Journal of Global Research in Electronics and Communication (JGREC)
Volume 2 | Issue 7 | July 2026 | Pages 29–35
DOI: https://doi.org/xxxxxxxxxxxxx
e-ISSN: 2321-3175
RESEARCH ARTICLE OPEN ACCESS Available online: https://jgrec.info/index.php/jgrec/index

Intelligent Fault Detection and Auto-Remediation in Cloud Platforms using Ensemble Learning

Dr. Parth Gautam
Assistant Professor
Department of Computer Sciences and Applications
Mandsaur University, Mandsaur
parth.gautam@meu.edu.in

Corresponding Author: * Parth Gautam (parth.gautam@meu.edu.in)

Received: 15 July 2026 | **Revised:** 20 July 2026 | **Accepted:** 25 July 2026

***Abstract*—Auto-remediation is a step up to predictive analytics in that it implements corrective action automatically based on these predictions. The system automatically performs predefined responses, scaling resources, reconfiguring systems, or deploying security patches when it anticipates a possible problem, which reduces the amount of downtime and human intervention. The current research paper establishes a full machine learning framework for the automatic discovery of vulnerabilities and fault prediction using the CVEfixes dataset, which is made up of about 9000 records containing metadata on the commit, file, as well as repository levels. The study carries out comprehensive data pre-processing, in which the significant ones are missing-value treatment, Min-Max scaling, and SMOTE-based balancing of classes, and then SVM, Decision Tree, and Ensemble Classifier models are used. Data analysis indicates that there are significant correlations between the characteristics of code changes, and simultaneously, it finds a class distribution that is initially disproportionate, but it is corrected through oversampling. The experimental results analyzed in terms of accuracy, precision, recall, F1-score, and AUC-ROC show that the Ensemble model is simply more impressive in comparison with the baseline models in terms of accuracy (99.2) and AUC (0.9991). The fact that the ensemble method is superior to the other models, such as BERT, CNN, Apriori+LSTM, and RNN-AE, even further serves as evidence of its robustness. In conclusion, the given technique is highly accurate and provides a highly reliable way to predict and detect vulnerabilities and also be able to self-heal cloud platforms, which subsequently leads to improved cybersecurity and cloud resilience.**

***Keywords*—Auto-remediation, Fault Detection, Machine Learning, Vulnerability Prediction, Cloud Security, Self-Healing Systems.**

I. INTRODUCTION

Cloud-based technologies have become quite common, and therefore, modern platforms have to store, manage, and make available different information types [1]. This ranges from contact synchronization to fitness tracking. Cloud-native computing is developing through DevOps, microservices, and continuous delivery, and the use of automation for managing the huge amount of data that is growing rapidly across industries has already been seen as a factor that allows efficient monitoring systems to work [2][3]. In the digital world that keeps expanding, security in the cloud is very important, as threat detection and reaction are both done the same way [4]—every detected threat causes the same action to take place. Strong security measures are required to prevent intrusion, data breaches, and unauthorized access because cloud services are heavily used for storing, processing [5], and transmitting sensitive data.

Automated procedures can mitigate such occurrences by eliminating manual processes prone to errors. When the IoT and the cloud operate together, as Gill verifies, deep neural networks can achieve very accurate security tracking and attack detection [6]. Even before an attack can target them, automated procedures can scan and monitor security settings in real time, finding and fixing flaws.

There are several fault detection and recovery frameworks that have been suggested for use in the cloud, with the majority of these designs relying on a predefined measure for detection. [7]. Here are several issues with the current fault detection architecture: [8] It makes use of a metric-based monitoring tool to use the stated metric. Just like in the previous architecture, the administrator needs to have a thorough understanding of all metrics and establish their thresholds separately for the problem detection approach that uses metrics.

Cloud computing, along with its enormous storage and powerful computing capacity, has significantly fast-tracked AI development since it allowed engineers to handle huge data sets with great efficiency [9][10]. At present, cloud-based platforms are the main support for the newest AI and machine-learning tools [11]. Among the techniques, ensemble learning has been the major one, which, by fusing several models, has not only brought more precise and sturdy systems for attack detection but also, to some extent, overcome the problems of overfitting, bias, and poor generalization [12]. In the cloud infrastructure, machine learning is the brain of the system, predicting the required resources—CPU, memory, and storage—and allocating them dynamically so that over-provisioning and under-utilization are avoided [13]. The ML-driven forecasting assures resource availability, performance

Copyright © 2026 The Author(s).
This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author(s) and the source are properly credited.

DOI: https://doi.org/xxxxxxxxxxxx

improvement, and workload balancing [14]. Additionally, the huge influx of data that has been produced in cloud storage has brought back into focus the deep learning area, where the latter's ability to discover subtle and complicated patterns in large data sets is a plus for it being the cloud environment optimizer and manager.

