibility for addressing imperfections in both discrete and real-valued data [20]. In addition, FRST requires no user-supplied input and possesses feature selection capabilities.

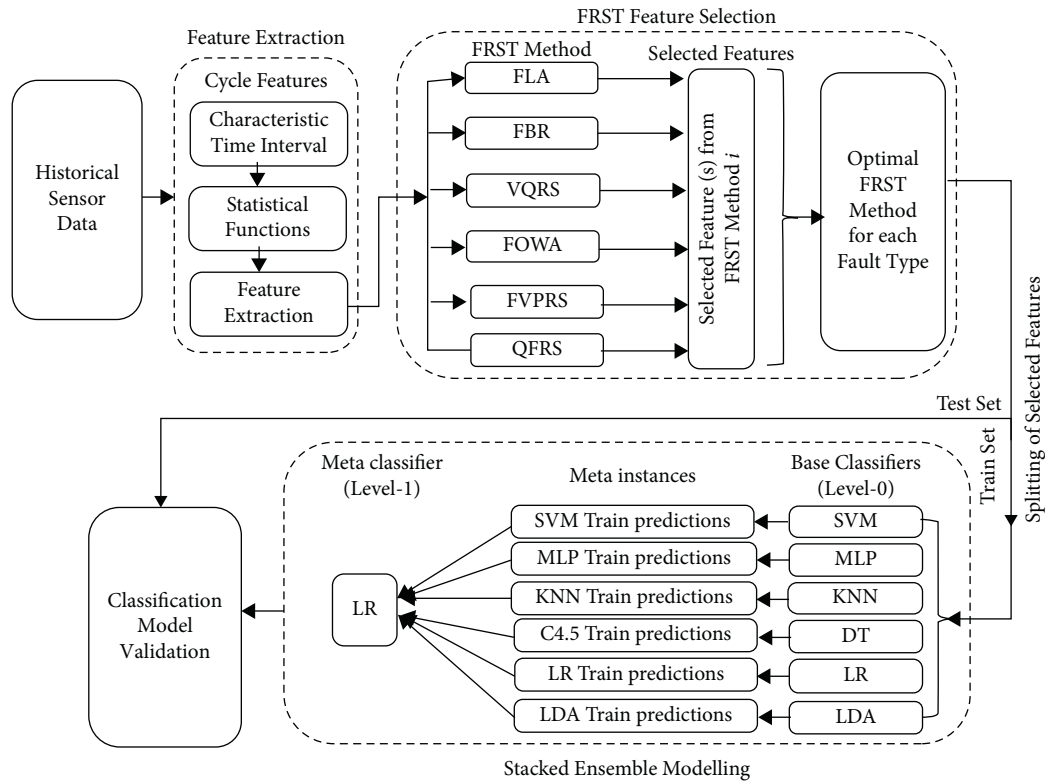


FIGURE 4: Schematic of the proposed hybrid technique.

From the discussion, the concept of the FRST can be considered as a generalisation of the rough set constructed based on two theories: the rough set and fuzzy set theories. The theories are briefly described as follows.

3.2.1. Fuzzy Set Theory (FST). The FST was proposed by Zadeh [7] and is an extension of the classical set theory for dealing with vagueness. Since its introduction, the theory has been developed and extended by other researchers with application in diverse disciplines of engineering and science [56–58] of which PdM is no exception [59–61].

The basic concept behind FST revolves around the membership status of an item [62]. That is whether an item, say x is a member or not a member of a set, say A ; $x \in A$ or $x \notin A$. As opposed to the classical set theory where an item should belong to a set or not, an item in FST can belong to a set to a degree $0 \leq k \leq 1$. Thus, the fuzzy membership function is expressed as

$$\mu_X(x) \in \langle 0, 1 \rangle, \quad (1)$$

where x denotes the item and X represents the set.

The fuzzy membership function has the following properties:

$$\mu_{U-X}(x) = 1 - \mu_X(x) \text{ for any } x \in U, \quad (2)$$

$$\mu_{X \cup Y}(x) = \max(\mu_X(x), \mu_Y(x)), \text{ for any } x \in U, \quad (3)$$

$$\mu_{X \cap Y}(x) = \min(\mu_X(x), \mu_Y(x)), \text{ for any } x \in U. \quad (4)$$

Hence, this suggests that the membership of a set determines whether an item belongs to the union and intersection of the set.

3.2.2. Rough Set Theory (RST). In the literature, the success in the implementation of RST [8] in addressing data imprecision, vagueness, and uncertainty has well been established [17, 55]. By exploiting the internal structure of the dataset, the basic RST is described based on the concept of indiscernibility.

Suppose that an information system I is expressed as

$$I = (U, A), \quad (5)$$

where $U = \{x_1, x_2, \dots, x_n\}$ and $A = \{a_1, a_2, \dots, a_m\}$ represent nonempty sets of finite objects as a universe of discourse (all instances) and features, respectively, such that $a: U \rightarrow V_a$ for every $a \in A$. V_a represents the set of possible values of the feature a . Given any $P \subseteq A$, there exists an associated equivalence relation $\text{IND}(P)$ as shown in

$$\text{IND}(P) = \{(x, y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}. \quad (6)$$

From (6), the partition of U generated, denoted by $U/\text{IND}(P)$, is estimated using

$$\frac{U}{\text{IND}(P)} = \otimes \left\{ \frac{U}{\text{IND}(\{a\})} \mid a \in P \right\}, \quad (7)$$

where the operator \otimes is defined as shown in (8) using two imaginary sets, say A and B :

$$A \otimes B = \{X \cap Y | X \in A, Y \in B, X \cap Y \neq \emptyset\}. \quad (8)$$

When $(x, y) = \text{IND}(P)$, it implies that x and y are indiscernible based on features from P . Denoting the equivalence classes of the P -indiscernibility relations by $[x]_P$ and assuming X as a subset of $U, X \subseteq U$, approximating X using information contained in P is achieved by constructing the P -upper and P -lower approximations of X as shown in (9) and (10), respectively:

$$\overline{P}X = \{x \in U | [x]_P \cap X \neq \emptyset\}, \quad (9)$$

$$\underline{P}X = \{x \in U | [x]_P \subseteq X\}, \quad (10)$$

where $\overline{P}X$ and $\underline{P}X$, respectively, are the upper and lower approximations of X with respect to P . That is, the $\overline{P}X$ comprises of possible objects classified in X while object certainly classified in X are listed in the $\underline{P}X$. The ordered pair $(\underline{P}X, \overline{P}X)$ is the rough set. Assuming P and Q are subsets of features from A , with equivalence relation over U , then the different regions (positive, negative, and boundary regions) are expressed as

$$\text{POS}_P(Q) = \bigcup_{X \in U/\text{IND}(Q)} \underline{P}X, \quad (11)$$

$$\text{NEG}_P(Q) = U - \bigcup_{X \in U/\text{IND}(Q)} \overline{P}X, \quad (12)$$

$$\text{BND}_P(Q) = \bigcup_{X \in U/\text{IND}(Q)} \overline{P}X - \bigcup_{X \in U/\text{IND}(Q)} \underline{P}X. \quad (13)$$

The positive region, $\text{POS}_P(Q)$, comprises of all objects classified into classes of $U/\text{IND}(Q)$, the boundary region, $\text{BND}_P(Q)$, comprises of possible but not definite objects classified, and the negative region, $\text{NEG}_P(Q)$, contains objects not classified into classes of $U/\text{IND}(Q)$.

In data analysis, discovering the dependencies within features is of paramount importance. Thus, a set of features Q is said to be totally dependent on a set of features P if the values of the feature in Q can be determined uniquely by the values of features from P . In RST, the dependency of $P, Q \subset A$ in a degree $0 \leq k \leq 1$, denoted by $P \Rightarrow_k Q$, is expressed in

$$k_{=_{\gamma P}}(Q) = \frac{|\text{POS}_P(Q)|}{|U|}. \quad (14)$$

Such that Q totally depends on P if $k = 1$, Q partially depends on P if $k < 1$ and Q does not depend on P if $k = 0$.

The RST process described here was proposed for crisp or discrete data, thus, was limited in its applicability to real-valued data which is most common [8, 18, 19]. A possible alternative around this limitation will be to create in advance new crisp data out of the real-valued data through discretisation. However, the procedure is still inadequate as the degree of membership or similarity between feature values is not exploited [63, 64]. Notwithstanding, the hybrid FRST addresses the aforementioned limitation by combining the concepts of fuzziness and indiscernibility for addressing imperfections in both discrete and real-valued data.

3.2.3. Fuzzy Rough Set Theory (FRST). The FRST can be thought of as a generalisation to the RST where the

approximation from fuzzy sets, lower and upper fuzzy approximations, is derived in a crisp rough set space. That is, fuzziness is integrated into rough sets by defining the lower and upper approximations of the set when U , the nonempty set of finite fuzzy sets, becomes rough because of the equivalence relation.

Suppose that the subsets of features from $A, P \subseteq A$, with equivalence relation over U denoted as $\text{IND}(P)$, the equivalence class can be expressed as fuzzy sets $F = \{F_1, F_2, \dots, F_h\}$ if the class to which F_i for all $i \in \{1, 2, \dots, h\}$ attribute are ambiguous. Thus, the fuzzy P - lower and P - upper approximations of X are expressed as

$$\mu_{\underline{P}X}(F_i) = \inf_x \max\{1 - \mu_{F_i}(x), \mu_X(x)\}, \quad \forall i, \quad (15)$$

$$\mu_{\overline{P}X}(F_i) = \sup_x \min\{\mu_{F_i}(x), \mu_X(x)\}, \quad \forall i, \quad (16)$$

where X is the fuzzy concept to be approximated with F_i being the fuzzy equivalence class belonging to $\text{IND}(P)$. The ordered pair $(\underline{P}X, \overline{P}X)$ is the fuzzy rough set. These definitions deviate a bit from the lower and upper approximations under the crisp rough set due to the inability to explicitly access the membership of individual objects to the approximations. As a result, the fuzzy lower and upper approximations are redefined by employing the concept of sup and inf as shown in

$$\mu_{\underline{P}X}(x) = \sup_{F \in U/\text{IND}(P)} \min\left(\mu_F(x), \inf_{y \in U} \max\{1 - \mu_F(y), \mu_X(y)\}\right), \quad (17)$$

$$\mu_{\overline{P}X}(F_i) = \sup_{F \in U/\text{IND}(P)} \min\left(\mu_F(x), \sup_{y \in U} \min\{\mu_F(y), \mu_X(y)\}\right). \quad (18)$$

It can be observed from (17) and (18) that every $y \in U$ is taken into account with instances where their corresponding $\mu_F(y) \neq 0$. A detailed discussion on the usage of the min and max operators can be found in Radzikowska and Kerre [65], where a comparative study of FRS represented by specific implication and t -norm have been presented.

3.2.4. Fuzzy Rough Set Feature Selection (FRFS). The feature selection or reduction ability of RST is perhaps a significant factor owing to its successful application in diverse disciplines. This unique ability can be exploited in the fuzzy rough set via the concept of fuzzy lower approximation for reducing datasets of real-valued features. Referring to the extension principle of Zadeh [66], the membership of an object $x \in U$ belonging to a fuzzy positive region is expressed as

$$\mu_{\text{POS}_P(Q)}(x) = \sup_{X \in U/\text{IND}(Q)} \mu_{\underline{P}X}(x). \quad (19)$$

From (19), it can be deduced that the object x fails to belong to the positive region if the equivalence class it belongs to is not a member of the positive region. Recall (14), the fuzzy rough dependency degree function is expressed from the definition of the positive region:

$$\begin{aligned} \gamma'_P(Q) &= \frac{|\mu_{\text{POS}_P(Q)}(x)|}{|U|} \\ &= \frac{\sum_{x \in U} \mu_{\text{POS}_P(Q)}(x)}{|U|}. \end{aligned} \quad (20)$$

However, for the FRFS to be useful in practice, it should be capable of handling high-dimensional datasets by means of estimating the dependencies of various feature subsets with the original dataset. This is relevant as the objects may belong to several equivalence classes. For instance, in the crisp case in (7), $U/\text{IND}(P)$ consist of groups of objects that are indiscernible based on the features from P . However, in the case of fuzzy, the Cartesian product of features from P is considered in estimating $U/\text{IND}(P)$, where each set in $U/\text{IND}(P)$ is an equivalence class. Hence, the extent to which an object belongs to the equivalence class could be estimated using the combination of constituent fuzzy equivalence classes, $F_i, \forall i = 1, 2, \dots, n$ as shown in

$$\mu_{F_1 \cap \dots \cap F_n}(x) = \min(\mu_{F_1}(x), \mu_{F_2}(x), \dots, \mu_{F_n}(x)). \quad (21)$$

Although the usefulness of FRFS in addressing uncertainty in data and also as a dimension reduction technique is widely known in the literature, the technique possesses some deficiencies [12, 67, 68]. First, the problem of complexity in estimating the Cartesian product for the equivalence classes renders the technique to be highly prohibitive and computationally expensive, especially when dealing with high-dimensional datasets. The Cartesian product of the fuzzy equivalence classes may not result in a family of fuzzy equivalence classes [12, 69]. Also, the scenario of obtaining a lower approximation outcome which may not necessarily be a subset of the upper approximation for some instances may occur [69]. This outcome is undesirable since it contradicts the theoretical viewpoint of the lower approximation being higher than its upper approximation, implying that there is less certainty in the lower than the upper approximations. Moreover, several research works have established that the classical FRS model is sensitive to misclassification (error or missing values) and perturbation (noisy information) which are noted to be the primary source of uncertainty in real-life applications [70, 71]. This limits the applicability of fuzzy rough sets in practice.

