ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20250321002502

Original scientific paper

Intelligent Deep Learning Based Fault Classification Using Dual-Stream Transformer-CNN with Self-Supervised Feature Refinement for Industrial Applications

K. RANJANI*, T. BARANIDHARAN, J. RAMAL

Abstract: Fault classification plays a crucial role in industrial engineering, particularly in manufacturing and power generation, where accurate fault detection is essential to prevent system failures, reduce maintenance costs, and enhance operational safety. With the advancement of Industry 4.0 and 5.0, intelligent fault classification techniques leveraging real-time data processing have become increasingly important. This study proposes a deep learning-based fault classification model integrating Dual-Stream Transformer-CNN with Self-Supervised Feature Refinement (DSTC-SSFR) to improve classification accuracy and robustness. The core architecture consists of two parallel processing streams designed to effectively extract both spatial and temporal features from multivariate industrial sensor signals. The spatial stream uses multi-scale 1D Convolutional Neural Networks (1D CNNs) with varying kernel sizes to capture localized fault-related features at different frequency scales. The Bobcat is used for hyperparameter tuning, further enhancing model performance. The proposed approach achieves an overall accuracy of 95.07%, with a precision of 95.10%, recall of 95.07%, and F1-score of 95.07%. Additionally, the model attains a logarithmic loss of 0.1070, a Matthews correlation coefficient (MCC) of 0.9343, and an area under the ROC curve (AUC) of 99.54%. These results demonstrate the model's effectiveness in fault classification, offering a robust and efficient solution for industrial applications in smart manufacturing environments.

Keywords: deep learning; fault classification; industry 4.0; sensor signals; transformer-CNN

1 INTRODUCTION

Fault classification plays a crucial role in maintaining the operational efficiency and reliability of industrial systems [1]. With the advent of Industry 4.0 and Industry 5.0, industrial environments have become increasingly interconnected, integrating various sensors and communication devices for real-time monitoring and decision-making [2, 3]. The data collected from these sensors facilitate fault detection, enabling predictive maintenance and minimizing unexpected failures. Intelligent systems, empowered by the Internet of Things (IoT), generate large-scale data for machine health monitoring, which is further processed using feature learning techniques for effective fault classification.

Accurate and timely fault classification is essential to prevent industrial failures, reduce downtime, and enhance safety. Automated fault classification systems are particularly critical in high-risk industries such as power grids, medical devices, and heavy machinery, where failures can lead to severe hazards [4]. These systems not only improve safety but also optimize resource allocation, ensuring efficient operations. By integrating predictive analytics, fault classification models can anticipate potential failures, facilitating proactive maintenance and reducing operational disruptions.

Recent advancements in deep learning (DL) have significantly improved fault classification by leveraging large datasets for predictive maintenance [5, 6]. DL techniques can identify potential issues early, enabling data-driven maintenance strategies that minimize downtime and extend the lifespan of industrial machinery. Moreover, robust fault classification models help prevent production halts and financial losses caused by unexpected failures.

Traditional DL methods such as Convolutional Neural Networks (CNNs) have been widely adopted for fault classification due to their ability to extract hierarchical spatial features. However, CNNs struggle to capture complex spatial relationships comprehensively. Similarly, Long Short-Term Memory (LSTM) networks, commonly used for sequential data processing, are not always optimal for modeling spatial dependencies [7, 8]. To address these limitations, hybrid models combining different deep learning architectures have been explored.

In this work, a new DL model called Dual-Stream Transformer-CNN with Self-Supervised Feature Refinement (DSTC-SSFR) is propsoed to improve diagnostic accuracy in industrial systems. The architecture combines multi-scale 1D Convolutional Neural Networks (CNNs) and Transformer encoders to simultaneously capture localized spatial patterns and long-range temporal dependencies in sensor signals. The spatial stream uses CNNs with varied kernel sizes to extract frequency-aware features. The temporal stream uses Transformer blocks with positional encoding for robust sequence modeling. These two feature streams are fused via an attention-based fusion mechanism that dynamically prioritizes the most informative components of the signal. To further enhance performance, a self-supervised learning module is integrated to learn meaningful representations through masked signal reconstruction and contrastive loss.

The remainder of this paper is structured as follows: Section 2 presents related work, Section 3 details the proposed fault classification model, Section 4 discusses performance evaluation, and Section 5 concludes the study.

2 RELATED WORK

Several fault classification models have been proposed for industrial applications, utilizing various machine learning and deep learning approaches. One study introduced a model for bearing fault classification, addressing the challenge of unknown bearing signal calculations. The approach initially processes data using autoregressive external input modeling for classification [11]. A hybrid fault classification model for induction machines has been developed, incorporating a decision tree with an advanced feature extraction technique, achieving an accuracy of 93.2% [12]. Another study proposed a fault classification model based on a probabilistic approach for industrial applications. The model utilizes Bayesian filtering for temporal feature learning, with residual data classified using a Support Vector Machine (SVM) model [13].

