

Artificial Intelligence-Driven Fault Tolerance Mechanisms for Distributed Systems Using Deep Learning Model

Anila Gogineni*

Citation: Gogineni A. Artificial Intelligence-Driven Fault Tolerance Mechanisms for Distributed Systems Using Deep Learning Model. *J Artif Intell Mach Learn & Data Sci* 2023, 1(4), 2401-2406. DOI: doi.org/10.51219/JAIMLD/anila-gogineni/519

Received: 03 December, 2023; **Accepted:** 28 December, 2023; **Published:** 30 December, 2023

*Corresponding author: Anila Gogineni, Independent Researcher, USA, E-mail: anila.ssn@gmail.com

Copyright: © 2023 Gogineni A., This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

The foundation of technical advancement recently has been distributed systems. New technological developments, such as the internet of things, have it as their basis. However, permitting the failure of a single component within the system does not bring the whole system down, which is easily achieved through the use of distributed systems. This paper analyzes the use of deep learning models in artificial intelligence (AI) driven fault tolerance mechanisms for fault detection and fault mitigations in distributed systems. Pre-processing techniques like feature extraction and normalization were employed in order to prepare a distracted driver dataset that would be useful to train and test a model. It was further compared with other standard machine learning algorithms like Naïve Bayes (NB) to determine that the application of deep learning produces better results in fault detection. The performance tested showed that the VGG16 model was able to achieve an accuracy of 92%, with a precision of 91.73%, recall of 91.70% and F1-score of 91.70% was higher than the traditional method used in this research. The performance of the classifiers was evaluated in different conditions involving faulty data and it was observed that the classification accuracy decreases with increased amount of missing or unknown data. However, there are still some constraints with the method, these include high computational costs and low numbers of datasets.

Keywords: Fault tolerance, Distributed systems, Deep learning, Fault detection, Reliability, Intelligent control, Cloud computing, Resilient computing, Adaptive fault management, Data-driven fault diagnosis

1. Introduction

Distributed systems have emerged as the key enablers in modern computing paradigm as the large-scale applications such as cloud computing, communication systems and industrial automation. These are systems intended to operate on large volumes of data with the considerations of scalability, high system speed and availability of the systems¹. However, as the size and complication of the component and its distributed structure enlarges, the possibility of fault and failure in the structure becomes higher. Fault tolerance must be incorporated into the design of a distributed system to reduce the impact of these defects such as breakdowns, slow operation and monetary loss. The capacity of a system to keep running in the face of

errors or malfunctions is called fault tolerance. Distributed systems, unlike traditional centralized architectures, often experience partial failures where some components fail while others remain operational². These failures can result from hardware malfunctions, software bugs, network disruptions or resource overloads³. A robust fault tolerance mechanism ensures system stability by detecting, diagnosing and mitigating faults in real time.

When one component of a distributed system fails, the system as a whole does not go down since faults or failures in distributed systems are usually partial^{4,5}. In light of these difficulties, it is critical to develop, independently of quadrator dynamics expertise, a data-driven fault-tolerant synchronization

controller for very nonlinear multi-quadrotor systems operating under several simultaneous actuator failures. Traditional fault tolerance techniques⁶, such as redundancy, checkpointing and consensus algorithms, provide some level of resilience but often struggle with scalability and adaptability in dynamic environments.

Intelligent fault tolerance techniques are a new and exciting alternative to old approaches that have arisen with the emergence of AI and ML⁷. ML-driven approaches analyze vast amounts of system data to identify fault patterns, predict potential failures and trigger preventive actions before disruptions occur. By leveraging AI, distributed systems can enhance fault detection accuracy, reduce false positives and improve recovery efficiency. The capacity of DL, a subset of ML⁸, to automatically extract complicated patterns from massive amounts of data further improves fault management.

1.1. Motivation and contribution

The research motivation for this publication arises from the fact that distributed systems have become more common and widespread, the reliability and fault tolerance are critical. Redundancy protection is crucial in order to have such systems running continuously in tasks including self-driving vehicles and monitoring services. The existing fault tolerance techniques that are raised in solving the complications of distributed applications are not proactive and do not take into account faults as it happens in a dynamic system. Deep learning which is a form of AI presents a possibility of improving fault tolerance through fault identification and eradication. This justifies the analysis of fault detection and diagnosis using Artificial Intelligence models especially for distributed systems which failure may take place at many levels and thus affect system performance. The main contributions are:

- Develop effective DL models for fault tolerance for distributed systems by driver distraction dataset.
- Data preprocessing, including normalization, feature extraction and splitting the dataset, is presented to prepare data for efficient training and evaluation of deep learning models.
- Presents AI-driven fault tolerance mechanisms where deep learning models, namely VGG16 and NB, are proposed for a distributed computing environment with respect to fault identification in driver distraction cases.
- Use measures like F1-score, recall, accuracy and precision to assess the model's performance.
- The study demonstrates the impact of data quality on model performance, highlighting the vulnerability of dl models to faulty data and stresses the importance of robust data preprocessing for maintaining high model accuracy in fault-tolerant systems.