For these reasons, this study employs a comparative study of six different variants of the lower and upper approximations that define the extent to which the set of elements can be classified into a certain class as strongly or weakly. The variants considered are the Fuzzy Lower Approximation (FLA) and Fuzzy Boundary Region (FBR) proposed by Jensen and Shen, Vaguely Quantified Rough Sets (VQRSs) [72], Ordered Weighted Average (OWA) [73], Fuzzy Variable Precision Rough Sets (FVPRSs) [70], and the β -Precision Fuzzy Rough Sets (β PFRSs) [74]. These variants were considered as they have been uniquely designed to address some limitations of the classical FRS model through enhanced estimation capabilities for the lower and upper approximations as well as their degree of dependency parameters for selecting relevant features.

(1) *Fuzzy Lower Approximation (FLA)-Based Feature Selection.* Unlike the previously discussed FRFS which employs a fuzzy partitioning of the input space for determining the fuzzy equivalence classes (refer to Section 3.2.4), the FLA proposed by Jensen and Shen [12] utilises a T -transitive fuzzy similarity relation found in Radzikowska and Kerre [65] as an alternative for estimating the fuzzy lower and upper approximations:

$$\mu_{\underline{R}_P X}(x) = \inf_{y \in U} I\{\mu_{R_P}(x, y), \mu_X(y)\}, \quad (22)$$

$$\mu_{\overline{R}_P X}(x) = \sup_{y \in U} T\{\mu_{R_P}(x, y), \mu_X(y)\}, \quad (23)$$

where R_P is the fuzzy similarity relation induced by a subset of features P , I and T are the fuzzy implicator and t -norm, respectively. Given a feature, say a , the degree to which objects x and y are similar to feature a and $\mu_{R_a}(x, y)$ is expressed as

$$\mu_{R_a}(x, y) = \bigcap_{a \in P} \{\mu_{R_a}(x, y)\}. \quad (24)$$

Based on (24), several fuzzy similarity (tolerance) relations such as (25)–(27) could be defined:

$$\mu_{R_a}(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|}, \quad (25)$$

$$\mu_{R_a}(x, y) = \exp\left(-\frac{(a(x) - a(y))^2}{2\sigma_a^2}\right), \quad (26)$$

$$\begin{aligned} \mu_{R_a}(x, y) &= \max\left(\min\left(\frac{(a(y) - (a(x) - \sigma_2))}{(a(x) - (a(x) - \sigma_2))}, \right. \right. \\ &\quad \left. \left. \frac{((a(x) + \sigma_2) - a(y))}{((a(x) + \sigma_2) - a(x))}, 0\right), 0\right), \end{aligned} \quad (27)$$

where σ_a^2 is the variance of feature a . Similar to the FRFS (Section 3.2.4), the fuzzy positive region and the dependency degree functions are estimated using the following:

$$\mu_{\text{POS}_{R_P}(Q)}(x) = \sup_{X \in U/\text{IND}(Q)} \mu_{\underline{R}_P X}(x), \quad (28)$$

$$\gamma'_P(Q) = \frac{\sum_{x \in U} \mu_{\text{POS}_{R_P}(Q)}(x)}{|U|}. \quad (29)$$

(2) *Fuzzy Boundary Region (FBR)-Based Feature Selection.* Proposed by Jensen and Shen [12], the FBR algorithm is based on the membership degree to the fuzzy boundary region expressed as the difference between the upper and lower approximations:

$$\mu_{\text{BND}_{R_P}} = \mu_{\overline{R}_P X}(x) - \mu_{\underline{R}_P X}(x), \quad (30)$$

with the fuzzy negative region for all decision concepts estimated using

$$\mu_{\text{NEG}_{R_P}(Q)}(x) = N\left(\sup_{X \in U/\text{IND}(Q)} \mu_{\overline{R}_P X}(x)\right). \quad (31)$$

From (30), the uncertainty for a concept X based on P features and the total uncertainty degree for all concepts are estimated using (32) and (33), respectively:

$$U_P(X) = \frac{\sum_{x \in U} \mu_{\text{BND}_{R_p}(x)}(x)}{|U|}, \tag{32}$$

$$\lambda_P(Q) = \frac{\sum_{x \in U} \text{IND}(Q) U_P(x)}{|U/Q|}. \tag{33}$$

Thus, the FBR algorithm utilises the total uncertainty degree $\lambda_P(Q)$ for all concepts of feature subset P and decision attribute Q for selecting the optimal features.

(3) *Vaguely Quantified Rough Set (VQRS)-Based Feature Selection.* The VQRS was proposed by Cornelis and Jensen [72] with the notion that the standard FRS as well as some extensions are sensitive to noise and hence may be highly influenced during the reduction process (specifically, on the lower and upper approximations, (22) and (23)) when there is a change in any single object. Hence, to remedy this shortfall, the VQRS replaces the fuzzy lower (22) and upper (23) approximations with (34) and (35), respectively:

$$\mu_{\underline{R}_p X}(x) = Q_l \left(\frac{|\mu_{R_p}(x, y) \cap \mu_X(y)|}{|\mu_{R_p}(x, y)|} \right), \tag{34}$$

$$\mu_{\overline{R}_p X}(x) = Q_u \left(\frac{|\mu_{R_p}(x, y) \cap \mu_X(y)|}{|\mu_{R_p}(x, y)|} \right), \tag{35}$$

where Q_l and Q_u are the fuzzy quantifiers for the lower and upper approximations, respectively. The performance of the VQRS model is dependent on the values of Q_l and Q_u parameters as expressed in

$$Q_{(\alpha, \beta)}(x) = \begin{cases} 0, & x \leq \alpha, \\ \frac{2(x - \alpha)^2}{(\beta - \alpha)^2}, & \alpha \leq x \leq \frac{\alpha + \beta}{2}, \\ 1 - \frac{2(x - \alpha)^2}{(\beta - \alpha)^2}, & \frac{\alpha + \beta}{2} \leq x \leq \beta, \\ 1, & x \geq \beta, \end{cases} \tag{36}$$

where $0 \leq \alpha \leq \beta \leq 1$ are parameter values.

(4) *Fuzzy Ordered Weighted Average (FOWA)-Based Feature Selection.* Similar to the VQRS, the Ordered Weighted Average (OWA) variant of FRS was proposed by Cornelis et al. [73] for addressing noisy and outlying samples. In FOWA computation, the lower and upper approximations are estimated using an aggregation technique of OWA estimators (39) as indicated in (37) and (38), respectively:

$$\mu_{\underline{R}_p X}(x) = \text{OWA}_{W_l} \langle I \{ \mu_{R_p}(x, y), \mu_X(y) \} \rangle, \tag{37}$$

$$\mu_{\overline{R}_p X}(x) = \text{OWA}_{W_u} \langle T \{ \mu_{R_p}(x, y), \mu_X(y) \} \rangle, \tag{38}$$

$$\text{OWA}_W(X) = \sum_{i=1}^n w_i c_i, \tag{39}$$

where c_i is the i^{th} largest value in X and $W = \langle w_i \rangle$ is the weighting vector such that for $m < n$, $\forall i = 1, 2, \dots, n$, it is possible to define $W_l = \langle w_i^l \rangle$ and $W_u = \langle w_i^u \rangle$ as (40) and (41), respectively:

$$w_{n+1-i}^l = \begin{cases} \frac{2^{m-i}}{2^{m-1}}, & i = 1, 2, \dots, m, \\ 0 & i = m + 1, \dots, n, \end{cases} \tag{40}$$

$$w_i^u = \begin{cases} \frac{2^{m-i}}{2^{m-1}}, & i = 1, 2, \dots, m, \\ 0 & i = m + 1, \dots, n. \end{cases} \tag{41}$$

(5) *Fuzzy Variable Precision Rough Set (FVPRS)-Based Feature Selection.* The FVPRS was developed based on the concept of addressing the sensitivity of the standard FRS to misclassification (error or missing values) and perturbation (noise) [70]. For these reasons, the FVPRS was proposed by hybridising the FRS with the Variable Precision Rough Set (VPRS) which is known as the first model for handling class noise in data [75]. Thus, the FVPRS addresses not only data uncertainty but also is less sensitive to misclassification and perturbation. In FVPRS, the lower and upper approximations are defined in (42) and (43), respectively:

$$\mu_{\underline{R}_p X}(x) = \inf_{\mu_X(y) \leq \alpha} I \{ \mu_{R_p}(x, y), \alpha \} \tag{42}$$

$$\wedge \inf_{\mu_X(y) > \alpha} I \{ \mu_{R_p}(x, y), \mu_X(y) \},$$

$$\mu_{\overline{R}_p X}(x) = \sup_{\mu_X(y) \geq N(\alpha)} T \{ \mu_{R_p}(x, y), N(\alpha) \} \tag{43}$$

$$\vee \sup_{\mu_X(y) < N(\alpha)} T \{ \mu_{R_p}(x, y), \mu_X(y) \},$$

where α , I , and T are the variable precision parameter, implicator, and t -norm operators, respectively.

(6) *β -Precision Fuzzy Rough Set (BPFRS)-Based Feature Selection.* The BPFRS algorithm as proposed by Salido and Murakami [74] uses the concept of β -precision aggregation as a generalisation to Zarinko's Variable Precision Rough Set (VPRS) [76] for addressing uncertainty in huge datasets. In its implementation, the β -precision quasi- T -norm and β -precision quasi- T -conorm are utilised in defining the β -precision versions of fuzzy lower and upper approximations of a fuzzy set X in U expressed as (44) and (45), respectively:

$$\mu_{\underline{R}_p X}(x) = T_{\beta, y \in U} S \{ \mu_{R_p}(x, y), \mu_X(y) \}, \tag{44}$$

$$\mu_{\overline{R}_p X}(x) = S_{\beta, x \in U} T \{ \mu_{R_p}(x, y), \mu_X(y) \}, \tag{45}$$

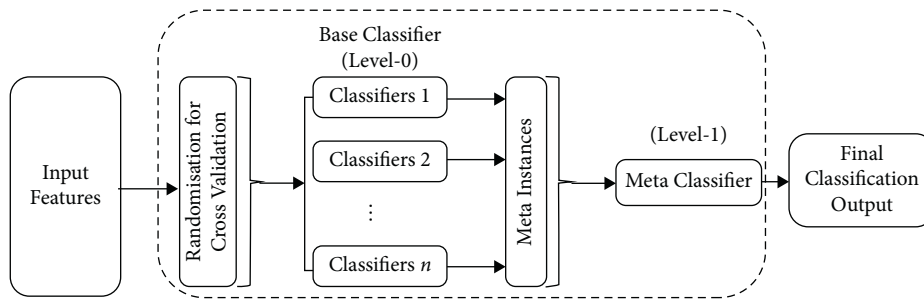


FIGURE 5: Schematic diagram of stacked generalisation.

where $T_{\beta, x \in U} = T(y_1, \dots, y_{n-m})$ and $S_\beta(x) = T(z_1, \dots, z_{n-k})$ are the β -precision quasi- T -norm and β -precision quasi- T -conorm, respectively, such that when given a t -norm T , a t -conorm S , $\beta \in [0, 1]$ and $n \in \mathbb{N}/\{0, 1\}$, T_β and S_β of the order n are $[0, 1]^n \rightarrow [0, 1]$ mappings for all X . y_i is the i^{th} greatest element of X , z_i is the i^{th} smallest element of X , $m = \max(i \in \{0, \dots, n\} | i \leq (1 - \beta) \sum_{j=1}^n x_j)$, and $k = \max(i \in \{0, \dots, n\} | i \leq (1 - \beta) \sum_{j=1}^n (\alpha - x_j))$.