A hybrid model integrating one-dimensional Convolutional Neural Networks (1D-CNN) has demonstrated improved accuracy in industrial fault diagnosis. The results show that the hybrid model outperforms single-model approaches in classifying industrial faults [14]. Additionally, a fault classification model leveraging an IoT-enabled sensor system was introduced, where statistical time-domain features were extracted from sensor signals, and the final classification was performed using an AdaBoost classifier. The model was validated using various k-fold cross-validation techniques [15].

A rolling bearing fault classification model based on SVM was proposed, where sensor signals were converted into time-domain features. These extracted features were used to train the SVM model, achieving an average accuracy of 92.4% [16]. Another study introduced a fault detection model for industrial machines using mathematical morphology (MM). Voltage and current sensor data were pre-processed using MM, followed by classification using a random forest algorithm [17].

A prediction model for gas discharge detection in gasoline engines was developed, combining machine learning algorithms with fuzzy logic. The model outperformed existing methods by an accuracy margin of 3.2% to 5.5% [18]. Furthermore, a modified One-Against-Rest (OAR) algorithm was proposed for fault classification, where a decision tree was integrated to enhance the accuracy and detection speed of conventional OAR methods [19].

A reservoir-based deep learning model was introduced for bearing fault detection, utilizing both time and frequency features for training. The model employs a large, sparsely connected reservoir with random weights, allowing it to process nonlinear patterns effectively [20]. Dimensionality reduction techniques are widely used to optimize fault classification performance. The t-distributed Stochastic Neighbor Embedding (t-SNE) technique has been applied for feature extraction, achieving accuracy improvements of 5.6% to 9.2% [21]. Similarly, a Principal Component Analysis (PCA)-based feature extraction method has been explored, reducing data dimensionality and identifying key principal components, with classification performed using the K-Nearest Neighbor (KNN) algorithm [22].

Empirical Mode Decomposition (EMD) has been utilized for feature extraction, addressing mode-mixing issues. A hybrid SVM and AdaBoost model was employed for classification, demonstrating enhanced performance [23]. In another study, a machine learning model integrated with data augmentation techniques was analyzed. The augmentation process increased the diversity of training data and mitigated data imbalance issues, leading to improved classification performance [24].

Optimized feature selection techniques have also been explored to enhance model efficiency while reducing computational complexity. One approach applied the Cuckoo Search Algorithm (CSA) for feature selection, leading to an accuracy improvement of approximately 12.4% [25]. A three-fold hybrid model for fault analysis was introduced, where machine signals were initially pre-processed using a short-time Fourier transform. A 1D-CNN model was applied for fault analysis, and an attention mechanism was incorporated to enhance feature extraction capability [26]. Similarly, the performance of different machine learning models was investigated using a Binary Grey Wolf Optimization-based feature selection approach. The optimized models achieved high-accuracy fault detection, with a performance of 93.8%, outperforming conventional techniques [27].

The literature highlights the effectiveness of deep learning and optimization-based models in fault classification, with several approaches demonstrating superior accuracy and robustness. However, existing models still face challenges in capturing spatial and temporal dependencies effectively. To address these limitations, this work proposes a hybrid model integrating Capsule Networks and Gated Recurrent Units (GRUs) for improved fault classification accuracy.

3 PROPOSED MODEL

In the proposed work, Dual-Stream Transformer-CNN with Self-Supervised Feature Refinement (DSTC-SSFR) is proposed for fault classification in industrial system. This system is designed to handle multi-class classification tasks. To further enhance model performance, the BOA is applied for hyperparameter tuning. The architecture of proposed model is shown in Fig. 1.

