International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

BMHMCRAS: Design of a Bio-inspired Model for High-Efficiency Multipurpose Data Compression & Representation via Adaptive Signal Analysis

Moumita Majumder Sarkar [1]* Dr Manish Dhananjay Sawale[2]

Submitted: 20/05/2023 **Revised**: 06/07/2023 **Accepted**: 25/07/2023

Abstract: Signal processing applications rely on pre-processing, context-based data selection, representation of data into features & their selection of input datasets. These operations enhance the effectiveness of classification, prediction, and other data-dependent applications. Researchers have proposed a variety of machine learning models to perform this task, each with its own nuances, advantages, limitations, and future research directions. However, these models are either application-specific or employ a black-box approach, limiting their scalability and context-specific performance across a variety of applications. This text proposes a novel bio-inspired model that combines high-efficiency multipurpose data compression and representation through adaptive signal analysis in order to circumvent this limitation. The proposed model combines two distinct bio-inspired methods for the parametric tuning of compression and feature extraction models, respectively. Initially, a Particle Swarm Optimizer (PSO) Model is used to determine the optimal compression model parameters. This compression can be lossy or lossless, and it can be altered according to the needs of the deployed applications. This text makes use of an ensemble compression layer that combines Huffman, Run Length Encoding (RLE), Wavelet, ZLib, BZ2, and LZMA, which aided in high-density compression and dataset aggregation. These compressed and aggregated signals are subjected to a feature extraction process that is controlled by a Multiple objective Genetic Algorithm (MGA) Model, which aids in the selection of window sizes, stride sizes, padding sizes, etc. of the feature extraction model. For large-scale feature extraction, this model employs a combination of statistical, convolutional, Fourier, and Cosine transformation techniques. Multiple objective GA models also perform the selection of these features. The fused MGA PSO Model was applied to a wide range of applications, such as electrocardiogram (ECG) classification, plant disease detection, stock market prediction, and credit card fraud detection using various test signals. It was observed that the proposed model could improve classification accuracy by 3.5 percent, classification precision by 4.1 percent, and classification recall by 3.8 percent, while it could also improve prediction accuracy by 5.4 percent, precision by 3.9 percent, and recall by 4.5 percent when averaged across different test sequences. Due to the use of compression layer, the model was also able to reduce the processing delay by 8.5% compared to other contemporary methods.

Keywords: *Feature, Extraction, Compression, Selection, Fusion, Bioinspired, MGA, PSO*

1. Introduction

Adaptive signal processing is a multidisciplinary field that involves data compression, representation, classification, post-processing, and other complex signal-based operations. A typical adaptive signal processing model is depicted in figure 1, wherein data collection, adaptive cancellation for denoising & compression, Cross Ambiguity Function (CAF) & Constant False Alarm Rate (CFAR) detection for continuous tuning, target feature variance detection, harmonic detection, & feature selection models are depicted [1].

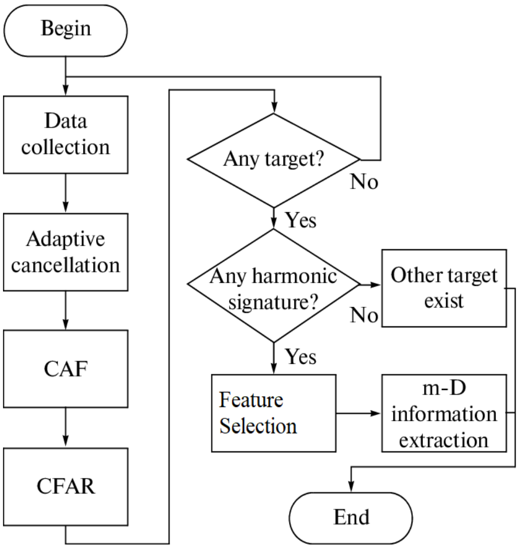

Fig 1. A typical Adaptive Signal Processing Model

[1]*Department of Electronics and Communication Engineering, Oriental University, Indore, India.*
mukunjsarkar1985@gmail.com
[2]*Department of Electronics and Communication Engineering, Oriental University, Indore, India*
manish.sawale@gmail.com

This model yields multidimensional features that are applicable to a wide range of application deployments. Researchers propose similar models by altering their internal characteristics. The following section of this text provides an overview of these models [2, 3, 4, 5, 6] in terms of their nuances, advantages, limitations, and future research scopes. On the basis of this discussion, it was determined that existing models are either application-specific, which enables high-accuracy operation but has limited scalability, or use a black-box framework for high-density but low-accuracy feature extraction, which limits their deployment capabilities. In order to overcome this limitation, Section 3 proposes the design of a novel bio-inspired model for high-efficiency multipurpose data compression and representation by means of adaptive signal analysis. The model combines Multiple objective Genetic Algorithm (MGA) and Particle Swarm Optimization (PSO) for feature extraction & selection and data compression optimizations, respectively. The fusion of both models facilitates the extraction of high-density features, which can be utilized in multiple applications. Different test datasets were used to evaluate the performance of this model for electrocardiogram (ECG) classification, plant disease detection, stock market prediction, and credit card fraud detection. In section 4, the performance of the model for these applications is evaluated in terms of accuracy, precision, recall, and delay metrics. On the basis of this evaluation, readers will be able to determine the performance enhancement percentage after implementing the proposed model. This text concludes with some insightful observations regarding the proposed model, as well as suggestions for improving its performance.

2. Literature Review

Feature extraction & data representation models allow researchers and system designers to easily identify patterns between different sets of data sources. For instance, work in [1, 2, 3] proposes methods for ECG classification via use of kernelized fuzzy rough sets (KFR), Genetic Algorithm (GA) with Support Vector Machine (SVM), and Slope Variation Measurement for better feature representation of these signals. The performance of these models is further extended via the work in [4, 5, 6], wherein researchers have used CNN with SVM, Multiple Leads-based Branch Fusion Network (MLBF-Net), and Multiple Feature Sparse Representations Learning with Collective Matrix Factorization for improving classification performance under different datasets.

