Received: 11 March 2022 | Revised: 6 April 2023 | Accepted: 8 April 2023

DOI: 10.1111/exsy.13312

ORIGINAL ARTICLE

Expert Systems WILEY

PrismPatNet: Novel prism pattern network for accurate fault classification using engine sound signals

Sakir Engin Sahin[1] | Gokhan Gulhan[2] | Prabal Datta Barua[3,4,5,6,7,8,9,10,11] | Turker Tuncer[12] | Sengul Dogan[12] | Oliver Faust[13] | U. Rajendra Acharya[14]

[1]Department of Computer Technologies, Arapgir Vocational School, Malatya Turgut Ozal University, Malatya, Turkey

[2]Department of Motor Vehicles and Transportation Technologies, Automotive Technology Program, Arapgir Vocational School, Malatya Turgut Ozal University, Malatya, Turkey

[3]Cogninet Australia, Sydney, New South Wales, Australia

[4]School of Business (Information System), University of Southern Queensland, Toowoomba, Australia

[5]Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, New South Wales, Australia

[6]Australian International Institute of Higher Education, Sydney, New South Wales, Australia

[7]School of Science and Technology, University of New England, Brisbane, Australia

[8]School of Biosciences, Taylor's University, Kuala Lumpur, Malaysia

[9]School of Computing, SRM Institute of Science and Technology, Chengalpattu, India

[10]School of Science and Technology, Kumamoto University, Kumamoto, Japan

[11]Sydney School of Education and Social work, University of Sydney, Sydney, Australia

[12]Department of Digital Forensics Engineering, College of Techology, Firat University, Elazig, Turkey

[13]School of Computing and Information Science, Anglia Ruskin University Cambridge Campus, Cambridge, UK

[14]School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Australia

Correspondence

Oliver Faust, School of Computing and Information Science, Anglia Ruskin University Cambridge Campus, UK.
Email: oliver.faust@gmail.com

Abstract

Engines are prone to various types of faults, and it is crucial to detect and indeed classify them accurately. However, manual fault type detection is time-consuming and error-prone. Automated fault type detection promises to reduce inter- and intra-observer variability while ensuring time invariant attention during the observation duration. We have proposed an automated fault-type detection model based on sound signals to realize these advantageous properties. We have named the detection model prism pattern network (PrismPatNet) to reflect the fact that our design incorporates a novel feature extraction algorithm that was inspired by a 3D prism shape. Our prism pattern model achieves high accuracy with low-computational complexity. It consists of three main phases: (i) prism pattern inspired multilevel feature generation and maximum pooling operator, (ii) feature ranking and feature selection using neighbourhood component analysis (NCA), and (iii) support vector machine (SVM) based classification. The maximum pooling operator decomposes the sound signal into six levels. The proposed prism pattern algorithm extracts parameter values from both the signal itself and its decompositions. The generated parameter values are merged and fed to the NCA algorithm, which extracts 512 features from that input. The resulting feature vectors are passed on to the SVM classifier, which labels the input as belonging to 1 of 27 classes. We have validated our model with a newly collected dataset containing the sound of (1) a normal engine and (2) 26 different types of engine faults. Our model reached an accuracy of 99.19% and 98.75% using 80:20 hold-out validation and 10-fold cross-validation, respectively. Compared with previous studies, our model achieved the highest overall classification accuracy even though our model was tasked with identifying significantly more fault classes. This performance indicates that our PrismPatNet model is ready to be installed in real-world applications.

KEYWORDS

engine fault classification, fault detection, machine learning, prism pattern, sound analysis, textural feature generation

1 | INTRODUCTION

Nowadays, engine fault detection plays an important role in modern engineering systems (Gelgele & Wang, 1998; Zhang & Deng, 2018). Several techniques have been proposed in the literature to detect and rapidly resolve abnormalities in motors or engines. Incorporating artificial intelligence (AI) usually results in good solutions for automated fault detection. Moreover, AI techniques are widely used in robotic applications, vehicle control, and medical systems (Gundogdu et al., 2018; Oniani et al., 2021; Yan, 2006).

Automatic engine fault detection systems are crucial for automotive and other vehicle industries (Zhang & Deng, 2020). Therefore, computer-aided fault diagnosis models have been used in recent automobiles. However, these models lack the ability to detect and, indeed, discriminate between a range of faults. Despite this shortcoming, intelligent systems for automatic fault detection in automobiles have greatly increased reliability and extended mobility (Wang et al., 2007). By implementing machine learning-based fault diagnosis systems, faults can be detected in real time. For example, signal-based fault detection systems are used to detect anomalies by continuously measuring and analysing engine parameters (Chen et al., 2015). These parameters can come from vibration, and pressure measurements (Adaileh, 2013; Wu & Chuang, 2005). A fault is detected when measured signals deviate from nominal values. However, the algorithms that detect faults are computationally complex. Moreover, they only exploit a linear relationship which makes them error-prone.

1.1 | Motivation

To address these shortcomings (see literature review), we propose a new method for engine fault detection based on the prism shape algorithm. Our work was motivated by three essential considerations: (i) Graph-based feature engineering methods have been used extensively in established research work. In this research, we used a graph-based feature extraction function that uses a prism shape (a well-known 3D shape). Our initial motivation was to investigate the feature extraction capability of the prism shape-based function. (ii) We established a new feature engineering architecture using the proposed prism shape-based feature extraction function. Herein, we mimicked the architecture of deep learning (especially CNN) models to create a multileveled feature generation model. Thus, this model was named PrismPatNet. (iii) Fault detection models are very valuable for Industry 4.0 applications to establish the required quality of service. Engine sound acquisition and analysis is cost-effective and non-invasive. Therefore, we have used engine sound signals as a basis for our fault classification system. During the development phase, we collected a big engine sound dataset that contains one normal and 26 fault categories. Moreover, there are 8614 sound observations in this dataset. We applied to the proposed PrismPatNet to get a high classification performance.

1.2 | Contributions

The main contributions of this work are:

a. To the best of our knowledge, this is the first work to classify 27 different engine sounds. Earlier work focused on fewer engine fault classes. In this respect, the sound dataset collected is the most comprehensive dataset to evaluate engine fault diagnosis systems thoroughly.
b. Developed a novel 27 classes engine fault classification dataset which is publicly available at: http://web.firat.edu.tr/turkertuncer/gasoline_engine_fault_dataset.rar
c. The proposed model mimics popular state-of-the-art deep learning models. The recommended prism pattern (it is a local histogram-based feature generator), maximum pooling, neighbourhood component analysis (NCA) selector, and support vector machine (SVM) were combined to create a new and effective machine learning model.

To support our claim about the efficacy of the PrismPatNet system, we have structured the remainder of the paper as follows. In Section 2, the literature review is presented. Section 3 defines material and method, the results of the proposed PrismPat-based feature engineering model for the used dataset are given in Section 4, the calculated results and findings are discussed in Section 5 and Section 6 presents conclusions.

2 | LITERATURE REVIEW

In the literature, most studies focus on systems that can detect and classify a limited number of faults. Furthermore, some studies did not develop automated systems. The absence of multi-class fault detection and in some cases the absence of automatization constitutes a research gap that creates a need to develop an automated system with many faults classes. Our PrismPatNet addresses that need by automating the classification of 27 engine sounds. Li and Zhao (Li & Zhao, 2020) presented a fault detection model for an aircraft engine. They used kernel extreme learning machine methods. They calculated an accuracy value of 98.68% for eight classes with 10-fold CV. In their study, the computational complexity is high. Zhao et al. (Zhao et al., 2020) used an extended least squares support vector machine for aircraft engine fault detection. They attained an accuracy rate of 98.07%. In their study, the number of classes is small. Wang et al. (Wang et al., 2014) proposed an engine fault detection approach. Their approach was based on Wavelet packet denoising and Hilbert–Huang transform. They attained an accuracy rate of 96.00% for 7 classes. However, the number of classes was low and the dataset was small. Ahmed et al. (Ahmed et al., 2014) tried using artificial neural networks to detect fault. Their method achieved an accuracy of 97.00%. The method had a high computational complexity and the data volume was small. Basir and Yuan (Basir & Yuan, 2007) developed an engine fault detection method. Their method was based on the Dempster–Shafer evidence theory. For 3 classes they obtained 94.00% accuracy. The number of classes was insufficient for practical applications and the dataset was small. Yong-Sheng et al. (Yang et al., 2017) utilized Discriminative non-negative matrix factorization for diesel engine fault diagnosis. They selected 8 classes for experimental results and obtained an accuracy of 93.38%. The number of classes and data used was small in their study. Jung (Jung, 2019) used machine learning techniques for data-driven fault diagnosis. For 9 classes, the accuracy rate was calculated as 90.00%. In their study, both the number of labelled data sets and the number of classes were small. Wu et al. (Wu et al., 2010) proposed an engine fault classification method using artificial neural networks. They selected 10 classes and attained an accuracy of 99.00%. The computational complexity is high for their proposed method.