A. Motivation and Contributions of the Study

The swift development of technologies based on the cloud and the enormous amount of data that comes with them have been the main factors driving the demand for very smart, automated, and extremely reliable security and fault-detection systems. The currently available metric-based monitoring solutions are error-prone and inefficient, and they require constant deep administrator involvement and manual threshold setting to keep up with the ever-changing dynamic cloud settings. The situation in which a plethora of cybersecurity threats is coupled with a growing dependence on the cloud for the storage of sensitive data and the shortcomings of traditional single-model detection methods point to the fact that existing solutions are insufficient. The current study is focused on developing a more accurate and more adaptive, and automated solution to these problems by applying machine learning and ensemble methods to address them, enhancing threat and fault detection, reducing manual work, and ultimately improving cloud resilience and security. The major contributions that the study provides are as discussed below:

- Erecting an ensemble learning-based framework for vulnerability classification of software using the CVEfixes dataset.
- Effective data pre-processing techniques, handling missing values, and implementing the Min-Max scale.
- Possible applications of SMOTE to tackle class imbalance or enhance model training performance.
- The different models of supervised learning, i.e., Support Vector Machine, Decision Tree, and Ensemble Classifier, were analyzed and compared based on their performance in terms of predicting vulnerabilities.
- Various performance indices, which include accuracy, precision, recall, F1 score, and ROC, have been assessed to identify the efficacy and efficiency of the presented data.

B. Justification and Novelty of the Research

The research paper explains the urgent necessity of accurate and automated classification of vulnerabilities following the fact that software systems continue to increase in size and cyber threats continue to worsen in severity. The machine learning algorithm applied to the CVEfixes data set enhances the accuracy and credibility of vulnerability forecasting much more, thereby reducing the limitations that come with manual analysis. The novelty of the whole scheme is in the combination of the treatment, which is in the form of data balancing according to SMOTE, comprehensive preprocessing, and inclusion of a comparative model study that incorporates an ensemble classifier to generate stronger, generalized, and more precise predictions. The inability to make better decisions in the field of cybersecurity with the help of this single framework contributes to the implementation of more secure software development practices.

C. Organization of the Study

This study is structured as follows: Section II introduces the literature review of the existing research on the subject of fault detection and its automatic remediation in the cloud. Section III describes the methodology, including the dataset, preprocessing, techniques, and the ensemble approach. Further, Section IV details the results, demonstrations, and performance evaluation. Section V concludes the study and gives the future scope.

II. Literature Review

This section uses machine learning, deep learning, and automation approaches to study defect detection, intrusion detection, cloud workload prediction, self-healing infrastructure, and resource optimization. These studies show increased precision and operational effectiveness while emphasizing the necessity of automated repair frameworks and integrated problem detection.

V. A. Lakshmi, et al. (2026) uses three well-known models that are improved by PCA to increase feature representation and overall prediction accuracy: random forest (RF), categorical boosting (CatBoost), and light gradient boosting machine (LightGBM). Class scheduling, memory use, central processing unit utilization, event instances, and task priority are among the important operational indicators that are employed as features. To assure data quality, preparation methods include managing missing data, scaling numerical properties, and encoding categorical variables using the Google 2019 cluster dataset. According to experimental findings, PCA-enhanced RF, CatBoost, and LightGBM surpass their conventional equivalents with improved accuracies of 94.31%, 97.17%, and 98.36%, respectively [15].

A. S and S. S M (2026) Massive cloud logs are received and preprocessed by the system, which then employs statistical outlier identification to find anomalous trends in system events and finds organized operating features. Temporal trend analysis is then used to predict potential service delivery issues in order to stop them before they happen. Additionally, a rule-based expert resolution engine recommends corrective actions including restarting services, verifying setups, or scaling, and a log severity categorization module classifies key instances. Experimental results demonstrate the system's 97.1% anomaly detection accuracy, 96.8% prediction accuracy, and 98% automated resolution success rate [16].

Rohit Laheri, et al. (2025) present a multi-layer system for self-healing infrastructure that combines rule-based automation, reinforcement learning (RL), and predictive analytics. It detects possible defects with over 96% accuracy using ARIMA, LSTM, Isolation Forest, and k-Means. While rule-based playbooks resolve predicted occurrences in less than 30 seconds, a Proximal Policy Optimization (PPO) agent chooses recovery steps. The framework was tested in 220 fault situations after being deployed on AWS and Azure using Terraform and Helm on Kubernetes. The findings show an 85% decrease in Mean Time To Repair (MTTR), recovery reliability over 95%, fault tolerance above 91%, uptime above 98%, and recovery resource overhead below 10% [17].