1.2. Structure of the paper

The following is the research protocol: Section II includes important research on distributed fault tolerance. In Section III, they detail everything from the procedures to the materials.

Section IV details the results of the tests conducted using the proposed system. This study is summarized and brought to a close in Section V.

2. Literature Review

This section discusses the surveys and reviews articles on Driven Fault Tolerance Mechanisms for Distributed Systems based on deep Learning Algorithms and AI.

Fallah, Ramezani and Mehrizi-Sani this study recommends a FTC and FDD approach using a supervised ML algorithm to detect, diagnose and classify grid faults; rectify input voltage prior to affecting grid-connected DER inverters; and, in the end, restore grid power. This controller is able to reduce the effect of grid failures on inverters by adjusting and forecasting the input voltage time series. In order to assess the suggested controller's efficacy and operational performance, simulation results are presented⁹.

Ishii and Namba this study have presented an error-tolerant approach to deep neural network (DNN) inference accelerated with field-programmable gate arrays (FPGAs) against Stuck-at-faults. They achieved a recognition rate of 99.7% for the error-free case by eliminating outliers using a threshold calculated from the median and deviation for the parameters and computing them as¹⁰.

Chen and Chakrabarty this study, demonstrate that ML models can identify millions of errors in minutes with an impressive classification accuracy of up to 99% by employing the ground truth obtained from MDT and forward inferencing. Additionally, they demonstrate that when applied to the ImageNet dataset, the ML model trained employing CIFAR-10 delivers very accurate fault classification results. They provide a fault-tolerance system that makes use of this high criticality-classification accuracy, which reduces the redundancy required for fault tolerance by 92%¹¹.

Hoang, Hanif and Shafique this study, use the VGG-16 and AlexNet DNNs that were trained on the CIFAR-10 dataset to assess their method. Evidence from experiments shows that their mitigation strategy greatly improves DNNs' ability to withstand errors. To illustrate, in comparison to the base network that does not use fault mitigation, the suggested method improves a classification accuracy of a resilience-optimized VGG-16 model by an average of 68.92% at a 1×10^{-5} fault rate¹².

Prathibha in this study technique guarantees that client data is unaffected in the event of a cloud failure. Fault tolerance measures may assist ensure that consumers get on-demand services as needed, which will increase cloud performance. This study compares and contrasts three ML algorithms—K-Means, DT and KNN—through the use of sensitivity and specificity metrics in a variety of scientific workflow application structures, including pipeline, merge, split and diamond¹³.

(Table 1) compares and contrasts several prior assessments of fault tolerance with respect to datasets, results, limitations and planned future research.

Table 1: Summary of background study on fault tolerance using deep learning algorithm.

Paper	Method	Dataset	Key Findings	Limitations / Future Work
Fallah, Ramezani and Mehrizi-Sani	Supervised ML for Fault Detection and Diagnosis (FDD) and Fault-Tolerant Control (FTC)	Grid-connected Distributed Energy Resources (DER) Inverters	The outcome of the simulation demonstrated that the suggested controller was successful.	Future work could involve testing the approach on real-world systems and optimizing the algorithm for scalability and robustness.
Ishii and Namba	Error-Tolerant Deep Neural Network (DNN) Inference with FPGA	Theoretical Approach	Presented an error-tolerant approach to DNN inference using FPGAs. Achieved 99.7% recognition rate	Future work could explore fault tolerance in larger-scale DNN models and investigate the impact of other fault types beyond Stuck-at faults.
Chen and Chakrabarty	Machine Learning for Fault Classification	CIFAR-10, ImageNet	ML models achieved high classification accuracy (up to 99%) for fault detection. The redundancy needed for fault tolerance was reduced by 92% as a result of the fault-tolerance solution that was offered.	Future research may include applying the method to more complex datasets and improving the model's adaptability to different fault types.
Hoang, Hanif and Shafique	Fault Mitigation in DNNs (AlexNet and VGG-16)	CIFAR-10	The proposed fault mitigation technique improved the resilience of DNNs, with a 68.92% improvement in classification accuracy for VGG-16 at a fault rate of 1×10^{-5} .	Future work could explore fault tolerance in more complex networks and assess the trade-offs between fault mitigation and computational overhead.
Prathibha	Comparative Analysis of ML Algorithms (K-Means, DT, KNN)	Scientific Workflow Applications (Pipeline, Merge, Split, Diamond)	Conducted a comparative analysis of ML algorithms for different workflow structures. Parameters such as sensitivity and specificity were evaluated.	Future work could focus on evaluating other ML algorithms and applying the framework to a broader set of applications.