Existing fault detection models have been developed for fewer fault classes. Some of the proposed engine fault diagnosis models have used vibrations or currents to detect/classify faults. When compared to engine sounds, these signals are more difficult to measure in real-world scenarios. We developed a new sound dataset with 27 groups to overcome these problems. The data was collected using a smartphone microphone. The dataset was used to train and test our new PrismPatNet machine learning model in a second step. The next section provides further details about the new model.

3 | MATERIALS AND METHODS

Machine learning methods have been used in many areas of real-life applications to detect machine faults accurately and thereby reduce the running cost (Chen et al., 2021; Tomasik et al., 2021; Xu et al., 2021). This work proposes an automated engine fault detection model to classify 27 engine sounds. Figure 1 shows a graphical layout of the proposed engine sound classification model.

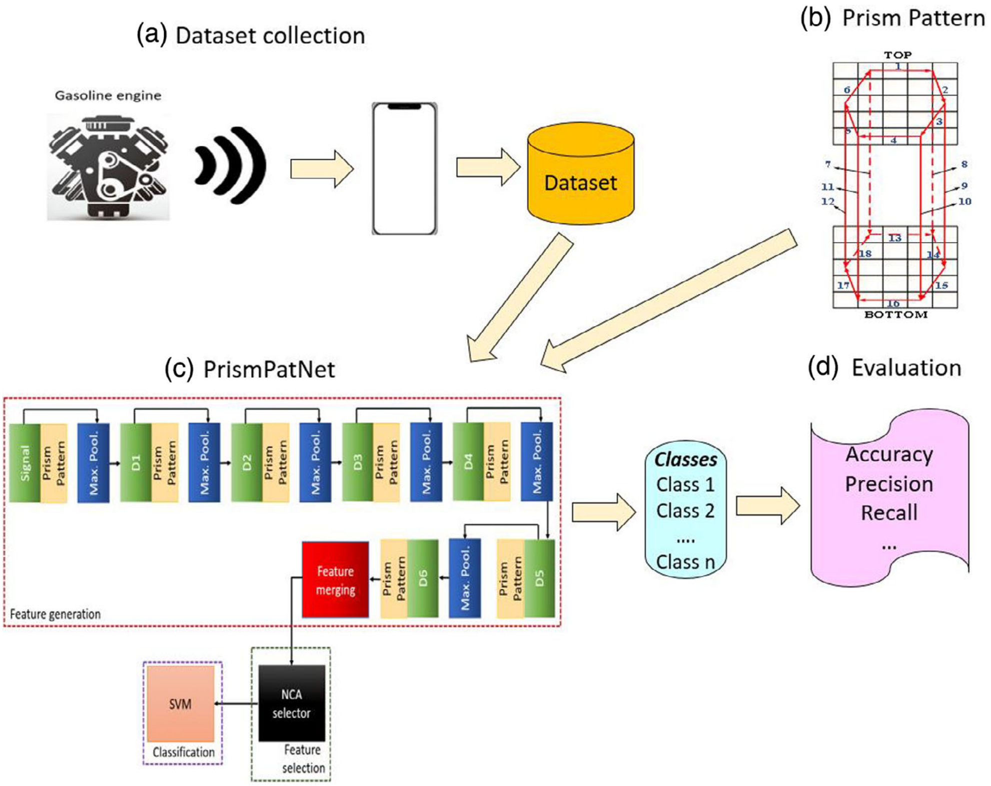


FIGURE 1 Layout of the proposed engine sound classification model.

3.1 | Material

Our first action was to collect an engine fault dataset comprising 26 faulty engine sounds and one healthy control (overall 27 classes). The sounds were collected from a Tofaş 131 gasoline engine. This gasoline engine was designed for automobiles and is used for educational purposes. The Tofaş 131 attributes are given as follows. It has 65 HP (48 kW) power, 1297 cc capacity, and 102 Newton/meter torque. The sound signals were collected in a workshop, the photos in Figure 2 document that process.

All the engine sound signals were collected using a smartphone (Figure 2). The sampling frequency was set to 44.1 kHz and the sound signals were source code into the m4a standard. The collected sound signals were segmented into the one-second segment to develop a novel machine learning model. Table 1 shows the sound attributes of the 26 faults and the normal control. After recording the dataset, we were ready to design the information extraction algorithms.

As can be seen from Table 1, there are 27 categories. The 26 categories contain faulty sounds and one category (22nd class) contains sounds of the healthy engine.

3.2 | PrismPatNet method

The main objective of the proposed PrismPatNet was to create a lightweight (has a low time burden) and efficient classification model for engine sounds. That was accomplished by structuring the PrismPatNet method into three phases: (i) feature extraction using a prism pattern and maximum pooling network, (ii) selecting the most significant 512 features with NCA (Goldberger et al., 2004), and (iii) classification using the SVM classifier (Vapnik, 1998).

Figure 3 shows a graphical representation of the proposed PrismPatNet method. This figure shows D1, D2, …, and D6 are the decomposed signals obtained using maximum pooling (Max. Pool) (Yao et al., 2021). The proposed prism pattern algorithm extracts parameters from the sound signal and *six* decomposition levels. Subsequently, the parameters were merged and processed with NCA. The 512 most significant parameters were selected from the NCA output to form a feature vector. These feature vectors were input to the SVM classifier for automated classification. The pseudocode of the proposed PrismPatNet classification model is given in Algorithm 1.

The main characteristics of the proposed PrismPatNet method are presented in Table 2.

3.2.1 | Iterative decomposer and prism pattern-based feature generation

In the initial design phase of the proposed PrismPatNet method, we established the prism pattern, and we created the feature extractor based on maximum pooling. The prime feature generation function is established through a novel prism shape pattern. Various graphs have been used to present patterns for local and histogram-based feature generators (Tuncer, Dogan, & Subasi, 2021; Tuncer, Dogan, & Acharya, 2021; Aydemir et al., 2021). The proposed feature generation function uses the prism shape as a pattern. To model the prism shape as a pattern, overlapping blocks with a length of 50 were used. These blocks were mapped into two 5 × 5 matrices. The prism shape is modelled by deploying these matrices, as shown in Figure 4.

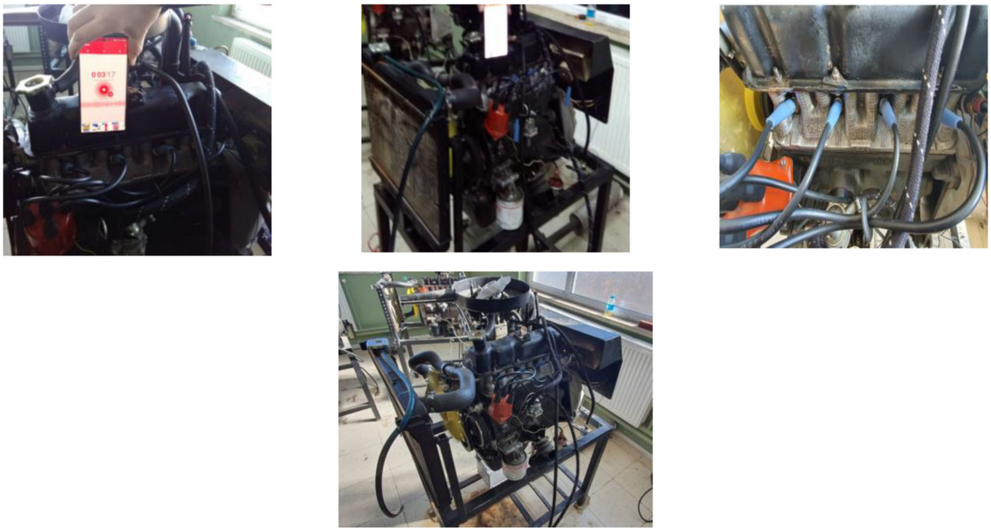


FIGURE 2 Photos of the Tofaş 131 gasoline engine and the data collection environment.