3.3. Stacked Ensemble. Considering the fact that the primary goal of every decision support system is to always produce reliable and accurate outcomes for every classification task, researchers are constantly being faced with the decision of selecting the most accurate classifier for various tasks even though they are task-specific [76, 77]. In practice, the ideal approach will be to use trial and error where the candidate classifiers will be tested on a given problem to ascertain and select the one which produces the most accurate result. While this trial-and-error approach may eventually yield an optimal outcome, it demands a lot of resources such as time. For these reasons, considerable resources have been dedicated to the combination of different classifiers (ensembles or metaclassification) as an alternative for achieving the desired results. Thus, the strategy of the ensemble is to systematically combine the output of classifiers such that it yields better results than any of the single classifiers.

Stacking, also known as stacked generalisation which was proposed by Wolpert [78], can be considered among the influential of such ensemble schemes [79]. That is, unlike other metaclassification schemes such as bagging and boosting which generate attributes via the same (homogeneous) learning algorithm and combining their outputs based on a predetermined scheme (nontrainable), stacking generates attributes from the predictions of multiple (heterogeneous) base learners, and subsequently combining these outputs through one metaclassifier [77]. Hence, each learner uses a different approach to obtaining knowledge, biases, etc., and as such explores various hypothetical spaces from a diverse perspective. As a result, the classification from the resulting stacked ensemble is known to yield a more accurate and robust outcome than any of the individual base classifiers. Consequently, the stacked ensembles provide a naturally suited remedy for many large-scale data analysis and autonomous systems which fuses heterogeneous data from multiple sources [80].

In combining the predictions from the multiple and heterogeneous base classifiers, stacking utilises the concept of a metaclassifier (Level-1) to generate the final output by modelling a k^{th} -fold-like cross-validated predictions (meta instances) from the base-classifiers (Level-0) as shown in Figure 5. First, the selected features from FRFS were split into training, validation, and testing sets. Using the training dataset, each base-classifier is trained, and predictions are made based on the validation set. The predictions from the base classifiers are used as features for training the metalearner. This implies that the metaclassifier is trained to synergistically combine the outputs of the base-classifiers for predicting out-of-sample instances (test set).

However, one major challenge with stacking is the optimal combination of base classifiers and the choice of a single metaclassifier since the configuration is application-specific [77, 81–83]. The challenge is further complicated when dealing with a large search space [77]. In the literature, the usage of diverse or heterogeneous base classifiers for obtaining the meta instances has been noted to efficiently address this challenge [84]. Hence, as the base learners, this research employs diverse learners such as the SVM, the Multilayer Perceptron (MLP), k -Nearest Neighbour (k -NN), C4.5 Decision Trees (C4.5 DT), Logistic Regression (LR), and the LDA. Regarding the metalearner, complex meta-classifiers are rarely used in the literature since they are likely to overfit the predictions from the base-classifiers. For this reason, simple models such as the Logistic Regression (LR) are utilised due to the advantage of producing piecewise linear approximations, less prone to overfitting and also for simpler interpretations of the resulting output [85]. Hence, the schematic for the proposed stacked ensemble is shown in Figure 6.

3.3.1. Support Vector Machine (SVM). The concept of SVM is basically to create hyperplane(s) (decision boundary) in a high-dimensional space, where the best hyperplane is denoted as the one that optimally partitions the data into distinct classes and with the largest separation of classes [25, 26, 28]. The technique is known for its high performance, efficiency, and robustness in both classification and regression tasks; thus, SVM is one of the highly preferred classifiers in the field of predictive maintenance.

In SVM computation, the extent of the separation (margins) is determined and maximised via a kernel

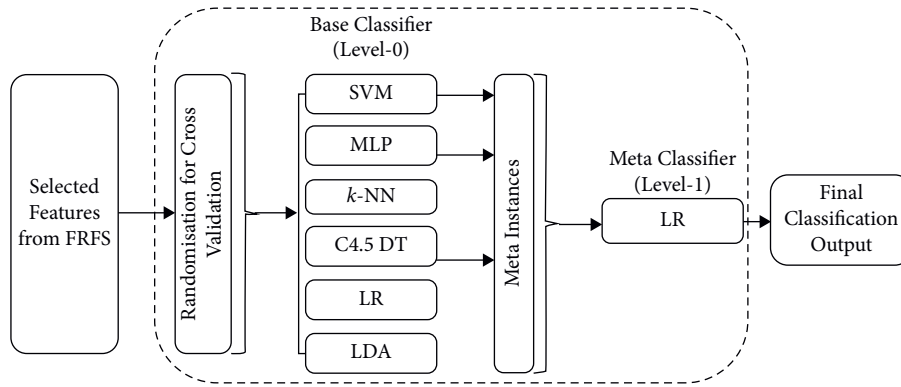


FIGURE 6: Proposed stacked ensemble.

function (i.e., linear, radial basis, polynomial, and sigmoid). However, the frequently used kernel function, the Radial Basis Function (RBF) is adopted for its excellent general performance, wider convergence domain, and high-resolution power and requires fewer parameters [86–88]. The RBF kernel function is expressed in

$$K_{\text{RBF}}(x, y) = \exp(-\gamma \|x - y\|^2), \quad (46)$$

where for a given training sample (x_i, y_i) , $x_i \in \mathbb{R}^{n \times p}$ and response vector $y_i \in \{-1, +1\}$, where -1 and $+1$ represent samples from the negative and positive classes, respectively.

The SVM problem is formulated as a minimisation problem as shown in

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i, \quad (47)$$

subject to

$$y_i (w^T w \phi(x_i) + b) \geq 1 - \xi_i. \quad (48)$$

Here, $\phi(x)$ represents a nonlinear mapping function for transforming x to a high-dimensional space, w and b are the weight vector and bias respectively, and $\xi_i, \forall i = 1, 2, \dots, n$ are slack variables. Hence, the SVM aims to estimate w and b such that the maximum separation (margins) of the classes into distinct groups is achieved. However, in cases where the maximum separation is suboptimal, soft margins which are represented as inequalities ((49)–(51)) are used:

$$w x_i - b \geq +1 - \xi_i \quad \text{if } y_i = +1, \quad (49)$$

$$w x_i - b \geq -1 + \xi_i \quad \text{if } y_i = -1, \quad (50)$$

$$\xi_i \geq 0. \quad (51)$$

When an error occurs, $\xi_i > 1$ and the upper bound on the training error $\sum_i \xi_i$ is controlled by the Lagrangian L_p shown in

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_i \alpha_i (y_i (x_i w - b) - 1 + \xi_i) - \sum_i \mu_i \xi_i, \quad (52)$$

where μ_i is the Lagrange multipliers for estimating the positive values of ξ_i .

3.3.2. *Multilayer Perceptron (MLP)*. The MLP neural networks are the most frequently used feedforward neural networks due to their simplicity, efficiency, and versatility in various research problems including predictive maintenance tasks [23, 24, 89]. The MLP is generally organised in three parallel layers: input, hidden, and output. The layers are connected as a summated linear combination of weight and bias vectors expressed mathematically as

$$s_j = \sum_{i=1}^n (W_{ij} \cdot X_i) + \beta_j, \quad j = 1, 2, \dots, h, \quad (53)$$

where the weight vector W_{ij} ranging between $[-1, 1]$ connects the i^{th} node of the input layer and the j^{th} node of the hidden layer of neuron(s), X_i is the i^{th} input feature with n number of inputs, and β_j is the bias (threshold) of the j^{th} hidden node.

The output of each hidden node is then estimated using an activation function. Though there are several forms of activation functions for MLP, this paper utilises the most commonly used sigmoid function as shown in

$$f_j = \frac{1}{(1 + \exp(-s_j))}, \quad j = 1, 2, \dots, h. \quad (54)$$

Based on the outputs from the hidden nodes f_j , the final outputs y_i are then estimated using

$$y_i = f_j \left(\sum_{i=1}^n (W_{ij} \cdot X_i) + \beta_j \right). \quad (55)$$

With this setup, the estimates of the weights and biases are fine-tuned and updated until minimal error is obtained.

3.3.3. *k-Nearest Neighbour (k-NN)*. The k -NN is a non-parametric supervised learner and perhaps the simplest form of classification as it is based on the concept of similarity (distance between objects). As such, it is often used in the field of predictive maintenance [90, 91]. In k -NN model computation, the commonly used distance measure is the Euclidean expressed as

$$d(X_i, x) = \sqrt{\sum_{i=1}^n (X_i - x_i)^2}, \quad (56)$$

where X is the n -dimensional input vector and x is an unknown or test sample.

3.3.4. C4.5 Decision Tree (C4.5 DT). C4.5 DT is a non-parametric supervised learner consisting of decision and leaf nodes. In C4.5 DTs, the model is developed in a flowchart-like scheme where the search space is broken down into smaller subspaces while an associated decision tree is incrementally developed [92]. C4.5 DT has been employed in various predictive maintenance problems [93, 94].

The computation of C4.5 DT follows the concept of divide-and-conquer using a set of training data $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{n \times p}$ with a corresponding class vector $Y = (y_1, y_2, \dots, y_k)$. The feature space of the training data is recursively partitioned such that instances with the same class output are grouped together [95]. Suppose that P_k denotes the data at node k comprising of S_k instances, the splitting of each candidate consisting of a feature j and threshold t_k is achieved by partitioning the data into $P_k^{\text{left}}(\theta)$ and $P_k^{\text{right}}(\theta)$ subsets as shown in (57) and (58), respectively:

$$P_k^{\text{left}}(\theta) = \{(x, y) | x_j \leq t_k\}, \quad (57)$$

$$P_k^{\text{right}}(\theta) = \frac{P_k}{P_k^{\text{left}}(\theta)}, \quad (58)$$

$$\theta = (j, t_k).$$

Using (57) and (58), the quality of candidate split of node k is then estimated using

$$G(P_m, \theta) = \frac{S_k^{\text{left}}}{S_k} H(P_k^{\text{left}}(\theta)) + \frac{S_k^{\text{right}}}{S_k} H(P_k^{\text{right}}(\theta)), \quad (59)$$

where H is the impurity function expressed by either of the three common impurity measures shown in (60) to (62), respectively:

$$\text{Gini} \Rightarrow H(P_k) = \sum_n p_{kn}(1 - p_{kn}), \quad (60)$$

$$\text{Entropy} \Rightarrow H(P_k) = - \sum_n p_{kn} \log(p_{kn}), \quad (61)$$

$$\text{Misclassification} \Rightarrow H(P_k) = 1 - \max(p_{kn}), \quad (62)$$

where p_{kn} is the proportion of class n observations in node k expressed as

$$p_{kn} = \frac{1}{S_k} \sum_{y \in P_k} I(y = n). \quad (63)$$

The objective of the algorithm is to select parameters that minimise the impurity θ^* as shown in

$$\theta^* = \text{argmin}_{\theta} G(P_K, \theta). \quad (64)$$

The subsets $P_k^{\text{left}}(\theta^*)$ and $P_k^{\text{right}}(\theta^*)$ are recursed until the maximum allowable depth, $S_k < \min_{\text{samples}}$ or $S_k = 1$, is achieved.

3.3.5. Logistic Regression (LR). LR is a commonly used machine learning technique for classifying responses with binary outcomes. The LR model was originally proposed for modelling population growth, yet its application over the years is found in numerous disciplines due to its simplicity and effectiveness. Despite the simple nature of the classifier, it possesses some desirable characteristics such as high accuracy, interpretability, and less influence from error in data when optimally trained [96]. Hence, it is commonly used in stacked generalisation, mostly as a metalearner due to its easier interpretability characteristics [97–100].

Given a matrix $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{n \times p}$ with a corresponding response vector $Y = (y_1, y_2, \dots, y_k)$, the LR model aims to minimise the negative log-likelihood of the data expressed as

$$\min_{\beta} f(\beta), \quad (65)$$

where $f(\beta)$ is defined as

$$f(\beta) = \sum_{i=1}^n -y_i(\beta'x_i) + \log(1 + \exp(\beta'x_i)). \quad (66)$$

The regression coefficients are expressed as $\beta \in \mathbb{R}^{p \times 1}$, and assume that $\beta'x_i = \log(P(y_i = 1|x_i)/P(y_i = 0|x_i))$.