3. Methodology

The methodology for Fault Tolerance Mechanisms for Distributed Systems Using Deep Learning Model involves multiple key steps. The driver distraction dataset is first collected and data preprocessing with data cleaning, where raw data from distributed systems is normalized (0,1) and split into training (80%) and testing (20%) sets. Feature extraction is applied to transform raw data into meaningful representations and deep learning models, including VGG16 is the proposed model with the comparison of NB, are employed to model fault tolerance mechanisms in distributed systems. The accuracy, precision, recall and F1-score are some of the performance indicators utilized to evaluate a model's ability to identify and mitigate problems. The best-performing model is selected for further analysis and deployment. Figure 1 displayed a data pipeline diagram for a whole study design process of fault tolerance for distributed systems.

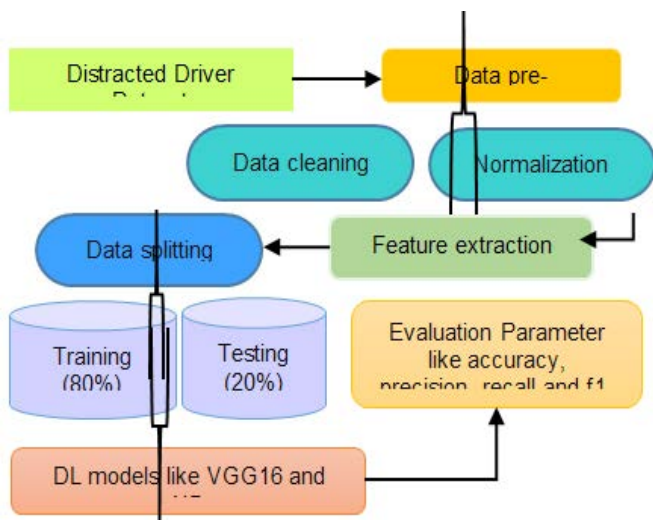


Figure 1: Flowchart for Fault Tolerance for Distributed System.

This section provides a concise explanation of the flowchart's subsequent steps:

3.1. Data collection

This study makes use of a distracted driver dataset, which contains 17,309 images sorted into the following groups: Safe Driving (3,686), Phone Right (1,223), Phone Left (1,361), Text Right (1,974), Text Left (1,301), Adjusting Radio (1,220), Drinking (1,612), Hair or Makeup (1,202), Reaching Behind (1,159) and Talking_to_Passenger (2,570). (Figure 2) shows a data correlation matrix.



Figure 2: Correlation matrix for Fault tolerance.

Figure 2 shows a heatmap of the correlation matrix for various car sensor readings, including rotation (rot x, rot y, rot z), acceleration (acc x, acc y, acc z), speed, accelerator pedal position, throttle, engine coolant temperature, engine load and engine speed. The linear correlations between these variables are shown in the heatmap along with their intensity and direction. There is a strong positive correlation of 1 and a significantly negative correlation of -1 in the set of correlation coefficients. There is no linear connection when the coefficient is 0.

3.2. Data preprocessing

Cleaning and normalizing data is a crucial part of data pre-processing, which is a crucial step for ML approaches¹⁴. A model's accuracy and performance are directly related to the quality of its pre-processing. In this research, the following procedures were taken to summarize the pre-processing of data related to driver distraction.

- Data cleaning, which involves handling missing or corrupt data¹⁵. Since the dataset is well-structured and does not contain missing values, no additional cleaning is required.
- The dimensions of each image are 640×480. There are images of 26 drivers in the collection.