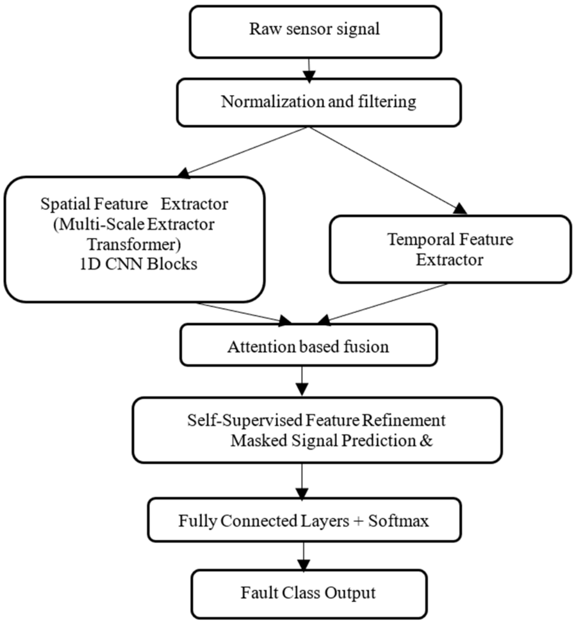

Figure 1 Proposed capsule - GRU model

The system architecture is designed to separately capture spatial and temporal patterns from multivariate sensor signals. Given an input signal sequence $X \in R^{T \times D}$ where T denotes time steps and D is the number of sensor channels, the signal is simultaneously processed through two parallel branches: a spatial stream using multi-scale 1D convolutional neural networks and a temporal stream using Transformer encoders.

In the spatial stream, the input signal X is convolved using multiple 1D kernels with different receptive fields to extract local features:

$$F_s = \text{Concat}\left(\text{Conv1}D_{K=3}\left(X\right),\right.$$
$$\left.\text{Conv1}D_{K=5}\left(X\right), \text{Conv1}D_{K=7}\left(X\right)\right) \tag{1}$$

where Conv1D denotes a 1D convolution with kernel size k, and $F_s \in R^{T\times \mathrm{d}s}$ is the spatial feature representation.

Concurrently, the temporal stream applies a Transformer encoder to capture long-range dependencies in the signal. Positional encoding PE is added to retain temporal order:

$$X_T = X + PE \tag{2}$$

$$F_t = \text{TransformerEncoder}\left(X_T\right) \tag{3}$$

where $F_t \in R^{T\times \mathrm{d}t}$, F is the temporal feature output.

To combine spatial and temporal features, an attention-based fusion mechanism is introduced:

$$\alpha = \text{Softmax}\left(W_f\left[F_s\middle\|F_t + b_f\right)\right. \tag{4}$$

$$F = \alpha\cdot\left(F_s + F_t\right) \tag{5}$$

where W_f and b_f are trainable parameters, and ‖\|‖ denotes feature concatenation. This fusion learns to weight the spatial and temporal components adaptively.

A key innovation is the self-supervised feature refinement module. During pretraining, a masked signal modeling task is applied where random time steps in X are masked, and the network is trained to reconstruct them:

$$L_{\text{recon}} = \frac{1}{\left|M\right|}\sum\nolimits_{t\in M}\left\|\hat{X}_t - X_t\right\|^2 \tag{6}$$

where M denotes the masked indices. Additionally, contrastive learning is applied on augmented views of the signal:

$$L_{\text{count}} == \log\frac{\exp\left(\text{sim}\left(z_i, z_j/\tau\right)\right)}{\sum\nolimits_{k=1}^{2N}\exp\left(\text{sim}\left(z_i, z_k/\tau\right)\right)} \tag{7}$$

where z_i and z_j are feature embeddings from different views of the same signal, sim(·) is cosine similarity, and τ is a temperature parameter.

Finally, the refined features F are passed through fully connected layers and a softmax classifier for fault label prediction:

$$\hat{y} = \text{Softmax}\left(W_o F + b_o\right) \tag{8}$$

$$L_{\text{class}} = -\sum_{c=1}^{c} y_c \log\left(\widehat{y_c}\right) \tag{9}$$

where C is the number of fault classes.

The total training loss is a weighted sum of all components:

$$L_{\text{total}} = \lambda_1 L_{\text{class}} + \lambda_2 L_{\text{recon}} + \lambda_3 L_{\text{cont}} \tag{10}$$

where λ_1, λ_2, λ_3 are hyperparameters.

This novel DSTC-SSFR model offers improved classification accuracy and robustness, especially in real-world industrial environments with noisy, multivariate, and imbalanced sensor data.

3.1. BOA Technique

The BobcatOptimizationAlgorithm (BOA) is a novel bio-inspired metaheuristic approach. It mimics the hunting behavior of bobcats in the wild. The algorithm is motivated by the No Free Lunch (NFL) theorem which emphasizes the need for designing new optimizers tailored to specific problem landscapes. BOA operates in two main phases: Exploration Phase and Exploitation Phase. In Exploration Phase, the models mimic the tracking movement of bobcats toward prey. In Exploitation Phase , the models imitate the chase process to catch the prey.