Methods that utilize plant diseases via feature variance estimation are discussed in [7], wherein multiple sets of features are extracted via different deep learning methods, and their performance is compared in terms of accuracy, precision and recall metrics. Specific models that perform similar tasks are discussed in [8, 9, 10], which propose use of Long Short-Term Memory (LSTM), Texture Features with CNN (TF CNN), and Multiple Task learning with Attention Features (MTL AF) showcase that extracting multimodal feature sets improves accuracy of classification for different applications. Similar observations are done in [11, 12], which propose that depth=wise separable convolution features with CNN (DSCNN), and extreme learning machine (ELM) with Kuan filtering (KF) has better performance when compared with linear feature extraction methods for the same datasets.

When considering credit card fraud detection applications, work in [13, 14, 15] proposes use of Temporal Transaction Scraping with Autoencoder Based Feature Engineering (TTS ABF), Neural Feature Aggregation Framework (NFAG), LSTM with Adaboost is used for improving classification performance under different datasets. Extensions to these models are discussed in [16, 17, 18], which propose use of Transfer Learning Strategies (TLS), topological features via Node2Vec, and Sparse Feature Selection with Overlap Minimization (SFS OM), methods for better performance under different input datasets. These methods utilize feature variance to reduce redundancies during classification & prediction phases. Another application which was considered for evaluation uses stock value features, for prediction of different inter-day and intra-day stock prices. Work in [19, 20, 21] propose use of Extreme Gradient Boosting (XGBoost) with Deep Neural Network (DNN) regression model, Hybrid Red Deer Grey Algorithm (HRDGA), and Candlestick Charting with Novelty Feature Engineering Scheme (CC NFES) in order to extract high-density stock indices, which can be used for low-complexity classification scenarios. Similarly, models proposed in [22, 23, 24] also discuss use of novel feature extraction & classification methods including, GA with LSTM, LSTM features with Multimodal augmentation process, and Spatial-Temporal Deep Neural Network (ST DNN) for improved prediction performance under different stock market types.

These models showcase low redundancy, and high efficiency feature extraction & classification methods, which are used for evaluation by this text. But these models have limited scalability due to their application-specific characteristics, which can be improved via use of the proposed MGA & PSO Model described in the next section of this text, and can be applied to varying application scenarios.

3. Proposed MGA & PSO Model for high-efficiency feature representation

Based on the literature survey it was observed a wide variety of feature extraction, and data representation models are proposed by researchers, and each of them have their own context-specific or application independent characteristics. But these models either use a customised method for individual applications, or utilize a general-purpose black-box method for higher scalability. In both cases, either performance is compromised, or scalability is reduced, which limits their application to a small sub-set of data sources. To overcome this limitation, a novel bioinspired model for high-efficiency multipurpose data compression & representation via adaptive signal analysis is discussed in this text. The proposed model initially uses a combination of parameter tuned Huffman, Run Length Encoding (RLE), Wavelet, ZLib, BZ2, and LZMA methods for high-density compression & data aggregation processes. This operation is followed by another bioinspired model that combines statistical, convolutional, Fourier, and Cosine transformation methods for high-density feature extraction, that can be used by multiple applications. Overall flow of the proposed model is depicted in figure 2, wherein data compression, feature extraction, and application-specific control models are visualized, which can be applied to multiple deployment scenarios.

The model design is divided into multiple sub modules, and each of these modules are discussed in different sub-sections of this text. Based on this discussion, readers will be able to implement these modules in part(s) or as whole, depending upon their application requirements on different types datasets.

3.1. Design of the PSO based data compression layer

The dataset of different applications is directly processed via a compression layer, which assists in dimensionality reduction for improving computational speed of underlying application deployments.

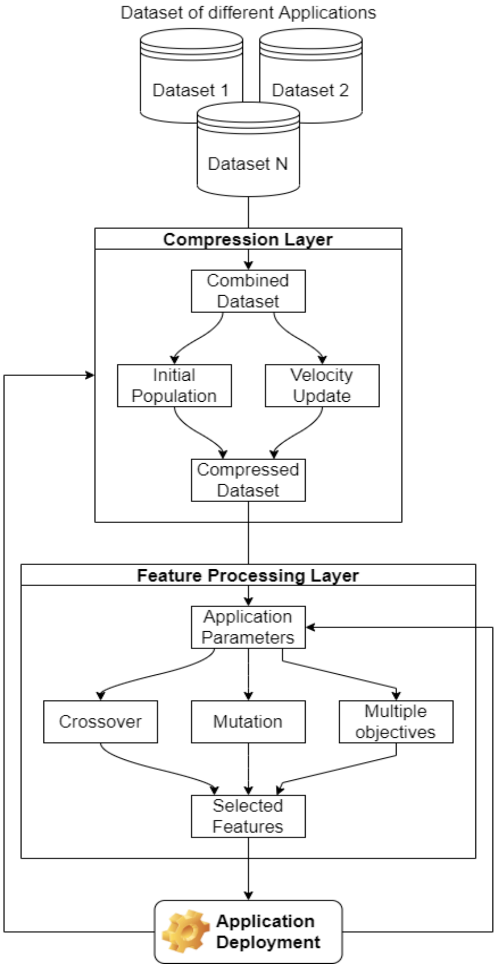

Fig 2. Overall flow of the proposed model for feature compression, extraction & selection process

This layer uses a combination of lossless compression models including, Huffman, Run Length Encoding (RLE), ZLib, BZ2, & LZMA, and lossy data compression via different Wavelet methods to design a novel high efficiency compression & aggregation layer. Table 1 summarizes different parameters for these models, which are used by PSO for application-specific optimizations.

Method	**Parameters**	**Reason for selection**
Huffman	Maximum length of code $Max_L(H)$	Produces highly compact codes for input dataset, and has good representation efficiency
RLE	None	Simple to encode and decode, has faster performance

ZLib	Compression Level $C_L(Z)$	Provides better compression performance when compared with RLE & Huffman
BZ2	Compression Level $C_L(B)$	Has higher compression ratio that ZLib
LZMA	Compression Level $C_L(L)$ Dictionary Size $D_L(L)$	Highly customizable lossless compression with good compression ratio under different datasets
Wavelet	Wavelet type W_T	Guarantees 75% compression with minimum distortion in data samples

Table 1. Methods & their parameters

Based on these methods, a PSO based Model is deployed, which assists in dataset specific compression for improving efficiency during application deployments. This PSO Model works via the following process,

- Initialize following PSO parameters,
  - Iterations needed for optimization (N_i)
  - Particles needed for optimization (N_p)
  - Cognitive Rate of Learning (L_c)
  - Social Learning Rate (L_s)
  - Application compression requirements (Lossy or Lossless)
- Based on these parameters, initial population is generated via the following process,
  - For each particle in 1 to N_p perform the following process,
    - If the compression type is lossy, then stochastically select different wavelet types from Haar, Daubechies, Bior, Coiflet, Symlet, Morlet, Meyer, and Reverse Bior
    - If the compression type is lossless, then evaluate values of compression models via equation 1,

$$CP_i = STOCH\big(Min(P_i), Max(P_i)\big) \ldots (1)$$

Where, $STOCH$ represents a stochastic process, while P_i represents tuneable parameter for the i^{th} compression model which is lossless.