3.3. Data normalization

Data normalization is applied to ensure consistency across different features, particularly pixel values, by scaling all pixel intensities to a range of [0, 1]. This enhances model convergence during training. By using Equation (1).

$$Y = a - \frac{\min(a)}{\max(a)} - \min(a) \quad Y = a - \frac{\min(a)}{\max(a)} - \min(a) \quad (1)$$

Where Y denotes a normalized value and a denotes an original value.

3.4. Feature extraction

Feature extraction is essential ML and DL, converting raw data into useful representations to improve model performance¹⁶. In image processing, convolutional neural networks (CNNs) automatically extract hierarchical features¹⁷. Effective feature extraction enhances classification accuracy, reduces dimensionality and improves computational efficiency, crucial for tasks like object recognition, medical imaging and driver distraction detection.

3.5. Data splitting

Separate sets of data were utilized for training and testing. To train models, that used a training set and to evaluate them, they utilized the test set. The data is split between training and testing, with 80% going into training and 20% into testing.

3.6. Proposed VGG16 models

A 16-layer deep CNN architecture, VGG16 is famous for its depth and simplicity thanks to its 13 convolutional layers and 3 fully linked layers. The model is able to capture hierarchical feature patterns because, after each convolutional layer, which utilizes tiny 3×3 filters, there are max-pooling layers that down-sample the spatial dimensions^{18,19}. The architecture is built to learn granular characteristics via a series of convolutional layers, starting with low-level edges and working its way up to high-level object representations. It is possible to depict the feature extraction procedure in VGG16 as follows (2):

$$F = Relu(W \cdot X + b) \quad F = Relu(W \cdot X + b) \quad (2)$$

In Equation (2), F stands for the features that were retrieved, W and b for the convolutional layer's weights and biases and ReLU for the Rectified Linear Unit activation function²⁰. The model's capacity to detect intricate patterns in the input scalograms is improved by introducing non-linearity using this activation function.

3.7. Performance metrics

The research in this study evaluated the efficacy of classification algorithm-built models with the use of a confusion matrix. To evaluate performance, four statistical metrics were used: F1-score, recall, accuracy and precision. The likelihood of accurately identifying the True Negative (TN) class is represented by specificity, while the likelihood of properly identifying the

True Positive (TP) class is represented by sensitivity. A false negative (FN) happens when a model predicts a negative class when the actual class is positive, while a false positive (FP) happens when a model predicts a positive class when the actual class is negative. The following metrics will be used to evaluate performance:

- **Accuracy:** The proportion of right guesses to the total number of forecasts is called accuracy, as in Equation (3).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

- **Precision:** A Positive Predictive Value (PPV) is another name for precision. As in Equation (4), it is the ratio of positive forecasts to the total number of positive class value predictions.

$$Precision = \frac{TP}{TP+FP} \quad Precision = \frac{TP}{TP+FP} \quad (4)$$

- **Recall:** Detection Rate (DR), True Positive Rate (TPR) and sensitivity are some of the other names for the recall measure. Equation (5) shows that it is defined as a fraction of class values in the test data that were correctly forecasted divided by the total number of accurate class values.

$$Recall = \frac{TP}{TP+FN} \quad Recall = \frac{TP}{TP+FN} \quad (5)$$

F1-score: A number of other names for the F1-score include F score and F measure. Precision and recall are said to be its harmonic means. Consequently, it communicates the equilibrium between recall and precision, as in Equation (6).

$$F1 - Score = \frac{2(Precision \cdot Recall)}{Precision + Recall}$$

$$F1 - Score = \frac{2(Precision \cdot Recall)}{Precision + Recall} \quad (6)$$

The models for deep learning are determined by these matrices.

4. Result Analysis and Discussion

A workstation equipped with an operating system, 32 GB of RAM, a 3.2 GHz CPU and a Tesla K80 GPU was used to evaluate the suggested concept. The experiment results DL models that are utilized for Fault tolerance are provided in this section. Accuracy, precision, recall and f1-score are some of the performance metrics used to assess the following suggested models, which were trained using the distracted driver dataset. The VGG16 model achieves the highest performance, as shown in (Table 2).

Table 2: VGG16 model Performance for Fault Tolerance on the distracted driver dataset.

Matrix	VGG16
Accuracy	92
Precision	91.73
Recall	91.70
F1-score	91.70

The above Table 2 and Figure 3 show the model performance for fault tolerance. VGG16 model achieves

excellent performance with an accuracy of 92%, precision of 91.73%, recall and F1-score both at 91.70%, indicating strong classification ability.