3.3.6. Linear Discriminant Analysis (LDA). LDA is among the predominantly used classifiers for predictive maintenance purposes [29, 31, 101]. Its concept exploits the statistical features such as the mean and covariance matrix of each class and then utilises mathematical processes and functions for classifying multiple classes. In LDA, the computation starts by calculating the within-class (67) and between-class (68) scatter matrices:

$$S_w = \sum_{i=1}^n \Pr(Y_i) \Sigma_i, \quad (67)$$

$$S_b = \sum_{i=1}^n \Pr(Y_i) (m_i - m)(m_i - m)^T. \quad (68)$$

However, Σ_i in (67) is the covariance matrix estimated using

$$\Sigma_i = E[(x - m_i)(x - m_i)^T]. \quad (69)$$

Hence, (67) is expanded as shown in

$$S_w = \sum_{i=1}^n \Pr(Y_i) E[(x - m_i)(x - m_i)^T], \quad (70)$$

where S_w and S_b are the scatter matrices within and between the classes, respectively, $\Pr(Y_i)$ is the priori probability of class Y_i , m_i and m are the mean of the different classes and grand mean, respectively, and x is the input vector.

Using (67) and (68), a discriminatory power value $j(\psi)$ is estimated for deriving a projection matrix ψ that maximizes the following equation:

$$j(\psi) = \frac{\|\psi^T S_b \psi\|}{\|\psi^T S_w \psi\|} \tag{71}$$

That is, (71) is maximised when ψ the eigenvectors of the matrix $S_w - 1S_b$ are estimated. This results in the linear discriminant function as shown in

$$d(x) = \psi^T \left(x - \frac{m}{\psi} \right) \tag{72}$$

3.4. Classification Performance Evaluation. For the purpose of reliability, nine evaluation metrics, namely, accuracy, error rate, sensitivity, specificity, precision, *F*-score, Matthews correlation coefficient (MCC), geometric mean, and area under curve (AUC) were used.

3.4.1. Accuracy. Accuracy is one of the most widely used evaluation metrics for assessing the performance of classification algorithms [102–104]. Classification accuracy is expressed as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{73}$$

where TP is the number of correct classification counts when there is a fault condition, TN is the number of correct classification counts when there is no fault condition, FP is the number of misclassification counts when there is a fault condition, and FN is the number of misclassification counts when there is no fault condition for a specific classification model.

3.4.2. Error Rate. This metric is one of the main indicators for evaluating classification performance by measuring the errors (misclassifications) incurred by a classifier [104–107]. The error rate is expressed as shown in

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{74}$$

3.4.3. Precision. The precision of a classifier measures the exactness of classification after prediction [104, 107, 108]. The evaluation metric is expressed as a ratio of true positives (TP) to the sum of true positives (TP) and false positives (FP) as shown in

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{75}$$

3.4.4. Sensitivity. Recall, also known as sensitivity, is a measure of a classifier’s capacity to determine positive instances. It measures the fraction of positive instances that are correctly classified [104, 108, 109] and is expressed as a ratio of true positives (TP) to the sum of true positives (TP) and false negatives (FN) as shown in

$$\frac{\text{Recall}}{\text{Sensitivity}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{76}$$

3.4.5. Specificity. Specificity measures the fraction of negative instances that are correctly classified [108, 110]. That is, the metric denotes the test’s ability to identify negative results as expressed in

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{77}$$

3.4.6. F-Score. The *F*-score describes the overall performance of a classification model as the harmonic mean of precision and recall [104, 107, 108]. The metric ranges from zero to one, with high values indicating high classification performance and vice versa. *F*-score is given in

$$F\text{Score} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \tag{78}$$

3.4.7. Matthews Correlation Coefficient (MCC). MCC (79) measures the relationship between the observed and the predicted classification and is generally regarded as a balanced metric for evaluating the performance of classifiers even with varying class sizes [111, 112]. MCC, as compared to other classification evaluation metrics, is known to be more informative as it considers the balance ratios of the four confusion matrix categories. An MCC coefficient ranges from -1 to $+1$, with $+1$ suggesting a perfect classification while -1 implies a total misclassification:

$$MCC = \frac{(\text{TP} * \text{TN}) - (\text{FP} * \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{79}$$

3.4.8. Geometric Mean (GM). GM (80) is an aggregate of both sensitivity and specificity evaluation metrics. The metric seeks to maximise the rate of true positive and negative instances while maintaining a balance between both rates [113, 114], thus making the metric suitable for imbalanced datasets:

$$GM = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} * \frac{\text{TN}}{\text{TN} + \text{FP}}} \tag{80}$$

3.4.9. Area under Curve (AUC). In evaluating the classification performance of classifiers, metrics such as accuracy, specificity, and precision are prone to the changes in the class distribution of the test data; thus, they are not always robust [115]. That is, they may underperform when the ratio of positive to negative instances changes. For this reason, the AUC [116] metric which is insensitive to changes in class distribution (nonparametric) is also employed. The AUC statistic is expressed as shown in

$$\text{AUC} = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} I(x_i > y_j), \quad (81)$$

where N_+ and N_- is the number of positive and negative instances, respectively. x_i ($i = 1, 2, \dots, N_+$) is the scores predicted by the model for the N_+ whilst y_j ($j = 1, 2, \dots, N_-$) is the scores predicted by the model for the N_- . $I(\cdot)$ is an indicator function satisfying the condition $I(\text{true}) = 1$ and $I(\text{false}) = 0$.

4. Results and Discussion

4.1. Fuzzy Rough Set Feature Selection (FRFS). The FRFS experiment was carried out in the R statistical environment. The algorithms were executed on an i3 ASUS computer with a 2.3 GHz processor and 16 GB memory. Table 2 shows the number of features selected with the six FRFS methods compared in this study. As observed, feature(s) ranging from 1 to 27 were deemed relevant to have contained the required variability in classifying accumulator conditions. Similarly, 1 to 8 feature(s) were selected for the classification of cooler conditions. Feature(s) ranging from 1 to 21 were noted as informative enough for classifying pump conditions by the various FRFS methods. Concerning the valve conditions, 1 to 5 feature(s) were selected. These selected features by the various FRFS methods were noted to contain enough variability for explaining the conditions of the hydraulic system. This subsequently will improve classification accuracy while reducing the computational cost, overfitting, training runtime, and uncertainty in modelling. However, regarding the number of features selected, the FLA, FBR, FOWA, and FVPRS produced on average 15 features while VQRS and BPFERS produced the least average selected feature(s) of 1 and 2, respectively.

Figure 7 shows the respective runtime during the implementation of the various FRFS methods for feature selection. As observed in Figure 7(a), the highest runtime range of 1884.98 to 18497.54 seconds was recorded by the various FRFS methods in producing the most informative feature(s) for classifying accumulator fault conditions while a minimum range of 1069.64 to 4141.75 seconds was recorded during valve conditions (Figure 7(d)). The high runtime during the FRFS implementation on the accumulator dataset suggests the high complexity and the high degree of uncertainty of the data. This supports the assertion made by prior works that the classification of accumulator conditions is noted as the most challenging among the monitored hydraulic components (accumulator, cooler, pump, and valve) [29, 44]. Similar but lesser complexity and degree of uncertainty were also seen in the pump (Figure 7(c)) and cooler (Figure 7(b)) datasets, respectively.

4.2. Optimal FRFS Method for Various Fault Types. Based on the varying complexity and uncertainty levels indicated by the datasets of the four monitored hydraulic components (accumulator, cooler, pump, and valve), it is prudent to ensure that the optimal FRFS method for selecting the most informative features for adequate classification is employed.

TABLE 2: Selected feature(s) from various FRFS methods.

| FRFS method | Feature(s) selected | | | | Average |
|-------------|---------------------|--------|------|-------|---------|
| | Accumulator | Cooler | Pump | Valve | |
| FLA | 27 | 8 | 21 | 5 | 15 |
| FBR | 27 | 8 | 21 | 5 | 15 |
| VQRS | 4 | 1 | 1 | 1 | 2 |
| FOWA | 27 | 8 | 21 | 5 | 15 |
| FVPRS | 26 | 7 | 20 | 5 | 15 |
| BPFERS | 1 | 1 | 1 | 1 | 1 |

For this reason, the selected features from each FRFS method for the various fault conditions were used as inputs for training the proposed stacked ensemble (ESB_(STK)) in order to ascertain their performance.

However, before training, the selected features were split into train and test sets with a ratio of 70 : 30. The train set was used for model development while the remaining test was used for model validation purposes. Due to the stochastic learning characteristics of the base classifiers used for the stacked ensemble, slightly different output may be achieved after each run. As a result, the proposed stacked ensemble (ESB_(STK)) was trained 10 times and the final test output is presented as the average from the 10 runs. Finally, to ensure that the optimal hyperparameters for the base classifiers as well as the generaliser are set, the random grid search algorithm [117] was implemented. Tables 3 to 6 show the test performance of the feature(s) selected from the various FRFS methods used as inputs for classifying the accumulator, cooler, pump, and valve conditions.

Table 3 shows the test performance of the various FRFS methods when their corresponding extracted features were used as inputs to classify accumulator conditions. As observed, all the competing FRFS methods except for BPFERS produced satisfactory classification accuracies greater than 90%. In comparison (Table 3), the VQRS proved superior in classifying the accumulator conditions to the other explored FRFS methods. The smaller number of four features (refer to Table 2) selected by the VQRS implies less complexity and runtime during the learning and implementation phase of the proposed stacked ensemble. Thus, comparing the FRFS methods (excluding BPFERS), the VQRS was executed with a low runtime of 3989.43 seconds (Figure 7(a)). It can be further deduced from the results (Table 3) that the high performance, low runtime, and low features selected by VQRS suggest that the method could produce optimal and most informative features for the adequate classification of accumulator fault conditions. Thus, the VQRS is chosen as the optimal FRFS method when it comes to dealing with the imperfections or uncertainty within the accumulator dataset.

Unlike the complex nature of the accumulator dataset, the complexity of classifying the cooler conditions was less. This was evident by the low minimum runtime range of 1554.47 to 6462.43 seconds during the feature selection phase by the FRFS methods (Figure 7(b)). Likewise, the variability in cooler conditions (class output) could be adequately explained by a low number of features (i.e., from 1 to 8 features, out of the pool of features). The less complex

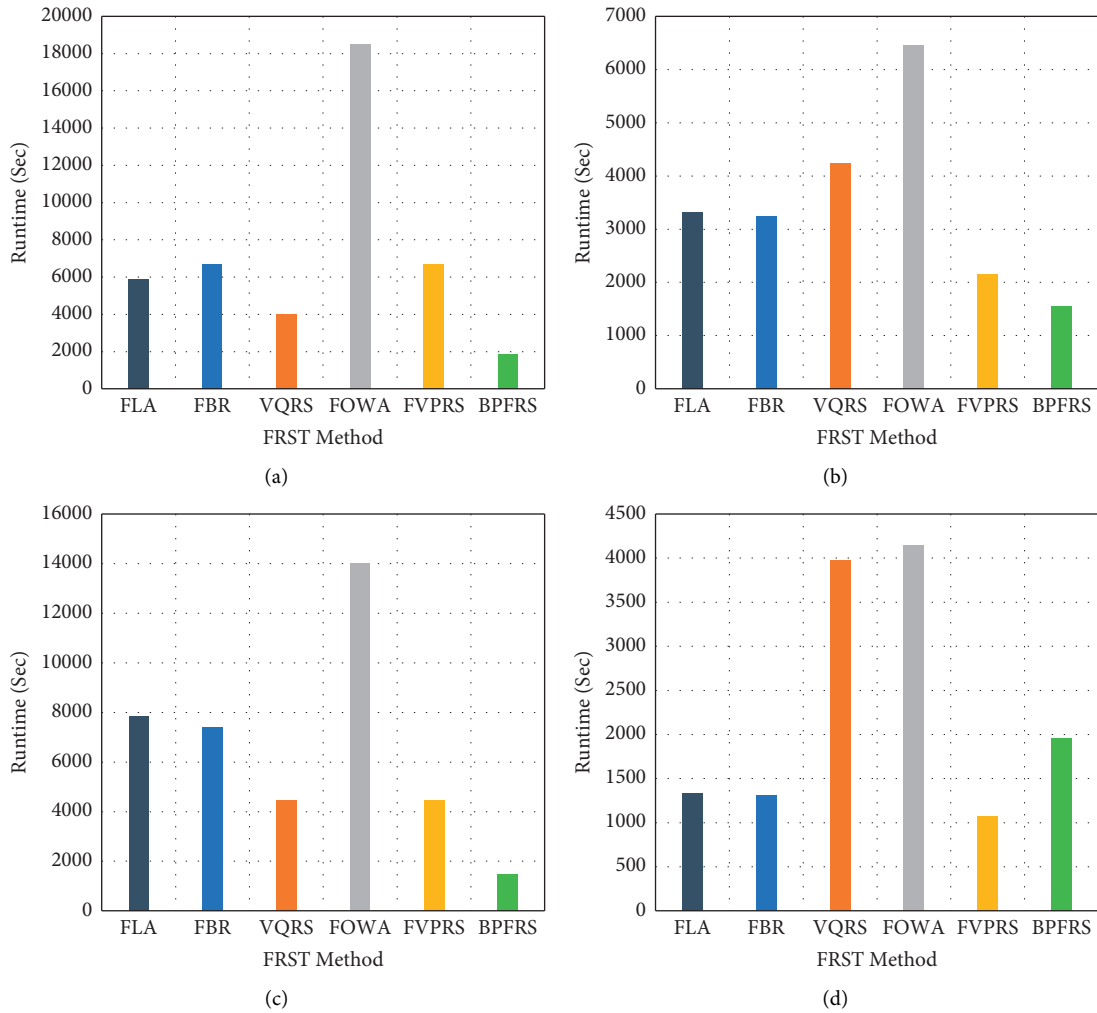


FIGURE 7: Runtime of FRFS methods for feature selection. (a) Accumulator. (b) Cooler. (c) Pump. (d) Valve.