- Perform cascaded compression via these methods, and evaluate particle fitness via equation 2 as follows,

$$f_p = \frac{OUT(L)}{IN(L)} \ldots (2)$$

Where, $OUT(L), IN(L)$ represents length of the output data, and length of the input data respectively, while f_p represents particle fitness, and it initially marked as particle best ($PBest$) value

- Maximum value of this fitness is marked as Global Best ($GBest$) via equation 3 as follows,

$$GBest = Max\left[\bigcup_{i=1}^{N_p} PBest_i\right] \ldots (3)$$

- Based on these values, new particle values are estimated by iterating the following process N_i times,
  - Evaluate new value of particle position via equation 4,

$$New(P) = Old(P) * r + L_c[Old(P) - PBest] + L_s[Old(P) - GBest] \ldots (4)$$

Where, $New(P), \& \, Old(P)$ represents new particle fitness, and old particle fitness respectively, while r represents a stochastic number between $(0,1)$, which is evaluated via a Gold Code Stochastic Number Generator process.

  - Update $PBest$, if $New(P) > PBest$, else go to the next iteration
  - If $PBest$ is updated, then modify the compression parameters stochastically to obtain the $PBest$ value
- At the end of N_i iterations, select the solution with maximum fitness value, and use its parameters for final data compression & aggregation process.

Based on this optimization model, the data is compressed with highest compression ratio, and can be used for feature extraction process. The feature optimization process is described in the next sub-section of this text.

3.2. Discussion about MGA based feature processing operations

The compressed features have higher density when compared with their non-compressed form, this is because compression aims at reducing redundancies in the datasets, which assists in improving its compression ratio for different input types. This compressed data is processed via a Multiple objective

Genetic Algorithm (MGA) Model, that combines convolutional, Fourier, Cosine, Wavelet and Statistical feature vectors for improving its data representation capabilities. The model also optimizes feature variance, which assists in improving its representation capabilities. It requires extracted features for optimization, which are evaluated via different Mathematical identities. For instance, Convolutional features are extracted via equation 5 as follows,

$$Conv_{out_{i,j}} = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} \sum_{b=-\frac{n}{2}}^{\frac{n}{2}} Data(i-a, j-b) * ReLU\left(\frac{m}{2}+a, \frac{n}{2}+b\right) \ldots (5)$$

Where, m, n represents dimensions of window for the input $Data$, while a, b represents stride sizes for the convolution process, and $ReLU$ represents a Rectilinear Unit, which is used for activation of features. These features can be evaluated for any dimension of input data, and thus contain some redundancies which must be removed via variance maximization process. These features are extended via use of Fourier transform on compressed data via equation 6,

$$F_{out} = \sum_{i=0}^{N-1} Data(i) * \exp\left[-j * 2 * \pi * w * \frac{i}{N}\right] \ldots (6)$$

Where, w represents window size for the Fourier transform, while N represents number of compressed data samples. These features are further extended via use of Cosine transform features which are evaluated via equation 7 as follows,

$$D_{out_w} = \frac{1}{\sqrt{2N}} * Data_w * \sum_{i=0}^{N-1} Data_i * \cos\left[(2*i+1)*j * \frac{\pi}{2*N}\right] \ldots (7)$$

Where, w represents size of window for the Cosine transform, and can be tuned via optimization processes. These features are combined with statistical features, that include variance, standard deviation, minimum value, maximum value, and mean value, which are evaluated via equations 8, 9, and 10 respectively as follows,

$$Std = \sqrt{\sum_{i=1}^{N} \frac{\left[Data_i - \sum_{j=1}^{N} \frac{Data_j}{N}\right]^2}{N-1}} \ldots (8)$$

$$Var = Std^2 \ldots (9)$$

$$Mean = \sum_{j=1}^{N} \frac{Data_j}{N} \ldots (10)$$

Similarly, approximate and diagonal wavelet features are evaluated via equations 11 and 12 as follows,

$$A(W)_i = \frac{Data_i + Data_{i+1}}{2} \ldots (11)$$

$$D(W)_i = \frac{Data_i - Data_{i+1}}{2} \ldots (12)$$

All the features are combined to form a super feature vector (SFV), which is processed via the MGA Model for improving feature representation efficiency, and variance levels. The MGA Model works via the following process,

- Initialize MGA parameters as follows,
  - Iterations needed for optimization (N_i)
  - Solutions needed for optimization (N_s)
  - Rate for solution learning (L_r)
- To start the process, mark every solution as 'should be modified'
- Go to individual iterations between 1 to N_i
  - Go to individual solutions between 1 to N_s
    - Modify all 'should be modified' solutions via the following process,
- Generate stochastic values for convolution window size, convolution padding size, Fourier window size, and Cosine transform window size
- Based on these sizes, evaluate the SFV, and estimate its fitness via equation 13 as follows,

$$f_i = \sum_{i=1}^{N} \frac{\left[SFV_i - \sum_{j=1}^{N} \frac{SFV_j}{N}\right]^2}{N-1} * AppM_i(SFV) \ldots (13)$$

Where, $AppM$ represents the application specific metric which requires optimization (maximization), and can be accuracy, precision, recall, fMeasure, Area Under the Curve (AUC), etc. This metric is evaluated via the selected feature vectors, and used for continuous performance improvements.

- Estimate solution fitness via equation 14,

$$f_{th} = \sum_{i=1}^{N_s} f_i * \frac{L_r}{N_s} \ldots (14)$$

○ Solutions are termed as 'should be modified', where $f_i \leq f_{th}$, else they are mark as 'should not be modified'

- Repeat this process for all iterations, and select the feature configuration with maximum fitness values.