Initialization Phase

The BOA begins with the random initialization of a population of bobcats, each representing a potential solution in the problem-solving space. The position of each bobcat is encoded as a vector within the boundaries defined by the problem domain. It can mathematically be expressed as follows:

$$x_{i,\,d} = lb_d + r\cdot\left(ub_d - lb_d\right),\; i = 1, 2, ..., m \tag{11}$$

where $x_{i,\,d}$ denotes the position of the i-th bobcat in the dth dimension, lb_d and ub_d are the lower and upper bounds of the dimension, and $r \in [0, 1]$ is a uniformly distributed random number. The complete population matrix $X \in R^{N\times m}$ is formed, where N is the number of bobcats and m is the number of dimensions. The fitness of each bobcat is then calculated using the objective function. It can be expressed as follows :

$$F = \left[F\left(X_1\right), F\left(X_2\right), ..., F\left(X_N\right)\right]^T \tag{12}$$

which evaluates the quality of each solution.

Phase 1: Exploration - Tracking and Moving Toward Prey

The first phase of BOA is focused on exploration, which allows bobcats to investigate various regions of the search space in pursuit of better solutions. This behavior mimics the initial tracking movement of a bobcat toward its prey. In this phase, each bobcat identifies a candidate prey set CP_i which includes solutions with better fitness than itself:

$$CP_i = \left\{X_k \mid F\left(X_k\right) < F\left(X_i\right)\right. \tag{13}$$

From this set, one prey SP_i is selected randomly. The position of the bobcat is then updated to move closer toward this prey using the following rule:

$$x_{i,j}^{P1} = x_{SP_i,j} + r\cdot I_{i,j}\left(x_{SP_i,j} - x_{i,j}\right) \tag{14}$$

where $r_{i,j} \in [0, 1]$ is a random number and $I_{i,j} \in \{1, 2\}$ introduces variability in movement. This model simulates diverse directional shifts, thereby increasing the global search capability of BOA. The updated position is retained only if it improves the objective function:

$$\text{If } F\left(x_i^{P1}\right) < F\left(x_i\right), \text{ then } x_i = x_i^{P1} \tag{15}$$

This strategy increases the algorithm's capability to escape local optima and enhances population diversity.

Phase 2: Exploitation - Chasing and Catching Prey

Once promising regions are located, BOA enters the exploitation phase. It is performed to fine-tune the solutions by mimicking the bobcat's chasing and catching behavior near the prey. This phase involves small adjustments to the bobcat's position. The position of the i-th bobcat in dimension j is updated as:

$$x_{i,j}^{P2} = x_{i,j} + x_{i,j}\left(\frac{1-2r_{i,j}}{1+t}\right) \tag{16}$$

where $r_{i,j} \in [0, 1]$ is a random number and t is the current iteration count. The term $\frac{1-2r_{i,j}}{1+t}$ ensures that the magnitude of positional change reduces as the number of iterations increases. It is used for precise convergence. The update is accepted only if it improves the objective function:

$$\text{If } F\left(x_i^{P2}\right) < F\left(x_i\right), \text{ then } x_i = x_i^{P2} \tag{17}$$

This localized search increases convergence near optimal solutions and complements the broad search from the exploration phase.

Parameter tuning

The performance of DSTC-SSFR is highly dependent on the appropriate selection of hyperparameters . The parameters include learning rate, filter sizes, number of attention heads, and transformer depth. The manual tuning of these hyperparameters is time-consuming and often suboptimal. To solve this challenge, the proposed system integrates the BOA as an efficient parameter tuning strategy. Each bobcat in the population represents a candidate configuration of hyperparameters. The fitness function is defined as the classification accuracy of the DSTC-SSFR model on a validation set. During each iteration, BOA dynamically updates the positions of bobcats based on the exploration and exploitation mechanisms. This automated tuning ensures that the DSTC-SSFR model reaches superior generalization performance without exhaustive manual trials.

BOA-Based Hyperparameter Tuning - Pseudocode

```
1. Initialize parameters:
  - $N$: number of bobcats (population size)
  - $\text{Max}_{\text{iter}}$: maximum number of iterations
  - $lb$, $ub$: lower and upper bounds of hyperparameters (e.g., [0.0001, 0.1] for learning rate)
  - $X$: initialize positions of $N$ bobcats randomly using $x_{i,d} = lb_d + r \times (ub_d - lb_d)$
  - Evaluate fitness $F_i = 1 - \text{Accuracy}(\text{DSTC-SSFR}(X_i))$ for each bobcat
2. For $t = 1$ to $\text{Max}_{\text{iter}}$:
    For each bobcat $i$:
        # Phase 1: Exploration
        a. Identify candidate prey $CP_i = \{\_k \mid F(X_k) < F(X_i)\}$
        b. Randomly select a prey $SP_i$ from $CP_i$
        c. Update position:
            $x_{i,j}^{P1} = xSP_{i,j} + r \cdot I_{i,j} \cdot \left(x_{SP_{i,j}} - x_{i,j}\right)$
        d. If $F\left(x_i^{P1}\right) < F\left(x_i\right)$: $x_i \leftarrow x_i^{P1}$
        # Phase 2: Exploitation
        e. Update position:
            $x_{i,j}^{P2} = x_{i,j} + x_{i,j} \cdot \left(\left(1 - 2 \cdot r_{i,j}\right) / \left(1+\text{t}\right)\right)$
        f. If $F\left(x_i^{P2}\right) < F\left(x_i\right)$: $x_i \leftarrow x_i^{P2}$
    g. Save the best solution (lowest fitness)
3. Return the best hyperparameter configuration found (i.e., $X_{\text{best}}$ with minimum $F$)
```