TABLE 3: Classification performance of FRFS methods on accumulator conditions.

| Metric | FLA | FBR | VQRS | FOWA | FVPRS | BPFERS |
|-------------|--------|--------|--------|--------|--------|--------|
| Accuracy | 0.9281 | 0.9092 | 0.9743 | 0.9370 | 0.9231 | 0.5393 |
| Error | 0.0719 | 0.0908 | 0.0257 | 0.0630 | 0.0769 | 0.4607 |
| Sensitivity | 0.9122 | 0.8879 | 0.9688 | 0.9222 | 0.9104 | 0.4260 |
| Specificity | 0.9765 | 0.9692 | 0.9917 | 0.9794 | 0.9746 | 0.8250 |
| Precision | 0.9162 | 0.9000 | 0.9695 | 0.9250 | 0.9120 | 0.4503 |
| F-score | 0.9140 | 0.8931 | 0.9691 | 0.9235 | 0.9111 | 0.4378 |
| MCC | 0.8909 | 0.8640 | 0.9608 | 0.9033 | 0.8859 | 0.2797 |
| GM | 0.9438 | 0.9277 | 0.9801 | 0.9504 | 0.9420 | 0.5928 |
| AUC | 0.9910 | 0.9875 | 0.9951 | 0.9926 | 0.9884 | 0.6937 |

TABLE 4: Classification performance of FRFS methods on cooler conditions.

| Metric | FLA | FBR | VQRS | FOWA | FVPRS | BPFERS |
|-------------|--------|--------|--------|--------|--------|--------|
| Accuracy | 0.9979 | 1.0000 | 0.9852 | 1.0000 | 1.0000 | 1.0000 |
| Error | 0.0021 | 0.0000 | 0.0148 | 0.0000 | 0.0000 | 0.0000 |
| Sensitivity | 0.9979 | 1.0000 | 0.9852 | 1.0000 | 1.0000 | 1.0000 |
| Specificity | 0.9989 | 1.0000 | 0.9926 | 1.0000 | 1.0000 | 1.0000 |
| Precision | 0.9979 | 1.0000 | 0.9853 | 1.0000 | 1.0000 | 1.0000 |
| F-score | 0.9979 | 1.0000 | 0.9852 | 1.0000 | 1.0000 | 1.0000 |
| MCC | 0.9968 | 1.0000 | 0.9779 | 1.0000 | 1.0000 | 1.0000 |
| GM | 0.9984 | 1.0000 | 0.9889 | 1.0000 | 1.0000 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 0.9973 | 1.0000 | 1.0000 | 1.0000 |

TABLE 5: Classification performance of FRFS methods on pump conditions.

| Metric | FLA | FBR | VQRS | FOWA | FVPRS | BPFRS |
|-------------|--------|--------|--------|--------|--------|--------|
| Accuracy | 0.9985 | 0.9940 | 0.9955 | 0.9970 | 1.0000 | 0.9849 |
| Error | 0.0015 | 0.0060 | 0.0045 | 0.0030 | 0.0000 | 0.0151 |
| Sensitivity | 0.9977 | 0.9910 | 0.9932 | 0.9955 | 1.0000 | 0.9775 |
| Specificity | 0.9994 | 0.9965 | 0.9976 | 0.9987 | 1.0000 | 0.9930 |
| Precision | 0.9978 | 0.9937 | 0.9946 | 0.9956 | 1.0000 | 0.9788 |
| F-score | 0.9977 | 0.9923 | 0.9939 | 0.9955 | 1.0000 | 0.9781 |
| MCC | 0.9971 | 0.9893 | 0.9917 | 0.9942 | 1.0000 | 0.9714 |
| GM | 0.9985 | 0.9937 | 0.9954 | 0.9971 | 1.0000 | 0.9852 |
| AUC | 0.9980 | 0.9974 | 0.9980 | 1.0000 | 1.0000 | 0.9935 |

TABLE 6: Classification performance of FRFS methods on valve conditions.

| Metric | FLA | FBR | VQRS | FOWA | FVPRS | BPFRS |
|-------------|--------|--------|--------|--------|--------|--------|
| Accuracy | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Error | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Sensitivity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Specificity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Precision | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| F-score | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| MCC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| GM | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

nature is further ascertained as most of the FRFS methods except for FLA and VQRS produced perfect test classification scores with 100% accuracy and AUC scores as shown in Table 4. Hence, any of the four FRFS methods (i.e., FBR, FOWA, FVPRS, and BPFRS) could be utilised as a feature selection method on the cooler dataset. However, for industrial applications, a high-performance system with the least computational cost and low runtime is highly preferred when developing frameworks for predictive maintenance purposes. On that note, the BPFRS having achieved high classification performance and the least runtime (1554.47 seconds in Figure 7(b)) is selected as the optimal method for classifying cooler conditions.

A similar data complexity observed in the accumulator was seen in the pump dataset. As a result, a high runtime ranging from 1484.64 to 14026.72 seconds (Figure 7(c)) was recorded during the implementation of the FRFS methods for feature selection. Although all the FRFS methods yielded satisfactory test classification results (Table 5), the FVPRS achieved a perfect accuracy and an AUC score of 100%. For this reason, the FVPRS is chosen as the optimal FRFS method for addressing the imperfections or uncertainty in the internal pump leakage dataset.

With regards to the classification of the valve conditions based on features selected by the various FRFS methods, all the methods accurately classified the reserved test dataset as shown in Table 6. This implies that the variability in valve conditions could be explained adequately by any of the feature(s) selected by the FRFS methods. However, based on minimum runtime and less complexity in learning, the FVPRS which recorded the least implementation runtime (1069.64 seconds in Figure 7(d)) is chosen to be the optimal method in the case of the valve dataset.

4.3. Comparison of Proposed Stacked Ensemble ($ESB_{(STK)}$) with Existing Classifiers. In order to ascertain the generalisability of the proposed stacked ensemble ($ESB_{(STK)}$), a comparative analysis is performed on the six base classifiers (standalone) and three existing ensemble classifiers. The existing ensembles compared with the proposed $ESB_{(STK)}$ are the Stochastic Gradient Boosting (SGB) [118], AdaBoost (ADB) [119], and Bagging (BAG) [120]. In all, the proposed $ESB_{(STK)}$ was compared with 9 well-established classifiers that are commonly used in numerous fields of autonomous systems such as pattern recognition, system monitoring, and fault identification, among others. Tables 7–10 show the classification test results for the four hydraulic components. It is important to note that the selected optimal FRFS methods for classifying the various conditions of the four monitored hydraulic components were used. Here, the VQRS selected features were used as input to train the nine classifiers (SVM, MLP, KNN, C4.5 DT, LR, LDA, SGB, ADB, and BAG) for classifying the operational conditions of the accumulator. Similarly, BPFRS selected features were used as input to classify cooler conditions. For the pump and valve, the FVPRS selected features were used in the aforementioned classifiers. For the purposes of clarity, the FRFS method that produced the optimal features extracted for each hydraulic component will be indicated with the stacked ensemble, $ESB_{(STK)}$.

Table 7 shows the classification performance of the proposed VQRS- $ESB_{(STK)}$ with the competing classifiers on accumulator conditions. As observed, though almost all the classifiers under consideration yielded satisfactory results, the proposed VQRS- $ESB_{(STK)}$ achieved the highest classification results in terms of accuracy (0.9743), AUC (0.9951), least error rate (0.0257), and the remaining performance

TABLE 7: Performance of VQRS-ESB_(STK) with competing classifiers on accumulator conditions.

| Metrics | Standalone | | | | | | Existing ensemble classifier | | | VQRS-ESB _(STK) |
|-------------|------------|--------|--------|---------|--------|--------|------------------------------|--------|--------|---------------------------|
| | SVM | MLP | KNN | C4.5 DT | LR | LDA | SGB | ADB | BAG | |
| Accuracy | 0.9698 | 0.9547 | 0.9653 | 0.9370 | 0.7568 | 0.7725 | 0.9637 | 0.4773 | 0.9562 | 0.9743 |
| Error | 0.0302 | 0.0453 | 0.0347 | 0.0630 | 0.2432 | 0.2275 | 0.0363 | 0.5227 | 0.0438 | 0.0257 |
| Sensitivity | 0.9636 | 0.9490 | 0.9577 | 0.9259 | 0.7018 | 0.7179 | 0.9570 | 0.5245 | 0.9434 | 0.9688 |
| Specificity | 0.9902 | 0.9852 | 0.9888 | 0.9792 | 0.9182 | 0.9254 | 0.9883 | 0.8380 | 0.9852 | 0.9917 |
| Precision | 0.9646 | 0.9476 | 0.9580 | 0.9282 | 0.7385 | 0.7404 | 0.9570 | 0.5720 | 0.9526 | 0.9695 |
| F-score | 0.9640 | 0.9478 | 0.9577 | 0.9269 | 0.6958 | 0.7121 | 0.9570 | 0.4741 | 0.9475 | 0.9691 |
| MCC | 0.9542 | 0.9333 | 0.9466 | 0.9063 | 0.6342 | 0.6518 | 0.9452 | 0.3744 | 0.9336 | 0.9608 |
| GM | 0.9768 | 0.9669 | 0.9731 | 0.9522 | 0.8027 | 0.8151 | 0.9725 | 0.6630 | 0.9641 | 0.9801 |
| AUC | 0.9918 | 0.9894 | 0.9919 | 0.9525 | 0.9170 | 0.9188 | 0.9926 | 0.8184 | 0.9916 | 0.9951 |

TABLE 8: Performance of FVPRS-ESB_(STK) with competing classifiers on pump conditions.

| Metrics | Standalone | | | | | | Existing ensemble classifier | | | FVPRS-ESB _(STK) |
|-------------|------------|--------|--------|---------|--------|--------|------------------------------|--------|--------|----------------------------|
| | SVM | MLP | KNN | C4.5 DT | LR | LDA | SGB | ADB | BAG | |
| Accuracy | 0.9955 | 0.9955 | 0.9940 | 0.9970 | 0.9773 | 0.9917 | 0.9985 | 0.9758 | 0.9955 | 1.0000 |
| Error | 0.0045 | 0.0045 | 0.0060 | 0.0030 | 0.0227 | 0.0083 | 0.0015 | 0.0242 | 0.0045 | 0.0000 |
| Sensitivity | 0.9973 | 0.9959 | 0.9923 | 0.9955 | 0.9662 | 0.9880 | 0.9977 | 0.9640 | 0.9946 | 1.0000 |
| Specificity | 0.9981 | 0.9981 | 0.9974 | 0.9987 | 0.9903 | 0.9964 | 0.9994 | 0.9896 | 0.9981 | 1.0000 |
| Precision | 0.9934 | 0.9934 | 0.9912 | 0.9955 | 0.9679 | 0.9878 | 0.9978 | 0.9659 | 0.9934 | 1.0000 |
| F-score | 0.9953 | 0.9946 | 0.9917 | 0.9955 | 0.9662 | 0.9879 | 0.9977 | 0.9639 | 0.9939 | 1.0000 |
| MCC | 0.9927 | 0.9922 | 0.9889 | 0.9942 | 0.9572 | 0.9842 | 0.9971 | 0.9544 | 0.9918 | 1.0000 |
| GM | 0.9977 | 0.9970 | 0.9949 | 0.9971 | 0.9782 | 0.9922 | 0.9985 | 0.9767 | 0.9963 | 1.0000 |
| AUC | 1.0000 | 0.9979 | 1.0000 | 0.9975 | 0.9971 | 0.9964 | 1.0000 | 0.9891 | 1.0000 | 1.0000 |