Based on this process, a SFV is evaluated via selection of most variant, and highly efficient feature vectors. The feature vectors are further tuned via a feedback layer, which assists in continuous tuning of the model as per the deployed application characteristics.

3.3. Design of the feedback layer for continuous performance optimizations

Once features are extracted via the MGA model, they are further optimized via a continuous learning process. This process uses a combination of correlation-based matching, along with continuous parameter update for achieving better application-specific performance under different input conditions. To perform this task, every new input entry from dataset is given to compression & feature extraction layers to generate a SFV, which is correlated with all training samples via equation 15 as follows,

$$Corr_j = \frac{\sum_{i=1}^{N} SFV(Train_i) - SFV(New)}{\sqrt{\sum_{i=1}^{N}[SFV(Train_i) - SFV(New)]^2}} \dots (15)$$

If the value of correlation with any training sample is higher than 0.999, then the sample is added to the training set, and optimization models are reevaluated with these new samples. Based on these optimizations, model variance is improved, and a continuous improvement in classification accuracy & prediction performance is observed. The proposed model was evaluated under different datasets, and its efficiency was compared with various state-of-the-art methods in the next section of this text.

4. Results & Comparison

Combining feature compression, feature extraction, and variance-based feature selection, the proposed model enhances the performance of various signal processing applications. To evaluate this performance, applications for electrocardiogram (ECG) classification, plant disease detection, stock market prediction, and credit card fraud detection were deployed and their accuracy, precision, recall, and delay were measured. The ECG data was extracted from the standard MITBIH dataset, the plant disease data from the Plant Pathology dataset, the stock market data from the Yahoo Finance dataset, and the credit card fraud detection data from the Kaggle datasets. All of these datasets were categorized using an Artificial Neural Network (ANN) model, and the model's performance was assessed using various test signal sets. Each set was divided into a 70:30 ratio, with 70% of the entries used for training and the remaining 30% for evaluating the deployed application scenarios. The accuracy of ECG classification for various datasets when compared to GA SVM [2], CNN SVM [4], and MLBF [5] can be viewed in table 2 as follows,

Test Entries	**A (%)** **GA SVM [2]**	**A (%)** **CNN SVM [4]**	**A (%)** **MLBF [5]**	**A (%)** **BMHM CRAS**
500	79.50	85.30	86.74	94.92
1000	79.80	85.60	87.05	95.27
1500	79.90	85.90	87.26	95.50
2000	80.20	86.20	87.58	95.84
2500	80.40	86.50	87.84	96.13
3000	80.62	86.80	88.12	96.43
3500	80.84	87.10	88.39	96.73
4000	81.06	87.40	88.66	97.03
4500	81.28	87.70	88.94	97.33
5000	81.50	88.00	89.21	97.63
6000	81.72	88.30	89.48	97.93
7000	81.94	88.60	89.76	98.23
8000	82.16	88.90	90.03	98.53
9000	82.38	89.20	90.31	98.82
10000	82.60	89.50	90.58	99.12

Table 2. Accuracy for ECG Classification

In terms of accuracy performance, it can be seen that the proposed model is 16.1% more accurate than GA SVM [2], 9.3% more accurate than CNN SVM [4, and 8.5% more accurate than MLBF [5]. Similarly, the precision performance for the same application can be observed in the following table 3,

Test Entries	**P (%)** **GA SVM [2]**	**P (%)** **CNN SVM [4]**	**P (%)** **MLBF [5]**	**P (%)** **BMHM CRAS**
500	75.86	81.38	82.76	90.56
1000	76.05	81.67	83.01	90.84
1500	76.24	81.95	83.26	91.11
2000	76.48	82.24	83.53	91.41
2500	76.68	82.52	83.79	91.69
3000	76.89	82.81	84.05	91.98
3500	77.10	83.10	84.31	92.26
4000	77.30	83.38	84.57	92.55
4500	77.51	83.67	84.83	92.84
5000	77.72	83.95	85.09	93.12
6000	77.93	84.24	85.35	93.41
7000	78.14	84.52	85.61	93.69
8000	78.35	84.81	85.87	93.98
9000	78.56	85.10	86.14	94.26
10000	78.77	85.38	86.40	94.55

Table 3. Precision for ECG Classification

In terms of precision performance, it can be observed that the proposed model is 15.5% superior to GA SVM [2], 8.3% superior to CNN SVM [4], and 7.9% superior to MLBF [5]. Similarly, the recall performance for the same application can be observed in the following table 4,

Test Entries	**R (%)** **GA SVM [2]**	**R (%)** **CNN SVM [4]**	**R (%)** **MLBF [5]**	**R (%)** **BMHM CRAS**
500	79.88	85.70	87.14	95.36
1000	80.13	86.00	87.43	95.68
1500	80.28	86.30	87.67	95.94
2000	80.55	86.60	87.98	96.28
2500	80.76	86.90	88.24	96.57
3000	80.98	87.20	88.52	96.87
3500	81.20	87.50	88.79	97.17
4000	81.42	87.81	89.07	97.47
4500	81.64	88.11	89.34	97.77
5000	81.86	88.41	89.62	98.07
6000	82.08	88.71	89.89	98.37
7000	82.30	89.01	90.17	98.67
8000	82.53	89.31	90.44	98.97
9000	82.75	89.61	90.71	99.27
10000	82.97	89.91	90.99	99.57

Table 4. Recall for ECG Classification

In terms of recall performance, it can be observed that the proposed model is 14.8% superior to GA SVM [2], 10.3% superior to CNN SVM [4], and 9.4% superior to MLBF [5]. Similarly, the delay performance for the same application can be observed in the following table 5,

Test Entries	**D (ms)** **GA SVM [2]**	**D (ms)** **CNN SVM [4]**	**D (ms)** **MLBF [5]**	**D (ms)** **BMHM CRAS**
500	64.77	47.62	43.36	19.15
1000	64.03	46.74	42.51	18.21
1500	63.59	45.85	41.81	17.45
2000	62.77	44.96	40.91	16.47
2500	62.16	44.07	40.13	15.61
3000	61.51	43.19	39.32	14.72
3500	60.86	42.30	38.51	13.84
4000	60.21	41.41	37.70	12.95
4500	59.56	40.53	36.89	12.07
5000	58.91	39.64	36.08	11.18
6000	58.26	38.75	35.27	10.30
7000	57.61	37.87	34.46	9.41
8000	56.96	36.98	33.65	8.53
9000	56.31	36.09	32.84	7.64
10000	55.66	35.21	32.04	6.76