The process of hyperparameter tuning using the BOA begins with the initialization phase, where key algorithmic parameters are defined. This includes setting the number of bobcats N, the maximum number of iterations Max_{iter}, and the lower and upper bounds for each hyperparameter in the model. Each bobcat in the population is assigned a random position within these bounds. The fitness of each bobcat is then evaluated by training the model with the corresponding hyperparameters and calculating the validation performance.

The optimization proceeds iteratively over a set number of iterations. In each iteration t, the algorithm applies two main behavioral phases, exploration and exploitation, to refine the hyperparameters. This movement promotes global search and diversity. If the new position improves the fitness, it replaces the current one. Following this, the exploitation phase begins, focusing on fine-tuning around promising areas. At the end of each iteration, the best-performing bobcat is stored. Once the maximum number of iterations is reached, the algorithm returns the optimal hyperparameter configuration X_{best}, which represents the best solution found across all iterations. This approach ensures an effective balance between broad exploration and local exploitation.

4 RESULT AND DISCUSSION

The dataset is collected from the publically available link https://www02.smt.ufrj.br/~offshore/mfs/page_01.html. It consists of 1951 multivariate timeseries acquired by sensors on SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT). This equipment allows the simulation of various mechanical faults in rotating machinery and gives data crucial for fault diagnosis. The data collection is performed using three industrial IMI sensors measuring in the radial, axial, and tangential directions. The different learning packages of Pandas, NumPy, Scikit-learn and TensorFlow are used for analysis. The entire data set is divided into training and test data sets. Tab. 1 shows the trainable parameters of the model. Initially, the hyperparameters of the DSTC-SSFR model are set within specified ranges to ensure an effective search for optimal values. Initially, the learning rate was bounded between 0.0001 and 0.1, the number of Transformer heads ranged from 2 to 8, and the convolutional kernel sizes were set between 3 and 11. The population size was set to N = 20, and the maximum number of iterations was Max_{iter} = 50**.** By iteration 45, BOA identified an optimal configuration with a learning rate of 0.0074**,** 5 Transformer heads, and kernel size 9, achieving a validation accuracy of 94.8%. These values represent the best balance between model complexity and classification accuracy.

Table 1 Model parameters

Layer	# Params
Multi-scale Conv1D (Spatial Stream)	6144
Transformer Encoder (Temporal Stream)	89600
Attention-Based Fusion Layer	4096
Self-Supervised Feature Refinement	12288
Fully Connected (Classifier)	1024
Total	113152

$$\text{Accuracy} = \frac{TP+TN}{\left(TP+TN+FP+FN'\right)} \tag{18}$$

$$\text{Recall} = \frac{TP}{\left(TP+FN'\right)} \tag{19}$$

$$\text{Precision} = \frac{TP}{\left(TP+FP'\right)} \tag{20}$$

$$\text{F1 score} = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\left(\text{Precision} + \text{Recall}\right)} \tag{21}$$

where TP and TN indicate True Positives and Negatives; FP and FN denote False Positives and Negatives.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i\log\left(p_i\right)+\left(1-y_i\right)\log\left(1-p_i\right)\right] \tag{22}$$

where N is the Number of samples, y_i is the True label (0 or 1), p_i is the predicted probability of the positive class.

The Cohen's Kappa (k) is expressed in below

$$k = \frac{p_0 - p_e}{1 - p_e} \tag{23}$$

where, p_0 denotes Observed agreement, p_e indicates Expected agreement by chance.

The Matthews Correlation Coefficient (MCC) validates the binary classifications expressed in the following:

$$MCC = \frac{\left(TP\times TN\right)-\left(FP\times FN\right)}{\sqrt{\left(TP+FP\right)\left(TP+FN\right\}\left\{TN+FP\right)\left(TN+FN\right)}} \tag{24}$$

The measured performance metrics are given in Tab. 2. For a fair comparison, the proposed model is compared with previously proposed fault classification models like 2D-CNN (Antía López Galdo et al.), tsfresh-Random Forest (Chu, Z et al.), 1D-CNN-LSTM (Siłka, J et al.) and Kernel Mean Matching-SVM (Ghorvei, M. et al.).