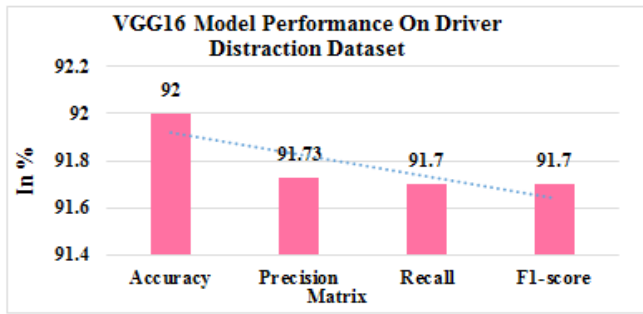


Figure 3: Performance of VGG16 Model.

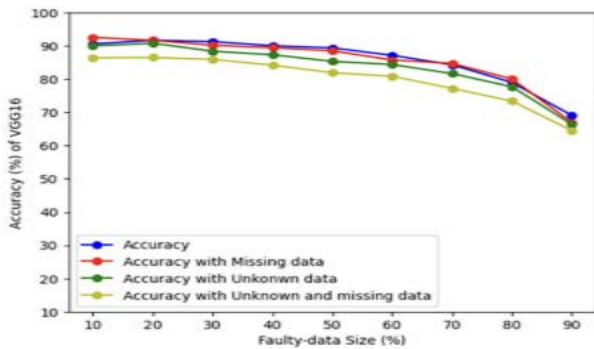


Figure 4: Accuracy graph of VGG16 model.

Figure 4 displayed the accuracy performance of the VGG16 model on the Driver Distraction dataset under varying levels of faulty data, where an x-axis displays a faulty data size (%) and a y-axis denotes accuracy (%). A graph compares four scenarios: standard accuracy (blue), accuracy with missing data (red), accuracy with unknown data (green) and accuracy with both unknown and missing data (yellow-green). As the faulty data size increases, the model’s accuracy declines across all cases, with the standard accuracy consistently outperforming the other conditions. The presence of both missing and unknown data leads to the most significant accuracy degradation, highlighting the impact of data quality on the robustness of the VGG16 model in driver distraction detection.

5. Comparison and Discussion

This section provides the comparative analysis between the proposed VGG16 model and the existing Naïve Bayes (NB)²¹ shown in (Table 3).

Table 3: ML and DL models comparison for fault tolerance.

Models	Accuracy
Naïve Bayes (NB) ²¹	90
Visual Geometry Group 16 (VGG16)	92

The following (Table 3 and Figure 5) shows the comparison of ML and DL models for fault tolerance, revealing that traditional ML models, such as NB, perform slightly better at 90%. However, using deep learning such as VGG16, these conventional models are enhanced with an accuracy rate of 92%.

The proposed DL-based fault-tolerant mechanism utilizing VGG16 model for fault detection has certain merits over conventional machine learning methods. It also showed better classification accuracy of 92% in contrast to NB, proves that it is capable of identifying and preventing fault occurrences in

distributed systems. Further, deep learning models, especially VGG16, can automatically extract the features from the data, which reduces the burden of manually creating features and improves the fault detection ability. It is also remarkable that, despite the insertion of faulty data of different intensities, the model can still perform with high classification accuracy. Further, deep learning models are capable of extrapolating well across the extensive and complex inputs hence are more appropriate in real-world failure tolerance in distributed systems.

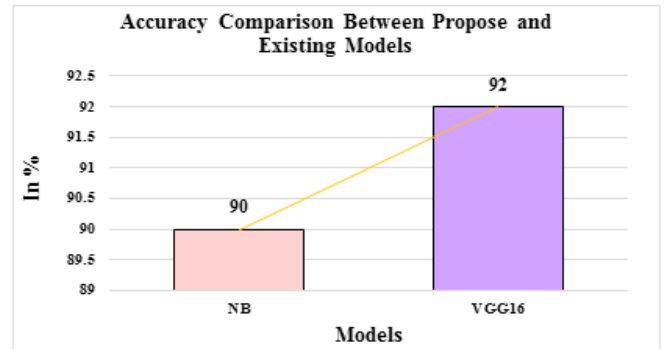


Figure 5: Comparison Bar Graph for Model performance.