Table 5. Delay for ECG Classification

On the basis of this performance, it can be seen that the proposed model is 31.5% better than GA SVM [2], 26.8% better than CNN SVM [4], and 25.5% better than MLBF [5], in terms of delay performance due to the use of compression models, making it useful for high efficiency and high-speed ECG classification applications. Similarly, the accuracy of plant disease classification for various datasets when compared with TF CNN [9], MTL AF [10], and ELM KF [12] can be observed in table 6 as follows,

Test Images	**A (%)** **TF CNN [9]**	**A (%)** **MTL AF [10]**	**A (%)** **ELM KF [12]**	**A (%)** **BMHM CRAS**
200	65.40	83.20	78.21	85.59
400	65.90	83.50	78.63	86.05
600	66.50	83.90	79.16	86.63
800	67.90	85.10	80.53	88.12
1000	68.50	85.40	81.00	88.64
1200	69.30	85.60	81.53	89.22
1400	70.12	86.34	82.35	90.12
1600	70.94	86.88	83.06	90.90
1800	71.76	87.42	83.78	91.68
2000	72.58	87.96	84.49	92.47
2200	73.40	88.50	85.21	93.25
2400	74.22	89.04	85.93	94.03
2600	75.04	89.58	86.64	94.82
2800	75.86	90.12	87.36	95.60
3000	76.68	90.66	88.07	96.38

Table 6. Accuracy for Plant Disease Classification

In terms of accuracy performance, it can be seen that the proposed model is 19.4% more accurate than TF CNN [9], 4.9% more accurate than MTL AF [10], and 7.4% more accurate than ELM KF [12]. Similarly, the precision performance for the same application can be seen in table 7,

Test Images	**P (%)** **TF CNN [9]**	**P (%)** **MTL AF [10]**	**P (%)** **ELM KF [12]**	**P (%)** **BMHM CRAS**
200	62.52	79.38	74.69	81.73
400	63.05	79.71	75.14	82.23
600	64.00	80.48	76.04	83.21
800	64.95	81.19	76.92	84.17
1000	65.62	81.43	77.39	84.69
1200	66.39	81.88	78.04	85.40
1400	67.17	82.49	78.77	86.20
1600	67.95	83.00	79.45	86.94
1800	68.73	83.51	80.13	87.69
2000	69.51	84.03	80.81	88.44
2200	70.30	84.54	81.49	89.18
2400	71.08	85.06	82.18	89.93
2600	71.86	85.57	82.86	90.67
2800	72.64	86.09	83.54	91.42
3000	73.42	86.60	84.22	92.17

Table 7. Precision for Plant Disease Classification

In terms of precision performance, it can be observed that the proposed model is 18.5% superior to TF CNN [9], 3.6% superior to MTL AF [10], and 8.3% superior to ELM KF [12]. Similarly, recall performance for the same application can be observed in the following table 8,

Test Images	**R (%)** **TF CNN [9]**	**R (%)** **MTL AF [10]**	**R (%)** **ELM KF [12]**	**R (%)** **BMHM CRAS**
200	65.77	83.59	78.61	86.03
400	66.30	83.91	79.06	86.52
600	67.10	84.51	79.79	87.32
800	68.30	85.50	80.95	88.58
1000	68.96	85.77	81.44	89.12
1200	69.76	86.11	82.04	89.78
1400	70.59	86.80	82.84	90.65
1600	71.41	87.34	83.55	91.44
1800	72.23	87.88	84.27	92.22
2000	73.06	88.43	84.99	93.01
2200	73.88	88.97	85.71	93.79
2400	74.70	89.51	86.43	94.58
2600	75.53	90.05	87.15	95.37
2800	76.35	90.59	87.86	96.15
3000	77.17	91.14	88.58	96.94

Table 8. Recall for Plant Disease Classification

In terms of recall performance, it can be observed that the proposed model is 16.5% superior to TF CNN [9], 4.9% superior to MTL AF [10], and 6.5% superior to ELM KF [12]. Similarly, the delay performance for the same application can be observed in the following table 9,

Test Images	**D (ms)** **TF CNN [9]**	**D (ms)** **MTL AF [10]**	**D (ms)** **ELM KF [12]**	**D (ms)** **BMHM CRAS**
200	106.31	53.83	68.49	46.65
400	104.76	52.87	67.17	45.21
600	102.40	51.11	65.01	42.84
800	98.84	48.21	61.61	39.12
1000	96.93	47.40	60.17	37.54
1200	94.55	46.42	58.40	35.61
1400	92.12	44.37	56.05	33.04
1600	89.70	42.78	53.93	30.72
1800	87.27	41.18	51.82	28.41
2000	84.85	39.59	49.70	26.09
2200	82.43	37.99	47.59	23.77
2400	80.00	36.39	45.47	21.46
2600	77.58	34.80	43.35	19.14
2800	75.15	33.20	41.24	16.83
3000	72.73	31.60	39.12	14.51

Table 9. Delay for Plant Disease Classification

In terms of delay performance, it can be seen that the proposed model is 41.8% better than TF CNN [9], 26.5% better than MTL AF [10], and 25.0% better than ELM KF [12], making it suitable for high-efficiency and high-speed plant disease classification applications. Table 10 displays the accuracy of credit card fraud detection for various datasets when compared to TTS ABF [13], NFAG [15], and SFS OM [18] as follows,

Test Sets	**A (%)** **TTS ABF [13]**	**A (%)** **NFAG [15]**	**A (%)** **SFS OM [18]**	**A (%)** **BMHM CRAS**
1000	90.20	85.60	96.50	97.25
2000	90.30	85.70	96.70	97.39
3000	90.50	85.90	96.90	97.61
4000	90.70	86.03	97.10	97.80
5000	90.80	86.18	97.30	97.96
6000	90.98	86.33	97.50	98.15
7000	91.14	86.48	97.70	98.33
8000	91.30	86.63	97.90	98.51
9000	91.46	86.78	98.10	98.69
10000	91.62	86.93	98.30	98.88
11000	91.78	87.08	98.50	99.06
12000	91.94	87.23	98.70	99.24
13000	92.10	87.38	98.90	99.42
14000	92.26	87.53	99.10	99.60
15000	92.42	87.68	99.30	99.79