Table 2 Performance analysis

Method	Accuracy	Precision	Recall	F1-Score	ROC-AUC Score	Logarithmic Loss	Cohen's Kappa	Matthews Correlation Coefficient
Proposed	95.07	95.1	95.07	95.07	99.54	0.107	0.9342	0.9343
34	93.04	93.03	93.04	93.03	98	0.15	0.9072	0.908
33	92.5	92.6	92.5	92.55	98.4	0.181	0.899	0.8995
32	91.72	91.84	91.72	91.75	98.69	0.1983	0.8895	0.8897
31	86.1	86.71	86.1	85.97	97.05	0.3069	0.8146	0.8177
30	84.12	84.5	84.2	84.27	96.5	0.254	0.844	0.8465

From the above Tab. 2, DSTC-SSFR model is more accurate in classifying faults. The results prove that DSTC-SSFR with BOA has better performance than all other models in fault classification, where the overall accuracy value is 95.07%, The precision value is 95.10%, the recall value is 95.07%, the F1-score is 95.07%, the Logarithmic Loss of 0.1070, the MCC value of 0.9343 and the ROC value is 99.54%. The overall Capsule -GRU model with OOA has given better performance in fault prediction.

The optimization using BOA improves the accuracy of the DSTC-SSFR model from 93.03% to 95.07%, Precision also increases from 98% to 99.54%, The F1-score rises from 0.9072 to 0.9342, Recall improves as well, from 0.9080 to 0.9343. Overall, BOA optimization significantly boosts the model's effectiveness across key metrics. DSTC-SSFR model showed an accuracy value of 93.04, precision of 93.03, recall of 93.04, F1-score of 93.03, Logarithmic Loss of 0.15 and MCC of 0.9080 for the test data.

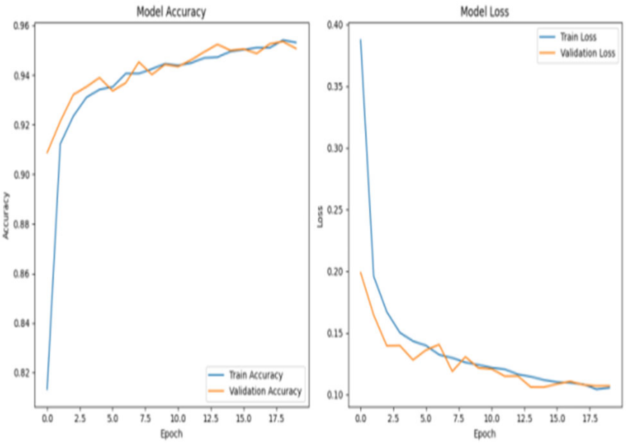

Figure 2 Model accuracy and loss curve

The 2D-CNN model showed an accuracy value of 92.60, precision of 92.50, recall of 92.55, F1-score of 98.40, Logarithmic Loss of 0.181 and MCC of 0.8995 for the test data. tsfresh-RF model showed an accuracy value of 91.72, precision of 91.84, recall of 91.72, F1-score of 91.75, Logarithmic Loss of 0.1983 and MCC of 0.8897 for the test data. 1D-CNN-LSTM showed an accuracy value of 86.10, precision of 86.71, recall of 86.10, F1-score of 85.97, Logarithmic Loss of 0.8146 and MCC of 0.8177 for the test data. KMM-SVM showed an accuracy value of 84.12, precision of 84.50, recall of 84.20, F1-score of 84.27, Logarithmic Loss of 0.8440 and MCC of 0.8465 for the test data.

The training and validation performance of a fault classification model is given in Fig. 2. In the accuracy graph, the blue line represents the training accuracy which consistently increases throughout the epochs and shows effective in training data. Also gap among training and validation loss is observed. This denotes the model is not overfitting. Likewise, in the loss graph, the validation loss decreases when epochs are increased. This denotes the model works well for given datasets. Fig. 7 shows the ROC curves for four different classes normal, overhang, under hang and imbalance. The Capsule-GRU model differentiates between all classes and achieves ROC scores of 0.99, 1, 0.99 and 01 for each class, respectively.