TABLE 9: Performance of BPFRS-ESB_(STK) with competing classifiers on cooler conditions.

| Metrics | Standalone | | | | | | Existing ensemble classifier | | | BPFRS-ESB _(STK) |
|-------------|------------|--------|--------|---------|--------|--------|------------------------------|--------|--------|----------------------------|
| | SVM | MLP | KNN | C4.5 DT | LR | LDA | SGB | ADB | BAG | |
| Accuracy | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Error | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Sensitivity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Specificity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Precision | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| F-score | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| MCC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| GM | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

TABLE 10: Performance of FVPRS-ESB_(STK) with competing classifiers on valve conditions.

| Metrics | Standalone | | | | | | Existing ensemble classifier | | | FVPRS-ESB _(STK) |
|-------------|------------|--------|--------|---------|--------|--------|------------------------------|--------|--------|----------------------------|
| | SVM | MLP | KNN | C4.5 DT | LR | LDA | SGB | ADB | BAG | |
| Accuracy | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9015 | 1.0000 | 0.8384 | 1.0000 | 1.0000 |
| Error | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0985 | 0.0000 | 0.1616 | 0.0000 | 0.0000 |
| Sensitivity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.8491 | 1.0000 | 0.7523 | 1.0000 | 1.0000 |
| Specificity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9706 | 1.0000 | 0.9517 | 1.0000 | 1.0000 |
| Precision | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9229 | 1.0000 | 0.8756 | 1.0000 | 1.0000 |
| F-score | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.8010 | 1.0000 | 0.6718 | 1.0000 | 1.0000 |
| MCC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.8067 | 1.0000 | 0.6812 | 1.0000 | 1.0000 |
| GM | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9078 | 1.0000 | 0.8462 | 1.0000 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9513 | 1.0000 | 1.0000 |

TABLE 11: Impact of base classifiers on the performance of VQRS-ESB_(STK) on accumulator conditions.

| Metrics | VQRS-ESB _(STK) | ESB _(STK) -base classifier (i.e., Modified ESB _(STK)) | | | | | |
|-------------|---------------------------|--|---------------------------|---------------------------|-------------------------------|--------------------------|---------------------------|
| | | ESB _(STK) -SVM | ESB _(STK) -MLP | ESB _(STK) -KNN | ESB _(STK) -C4.5 DT | ESB _(STK) -LR | ESB _(STK) -LDA |
| Accuracy | 0.9743 | 0.9727 | 0.9600 | 0.9592 | 0.9698 | 0.9644 | 0.9677 |
| Error | 0.0257 | 0.0273 | 0.0400 | 0.0408 | 0.0302 | 0.0356 | 0.0323 |
| Sensitivity | 0.9688 | 0.9667 | 0.9531 | 0.9533 | 0.9656 | 0.9550 | 0.9640 |
| Specificity | 0.9917 | 0.9910 | 0.9870 | 0.9862 | 0.9899 | 0.9882 | 0.9890 |
| Precision | 0.9695 | 0.9683 | 0.9527 | 0.9559 | 0.9664 | 0.9595 | 0.9653 |
| F-score | 0.9691 | 0.9675 | 0.9529 | 0.9545 | 0.9660 | 0.9571 | 0.9646 |
| MCC | 0.9608 | 0.9586 | 0.9399 | 0.9410 | 0.9560 | 0.9457 | 0.9538 |
| GM | 0.9801 | 0.9788 | 0.9699 | 0.9696 | 0.9777 | 0.9715 | 0.9764 |
| AUC | 0.9951 | 0.9944 | 0.9925 | 0.9888 | 0.9385 | 0.9923 | 0.9925 |

metrics. This was followed by the SVM (0.9698), KNN (0.9653), and SGB (0.9637), respectively. On average, the proposed VQRS-ESB_(STK) gains an improvement in accuracy and AUC of about 11.28% and 4.35%, respectively.

Similar performance (Table 8) was obtained for FVPRS-ESB_(STK) for classifying the internal pump conditions. It can be seen that the proposed FVPRS-ESB_(STK) produced the best classification results as compared with the other competing classifiers. Among the compared classifiers, SVM, KNN, SGB, and BAG produced relatively similar results as the proposed FVPRS-ESB_(STK) in terms of AUC scores (1.0000). This suggests that the SVM, KNN, SGB, and BAG classifiers could be considered alternatives to the proposed ESB_(STK) when considering the probability thresholds of the classification results.

Regarding the cooler and valve conditions classification results (Tables 9 and 10), very competitive outputs were recorded for all the standalone classifiers, ensemble classifiers, and the proposed BPFERS-ESB_(STK) and FVPRS-ESB_(STK), respectively. In comparison, all the considered classifiers accurately classified the conditions of the cooling system of the hydraulic machinery. Similar competitive classification except for LDA and ADB classifiers was witnessed when classifying the valve conditions. These high accuracies and competitive results could be attributed to the less complicated variability in the class outputs of the cooler and valve conditions as discussed earlier in Section 4.1.

Taking into account the results from Tables 7–10, it can be ascertained that none of the standalone classifiers which served as the based learners in the stacked ensemble achieved the best classification across all monitored hydraulic components (accumulator, cooler, pump, and valve). However, the proposed framework of FRFS method coupled with stacked ensemble (ESB_(STK)) has proven to be generalisable and versatile in the classification of all conditions of the various hydraulic components considered. Based on the generalisability and versatility of the proposed framework, one can opt for the proposed framework as opposed to investing resources into determining which standalone classifier to select for a specific task.

4.4. Impact of Base Classifiers on the Proposed ESB_(STK). In building a stacked ensemble, the selection of appropriate base classifiers is critical for ensuring goodness of fit. For this

reason, it is imperative to ascertain how the proposed stacked ensemble (ESB_(STK)) is affected by the quality of base classifiers. This objective is achieved by dropping each base classifier at a time and then assessing its classification performance on the four hydraulic components (accumulator, pump, cooler, and valve). Using this approach, the base classifiers with high impact on the overall performance of the proposed ESB_(STK) were determined. Here, the developed stacked ensemble is defined as modified ESB_(STK).

Table 11 presents the impact of base classifiers on the overall performance of the proposed VQRS-ESB_(STK) in classifying accumulator conditions. As observed, all the base classifiers contribute to the overall performance of ESB_(STK) since none of the modified ESB_(STK) achieved the same performance as the proposed VQRS-ESB_(STK). This can be confirmed in Table 11 where a maximum deterioration in accuracy (1.55%) is observed when KNN is dropped, followed by MLP (1.47%), LR (1.02%), LDA (0.68%), C4.5 DT (0.46%), and SVM (0.16%), respectively. This suggests the relative contribution made by each of the base classifiers to the overall VQRS-ESB_(STK). A similar observation is seen in the case of the pump conditions where the classification performance is reduced when each base learner is dropped (Table 12).

In the classification of valve conditions (Table 13), all the modified ESB_(STK) showed no improvement after dropping any of the base classifiers. Similar results were observed for cooler conditions (Table 14); however, a 0.15% deterioration in accuracy was recorded when either MLP or LR was dropped from the proposed BPFERS-ESB_(STK).

The summary of the relative impacts of the base classifiers on the proposed ESB_(STK) in classifying the condition of the four monitored hydraulic components is shown in Figure 8. Using the accuracy estimates from Tables 11 to 14, Figure 8 was generated with the relative impact of each base learner (i.e., the difference in accuracy between the proposed FRFS-ESB_(STK) and the Modified ESB_(STK)). A critical observation of Figure 8 ascertains that the impact of each base classifier varies among the hydraulic components. The highest impact was seen when classifying the conditions of the accumulator followed by the internal pump leakage. Lesser impact was observed for cooler conditions whilst no impact was seen for valve classification when any of the base classifiers was removed. Figure 8 has proven the proposed ESB_(STK), to be generalisable and versatile in the classification of all conditions of the various hydraulic components.

TABLE 12: Impact of base classifiers on the performance of FVPRS-ESB_(STK) on pump conditions.

| Metrics | FVPRS-ESB _(STK) | ESB _(STK) -base classifier (i.e., Modified ESB _(STK)) | | | | | |
|-------------|----------------------------|--|---------------------------|---------------------------|-------------------------------|--------------------------|---------------------------|
| | | ESB _(STK) -SVM | ESB _(STK) -MLP | ESB _(STK) -KNN | ESB _(STK) -C4.5 DT | ESB _(STK) -LR | ESB _(STK) -LDA |
| Accuracy | 1.0000 | 0.9937 | 0.9955 | 0.9964 | 0.9970 | 0.9953 | 0.9985 |
| Error | 0.0000 | 0.0063 | 0.0045 | 0.0036 | 0.0030 | 0.0047 | 0.0015 |
| Sensitivity | 1.0000 | 0.9907 | 0.9934 | 0.9946 | 0.9958 | 0.9930 | 0.9977 |
| Specificity | 1.0000 | 0.9963 | 0.9974 | 0.9980 | 0.9987 | 0.9976 | 0.9989 |
| Precision | 1.0000 | 0.9933 | 0.9950 | 0.9960 | 0.9955 | 0.9941 | 0.9991 |
| F-score | 1.0000 | 0.9920 | 0.9942 | 0.9953 | 0.9956 | 0.9935 | 0.9984 |
| MCC | 1.0000 | 0.9888 | 0.9919 | 0.9935 | 0.9943 | 0.9914 | 0.9975 |
| GM | 1.0000 | 0.9935 | 0.9954 | 0.9963 | 0.9972 | 0.9953 | 0.9983 |
| AUC | 1.0000 | 0.9963 | 0.9977 | 0.9979 | 1.0000 | 0.9974 | 1.0000 |

TABLE 13: Impact of base classifiers on the performance of FVPRS-ESB_(STK) on valve conditions.

| Metrics | FVPRS-ESB _(STK) | ESB _(STK) -base classifier (i.e., Modified ESB _(STK)) | | | | | |
|-------------|----------------------------|--|---------------------------|---------------------------|-------------------------------|--------------------------|---------------------------|
| | | ESB _(STK) -SVM | ESB _(STK) -MLP | ESB _(STK) -KNN | ESB _(STK) -C4.5 DT | ESB _(STK) -LR | ESB _(STK) -LDA |
| Accuracy | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Error | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Sensitivity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Specificity | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Precision | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| F-score | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| MCC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| GM | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

TABLE 14: Impact of base classifiers on the performance of BPFRRS-ESB_(STK) on cooler conditions.

| Metrics | BPFRRS-ESB _(STK) | ESB _(STK) -base classifier (i.e., Modified ESB _(STK)) | | | | | |
|-------------|-----------------------------|--|---------------------------|---------------------------|-------------------------------|--------------------------|---------------------------|
| | | ESB _(STK) -SVM | ESB _(STK) -MLP | ESB _(STK) -KNN | ESB _(STK) -C4.5 DT | ESB _(STK) -LR | ESB _(STK) -LDA |
| Accuracy | 1.0000 | 1.0000 | 0.9985 | 1.0000 | 1.0000 | 0.9985 | 1.0000 |
| Error | 0.0000 | 0.0000 | 0.0015 | 0.0000 | 0.0000 | 0.0015 | 0.0000 |
| Sensitivity | 1.0000 | 1.0000 | 0.9985 | 1.0000 | 1.0000 | 0.9985 | 1.0000 |
| Specificity | 1.0000 | 1.0000 | 0.9992 | 1.0000 | 1.0000 | 0.9992 | 1.0000 |
| Precision | 1.0000 | 1.0000 | 0.9985 | 1.0000 | 1.0000 | 0.9985 | 1.0000 |
| F-score | 1.0000 | 1.0000 | 0.9985 | 1.0000 | 1.0000 | 0.9985 | 1.0000 |
| MCC | 1.0000 | 1.0000 | 0.9977 | 1.0000 | 1.0000 | 0.9977 | 1.0000 |
| GM | 1.0000 | 1.0000 | 0.9989 | 1.0000 | 1.0000 | 0.9989 | 1.0000 |
| AUC | 1.0000 | 1.0000 | 0.9985 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

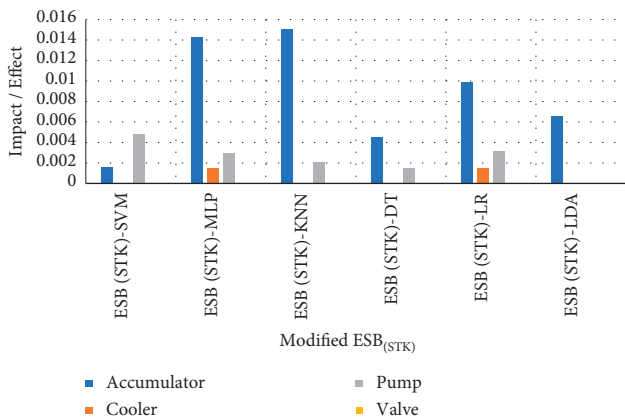


FIGURE 8: Summarised impact of base classifiers on ESB_(STK).