Table 10. Accuracy for Credit card fraud detection

In terms of accuracy performance, it can be observed that the proposed model is 6.5% more accurate than TTS ABF [13], 10.6% more accurate than NFAG [15], and 0.5% more accurate than SFS OM [18]. Similarly, the precision performance for the same application can be observed in the following table 11,

Test Sets	**P (%)** **TTS ABF [13]**	**P (%)** **NFAG [15]**	**P (%)** **SFS OM [18]**	**P (%)** **BMHM CRAS**
1000	85.95	81.57	92.00	92.69
2000	86.10	81.71	92.19	92.86
3000	86.29	81.87	92.38	93.05
4000	86.43	82.01	92.57	93.22
5000	86.56	82.15	92.76	93.38
6000	86.72	82.29	92.95	93.56
7000	86.88	82.44	93.14	93.73
8000	87.03	82.58	93.33	93.91
9000	87.18	82.72	93.52	94.08
10000	87.33	82.87	93.71	94.25
11000	87.49	83.01	93.90	94.43
12000	87.64	83.15	94.10	94.60

13000	87.79	83.29	94.29	94.77
14000	87.94	83.44	94.48	94.95
15000	88.10	83.58	94.67	95.12

Table 11. Precision for Credit card fraud detection

On the basis of this precision performance, it can be observed that the proposed model outperforms TTS ABF [13], NFAG [15], and SFS OM [18] by 5.9%, 8.5%, and 0.4%, respectively. Similarly, recall performance for the same application can be observed in the following table 12,

Test Sets	**R (%)** **TTS ABF [13]**	**R (%)** **NFAG [15]**	**R (%)** **SFS OM [18]**	**R (%)** **BMHM CRAS**
1000	90.57	85.95	96.92	97.40
2000	90.69	86.07	97.12	97.56
3000	90.89	86.26	97.32	97.77
4000	91.07	86.40	97.52	97.96
5000	91.19	86.55	97.72	98.12
6000	91.36	86.70	97.92	98.31
7000	91.53	86.85	98.12	98.49
8000	91.69	87.00	98.32	98.68
9000	91.85	87.15	98.52	98.86
10000	92.01	87.30	98.72	99.04
11000	92.17	87.45	98.92	99.22
12000	92.33	87.60	99.12	99.41
13000	92.49	87.75	99.32	99.59
14000	92.65	87.90	99.53	99.77
15000	92.81	88.05	99.73	99.95

Table 12. Recall for Credit card fraud detection

In terms of recall performance, it can be observed that the proposed model is 6.8% superior to TTS ABF [13], 10.6% superior to NFAG [15], and 0.2% superior to SFS OM [18]. Similarly, the delay performance of the same application can be observed in the following table 13,

Test Sets	**D (ms)** **TTS ABF [13]**	**D (ms)** **NFAG [15]**	**D (ms)** **SFS OM [18]**	**D (ms)** **BMHM CRAS**
1000	33.28	46.88	14.58	12.66
2000	32.91	46.51	13.99	12.19
3000	32.32	45.97	13.40	11.57
4000	31.80	45.56	12.81	11.03
5000	31.45	45.12	12.22	10.53
6000	30.93	44.68	11.63	9.98
7000	30.46	44.23	11.04	9.44
8000	29.99	43.79	10.45	8.90
9000	29.51	43.35	9.85	8.37
10000	29.04	42.90	9.26	7.83
11000	28.57	42.46	8.67	7.29
12000	28.09	42.01	8.08	6.75
13000	27.62	41.57	7.49	6.21
14000	27.15	41.13	6.90	5.68
15000	26.67	40.68	6.31	5.14

Table 13. Delay for Credit card fraud detection

On the basis of this performance, it can be seen that the proposed model is 25.3% better than TTS ABF [13], 46.5% better than NFAG [15], and 6.1% better than SFS OM [18] in terms of delay performance, making it suitable for high-efficiency and high-speed credit card fraud detection applications. Similarly, the accuracy of Stock Market Prediction for various datasets when compared to DNN [19], GA LSTM [21], and ST DNN [24] can be observed in Table 14 as follows,

Test Sets	**A (%)** **DNN [19]**	**A (%)** **GA LSTM [21]**	**A (%)** **ST DNN [24]**	**A (%)** **BMHM CRAS**
400	65.60	71.50	61.80	79.56
800	65.80	71.90	61.90	79.84
1200	66.50	72.30	62.30	80.44
1600	66.87	72.50	62.50	80.75
2000	67.32	72.90	62.80	81.21
2400	67.77	73.24	63.04	81.62
2600	68.22	73.58	63.30	82.04
3000	68.67	73.92	63.56	82.46
3400	69.12	74.26	63.82	82.88

3800	69.57	74.60	64.08	83.30
4200	70.02	74.94	64.34	83.72
4600	70.47	75.28	64.60	84.14
5000	70.92	75.62	64.86	84.56
5400	71.37	75.96	65.12	84.98
6000	71.82	76.30	65.38	85.40

Table 14. Accuracy for Stock Market Prediction

In terms of accuracy performance, it can be seen that the proposed model is 14.1% more accurate than DNN [19], 8.3% more accurate than GA LSTM [21], and 19.5% more accurate than ST DNN [24]. Similarly, the precision performance for the same application can be observed in the following table 15,

Test Sets	**P (%)** **DNN [19]**	**P (%)** **GA LSTM [21]**	**P (%)** **ST DNN [24]**	**P (%)** **BMHM CRAS**
400	62.57	68.29	58.90	75.90
800	63.00	68.67	59.14	76.32
1200	63.51	68.95	59.43	76.76
1600	63.90	69.24	59.67	77.12
2000	64.33	69.59	59.92	77.54
2400	64.75	69.91	60.16	77.93
2600	65.18	70.24	60.41	78.33
3000	65.61	70.56	60.66	78.73
3400	66.04	70.89	60.90	79.13
3800	66.47	71.21	61.15	79.53
4200	66.90	71.53	61.40	79.93
4600	67.33	71.86	61.65	80.33
5000	67.75	72.18	61.90	80.73
5400	68.18	72.50	62.14	81.13
6000	68.61	72.83	62.39	81.53