TABLE 1 Overview of 26 engine faults

Number	Fault name	Observation numbers
1	Too much valve clearance (0.15 mm)	316
2	Too much valve clearance (0.15 mm) and One spark plug failure (First spark plug)	316
3	Too much valve clearance (0.15 mm) and two spark plugs failure (First and third spark plugs)	318
4	Too much valve clearance (0.15 mm) and two spark plugs failure (First and fourth spark plugs)	316
5	Too much valve clearance (0.15 mm) and too much-advanced ignition timing	316
6	Too much valve clearance (0.15 mm), too much-advanced ignition timing, two spark plugs failure (First and third spark plugs)	316
7	Too much valve clearance (0.15 mm), too much-advanced ignition timing, two spark plugs failure (First and fourth spark plugs)	315
8	Too much valve clearance (0.15 mm), too much-advanced ignition timing, and incorrect engine timing	309
9	Too much valve clearance (0.15 mm), too much-advanced ignition timing, and one spark plug failure (First spark plug)	315
10	Too much valve clearance (0.15 mm), incorrect engine timing, and two spark plugs failure (first and third spark plugs)	317
11	Too much valve clearance (0.15 mm), incorrect engine timing, and two spark plugs failure (First and fourth spark plugs)	315
12	Too much valve clearance (0.15 mm), incorrect engine timing, and one spark plug failure (first spark plug)	316
13	Too much valve clearance (0.15 mm), incorrect engine timing, and too much retarded ignition timing	315
14	Too much valve clearance (0.15 mm) and incorrect engine timing	315
15	Two spark plugs failure (first and fourth spark plugs)	316
16	Two spark plugs failure (first and third spark plugs)	321
17	Too much retarded ignition timing	310
18	Too much-advanced ignition timing and two spark plug failure (first and third spark plugs)	312
19	Too much-advanced ignition timing and two spark plug failure (first and fourth spark plugs)	317
20	Too much-advanced ignition timing and One spark plug failure (first spark plug)	316
21	Too much-advanced ignition timing	318
22	Normal	310
23	One spark plug failure (first spark plug)	314
24	Incorrect engine timing	316
25	Incorrect engine timing and one spark plug failure (first spark plug)	422
26	Incorrect engine timing and two spark plug failure (first and third spark plugs)	317
27	Incorrect engine timing and two spark plug failure (first and fourth spark plugs)	310
Total		8614

The prism shape, shown in Figure 4, has 18 edges. Binary features are generated by deploying these edges together with a signum function. To be specific, deploying an edge involves first and second parameters of the signum function. Therefore, 18 bits are generated using this shape (prism pattern). The generated 18 bits are divided into two groups, and these groups are named left and right. Two feature map signals are created using the left and right bits. Finally, histograms of these map signals are extracted, and these histograms are merged to calculate the prism pattern feature vector. The recommended iterative feature generator uses the presented prism pattern and the maximum pooling decomposer extracts both low-level and high-level features. The steps of the iterative feature generator are given below:

Step 1: Decompose the signal using maximum pooling with 1 × 3 sized non-overlapping blocks. In this work, we present a six-level feature generator. The number of levels is calculated using Equation 1.

$$L = \left\lfloor log_b\left(\frac{ln}{w}\right)\right\rfloor \quad (1)$$

In Equation 1, L is the number of levels, b defines the reduction factor, ln represents the length of the sound, and w is the length of the used overlapping block.

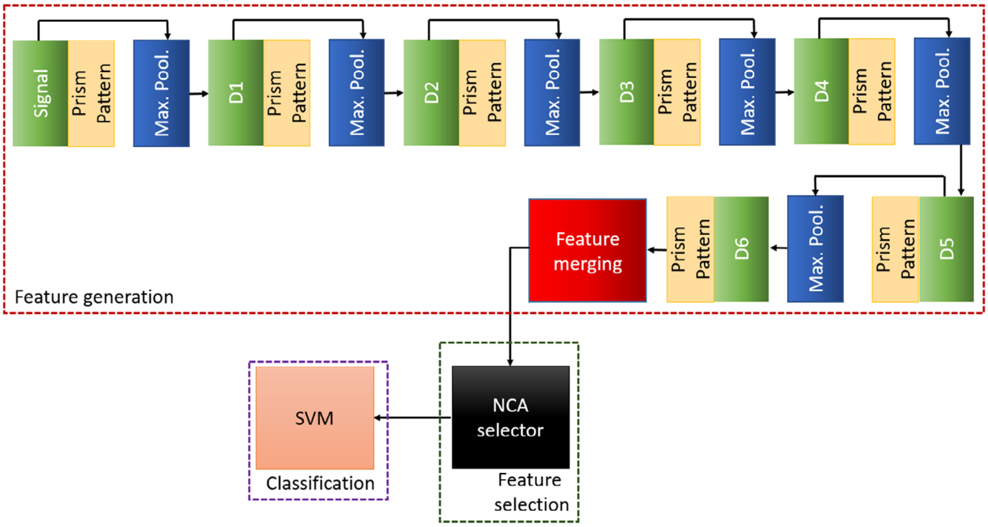


FIGURE 3 Graphical representation of the proposed PrismPatNet method.

Algorithm 1 Pseudocode of the PrismPatNet classification model.

```
Input: Used sound fault corpus (C), real output (y)
Output: Predicted values (prv).
00: Load C
01: for k = 1 to TI do
// TI is number of the training observation and it is equal to 6892 for this dataset using 80:20. For ten-fold CV, this number is 7753 for six folds and TI is 7752 for remainder four folds.
02:   $sound = C(k, :)$; //Read sound signal.
03:   for i = 1 to 7 do
04:     $X(k,(i-1)*1024+1 : i*1024) = prism-pat(sound)$; // Generate features
// The line 04 both generate features employing the proposed prism pattern ($prism-pat(.)$) and merge features. The proposed PrismPatNet has iterative feature generation.
05:     $D = maxp(signal,3)$; // Apply maximum pooling to sound for decomposition.
// Here, 1 × 3 sized non-overlapping blocks are used in the maximum pooling.
06:     $sound = D$; //Update signal
07:   end for i
08: end for k
09: Employ NCA to feature matrix and calculate sorted indices (id).
10: for i = 1 to TI do
11:   for j = 1 to 512 do
12:     $X^{sel}(i,j) = X(i,id(j))$; // Top 512 features selection
13:   end for j
14: end for i
15: Classify selected features ($X^{sel}$) deploying SVM and obtain prv
```

In Equation 1, the number of levels is calculated as $\lfloor log_3\left(\frac{44100}{50}\right) \rfloor = 6$. Therefore, *six* decomposed signals are generated. This generation process is shown in Equations 2 and 3.

$$D^1 = maxp(sound,3) \tag{2}$$

TABLE 2 Main characteristics of the proposed PrismPatNet method

Input	Length	Operator	Output
Raw sound	44,100	Prism pattern	1024 features
Sound	44,100	Maximum pooling	D1 with a length of 14,700
D1	14,700	Prism pattern	2048 features
D1	14,700	Maximum pooling	D2 with a length of 4900
D2	4900	Prism pattern	3072 features
D2	4900	Maximum pooling	D3 with a length of 1633
D3	1633	Prism pattern	4096 features
D3	1633	Maximum pooling	D4 with a length of 544
D4	544	Prism pattern	5120 features
D4	544	Maximum pooling	D5 with a length of 181
D5	181	Prism pattern	6144 features
D5	181	Maximum pooling	D6 with a length of 60
D6	60	Prism pattern	7168 features
Feature vector	7168	NCA	Selected feature vector with a length of 512
Classification	512	SVM	Results

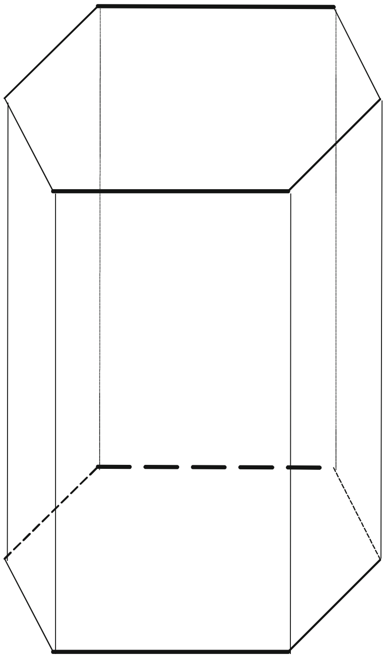


FIGURE 4 Typical prism shape.

$$D^{t+1} = maxp\left(D^t, 3\right), t \in \{1, 2, \ldots, 5\} \tag{3}$$

where D^t is t th decomposed signal, $maxp(.,.)$ is maximum pooling.

Step 2: Engages the prism pattern feature generation function on the decomposed sound signals for feature extraction.

$$feat^1 = prism - pat(sound) \tag{4}$$

$$feat^{l+1} = prism - pat\left(D^l\right), l \in \{1, 2, \ldots, 6\} \tag{5}$$

Equations 4 and 5 indicate that, seven feature vectors (*feat*) are generated using the recommended prism pattern ($prism-pat(.)$). The recommended prism pattern is defined below.

Step 2.1: Create overlapping blocks with a length of 50.

$$block(i)=signal(j+i-1), j\in\{1,2,\ldots,len-49\}, i\in\{1,2,\ldots,50\} \tag{6}$$

where *block* is the overlapping block with a length of 50, *len* defines the length of the used signal (*signal*).

Step 2.2: Generate two 5 × 5 matrices deploying the 50-sized overlapping blocks.

$$bottom(x,y)=block(i), i\in\{1,2,\ldots,25\}, x\in\{1,2,\ldots,5\}, y\in\{1,2,\ldots,5\} \tag{7}$$

$$top(x,y)=block(i+25) \tag{8}$$

Herein, *bottom* and *top* are created with 5 × 5 matrices using the divided overlapped blocks. A prism pattern was created using the bottom and top, as shown in Figure 5.