Although the proposed FRFS-ESB_(STK) framework proved superior in classifying the operational conditions of the hydraulic components, the implementation of FRFS is highly constrained by the available computational resource and the number of input features. Thus, the selection of relevant features using FRFS may be challenging when implemented on high dimensional datasets, especially with the unavailability of substantial computational resources such as memory and space. For these reasons, future works should explore FRFS alternatives with lesser computational demands or autonomous feature selectors with no human intervention. Also, regardless of the performance of ESB_(STK), the concept of stacking assumes that each base learner contributes equally to the ensemble classification. However, the discussion from Section 4.4 indicates otherwise. Hence, future works should explore stacking with

varying weighing schemes. This will ensure that high performing base learners contribute more while less performing contribute less to the ensemble classification.

5. Conclusion

A systematic and synergistic PdM framework based on the hybrid multisensor FRFS and the stacked ensemble ($ESB_{(STK)}$) has been proposed for the efficient classification of fault conditions characterised by uncertainties. The proposed hybrid framework (FRFS- $ESB_{(STK)}$) improved the classification accuracy with optimal feature subset size while reducing the computational cost, overfitting, training runtime, and uncertainty in modelling. It was however observed that the optimal FRFS method was task-specific, that is, no individual FRFS method produced an optimal number of features for all fault conditions due to their respective degree of complexity and uncertainty. It was also observed that no single base classifier (independent) achieved the best classification results across all the monitored hydraulic components as frequently observed in research works in fault classification. However, the proposed FRFS- $ESB_{(STK)}$ framework proved to be generalisable and versatile in the classification of the condition of all the hydraulic components considered. Based on the results obtained, it was concluded that one can opt for the proposed framework as opposed to investing resources into determining which standalone classifier to select for a specific task since the impact of the base classifiers varied among the hydraulic components. In summary, the proposed hybrid framework (FRFS- $ESB_{(STK)}$) advances the field of PdM by effectively and efficiently fusing recorded data from multiple and varying sensors. Thus, the proposed FRFS- $ESB_{(STK)}$ framework has enormous potential for enhancing the reliability of autonomous systems during an unexpected sensor failure. This enhances the robustness and confidence in fault detections as well expanding the sensitivity and specificity ability of systems to efficiently discriminate between operating conditions. Despite the high classification rate of the proposed FRFS- $ESB_{(STK)}$, the framework may be limited by computational resources for FRFS implementation and the nonvarying weighting scheme of $ESB_{(STK)}$. Hence, future works should explore alternative feature selectors with no human intervention and stacked ensembles with varying weighing schemes.

Data Availability

The data used to support the study can be accessed publicly at the University of California Irvine (UCI) machine learning repository via <http://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems>.

Disclosure

This manuscript is an extension of prior studies by Buabeng et al. [45] and a preprint available at Research Square, Buabeng et al. [46] which can currently be accessed via

<https://www.researchsquare.com/article/rs-600110/v1.pdf?c=1631887023000>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Albert Buabeng and Yao Yevenyo Ziggah contributed to the conceptualization of the research and the development of the methodology. Albert Buabeng performed the analysis and wrote the original draft. Anthony Simons, Nana Kena Frempong, and Yao Yevenyo Ziggah reviewed and suggested comments to improve the standard of the manuscript. All the authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research is part of a PhD research funded by the Staff Development Policy at the University of Mines and Technology, Tarkwa, Ghana.

References

- [1] G. Sinsley, "Distributed data fusion across multiple hard and soft mobile sensor platforms," 2012, <https://etda.libraries.psu.edu/catalog/16157>.
- [2] W. A. Abdulhafiz and A. Khamis, "Handling data uncertainty and inconsistency using multisensor data fusion," *Advances In Artificial Intelligence*, vol. 2013, Article ID 241260, 2013, <https://www.hindawi.com/journals/aai/2013/241260>.
- [3] R. Nowak, R. Biedrzycki, and J. Misiurewicz, "Machine learning methods in data fusion systems," in *Proceedings of the 2012 13th International Radar Symposium*, pp. 400–405, Warsaw, Poland, May 2012.
- [4] S. B. Ayed, H. Trichili, and A. M. Alimi, "Data fusion architectures: a survey and comparison," in *Proceedings of the 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 277–282, Marrakech, Morocco, December 2015.
- [5] A. Alofi, A. Alghamdi, R. Alahmadi, N. Aljuaid, and M. Hemalatha, "A review of data fusion techniques," *International Journal of Computer Application*, 2017.
- [6] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, "A survey on machine learning for data fusion," *Information Fusion*, vol. 57, pp. 115–129, 2020.
- [7] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [8] Z. Pawlak, "Rough sets: theoretical aspects of reasoning about data," *Springer Science & Business Media*, vol. 9, 2012.
- [9] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.
- [10] L. Yong, X. Congfu, and P. Yunhe, "A new approach for data fusion: implement rough set theory in dynamic objects distinguishing and tracing," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, pp. 3318–3322, The Hague, Netherlands, October 2004.

- [11] R. Jensen, *Combining Rough and Fuzzy Sets for Feature Selection*, PhD Thesis, Citeseer, PL, USA, 2005.
- [12] R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 824–838, 2009.
- [13] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, Berlin, Germany, 1st edition, 2013.
- [14] E.-L. Bonaldi, L. B. Da Silva, G. Lambert-Torres, L. E. L. Oliveira, and F. O. Assunco, "Using rough sets techniques as a fault diagnosis classifier for induction motors," in *Proceedings of the IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, vol. 4, pp. 3383–3388, Sevilla, Spain, November 2002.
- [15] E. L. Bonaldi, L. B. Da Silva, G. Lambert-Torres, and L. E. L. Oliveira, "A rough sets based classifier for induction motors fault diagnosis," *WSEAS Transactions on Systems*, vol. 2, no. 2, pp. 320–327, 2003.
- [16] M. C. Magro and P. Pinceti, "A confirmation technique for predictive maintenance using the Rough Set Theory," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1319–1327, 2009.
- [17] F. Thompson, *Condition Monitoring for Machine Health Prognosis Using Dominance Based Rough Sets*, PhD Thesis, University of Portsmouth, Portsmouth, England, 2016.
- [18] Q. H. Qinghua, D. Yu, Z. Zongxia Xie, and J. L. Jinfu, "Fuzzy probabilistic approximation spaces and their information measures," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 191–201, 2006.
- [19] L. S. Riza, A. Janusz, C. Bergmeir et al., "Implementing algorithms of rough set theory and fuzzy rough set theory in the R package "RoughSets"," *Information Sciences*, vol. 287, pp. 68–89, 2014.
- [20] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *International Journal of General Systems*, vol. 17, no. 2–3, pp. 191–209, 1990.
- [21] H. Fathabadi, "Novel filter based ANN approach for short-circuit faults detection, classification and location in power transmission lines," *International Journal of Electrical Power & Energy Systems*, vol. 74, pp. 374–383, 2016.
- [22] W. Chine and A. Mellit, "ANN-based fault diagnosis technique for photovoltaic stings," in *Proceedings of the 2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, pp. 1–4, Boumerdes, Algeria, October 2017.
- [23] A. Maheshwari, V. Agarwal, and S. K. Sharma, "Transmission line fault classification using artificial neural network based fault classifier," *International Journal of Electrical Engineering & Technology*, vol. 9, pp. 170–181, 2018.
- [24] H. Kaur and M. Kaur, "fault classification in a transmission line using levenberg-marquardt algorithm based artificial neural network," in *Data Communication and Networks*, pp. 119–135, Springer, Berlin, Germany, 2020.
- [25] M. N. Mahmud, M. N. Ibrahim, M. K. Osman, and Z. Hussain, "Support vector machine (SVM) for fault classification in radial distribution network," *Advanced Science Letters*, vol. 23, no. 5, pp. 4124–4128, 2017.
- [26] Z. Di, Q. Kang, D. Peng, and M. Zhou, "Density peak-based pre-clustering support vector machine for multi-class imbalanced classification," in *Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 27–32, Bari, Italy, October 2019.
- [27] J. Yu, Q. Li, W. Chen, J. Liu, J. Li, and C. Xu, "Fault classification of PEM fuel cell systems based on CS-SVM method," in *Proceedings of the 2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 2265–2269, Beijing, China, November 2019.
- [28] X. Gao, H. Wei, T. Li, and G. Yang, "A rolling bearing fault diagnosis method based on LSSVM," *Advances in Mechanical Engineering*, vol. 12, no. 1, pp. 1–10, 2020.
- [29] N. Helwig, E. Pignatelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *Proceedings of the Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 210–215, Pisa, Italy, May 2015.
- [30] R. Z. Haddad and E. G. Strangas, "On the accuracy of fault detection and separation in permanent magnet synchronous machines using MCSA/MVSA and LDA," *IEEE Transactions on Energy Conversion*, vol. 31, no. 3, pp. 924–934, 2016.
- [31] L. Xiao, H. Sun, L. Zhang, F. Niu, L. Yu, and X. Ren, "Applications of a strong track filter and LDA for on-line identification of a switched reluctance machine stator inter-turn shorted-circuit fault," *Energies*, vol. 12, no. 1, p. 134, 2019.
- [32] G. Chen, Y. Liu, and Z. Ge, "K-means Bayes algorithm for imbalanced fault classification and big data application," *Journal of Process Control*, vol. 81, pp. 54–64, 2019.
- [33] M. Suo, B. Zhu, R. An, H. Sun, S. Xu, and Z. Yu, "Data-driven fault diagnosis of satellite power system using fuzzy Bayes risk and SVM," *Aerospace Science and Technology*, vol. 84, pp. 1092–1105, 2019.
- [34] E. Aker, M. L. Othman, V. Veerasamy, I. bin Aris, N. I. A. Wahab, and H. Hizam, "fault detection and classification of shunt compensated transmission line using discrete wavelet transform and naive Bayes classifier," *Energies*, vol. 13, no. 243, pp. 1–24, 2020.
- [35] Y. Wu, Z. Fu, and J. Fei, "Fault diagnosis for industrial robots based on a combined approach of manifold learning, treelet transform and Naive Bayes," *Review of Scientific Instruments*, vol. 91, no. 1, Article ID 015116, 2020.
- [36] T. Chakraborty, "EC3: combining clustering and classification for ensemble learning," in *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, pp. 781–786, New Orleans, LA, USA, November 2017.
- [37] S. Behera, A. Choubey, C. S. Kanani, Y. S. Patel, R. Misra, and A. Sillitti, "Ensemble trees learning based improved predictive maintenance using iiot for turbofan engines," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 842–850, Limassol, Cyprus, April 2019.
- [38] Z. Li, D. Wu, C. Hu, and J. Terpenny, "An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 184, pp. 110–122, 2019.
- [39] P. S. Mung and S. Phyu, "Ensemble Learning Method for Enhancing Healthcare Classification," 2020, <https://onlineresource.ucsy.edu.mm:443/handle/123456789/2501>.
- [40] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
- [41] U. Agrawal, D. Soria, C. Wagner et al., "Combining clustering and classification ensembles: a novel pipeline to identify breast cancer profiles," *Artificial Intelligence in Medicine*, vol. 97, pp. 27–37, 2019.
- [42] F. Herrmann, M. Friedlander, and O. Yilmaz, "Fighting the curse of dimensionality: compressive sensing in exploration seismology," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 88–100, 2012.