Table 15. Precision for Stock Market Prediction

In terms of precision performance, it can be observed that the proposed model is 12.8% superior to DNN [19], 6.5% superior to GA LSTM [21], and 18.3% superior to ST DNN [24]. Similarly, the recall performance for the same application can be observed in the following table 16,

Test Sets	**R (%)** **DNN [19]**	**R (%)** **GA LSTM [21]**	**R (%)** **ST DNN [24]**	**R (%)** **BMHM CRAS**
400	65.90	71.87	62.06	79.73
800	66.22	72.27	62.23	80.08
1200	66.84	72.62	62.59	80.61
1600	67.23	72.87	62.81	80.96
2000	67.68	73.26	63.10	81.41
2400	68.13	73.60	63.34	81.82
2600	68.59	73.94	63.60	82.24
3000	69.04	74.28	63.86	82.66
3400	69.49	74.63	64.13	83.08
3800	69.94	74.97	64.39	83.50
4200	70.39	75.31	64.65	83.92
4600	70.84	75.65	64.91	84.34
5000	71.30	75.99	65.17	84.76
5400	71.75	76.33	65.43	85.18
6000	72.20	76.67	65.69	85.61

Table 16. Stock Market Prediction

In terms of recall performance, it can be observed that the proposed model is 12.3% superior to DNN [19], 9.4% superior to GA LSTM [21], and 19.1% superior to ST DNN [24]. Similarly, the delay performance of the same application can be observed in the following table 17,

Test Sets	**D (ms)** **DNN [19]**	**D (ms)** **GA LSTM [21]**	**D (ms)** **ST DNN [24]**	**D (ms)** **BMHM CRAS**
400	105.93	88.35	117.24	64.81
800	104.98	87.16	116.72	63.75
1200	103.15	86.12	115.69	62.19
1600	102.01	85.39	115.02	61.18
2000	100.68	84.25	114.18	59.85
2400	99.35	83.24	113.46	58.63
2600	98.02	82.24	112.69	57.39
3000	96.68	81.23	111.92	56.15

3400	95.35	80.23	111.15	54.91
3800	94.02	79.22	110.38	53.67
4200	92.69	78.22	109.61	52.43
4600	91.36	77.21	108.84	51.19
5000	90.03	76.21	108.07	49.94
5400	88.70	75.20	107.31	48.70
6000	87.37	75.90	106.54	47.46

Table 17. Delay for Stock Market Prediction

In terms of delay performance, it can be seen that the proposed model is 46.2% better than DNN [19], 39.4% better than GA LSTM [21], and 59.3% better than ST DNN [24], making it suitable for high-efficiency and high-speed stock market prediction applications. Due to its consistently high performance across a variety of applications, the proposed model exhibits high scalability and can be applied to a variety of scenarios for accuracy, precision, recall, and delay metric efficiency optimization. This makes the model highly applicable to a variety of deployment scenarios for real-time applications.

5. **Conclusion and Future Scope**

The proposed BMHMCRAS model initially compresses data by optimizing compression model parameters. This optimization was accomplished by utilizing PSO-based internal parameter selection for Huffman, Run-Length Encoding (RLE), Wavelet, ZLib, BZ2, and LZMA methods. The compressed data was processed using the MGA Model, which aided in the performance-aware extraction and selection of features for various application types. The proposed model was fine-tuned by employing a correlation-based layer, which facilitated continuous training-set updates. This model was tested and validated against 12 different cutting-edge techniques and 4 different applications. It was observed that the proposed model is 16.1% more accurate than GA SVM [2], 9.3% more accurate than CNN SVM [4], and 8.5% more accurate than MLBF [5] in terms of ECG classification accuracy; and 31.5% more accurate than GA SVM [2], 26.8% more accurate than CNN SVM [4], and 25.4% more accurate than MLBF [5] in terms of delay performance due to the use of compression models, making it useful for high efficiency, and high-speed ECG classification Similarly, it was observed that the proposed model was 19.4% better than TF CNN [9], 4.9% better than MTL AF [10], and 7.4% better than ELM KF [12], in terms of accuracy performance, and 41.8% better than TF CNN [9], 26.5% better than MTL AF [10], and 25.1% better than ELM KF [12], in terms of delay performance, making it suitable for high-efficiency and high-speed plant disease classification applications. It was also observed that the proposed model was 6.5% better than TTS ABF [13], 10.6% better than NFAG [15], and 0.5% better than SFS OM [18], in terms of accuracy performance, and that it is 25.3% better than TTS ABF [13], 46.5% better than NFAG [15], and 6.1% better than SFS OM [18], in terms of delay performance, making it useful for high-efficiency and high-speed credit card fraud detection applications. Similarly, it was found that the proposed model is 14.1% better than DNN [19], 8.3% better than GA LSTM [21], and 19.5% better than ST DNN [24], in terms of accuracy performance, and 46.2% better than DNN [19], 39.4% better than GA LSTM [21], and 59.3% better than ST DNN [24], in terms of delay performance, making it suitable for high-efficiency and high-speed stock market prediction applications. Indicating that the model is highly scalable and accurate across a variety of application types. In the future, researchers will be able to add a larger number of compression and feature extraction models for optimization via PSO and MGA, which will contribute to a higher density of feature extraction and a reduction in application deployment errors. In addition, researchers can validate the model's performance in terms of applicability and scalability under a variety of test conditions by evaluating its applicability and scalability under various application types.

6. References

[1] Y. Li, Z. Zhang, F. Zhou, Y. Xing, J. Li and C. Liu, "Multi-Label Classification of Arrhythmia for Long-Term Electrocardiogram Signals With Feature Learning," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021, Art no. 2512611, doi: 10.1109/TIM.2021.3077667.

[2] J. Yang and R. Yan, "A Multidimensional Feature Extraction and Selection Method for ECG Arrhythmias Classification," in IEEE Sensors Journal, vol. 21, no. 13, pp. 14180-14190, 1 July1, 2021, doi: 10.1109/JSEN.2020.3047962.

[3] X. Tang and W. Tang, "An ECG Delineation and Arrhythmia Classification System Using Slope Variation Measurement by Ternary Second-Order Delta Modulators for Wearable ECG Sensors," in IEEE Transactions on Biomedical

Circuits and Systems, vol. 15, no. 5, pp. 1053-1065, Oct. 2021, doi: 10.1109/TBCAS.2021.3113665.