Step 2.3: Generate 18 bits by deploying signum (basic comparison function), as shown in the prism edge schematic presented in Figure 5. The basic bit extractor is defined in Equation 8.

$$S(f,s)=\begin{cases}0, f-s<0\\1, f-s\geq 0\end{cases} \tag{9}$$

where $S(.,.)$ is a comparison function for bit generation and f,s are the first and second input parameters of the signum function. The mathematical notation of the proposed bit generation is explained below.

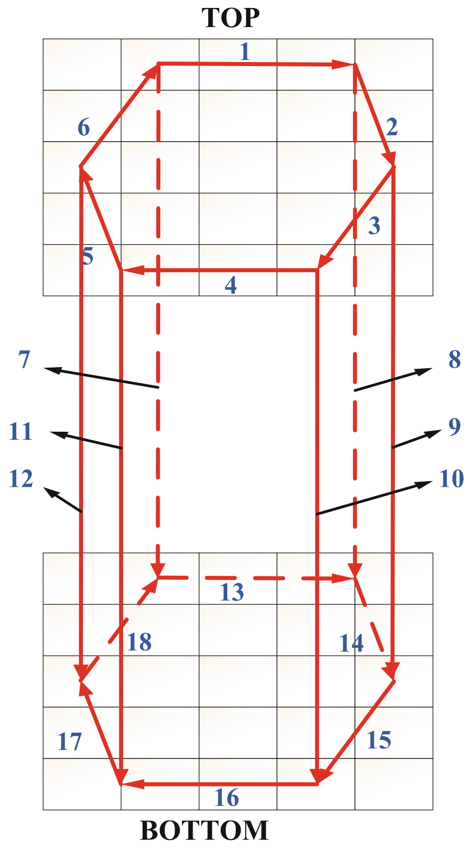


FIGURE 5 Schematic diagram of the prism edges. The edge labels have been defined according to the bit order and Equation (10) provides a mathematical description how this pattern was used for binary feature extraction.

$$\left|\begin{array}{c} bit(1) \\ bit(2) \\ bit(3) \\ bit(4) \\ bit(5) \\ bit(6) \\ bit(7) \\ bit(8) \\ bit(9) \\ bit(10) \\ bit(11) \\ bit(12) \\ bit(13) \\ bit(14) \\ bit(15) \\ bit(16) \\ bit(17) \\ bit(18) \end{array}\right| = S\left(\left|\begin{array}{c} top(1,2), top(1,4) \\ top(1,4), top(3,5) \\ top(3,5), top(5,4) \\ top(5,4), top(5,2) \\ top(5,2), top(3,1) \\ top(3,1), top(1,2) \\ top(1,2), bottom(1,2) \\ top(1,4), bottom(1,4) \\ top(3,5), bottom(3,5) \\ top(5,4), bottom(5,4) \\ top(5,2), bottom(5,2) \\ top(3,1), bottom(3,1) \\ bottom(1,2), bottom(1,4) \\ bottom(1,4), bottom(3,5) \\ bottom(3,5), bottom(5,4) \\ bottom(5,4), bottom(5,2) \\ bottom(5,2), bottom(3,1) \\ bottom(3,1), bottom(1,2) \end{array}\right|\right) \tag{10}$$

By using Equation (9), 18 bits have been generated.

Step 2.4: Generate left and right bits using the 18 bits created in the previous step.

$$left(y) = bit(y), y \in \{1,2,\ldots,9\} \tag{11}$$

$$right(y) = bit(y+9) \tag{12}$$

Step 2.5: Calculate map signals using right and left bits.

$$map^{left}(i) = \sum_{g=1}^{9} left(g) * 2^{g-1}, i \in \{1,2,\ldots,ln-49\} \tag{13}$$

$$map^{right}(i) = \sum_{g=1}^{9} right(g) * 2^{g-1} \tag{14}$$

In this work, map^{left}, map^{right} define right and left map signals and ln defines the length of the used sound.

Step 2.6: Extract histograms of the calculated map^{left} and map^{right} signals. According to Equations 11 and 12, these signals are coded with 9 bits. Thus, each signal has a histogram length of 512.

Step 2.7: Merge the two generated histograms and obtain a feature vector of length $= 1024$.

Steps 2.1–2.6 define the proposed $prism-pat(.)$ feature generation function. Furthermore, the real MATLAB code of this feature extractor has been given in Appendix A.

Step 3: Concatenate the calculated feature vectors (*feat*) to obtain a feature vector length $= 1024 \times 7 = 7168$.

3.2.2 | Feature selection

The proposed multileveled prism pattern feature generator extracted 7168 features for each sound signal. This leads to a 7168 feature matrix. To choose the top 512 features from the 7168 generated features, an NCA (Goldberger et al., 2004) based feature selector was used. The NCA uses common weights to generate feature selectors. The optimal weights were calculated based on city block distance and stochastic gradient descends (SGD) optimizer. By deploying the generated weights, the top features were selected. The two primary objectives of feature selection were: (i) increasing the performance of the model using the most informative features and eliminating redundant ones, and (ii) reducing the time complexity during classification (Dogan et al., 2020; Tuncer et al., 2021).

The following steps were used to choose the top features:

Step 4: Use NCA to select the feature vector and obtain the weights.

Step 5: Sort the obtained weights in descending order.

Step 6: Choose the top 512 out of 7168 features.

3.2.3 | Classification

SVM (Jain et al., 2021) was utilized as a classifier in this model. In this work, we have used a shallow classifier to obtain high classification accuracy using the presented PrismPatNet with hold-out validation (the split ratio is 80:20) and a 10-fold cross-validation strategy. The SVM attributes are: kernel = polynomial of order 3, kernel scale = automatic, coding = one-versus-all, and box-constraint (C) level = 1.

Step 7: Classify the chosen features using the 3rd-order polynomial kernel SVM classifier with 10-fold cross-validation and 80:20 hold-out validation techniques.

4 | EXPERIMENTAL RESULTS

This section details the experimental setup that was used to obtain the PrismPatNet performance results. The GEFC27 dataset was collected from an automotive workshop of Arapgir Vocational School, and the sound signals were acquired using a smartphone microphone. The collected sounds were segmented into blocks of one-second duration. The PrismPatNet method was implemented using MATLAB 2020b on a standard personal computer. We did not use parallel processing to execute the PrismPatNet algorithm. The model quality was evaluated with the following seven performance measures: Accuracy, overall precision, overall recall, overall F1-score, Cohen's Kappa, geometric mean, and Mathew correlation coefficient metrics (Tuncer et al., 2019). Table 3 shows the seven performance measures that were obtained by assessing the developed PrismPatNet model with hold-out and ten-fold cross-validation strategies.

Figure 6 shows the performance measures for each of the 27 classes using: (a) hold-out and (b) 10-fold cross-validation strategies.

It can be noted from Figure 6 that the developed model achieved over 94% for F1, precision, and recall. Furthermore, Figure 6 indicates that this performance threshold holds for both hold-out validation (80:20) and ten-fold cross-validation. Figure 7 shows the fold-wise accuracy obtained for the developed PrismPatNet method.

The model achieved the highest accuracy of 99.77% in the 10th fold and the lowest accuracy of 97.45% in fold 4.

5 | DISCUSSION

Automated and accurate fault classification is very useful for transportation and manufacturing systems. Therefore, many automated fault classification methods have been developed (Akbal, 2021; Aljemely et al., 2021; Yaman, 2021). To extend current knowledge and progress the technology, we developed a new engine sound dataset which contains 8614 observations. Each observation was labelled as belonging to one of 27 distinct engine sounds. This sound signal dataset was easy to collect at a low cost. In the second phase, a new hand-modelled features-based PrismPatNet was developed. The prism shape was modelled to generate local features, and maximum pooling was employed to decompose signals. Both low-level and high-level features were generated by deploying both the presented prism pattern and maximum pooling. NCA was used

TABLE 3 Performance measures to assess the presented PrismPatNet model based on hold-out and ten-fold cross-validation strategies.