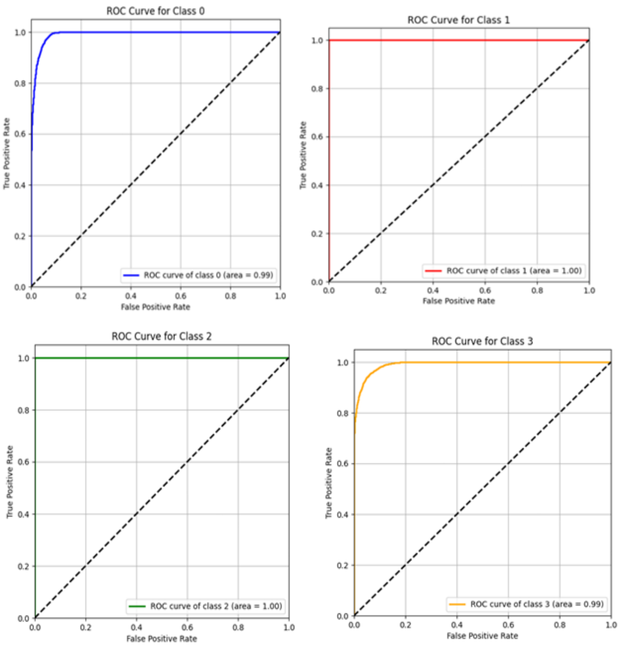

Figure 3 ROC of the proposed model for different classifications

5 CONCLUSIONS

This research introduced an innovative hybrid model integrating DSTC-SSFR for multi-class industrial fault classification. By integrating multi-scale convolutional layers for spatial feature extraction and Transformer encoders for temporal pattern modeling, the architecture effectively captures comprehensive fault characteristics from complex sensor data. The attention-based fusion mechanism further enhances the model's ability to prioritize the most salient features across both domains. Additionally, the integration of the BOA enhances parameter tuning, improving classification accuracy and robustness. The proposed model demonstrated superior performance in distinguishing multiple fault categories, achieving high accuracy and resilience to variations in input signals. Comparative analysis with existing methods further validated the model's effectiveness, highlighting its suitability for industrial environments requiring precise and timely fault detection. One key advantage of this approach is its adaptability to dynamic and diverse datasets, making it applicable for predictive maintenance and real-time fault diagnosis in industrial automation. While the results are promising, future research will focus on expanding datasets to include more fault conditions and testing the model's robustness with noisy and partially labeled data. The integration of explainable AI techniques will improve interpretability, enhancing transparency in industrial applications. Further investigations will explore deep learning architectures such as Convolutional Transformers to refine feature extraction. Additionally, hybrid optimization techniques and federated learning approaches will be explored to enhance hyperparameter tuning and support decentralized fault diagnosis in real-world industrial environments.

6 REFERENCES

[1] Anon. (2022). The use of sensor-based technology for enhancing maintenance operations. *Advances in Manufacturing Technology and Production Engineering*, 89-100. https://doi.org/10.2174/9789815039771122010012

[2] Joshi, S. J., Mamaniya, S., & Shah, R. (2022). Integration of intelligent manufacturing in smart factories as part of Industry 4.0 - A review. *2022 Sardar Patel International Conference on Industry 4.0 - Nascent Technologies and Sustainability for 'Make in India' Initiative*, 1-5. https://doi.org/10.1109/SPICON56577.2022.10180471

[3] Gómez, M., Castejón, C., & García-Prada, J. (2016). Automatic condition monitoring system for crack detection in rotating machinery. *Reliability Engineering & System Safety*, *152*, 239-247. https://doi.org/10.1016/j.ress.2016.03.013

[4] Mourtzis, D. & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop floor scheduling and condition-based maintenance. *Journal of Manufacturing Systems*, *47*, 179-198. https://doi.org/10.1016/j.jmsy.2018.05.008

[5] Nguyen, K. T. & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety*, *188*, 251-262. https://doi.org/10.1016/j.ress.2019.03.018

[6] Etebari, A., Arab, R., & Amirkhan, M. (2023). Forecasting demand level using time series and machine learning under uncertainty condition. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *30*(5).

[7] Pajić, N., Djapan, M., Buluschek, E., Fahrenbruch, W., Đorđević, A., & Stefanović, M. (2023). Machine learning prediction model for small data sets instead of destructive tests for resistance brazing process verification. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *30*(3).

[8] Lee, H. & Lee, D.-H. (2019). Prediction of times-to-failure of semiconductor chips using Vmin data. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *26*(1).

[9] Kong, X., Li, X., Zhou, Q., Hu, Z., & Shi, C. (2021). Attention recurrent autoencoder hybrid model for early fault diagnosis of rotating machinery. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1-10. https://doi.org/10.1109/TIM.2021.3051948

[10] TayebiHaghighi, S. & Koo, I. (2020). Fault diagnosis of rotating machine using an indirect observer and machine learning. *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 277-282. https://doi.org/10.1109/ICTC49870.2020.9289590

[11] Quabeck, S., Shangguan, W., Scharfenstein, D., & De Doncker, R. W. (2020). Detection of broken rotor bars in induction machines using machine learning methods. *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, 620-625. https://doi.org/10.23919/ICEMS50442.2020.9291033