[4] Z. Ahmad, A. Tabassum, L. Guan and N. M. Khan, "ECG Heartbeat Classification Using Multimodal Fusion," in IEEE Access, vol. 9, pp. 100615-100626, 2021, doi: 10.1109/ACCESS.2021.3097614.

[5] J. Zhang et al., "MLBF-Net: A Multi-Lead-Branch Fusion Network for Multi-Class Arrhythmia Classification Using 12-Lead ECG," in IEEE Journal of Translational Engineering in Health and Medicine, vol. 9, pp. 1-11, 2021, Art no. 1900211, doi: 10.1109/JTEHM.2021.3064675.

[6] C. Liu, J. Yu, Y. Huang and F. Huang, "Multi-Feature Sparse Representations Learning via Collective Matrix Factorization for ECG Biometric Recognition," in IEEE Access, vol. 9, pp. 163233-163241, 2021, doi: 10.1109/ACCESS.2021.3133482.

[7] L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[8] X. Liu, W. Min, S. Mei, L. Wang and S. Jiang, "Plant Disease Recognition: A Large-Scale Benchmark Dataset and a Visual Region and Loss Reweighting Approach," in IEEE Transactions on Image Processing, vol. 30, pp. 2003-2015, 2021, doi: 10.1109/TIP.2021.3049334.

[9] S. Barburiceanu, S. Meza, B. Orza, R. Malutan and R. Terebes, "Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture," in IEEE Access, vol. 9, pp. 160085-160103, 2021, doi: 10.1109/ACCESS.2021.3131002.

[10] R. Dwivedi, S. Dey, C. Chakraborty and S. Tiwari, "Grape Disease Detection Network Based on Multi-Task Learning and Attention Features," in IEEE Sensors Journal, vol. 21, no. 16, pp. 17573-17580, 15 Aug.15, 2021, doi: 10.1109/JSEN.2021.3064060.

[11] S. M. Hassan and A. K. Maji, "Plant Disease Identification Using a Novel Convolutional Neural Network," in IEEE Access, vol. 10, pp. 5390-5401, 2022, doi: 10.1109/ACCESS.2022.3141371.

[12] R. Dwivedi, T. Dutta and Y. -C. Hu, "A Leaf Disease Detection Mechanism Based on L1-Norm Minimization Extreme Learning Machine," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 8019905, doi: 10.1109/LGRS.2021.3110287.

[13] F. Ogme, A. G. Yavuz, M. A. Guvensan and M. E. Karsligil, "Temporal Transaction Scraping Assisted Point of Compromise Detection With Autoencoder Based Feature Engineering," in IEEE Access, vol. 9, pp. 109536-109547, 2021, doi: 10.1109/ACCESS.2021.3101738.

[14] Ghosh Dastidar, K., Jurgovsky, J., Siblini, W. et al. NAG: neural feature aggregation framework for credit card fraud detection. Knowl Inf Syst 64, 831–858 (2022). https://doi.org/10.1007/s10115-022-01653-0.

[15] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," in IEEE Access, vol. 10, pp. 16400-16407, 2022, doi: 10.1109/ACCESS.2022.3148298.

[16] B. Lebichot, T. Verhelst, Y. -A. Le Borgne, L. He-Guelton, F. Oblé and G. Bontempi, "Transfer Learning Strategies for Credit Card Fraud Detection," in IEEE Access, vol. 9, pp. 114754-114766, 2021, doi: 10.1109/ACCESS.2021.3104472.

[17] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu and Y. Gao, "Internet Financial Fraud Detection Based on a Distributed Big Data Approach With Node2vec," in IEEE Access, vol. 9, pp. 43378-43386, 2021, doi: 10.1109/ACCESS.2021.3062467.

[18] E. B. Fatima, B. Omar, E. M. Abdelmajid, F. Rustam, A. Mehmood and G. S. Choi, "Minimizing the Overlapping Degree to Improve Class-Imbalanced Learning Under Sparse Feature Selection: Application to Fraud Detection," in IEEE Access, vol. 9, pp. 28101-28110, 2021, doi: 10.1109/ACCESS.2021.3056285.

[19] N. Naik and B. R. Mohan, "Novel Stock Crisis Prediction Technique—A Study on Indian Stock Market," in IEEE Access, vol. 9, pp. 86230-86242, 2021, doi: 10.1109/ACCESS.2021.3088999.

[20] S. S. Alotaibi, "Ensemble Technique With Optimal Feature Selection for Saudi Stock Market Prediction: A Novel Hybrid Red Deer-Grey Algorithm," in IEEE Access, vol. 9, pp.

64929-64944, 2021, doi: 10.1109/ACCESS.2021.3073507.

[21] Y. Lin, S. Liu, H. Yang and H. Wu, "Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme," in IEEE Access, vol. 9, pp. 101433-101446, 2021, doi: 10.1109/ACCESS.2021.3096825.

[22] S. Chen and C. Zhou, "Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network," in IEEE Access, vol. 9, pp. 9066-9072, 2021, doi: 10.1109/ACCESS.2020.3047109.

[23] Q. Li, J. Tan, J. Wang and H. Chen, "A Multimodal Event-Driven LSTM Model for Stock Prediction Using Online News," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 10, pp. 3323-3337, 1 Oct. 2021, doi: 10.1109/TKDE.2020.2968894.

[24] X. Hou, K. Wang, C. Zhong and Z. Wei, "ST-Trader: A Spatial-Temporal Deep Neural Network for Modeling Stock Market Movement," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 5, pp. 1015-1024, May 2021, doi: 10.1109/JAS.2021.1003976.

[25] Mr. Ashish Uplenchwar. (2017). Modern Speech Identification Model using Acoustic Neural approach . International Journal of New Practices in Management and Engineering, 6(03), 01 - 06. https://doi.org/10.17762/ijnpme.v6i03.58

[26] Botha, D., Dimitrov, D., Popović, N., Pereira, P., & López, M. Deep Reinforcement Learning for Autonomous Robot Navigation. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/140

[27] Dhabliya, D., Soundararajan, R., Selvarasu, P., Balasubramaniam, M. S., Rajawat, A. S., Goyal,S. B., . . . Suciu, G. (2022). Energy-efficient network protocols and resilient datatransmission schemes for wireless sensor Networks—An experimental survey. Energies, 15(23)doi:10.3390/en15238883