Performance parameters	Results (80:20) (%)	Results (10-fold cross-validation) (%)
Accuracy	99.19	98.75
Overall precision	99.18	98.74
Overall recall	99.18	98.74
Overall F1-score	99.18	98.74
Cohen's Kappa	99.16	98.70
Geometric mean	99.17	98.74
Mathew correlation coefficient	99.15	98.69

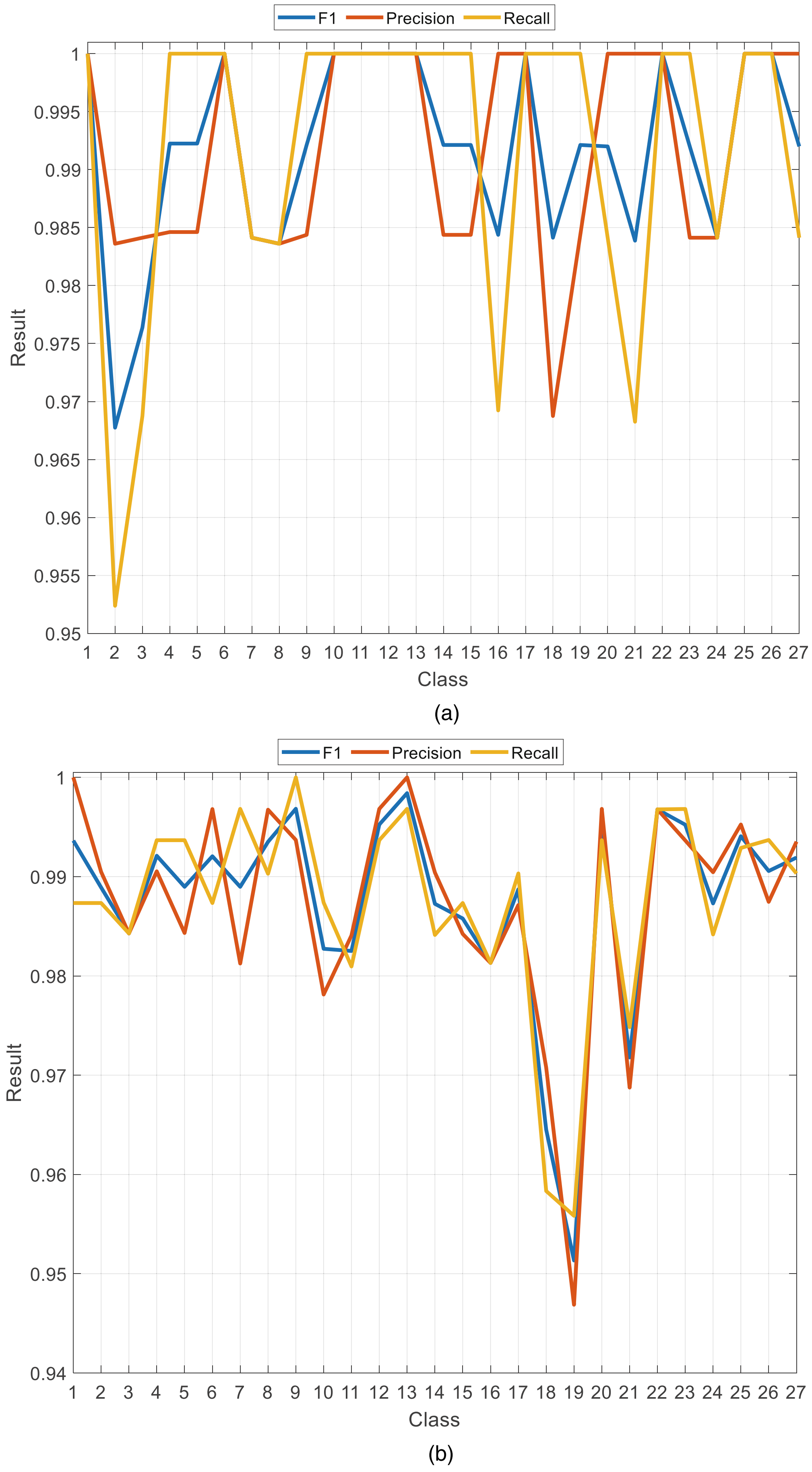


FIGURE 6 Performance measures obtained for each of the 27 classes using: (a) hold-out and (b) 10-fold cross-validation strategies.

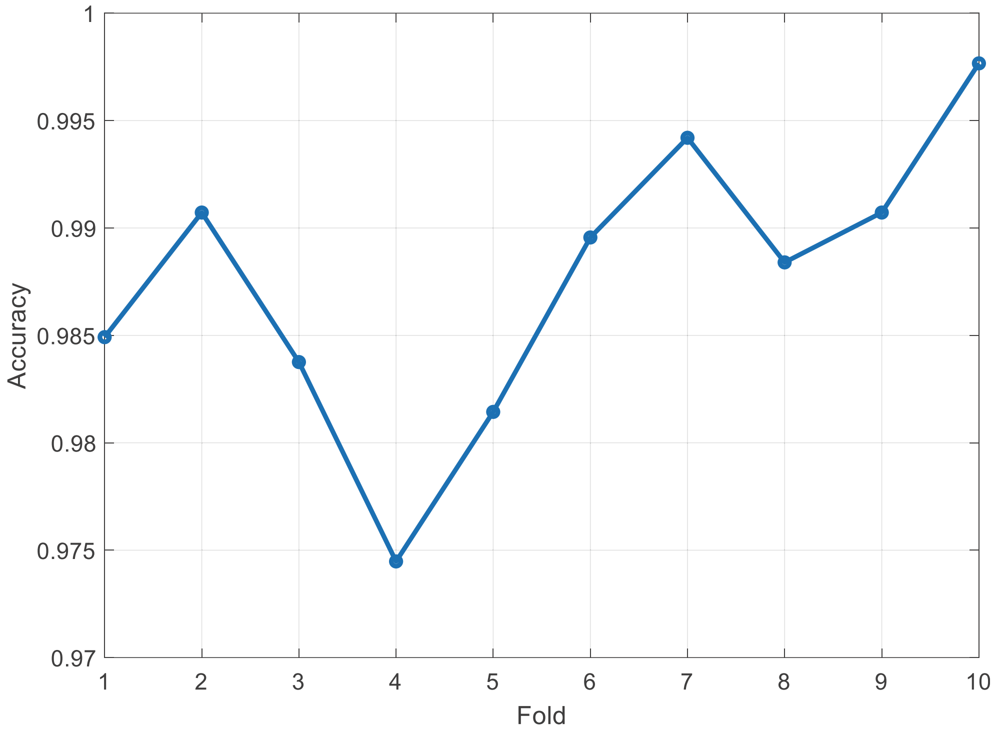


FIGURE 7 Fold-wise accuracy for the proposed PrismPatNet method.

to select the 512 most significant features. The resulting feature vectors were fed to a shallow SVM classifier which labels each input as belonging to one of 27 classes. The main objective of the presented prism pattern is to extract hidden patterns using a 3D shape relation. Deep networks inspire the presented PrismPatNet. The deep networks can extract multilevel features from a signal. Therefore, a multileveled feature generator is employed in this work. The proposed PrismPatNet model is simple and efficient. Our model was developed using both hold-out validation and 10-fold cross-validation. We have obtained an accuracy of 99.19% and 98.75% using 80:20 hold-out and 10-fold cross-validation, respectively. Figure 8 shows the confusion matrices obtained using our proposed method for (a) 80:20 hold-out validation and (b) 10-fold cross-validation strategy.

The confusion matrices in Figure 8 present a result summary.

In this model, the NCA was used as a feature selector. Without NCA, the presented model would need to process 7168 features. We selected the top 512 out of the generated 7168 features using NCA. That decreases the computation cost of the SVM and it increases classification ability. Figure 9 documents the positive effect NCA has on classification accuracy.

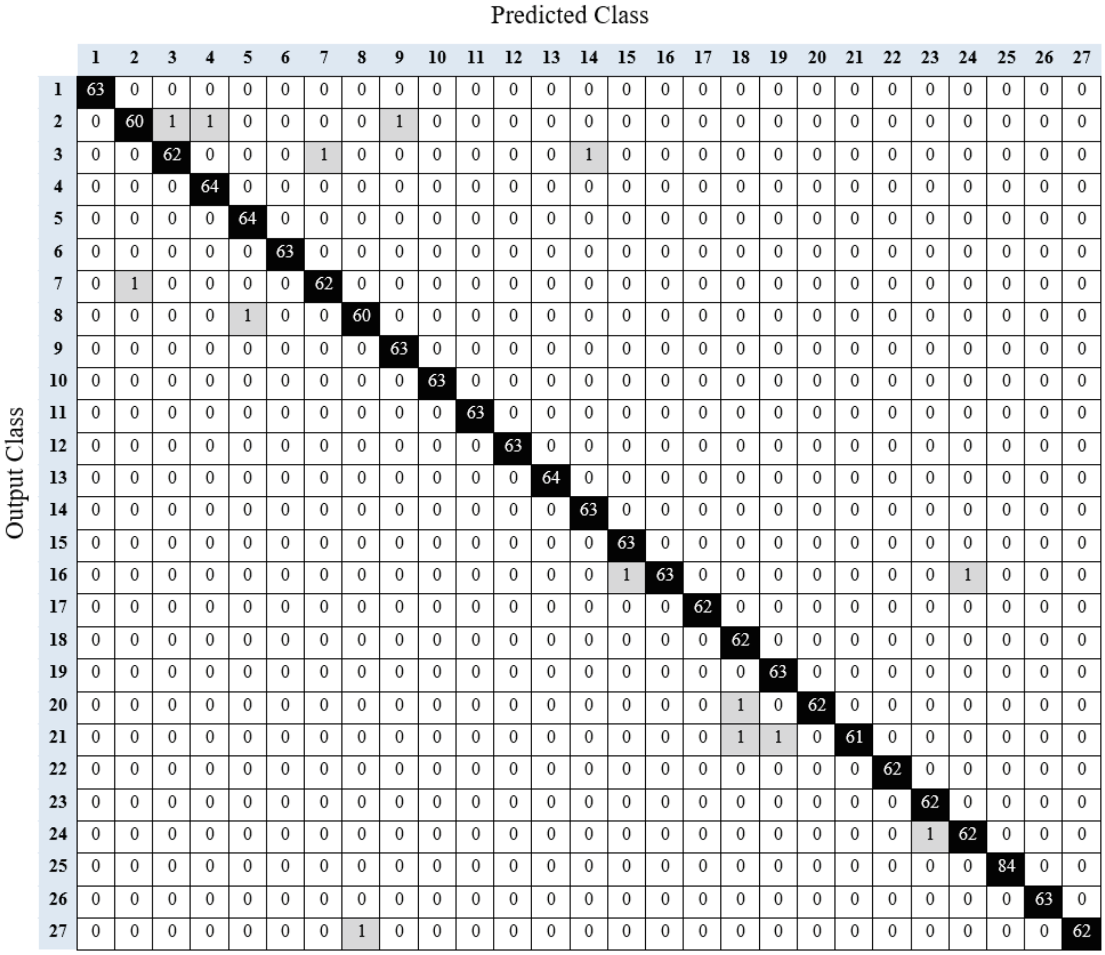
By using NCA, our model attained 99.19% classification accuracy. 96.44% classification accuracy was calculated without NCA and a bigger feature vector. Thus, we used the NCA selector in this research to get higher classification accuracy with lower running time.