6. Conclusion And Future Work

Fault tolerance is among the most important components of distributed systems, which ensures the proper operation in the presence of faults or failures within the system. AI agents are identified to be the key component in enhancing the reliability of fault-tolerant distributed systems as well as improving the system’s efficiency. This paper discussed an approach to the implementation of fault tolerance mechanisms in distributed systems through the use of DL techniques and featured the VGG16 model. Studies showed that VGG16 was more effective than the earlier ML models like RF (75%), DT (88.3%) and NB (90%), attaining an accuracy of 92%. From the precision around 91.73 %, recall of 91.70 %, F1 score of 91.70% it was evident that the model was effective in classifying different faults in distributed systems. The promising research outcomes have clear restrictions despite their favorable results. The research team analyzed only one database so their results may not work effectively throughout all distributed systems. The researchers did not analyze how well their system guards against attacks or how it functions in real time with changing conditions. VGG16 needs a lot of processing power and this creates deployment problems for devices with limited resources. Future researchers will analyze ways to make the model run faster without losing its performance advantages. Research into transformer networks offers chances to boost their fault management strategies.

7. References

- Gour L and Wao DAA. "Fault Tolerating Mechanism in Distributed Computing Environment," Int. J. Eng. Appl. Sci. Technol, 2020;5: 610-615.
- Sari A and Akkaya M. "Fault Tolerance Mechanisms in Distributed Systems," Int. J. Commun. Netw. Syst. Sci, 2015.
- Mehmood G, Khan MZ, Abbas S, Faisal M and Rahman HU. "An Energy-Efficient and Cooperative Fault-Tolerant Communication Approach for Wireless Body Area Network," IEEE Access, 2020.
- Kochhar D, Kumar A and Hilda J. "An Approach for Fault Tolerance in Cloud Computing Using Machine Learning Technique," 2017.

5. Thokala VS. "A Comparative Study of Data Integrity and Redundancy in Distributed Databases for Web Applications," IJRAR, 2021;8: 383-389.
6. Kolluri V. "A Comprehensive Analysis on Explainable and Ethical Machine: Demystifying Advances in Artificial Intelligence," Int Res J, 2015;2.
7. Peng B, Xia H, Lv X, Zhu S, Liu Y and Zhang J. "An intelligent fault diagnosis method for rotating machinery based on data fusion and deep residual neural network," Appl. Intell, 2022;52.
8. Vishwakarma PK. "An Efficient Machine Learning Based Solutions for Renewable Energy System," Int J Res Anal Rev, 2022;9: 951-958.
9. Fallah F, Ramezani A and Mehrizi-Sani A. "Integrated Fault Diagnosis and Control Design for DER Inverters using Machine Learning Methods," in IEEE Power and Energy Society General Meeting, 2022.
10. Ishii T and Namba K. "Stuck-at Fault Tolerance in DNN Using Statistical data," in Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC, 2022.
11. Chen CY and Chakrabarty K. "Efficient Identification of Critical Faults in Memristor-Based Inferencing Accelerators," IEEE Trans. Comput. Des. Integr. Circuits Syst, 2022.
12. Hoang LH, Hanif MA and Shafique M. "FT-ClipAct: Resilience Analysis of Deep Neural Networks and Improving their Fault Tolerance using Clipped Activation," in Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020.
13. Prathibha S. "Investigating the Performance of Machine Learning Algorithms for Improving Fault Tolerance for Large Scale Workflow Applications in Cloud Computing," in Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019.
14. Neeli SSS. "Key Challenges and Strategies in Managing Databases for Data Science and Machine Learning," Int J Lead Res Publ, 2021;2: 9.
15. Boddu B. "Ensuring Data Integrity and Privacy: A Guide for Database Administrators," Int J Multidiscip Res, 2022;4: 1-6.
16. Humeau-Heurtier A. "Texture feature extraction methods: A survey," IEEE Access, 2019.
17. Ammar M, El Habib Daho M, Harrar K and Laidi A. "Feature Extraction using CNN for Peripheral Blood Cells Recognition," EAI Endorsed Trans. Scalable Inf. Syst, 2022.
18. Gandhi Krishna JQ. "Implementation Problems Facing Network Function Virtualization and Solutions," IARIA, 2018: 70-76.
19. Adhinata FD, Tanjung NAF, Widayat W, Pasfica GR and Satura FR. "Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition," J. Ilm. Tek. Elektro Komput. dan Inform., 2021.
20. Tyagi S. "Analyzing Machine Learning Models for Credit Scoring with Explainable AI and Optimizing Investment Decisions," Am Int J Bus Manag, 2022;5: 5-19.
21. Harini Krishna S, Niveditha G and Gnana Mayuri K. "Reliability of fault tolerance in cloud using machine learning algorithm," Int J Innov Technol Explor Eng, 2019.