[12] Jung, D. (2020). Data-driven open-set fault classification of residual data using Bayesian filtering. *IEEE Transactions on Control Systems Technology*, *28*(5), 2045-2052. https://doi.org/10.1109/TCST.2020.2997648

[13] Wang, G. & Wei, H.-L. (2022). A hybrid 1DCNN-SVM model for bearing fault recognition and classification. *2022 27th International Conference on Automation and Computing (ICAC)*, 1-6. https://doi.org/10.1109/ICAC55051.2022.9911115

[14] Jan, S. U., Lee, Y.-D., Shin, J., & Koo, I. (2017). Sensor fault classification based on support vector machine and statistical time-domain features. *IEEE Access*, *5*, 8682-8690. https://doi.org/10.1109/ACCESS.2017.2705644

[15] Kumar, S. (2021). Intelligent bearing fault diagnosis and classification based on support vector machine. *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 1-6. https://doi.org/10.1109/GCAT52182.2021.9587721

[16] Kang, J., Wu, K., Chi, K., & Du, Y. (2016). Multi-class intelligent fault diagnosis approach based on modified relevance vector machine. *International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 27-30. https://doi.org/10.1109/INCoS.2016.66

[17] Senanayaka, J. S. L., Kandukuri, S. T., Van Khang, H., & Robbersmyr, K. G. (2017). Early detection and classification of bearing faults using support vector machine algorithm. *IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, 250-255. https://doi.org/10.1109/WEMDCD.2017.7947755

[18] Fu, Y., Gao, Z., Zhang, Y., Zhang, A., & Yin, X. (2021). Data-driven fault classification for three-phase induction machines under stator inter-turn faults. *4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, 306-314. https://doi.org/10.1109/RCAE53607.2021.9638921

[19] Yinghua, Y., Guoqiang, S., & Xiang, S. (2018). Fault monitoring and classification of rotating machines based on PCA and KNN. *Chinese Control and Decision Conference (CCDC)*, 1795-1800. https://doi.org/10.1109/CCDC.2018.8407418

[20] Ding, X., Wang, J., Wang, J., Li, L., Chen, B., & Li, N. (2023). Fault diagnosis of multistage gearbox based on Adaboost ensemble learning. *Global Reliability and Prognostics and Health Management Conference (PHM-Hangzhou)*, 1-5. https://doi.org/10.1109/PHM-Hangzhou58797.2023.10482610

[21] Zhao, C., Zio, E., & Shen, W. (2024). Multidomain class-imbalance generalization with fault relationship-induced augmentation for intelligent fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *73*, 3526311. https://doi.org/10.1109/TIM.2024.3428618

[22] Ayana, Ö. & İnan, A. (2024). Industrial fault detection and classification with the optimal windows size approach. *32nd Signal Processing and Communications Applications Conference (SIU)*, 1-4. https://doi.org/10.1109/SIU61531.2024.10601128

[23] Yan, S., Wei, H., Yang, C., Liu, Z., Zhang, S., & Zhao, L. (2024). Fault diagnosis of rolling bearing based on CBAM-ICNN. *39th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 1420-1425. https://doi.org/10.1109/YAC63405.2024.10598778

[24] Phan, Q. N. X., Le, T. M., Tran, H. M., Tran, L. V., & Dao, S. V. T. (2024). Novel machine learning techniques for classification of rolling bearings. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3431040

[25] Wu, W. (2024). Fault monitoring and diagnosis of motor operation status based on LBP-SVM. *IEEE Access*, *12*, 104204-104214. https://doi.org/10.1109/ACCESS.2024.3434635

[26] Ganesamoorthy, N., Sakthivel, B., Subbramania, D., & Balasubadra, K. (2024). Hen maternal care inspired optimization framework for attack detection in wireless smart grid networks. *International Journal of Informatics and Communication Technology (IJ-ICT)*, *13*(1), 123-130. https://doi.org/10.11591/ijict.v13i1.pp123-130

[27] Dehghani, M. & Trojovský, P. (2023). Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Frontiers in Mechanical Engineering*, *8*, 1126450. https://doi.org/10.3389/fmech.2022.1126450

Contact information:

K. RANJANI, Assistant Professor
(Corresponding Author)
Department of Electronics & Communication Engineering,
SNS College of Technology, Coimbatore-641035 Tamilnadu, India
E-mail: ranjani.ece.snsce@gmail.com

T. BARANIDHARAN, Professor
Department of Electronics & Communication Engineering,
KSR College of Technology, Thiruchengode-637215 Tamilnadu, India
E-mail: baraniinst@gmail.com

J. RAMAL
Saveetha Institute of Medical Technology Sciences,
Saveetha University, Chennai, 603203 India