We selected the SVM algorithm as the engine sound classification method. The selection process involved training and testing of 6 classifiers and these classifiers were linear discriminant (LD), naïve Bayes (NB), support vector machine (SVM), k nearest neighbours (kNN), bagged tree (BT) and artificial neural network (ANN). Classification accuracies of these classifiers are depicted in Figure 10.

Figure 10 demonstrates that the support vector machine (SVM) outperforms the other six algorithms in terms of classification accuracy, achieving a remarkable accuracy of 99.19%. The k-nearest neighbour (kNN) method comes in second with an accuracy of 97.81%. In contrast, the naive Bayes (NB) classifier shows the weakest performance with an overall accuracy of only 71.42%.

Table 4 summarizes state-of-the-art techniques developed for automated engine fault detection. That allows the reader to compare the performance of our method with related scientific work.

Table 4 shows that the collected dataset contains 27 different engine sounds, and it is the most comprehensive dataset when compared to other datasets. Based on this dataset, the PrismPatNet model obtained y of 98.75% and 99.19%98.75% and 99.19% with 10-fold cross-validation and hold-out validation, respectively. Recently, Wu and Chiang have obtained an accuracy of 99.4% in classifying five faults using their model (Wu & Chiang, 2009). In this respect, our proposed PrismPatNet was tested on the biggest dataset in the literature for engine fault detection. To be specific, our dataset contains 27 classes (fault types) and previous studies used up to 15 classes. Our PrismPatNet model reached an accuracy of 99.19% for the task of classifying 27 different engine sounds. Moreover, our model has low complexity. This positive property comes from the fact that we mimicked a basic CNN architecture and only used space domain. Our compression model is maximum pooling which is a basic

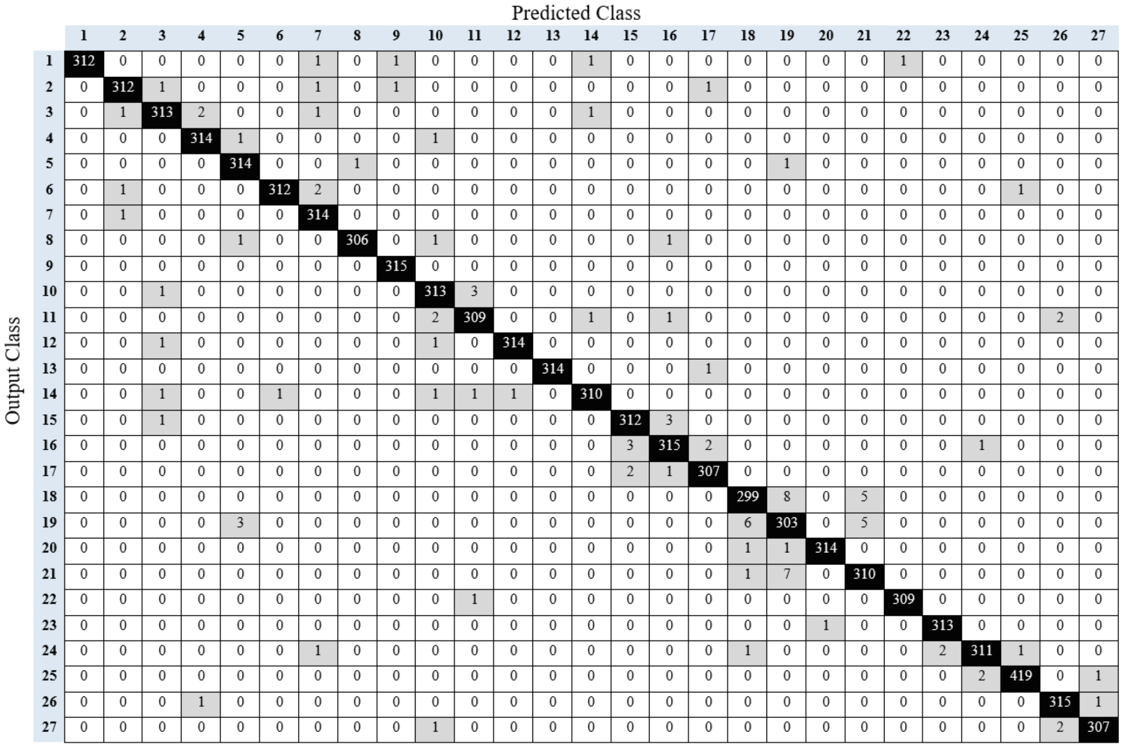


Predicted Class

Output Class

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	60	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	62	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	63	0	0	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	62	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	61	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	62	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0
27	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62

(a)



Predicted Class

Output Class

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	312	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
2	0	312	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	1	313	2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	314	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	314	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	1	0	0	0	312	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
7	0	1	0	0	0	0	314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	306	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	315	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	313	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	2	309	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2	0
12	0	0	1	0	0	0	0	0	0	1	0	314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	314	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	0	1	0	0	0	1	1	1	0	310	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	312	3	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	315	2	0	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	307	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	299	8	0	5	0	0	0	0	0	0
19	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	6	303	0	5	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	314	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	310	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	309	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	313	0	0	0	0
24	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	311	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	419	0	1
26	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	315	1
27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	307

(b)

FIGURE 8 Confusion matrices obtained using: (a) 80:20 hold-out validation and (b) 10-fold cross-validation strategy.

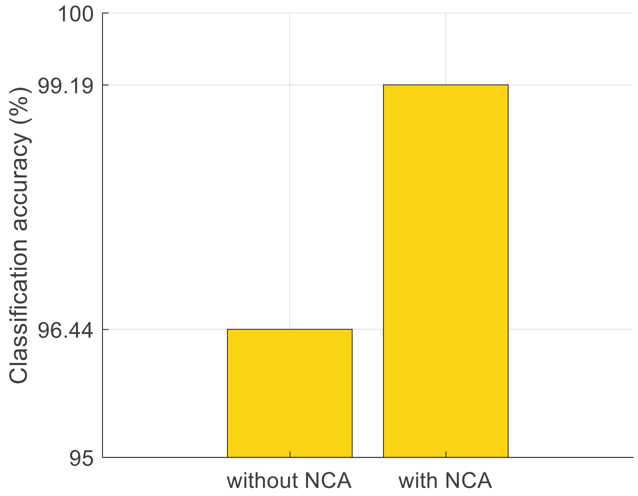


FIGURE 9 Classification accuracy with and without NCA.

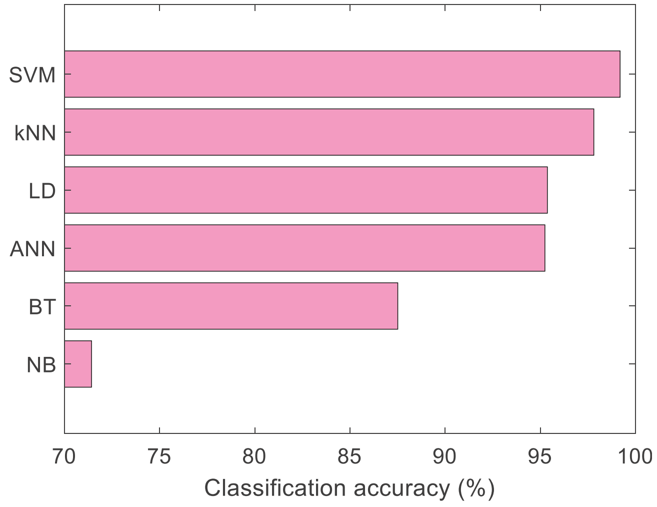


FIGURE 10 Accuracies of the six tested classification algorithms.

method. We have shown no need to use wavelet or other frequency transformations to get high classification accuracies. Pooling functions are good solutions for classification problems that require multiple feature levels. Furthermore, we tested the feature extraction capability of the prism pattern and showed a good performance for acoustic signal classification. Also, the proposed PrismPatNet can easily be implemented in embedded systems.