- [43] F. Bach, "Breaking the curse of dimensionality with convex neural networks," *Journal of Machine Learning Research*, vol. 18, no. 19, pp. 1–53, 2017.
- [44] S. S. Chawathe, "Condition monitoring of hydraulic systems by classifying sensor data streams," in *Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0898–0904, Las Vegas, NV, USA, January 2019.
- [45] A. Buabeng, A. Simons, N. K. Frempong, and Y. Y. Ziggah, "A novel hybrid predictive maintenance model based on clustering, smote and multi-layer perceptron neural network optimised with grey wolf algorithm," *SN Applied Sciences*, vol. 3, no. 5, p. 593, 2021.
- [46] A. Buabeng, A. Simons, N. K. Frimpong, and Y. Y. Ziggah, "Hybrid intelligent predictive maintenance model for multiclass fault classification," , 2021.
- [47] S. Binsaeid, S. Asfour, S. Cho, and A. Onar, "Machine ensemble approach for simultaneous detection of transient and gradual abnormalities in end milling using multisensor fusion," *Journal of Materials Processing Technology*, vol. 209, no. 10, pp. 4728–4738, 2009.
- [48] E. Azimirad, J. Haddadnia, and A. L. I. Izadipour, "A comprehensive review of the multi-sensor data fusion architectures," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, 2015.
- [49] X. Jing, Z. Yan, X. Jiang, and W. Pedrycz, "Network traffic fusion and analysis against DDoS flooding attacks with a novel reversible sketch," *Information Fusion*, vol. 51, pp. 100–113, 2019.
- [50] G. Li, Z. Yan, Y. Fu, and H. Chen, "Data fusion for network intrusion detection: a review," *Security and Communication Networks*, vol. 2018, Article ID 8210614, 16 pages, 2018, <https://www.hindawi.com/journals/scn/2018/8210614/>.
- [51] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang, and Z. Zhao, "An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure," *Information Fusion*, vol. 55, pp. 269–280, 2020.
- [52] J.-z. Wang, Z. Yan, L. T. Yang, and B.-x. Huang, "An approach to rank reviews by fusing and mining opinions based on review pertinence," *Information Fusion*, vol. 23, pp. 3–15, 2015.
- [53] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Pother, "Probabilistic fusion of stereo with color and contrast for bilayer segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1480–1492, 2006.
- [54] G. Shafer, *A mathematical theory of evidence*, Princeton university press, Princeton, New Jersey, US, 1976.
- [55] R. Geng, X. Wang, N. Ye, and J. Liu, "A fault prediction algorithm based on rough sets and back propagation neural network for vehicular networks," *IEEE Access*, vol. 6, Article ID 74992, 2018.
- [56] J. Maiers and Y. S. Sherif, "Applications of fuzzy set theory," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 175–189, 1985.
- [57] D. W. Roberts, "Ordination on the basis of fuzzy set theory," *Vegetatio*, vol. 66, no. 3, pp. 123–131, 1986.
- [58] H.-J. Zimmermann, "Fuzzy set theory," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 317–332, 2010.
- [59] C. F. Baban, M. Baban, and M. D. Suteu, "Using a fuzzy logic approach for the predictive maintenance of textile machines," *Journal of Intelligent and Fuzzy Systems*, vol. 30, no. 2, pp. 999–1006, 2016.
- [60] A. Chaudhuri, "Predictive Maintenance for Industrial Iot of Vehicle Fleets Using Hierarchical Modified Fuzzy Support Vector Machine," 2018, <https://arxiv.org/abs/1806.09612>.
- [61] M. Karakose and O. Yaman, "Complex fuzzy system based predictive maintenance approach in railways," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6023–6032, 2020.
- [62] Z. a. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [63] J. R. Anaraki and M. Eftekhari, "Improving fuzzy-rough quick reduct for feature selection," in *Proceedings of the 2011 19th Iranian Conference on Electrical Engineering*, pp. 1–6, Tehran, Iran, May 2011.
- [64] A. L. Jabar and T. A. Rashid, "Combining fuzzy rough set with salient features for HRM classification," in *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 244–251, Liverpool, UK, October 2015.
- [65] A. M. Radzikowska and E. E. Kerre, "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 126, no. 2, pp. 137–155, 2002.
- [66] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.
- [67] R. B. Bhatt and M. Gopal, "On fuzzy-rough sets approach to feature selection," *Pattern Recognition Letters*, vol. 26, no. 7, pp. 965–975, 2005.
- [68] R. Jensen and Q. Shen, "Fuzzy-rough sets assisted attribute selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 73–89, 2007.
- [69] E. C. C. Tsang, D. Degang Chen, D. S. Yeung, X. Z. Xi-Zhao Wang, and J. Lee, "Attributes reduction using fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 5, pp. 1130–1141, 2008.
- [70] S. Suyun Zhao, E. Tsang, and D. C. Degang, "The model of fuzzy variable precision rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 2, pp. 451–467, 2009.
- [71] Q. Hu, L. Zhang, S. An, D. Zhang, and D. Yu, "On robust fuzzy rough set models," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 4, pp. 636–651, 2011.
- [72] C. Cornelis and R. Jensen, "A noise-tolerant approach to fuzzy-rough feature selection," in *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pp. 1598–1605, Hong Kong, China, June 2008.
- [73] C. Cornelis, N. Verbiest, and R. Jensen, "Ordered weighted average based fuzzy rough sets," in *Proceedings of the International Conference on Rough Sets and Knowledge Technology*, pp. 78–85, Beijing, China, October 2010.
- [74] J. F. Salido and S. Murakami, "Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relations," *Fuzzy Sets and Systems*, vol. 139, no. 3, pp. 635–660, 2003.
- [75] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39–59, 1993.
- [76] R. Ranawana and V. Palade, "Multi-classifier systems: review and a roadmap for developers," *International Journal of Hybrid Intelligent Systems*, vol. 3, no. 1, pp. 35–61, 2006.
- [77] M. P. Sesmero, A. I. Ledezma, and A. Sanchis, "Generating ensembles of heterogeneous classifiers using stacked generalization," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 21–34, 2015.

- [78] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [79] P. Shunmugapriya and S. Kanmani, "Optimization of stacking ensemble configurations through Artificial Bee Colony algorithm," *Swarm and Evolutionary Computation*, vol. 12, pp. 24–32, 2013.
- [80] C. Zhang and Y. Ma, Eds., *Ensemble Machine Learning: Methods and Applications*, Springer, Berlin, Germany, 2012 edition, 2012.
- [81] B. Zenko, L. Todorovski, and S. Dzeroski, "A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 669–670, San Jose, CA, USA, November 2001.
- [82] S. Dzeroski and B. Zenko, "Is combining classifiers better than selecting the best one?" *ICML*, vol. 2002, Article ID 123e30, 2002.
- [83] A. K. Seewald, "How to make stacking better and faster while also taking care of an unknown weakness," in *Proceedings of the nineteenth international conference on machine learning*, pp. 554–561, Sydney, Australia, July 2002.
- [84] S. P. Healey, W. B. Cohen, Z. Yang et al., "Mapping forest change using stacked generalization: an ensemble approach," *Remote Sensing of Environment*, vol. 204, pp. 717–728, 2018.
- [85] S.-A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking strong ensembles of classifiers," in *Artificial Intelligence Applications and Innovations*, pp. 545–556, Cham, 2019.
- [86] S. S. Keerthi and C. J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.
- [87] D. Du, X. Jia, and C. Hao, "A new least squares support vector machines ensemble model for aero engine performance parameter chaotic prediction," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4615903, 8 pages, 2016.
- [88] K. Wang, X. Wen, D. Hou et al., "Application of least-squares support vector machines for quantitative evaluation of known contaminant in water distribution system using online water quality parameters," *Sensors*, vol. 18, no. 4, p. 938, 2018.
- [89] R. Resmi, V. Vanitha, E. Aravind, and S. Harithaa, "Detection, classification and zone location of fault in transmission line using artificial neural network," in *Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–5, Coimbatore, Tamil Nadu, India, February 2019.
- [90] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, pp. 1–42, 2020.
- [91] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: a multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2014.
- [92] N. Kardani, A. Zhou, M. Nazem, and S.-L. Shen, "Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data," *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 13, no. 1, pp. 188–201, 2021.
- [93] Z. Allah Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, "Predictive maintenance using tree-based classification techniques: a case of railway switches," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 35–54, 2019.
- [94] S. Kaparthi and D. Bumblauskas, "Designing predictive maintenance systems using decision tree-based machine learning techniques," *International Journal of Quality & Reliability Management*, vol. 37, no. 4, 2020.
- [95] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [96] D. Bertsimas and A. King, "Logistic regression: from art to science," *Statistical Science*, vol. 32, no. 3, pp. 367–384, 2017.
- [97] M. Chang, M. Maguire, and Y. Sun, "Stochastic modeling of bridge deterioration using classification tree and logistic regression," *Journal of Infrastructure Systems*, vol. 25, no. 1, Article ID 04018041, 2019.
- [98] J. Phillips, E. Cripps, J. W. Lau, and M. R. Hodkiewicz, "Classifying machinery condition using oil samples and binary logistic regression," *Mechanical Systems and Signal Processing*, vol. 60–61, pp. 316–325, 2015.
- [99] E. Skordilis and R. Moghaddass, "A condition monitoring approach for real-time monitoring of degrading systems using Kalman filter and logistic regression," *International Journal of Production Research*, vol. 55, no. 19, pp. 5579–5596, 2017.
- [100] J. Yan and J. Lee, "Degradation Assessment and Fault Modes Classification Using Logistic Regression," *Journal of Manufacturing Science and Engineering*, vol. 127, no. 4, 2005.
- [101] T. Schneider, N. Helwig, and A. Schütze, "Automatic Feature Extraction and Selection for Condition Monitoring and Related Datasets," in *Proceedings of the 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, Houston, TX, USA, May 2018.
- [102] S. Paul and P. Maji, "Rough set based gene selection algorithm for microarray sample classification," in *Proceedings of the 2010 International Conference on Methods and Models in Computer Science (ICM2CS-2010)*, pp. 7–13, Hong Kong, China, December 2010.
- [103] M. Qasem and M. Nour, "Improving accuracy for classifying selected medical datasets with weighted nearest neighbors and fuzzy nearest neighbors algorithms," in *Proceedings of the 2015 International Conference on Cloud Computing (ICCC)*, pp. 1–9, Riyadh, Saudi Arabia, April 2015.
- [104] M. A. Alsalem, A. A. Zaidan, B. B. Zaidan et al., "Systematic review of an automated multiclass detection and classification system for acute Leukaemia in terms of evaluation and benchmarking, open challenges, issues and methodological aspects," *Journal of Medical Systems*, vol. 42, no. 11, p. 204, 2018.
- [105] J.-H. Shing-Hua Ho and S.-H. Ho, "A combination of rough-based feature selection and RBF neural network for classification using gene expression data," *IEEE Transactions on NanoBioscience*, vol. 7, no. 1, pp. 91–99, 2008.
- [106] J. F. De Paz, J. Bajo, V. F. López, and J. M. Corchado, "Biomedical Organizations: an intelligent dynamic architecture for KDD," *Information Sciences*, vol. 224, pp. 49–61, 2013.
- [107] P. Mohapatra and S. Chakravarty, "Modified PSO based feature selection for Microarray data classification," in *Proceedings of the 2015 IEEE Power, Communication and Information Technology Conference (PCITC)*, pp. 703–709, Bhubaneswar, India, October 2015.
- [108] S. Agaian, M. Madhukar, and A. T. Chronopoulos, "Automated screening system for acute myelogenous leukemia

- detection in blood microscopic images,” *IEEE Systems Journal*, vol. 8, no. 3, pp. 995–1004, 2014.
- [109] W.-L. Tai, R.-M. Hu, H. C. Hsiao, R.-M. Chen, and J. J. Tsai, “Blood Cell Image Classification Based on Hierarchical SVM,” in *Proceedings of the 2011 IEEE International Symposium on Multimedia*, pp. 129–136, CA, USA, December 2011.
- [110] J. Laosai and K. Chamnongthai, “Acute leukemia classification by using SVM and K-Means clustering,” in *Proceedings of the 2014 International Electrical Engineering Congress (iEECON)*, pp. 1–4, Pattaya City, Thailand, March 2014.
- [111] S. Boughorbel, F. Jarray, and M. El-Anbari, “Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric,” *PLoS One*, vol. 12, no. 6, Article ID e0177678, 2017.
- [112] D. Chicco, “Ten quick tips for machine learning in computational biology,” *BioData Mining*, vol. 10, no. 1, p. 35, 2017.
- [113] M. Hossin and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, p. 1, 2015.
- [114] L. I. Kuncheva, Á. Arnaiz-González, J.-F. Díez-Pastor, and I. A. D. Gunn, “Instance Selection Improves Geometric Mean Accuracy: A Study on Imbalanced Data Classification,” 2018, <http://arxiv.org/abs/1804.07155>.
- [115] S. Wu and P. Flach, “A Scored AUC Metric for Classifier Evaluation and Selection,” in *Proceedings of the Second Workshop on ROC Analysis in ML*, Bonn, Germany, 2005, <http://users.dsic.upv.es/~flip/ROCML2005/papers/wuCRC.pdf>.
- [116] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve,” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [117] S. Chan and P. Treleaven, “Continuous model selection for large-scale recommender systems,” in *Handbook of Statistics*, V. Govindaraju, V. V. Raghavan, and C. R. Rao, Eds., vol. 33, pp. 107–124, 2015.
- [118] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
- [119] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [120] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.