The main advantages of the proposed method are as follows:

- Developed a novel and accurate PrismPatNet model to classify 27 types of engine sounds.
- Obtained an accuracy of 99.19% and 98.75% using hold-out and 10-fold cross-validation, respectively.
- Created a big gasoline engine fault GEFC27 dataset of engine sounds to validate the developed model.
- The prism pattern has a 3D shape that can extract subtle features from the sound signal and yield high classification performance.
- Achieved an efficient, simple, economical, and lightweight learning network Figure 10 demonstrated that the best classifier among the six tested algorithms is SVM since it reached 99.19% classification accuracy. The second best algorithm is kNN with an accuracy of 97.81%. Finally, the NB showed the weakest performance, with an overall accuracy of 71.42%. (PrismPatNet).
- The proposed model is the first to classify 27 different types of engine sounds.

A limitation of this research is:

- We used only a single motor type. To get a larger dataset, different motor types and different environmental conditions should be used.

TABLE 4 Summary of state-of-the-art techniques developed for automated engine fault detection using sound signals

Study	Method	Number of classes	Split ratio	Results (%)
Wu and Chiang (Wu & Chiang, 2009)	Wigner–Ville distribution	5	40 testing 8 training	Acc: 99.40
Wu and Liu (Wu & Liu, 2008)	Discrete wavelet transform	6	120 testing 30 training	Acc: 95.00
Wong et al. (Wong et al., 2016)	Sparse Bayesian extreme learning machine	15	5-fold cross-validation	Acc: 76.94
Wang et al. (Wang et al., 2020)	Wavelet packet analysis, Artificial neural network	9	30 testing 30 training 30 validation	Acc: >90.00
Our method	PrismPatNet	27	80:20	Acc: 99.19 Pre: 99.18 Rec: 99.18 F1: 99.18 CK: 99.16 GM: 99.17 MCC: 99.15
			10-fold cross-validation	Acc: 98.75 Pre: 98.74 Rec: 98.74 F1: 98.74 CK: 98.70 GM: 98.74 MCC: 98.69

Abbreviations: Acc, accuracy; CK, Cohen's Kappa; F1: F1-score; GM, geometric mean; MCC, Mathew correlation coefficient; Pre, precision; Rec, recall.

In the future, we plan to collect data related to diesel, electricity, and gasoline motors/engines. Our model can also be used in various vehicles to automate motor fault detection.

Moreover, PrismPatNet is simple to use and based on mobile phone technology, which enables a layman to detect and subsequently repair engine faults. It can also be employed to assess engine faults in used and modified cars.

6 | CONCLUSIONS

In this work, we have developed a novel PrismPatNet model to automate the classification of 27 engine sounds. The generated system was validated with a newly collected GEFC27 dataset which comprises of 8614 observations, each belonging to one of 27 classes. The presented model was assessed with the GEFC27 dataset, and it reached 99.19% and 98.75% accuracy using 80:20 hold-out validation and 10-fold cross-validation, respectively. Our model is simple, accurate, fast, and easy to implement. It can also be used to classify other types of one-dimensional signals. In the future, we intend to explore new 3D shape feature generators based on deep models to classify sound signals.

The sounds used were collected from a noisy environment. Therefore, the test results reflect the expected performance for real-world applications. In the future, a big dataset can be collected from various motors and a big, labelled dataset can be created using the deep network models or the recommended PrismPatNet. By using these big, labelled datasets, automatic fault diagnosis applications can be developed. Moreover, these applications can be used by members of the public.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in gasoline_engine_fault_dataset.rar at http://web.firat.edu.tr/turkertuncer/gasoline_engine_fault_dataset.rar.

ORCID

Turker Tuncer https://orcid.org/0000-0002-5126-6445
Oliver Faust https://orcid.org/0000-0002-3979-4077

REFERENCES

Adaileh, W. M. (2013). Engine fault diagnosis using acoustic signals. In *Applied mechanics and materials* (Vol. 295-298, pp. 2013–2020). Trans Tech Publ.

Ahmed, R., el Sayed, M., Gadsden, S. A., Tjong, J., & Habibi, S. (2014). Automotive internal-combustion-engine fault detection and classification using artificial neural network techniques. *IEEE Transactions on Vehicular Technology*, *64*(1), 21–33.

Akbal, A. (2021). A local knit pattern-based automated fault classification method for the cooling system of the data center. *Applied Acoustics*, *176*, 107888.

Aljemely, A. H., Xuan, J., Xu, L., Jawad, F. K., & Al-Azzawi, O. (2021). Wise-local response convolutional neural network based on Naïve Bayes theorem for rotating machinery fault classification. *Applied Intelligence*, *51*, 1–19.

Aydemir, E., Tuncer, T., Dogan, S., & Unsal, M. A. (2021). Novel biometric recognition method based on multi kernelled bijection octal pattern using gait sound. *Applied Acoustics*, *173*, 107701.

Basir, O., & Yuan, X. (2007). Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory. *Information Fusion*, *8*(4), 379–386.

Chen, M., Quan, H. L., & Tang, H. (2015). An approach for optimal measurements selection on gas turbine engine fault diagnosis. *Journal of Engineering for Gas Turbines and Power*, *137*(7), 071203-1–071203-9.

Chen, Z., Yao, H., Lou, Y., Cao, Y., Liu, Y., Wang, H., & Liu, X. (2021). An empirical study on deployment faults of deep learning based mobile applications. *arXiv*, 210104930.

Dogan, S., Akbal, E., & Tuncer, T. (2020). A novel ternary and signum kernelled linear hexadecimal pattern and hybrid feature selection based environmental sound classification method. *Measurement*, *166*, 108151.

Gelgele, H. L., & Wang, K. (1998). An expert system for engine fault diagnosis: Development and application. *Journal of Intelligent Manufacturing*, *9*(6), 539–545.

Goldberger, J., Hinton, G. E., Roweis, S., & Salakhutdinov, R. R. (2004). Neighbourhood components analysis. *Advances in Neural Information Processing Systems*, *17*, 513–520.

Gundogdu, K., Bayrakdar, S., & Yucedag, I. (2018). Developing and modeling of voice control system for prosthetic robot arm in medical systems. *Journal of King Saud University-Computer and Information Sciences*, *30*(2), 198–205.

Jain, A., Singh, S. K., & Pratap Singh, K. (2021). Multi-task learning using GNet features and SVM classifier for signature identification. *IET Biometrics*, *10*(2), 117–126.

Jung, D. (2019). Engine fault diagnosis combining model-based residuals and data-driven classifiers. *IFAC-Papers Online*, *52*(5), 285–290.

Li, B., & Zhao, Y.-P. (2020). Group reduced kernel extreme learning machine for fault diagnosis of aircraft engine. *Engineering Applications of Artificial Intelligence*, *96*, 103968.

Oniani, S., Marques, G., Barnovi, S., Pires, I. M., & Bhoi, A. K. (2021). Artificial intelligence for internet of things and enhanced medical systems. In *Bio-inspired neurocomputing* (pp. 43–59). Springer.

Tomasik, J., Han, S. Y. S., Barton-Owen, G., Mirea, D.-M., Martin-Key, N. A., Rustogi, N., Lago, S. G., Olmert, T., Cooper, J. D., & Ozcan, S. (2021). A machine learning algorithm to differentiate bipolar disorder from major depressive disorder using an online mental health questionnaire and blood biomarker data. *Translational Psychiatry*, *11*(1), 1–12.

Tuncer, T., Akbal, E., & Dogan, S. (2021). An automated snoring sound classification method based on local dual octal pattern and iterative hybrid feature selector. *Biomedical Signal Processing and Control*, *63*, 102173.

Tuncer, T., Dogan, S., & Acharya, U. R. (2021). Application of Petersen graph pattern technique for automated detection of heart valve diseases with PCG signals. *Information Sciences*, *565*, 91–104.

Tuncer, T., Dogan, S., Pławiak, P., & Acharya, U. R. (2019). Automated arrhythmia detection using novel hexadecimal local pattern and multilevel wavelet transform with ECG signals. *Knowledge-Based Systems*, *186*, 104923.

Tuncer, T., Dogan, S., & Subasi, A. (2021). A new fractal pattern feature generation function based emotion recognition method using EEG. *Chaos, Solitons & Fractals*, *144*, 110671.

Vapnik, V. (1998). The support vector method of function estimation. In *Nonlinear modeling* (pp. 55–85). Springer.

Wang, X., Kruger, U., Irwin, G. W., McCullough, G., & McDowell, N. (2007). Nonlinear PCA with the local approach for diesel engine fault detection and diagnosis. *IEEE Transactions on Control Systems Technology*, *16*(1), 122–129.

Wang, Y., Liu, N., Guo, H., & Wang, X. (2020). An engine-fault-diagnosis system based on sound intensity analysis and wavelet packet pre-processing neural network. *Engineering Applications of Artificial Intelligence*, *94*, 103765.

Wang, Y., Ma, Q., Zhu, Q., Liu, X., & Zhao, L. (2014). An intelligent approach for engine fault diagnosis based on Hilbert–Huang transform and support vector machine. *Applied Acoustics*, *75*, 1–9.

Wong, P. K., Zhong, J., Yang, Z., & Vong, C. M. (2016). Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis. *Neurocomputing*, *174*, 331–343.

Wu, J.-D., & Chiang, P.-H. (2009). Application of Wigner–Ville distribution and probability neural network for scooter engine fault diagnosis. *Expert Systems with Applications*, *36*(2), 2187–2199.

Wu, J.-D., & Chuang, C.-Q. (2005). Fault diagnosis of internal combustion engines using visual dot patterns of acoustic and vibration signals. *NDT & e International*, *38*(8), 605–614.

Wu, J.-D., Huang, C.-K., Chang, Y.-W., & Shiao, Y.-J. (2010). Fault diagnosis for internal combustion engines using intake manifold pressure and artificial neural network. *Expert Systems with Applications*, *37*(2), 949–958.

Wu, J.-D., & Liu, C.-H. (2008). Investigation of engine fault diagnosis using discrete wavelet transform and neural network. *Expert Systems with Applications*, *35*(3), 1200–1213.

Xu, K., Qian, J., Hu, Z., Duan, Z., Chen, C., Liu, J., Sun, J., Wei, S., & Xing, X. (2021). A new machine learning approach in detecting the oil palm plantations using remote sensing data. *Remote Sensing*, *13*(2), 236.

Yaman, O. (2021). An automated faults classification method based on binary pattern and neighborhood component analysis using induction motor. *Measurement*, *168*, 108323.

Yan, W. (2006). Application of random forest to aircraft engine fault diagnosis. In: The proceedings of the multiconference on" computational engineering in systems applications", IEEE.

Yang, Y.-S., Ming, A.-b., Zhang, Y.-y., & Zhu, Y.-S. (2017). Discriminative non-negative matrix factorization (DNMF) and its application to the fault diagnosis of diesel engine. *Mechanical Systems and Signal Processing*, *95*, 158–171.

Yao, D., Li, B., Liu, H., Yang, J., & Jia, L. (2021). Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. *Measurement*, *109166*, 109166.

Zhang, H., & Deng, Y. (2018). Engine fault diagnosis based on sensor data fusion considering information quality and evidence theory. *Advances in Mechanical Engineering*, *10*(11), 1687814018809184.

Zhang, H., & Deng, Y. (2020). Weighted belief function of sensor data fusion in engine fault diagnosis. *Soft Computing*, *24*(3), 2329–2339.

Zhao, Y.-P., Wang, J.-J., Li, X.-Y., Peng, G.-J., & Yang, Z. (2020). Extended least squares support vector machine with applications to fault diagnosis of aircraft engine. *ISA Transactions*, *97*, 189–201.

How to cite this article: Sahin, S. E., Gulhan, G., Barua, P. D., Tuncer, T., Dogan, S., Faust, O., & Acharya, U. R. (2023). PrismPatNet: Novel prism pattern network for accurate fault classification using engine sound signals. *Expert Systems*, *40*(8), e13312. https://doi.org/10.1111/exsy.13312

APPENDIX A

The MATLAB code of the proposed PrismPat has been given below.

```
function histo=prism_pat(signal)
h1=zeros(1512); h2=zeros(1512);
for i=1:length(signal)-49
  pp = signal(i:i + 49);
  m1 = reshape(pp(1:25),[5,5])’; % m1 is bottom.
  m2 = reshape(pp(26:50),[5,5])’; % m2 is top.
  f(1) = m2(1,2)-m2(1,4); f(2) = m2(1,4)-m2(3,5);
  f(3) = m2(3,5)-m2(5,4); f(4) = m2(5,4)-m2(5,2);
  f(5) = m2(5,2)-m2(3,1); f(6) = m2(3,1)-m2(1,2);
  f(7) = m1(1,2)-m2(1,2); f(8) = m1(1,4)-m2(1,4);
  f(9) = m1(3,5)-m2(3,5); f(10) = m1(5,4)-m2(5,4);
  f(11) = m1(5,2)-m2(5,2); f(12) = m1(3,1)-m2(3,1);
  f(13) = m1(1,2)-m1(1,4); f(14) = m1(1,4)-m1(3,5);
  f(15) = m1(3,5)-m1(5,4); f(16) = m1(5,4)-m1(5,2);
  f(17) = m1(5,2)-m1(3,1); f(18) = m1(3,1)-m1(1,2);
  map1(i) = 0; map2(i) = 0;
  for j = 1:9
    map1(i) = map1(i) + double(f(j) > =0)*2^(j-1);
    map2(i) = map2(i) + double(f(j + 9) > =0)*2^(j-1);
  end
  h1(map1(i) + 1) = h1(map1(i) + 1) + 1; h2(map2(i) + 1) = h2(map2(i) + 1) + 1;
end
histo=[h1 h2];
```

AUTHOR BIOGRAPHIES

Sakir Engin Sahin received Bsc. Degree in the Firat University. Now, He is working in the Department of Computer Technologies, Arapgir Vocational School at Malatya Turgut Ozal University. He is working on the development of machine learning and feature engineering techniques.

Gokhan Gulhan is working in the Department of Motor Vehicles and Transportation Technologies, Automotive Technology Program, Arapgir Vocational School at Malatya Turgut Ozal University. His interest areas are engine fault detection and classification.

Prabal Datta Barua PhD obtained PhD (Information System) from the University of Southern Queensland. He is an academic and accredited research supervisor at the University of Southern Queensland with 12 years of teaching experience. Dr Barua received research support from the Queensland Government Innovation Connections under the Entrepreneurs program to research "Cancer Recurrence Using Innovative Machine Learning Approaches". Dr Barua is interested in AI technology development in health, education, agriculture and environmental science and published several papers in the Q1 journal. He is an industry leader in ICT entrepreneurship in Australia and sitting as an ICT advisory panel member of many organisations. Dr Barua is an Adjunct Professor at the University of Southern Queensland and an Honourary Industry Fellow at the University of Technology Sydney.

Turker Tuncer received the master's degree in electronics and computer sciences and the PhD degree in software engineering from Firat University, Elazig, Turkey, in 2011 and 2016, respectively. He is currently an Associate Professor with the Digital Forensics Engineering, Technology Faculty, Firat University. His main research interests include feature engineering, image processing, signal processing, information security, and pattern recognition. He has been working actively on developing algorithms in machine learning applied to visual surveillance and biomedical data.

Sengul Dogan received the master's degree in bioengineering the PhD degree in electrical and electronics engineering from the Firat University, Elazig, Turkey, in 2007 and 2011, respectively. She is currently an Associate Professor with the Digital Forensics Engineering, Technology Faculty, Firat University. Her main research interests include computer forensics, mobile forensics, image processing, and signal processing. She has been working actively on developing algorithms in machine learning for biomedical data.

Oliver Faust is a well-known author in the field of medical engineering and computing. His research interests include medical processing and physiological signal analysis. He is also an expert in artificial intelligence for medical decision support. Editorial board member for MDPI International Journal of Environmental Research and Public Health as well as associate editor for Biomedical Signal Processing in Frontiers in signal processing.

U Rajendra Acharya MTech, PhD, DEng, DSc is a senior faculty member at Ngee Ann Polytechnic, Singapore. He is also (i) Adjunct Professor at University of Malaya, Malaysia, (ii) Adjunct Professor at Asia University, Taiwan, (iii) Associate faculty at Singapore University of Social Sciences, Singapore and (iv) Adjunct Professor at University of Southern Queensland, Australia. He received his PhD from National Institute of Technology Karnataka (Surathkal, India), DEng from Chiba University (Japan) and DSc from AGH University of Science and Technology, Poland. He has published more than 500 papers, in refereed international SCI-IF journals (345), international conference proceedings (42), books (17) with more than 65,000 citations in Google Scholar (with h-index of 128). He has worked on various funded projects, with grants worth more than 6 million SGD. He is ranked in the top 1% of the Highly Cited Researchers for the last seven consecutive years (2016 to 2022) in Computer Science according to the Essential Science Indicators of Thomson. He is on the editorial board of many journals and has served as Guest Editor for many journals. He is on the editorial board of many journals and has served as Guest Editor for many journals. Please visit https://scholar.google.com.sg/citations?user=8FjY99sAAAAJ&hl=en for more details.