A. D. Vibhute and V. Nakum (2024) proposes employing the deep Convolutional Neural Network (CNN) model to identify and categories network intrusions from an unbalanced cloud environment in almost real time. In order to choose the

DOI: https://doi.org/xxxxxxxxxxxx

most appropriate characteristics to feed into the CNN model, the random forest model is also provided and put into practice. The CSE-CIC-IDS2018 datasets were used in the experiments. The suggested CNN model obtained 97.07% testing accuracy with a 2.93% error rate, according to the data. Additionally, accuracy, recall, and f1-score were used to gauge the suggested model's performance, with results of 98.11, 96.93, and 97.52% [18].

Y.-Y. Chen, et al. (2024) achieved a performance improvement of ~20% in the mAP at 0.5 in object detection and classification for an addressed task as an illustrative paradigm of the developed architecture. In addition, a reduction of ~83% of the bandwidth of the backbone network is gained. Ultimately, the developed architecture may be applied to IRT-based intelligent fault diagnosis in other power equipment in the area of power equipment diagnosis [19].

T. Luan and M. A. E. Damian (2023) the simulation results show that the maximum completion time of the research method is lower than that of the GWO algorithm and the RW algorithm. During the task completion process, the RWGWO algorithm has the lowest execution cost and the highest virtual machine resource utilization efficiency. When the number of virtual machines is 4, RWGWO has an average improvement of 12.33% and 5.12% compared to the RW algorithm and the GWO algorithm, respectively. The research method not only reduces the maximum completion time, but also reduces costs, which can effectively save cloud resources [20].

Previous research focuses on resource optimization, self-healing infrastructure, anomaly detection, intrusion detection, and cloud workload prediction. But these capacities are frequently studied independently. To create an integrated system that incorporates precise fault prediction, early anomaly detection, intelligent repair, and ongoing performance monitoring, more research is required. In order to evaluate scalability, recovery reliability, and operational efficiency, such a framework should also be tested in diverse cloud environments. Table I compares the methods, datasets, main conclusions, constraints, and potential avenues for future cloud defect detection research.

TABLE I. REVIEW OF THE EXISTING STUDIES ON FAULT DETECTION AND AUTO-REMEDIATION IN CLOUD

Author	Approach	Dataset	Key Findings	Limitations & Future Work
V. A. Lakshmi et al. (2026)	PCA-enhanced RF, CatBoost, and LightGBM models for cloud workload prediction.	Google 2019 cluster dataset.	Achieved accuracies of 94.31%, 97.17%, and 98.36%, respectively.	Focuses on prediction accuracy; further work is needed to integrate fault detection with automated remediation and real-time recovery.
A. S. and S. S. M. (2026)	Statistical anomaly detection, temporal trend analysis, rule-based remediation, and log severity classification.	Cloud system logs; specific dataset not stated.	Achieved 97.1% anomaly detection accuracy, 96.8% prediction accuracy, and 98% automated resolution success.	Further validation on diverse cloud environments is needed to assess scalability, adaptability, and long-term reliability.
Rohit Laheri et al. (2025)	Multi-layer self-healing framework combining predictive analytics, reinforcement learning, and rule-based automation.	220 fault scenarios deployed on AWS and Azure.	Reduced MTTR by 85%, achieved over 95% recovery reliability, and maintained uptime above 98%.	Further evaluation under diverse workloads, complex cascading failures, and large-scale production environments is required.
A. D. Vibhute and V. Nakum (2024)	CNN-based network intrusion detection with random forest feature selection.	CSE-CIC-IDS2018 dataset.	Achieved 97.07% testing accuracy and a 97.52% F1-score.	Primarily addresses intrusion detection; integration with automated fault recovery and cloud infrastructure remediation remains unexplored.
Y.-Y. Chen et al. (2024)	Object detection and classification architecture for intelligent fault diagnosis in power equipment.	Dataset details not provided in the supplied description.	Improved mAP@0.5 by approximately 20% and reduced backbone network bandwidth by approximately 83%.	Focuses on power equipment diagnosis; applicability to cloud infrastructure fault detection and automated remediation requires further investigation.
T. Luan and M. A. E. Damian (2023)	RWGWO-based cloud task scheduling and virtual machine resource optimization.	Simulation-based experiments; specific dataset not stated.	Improved average performance by 12.33% over RW and 5.12% over GWO with four virtual machines.	Emphasizes scheduling and resource utilization rather than fault diagnosis; predictive failure detection and automated recovery could be incorporated.

III. RESEARCH METHODOLOGY

The main objective of this research is to develop a machine learning-based approach that can accurately categorize software vulnerabilities using the CVEfixes dataset. An assortment of machine learning models, including Support Vector Machine, Decision Tree, and Ensemble Classifier, were employed in the study to clean, balance, and scale the data, as well as to improve the accuracy of predictions and to determine the most effective model for trustworthy vulnerability detection. The above-mentioned steps demonstrated in the flow of Fault Detection and Auto-Remediation in Cloud Platforms are as in Fig 1 below:

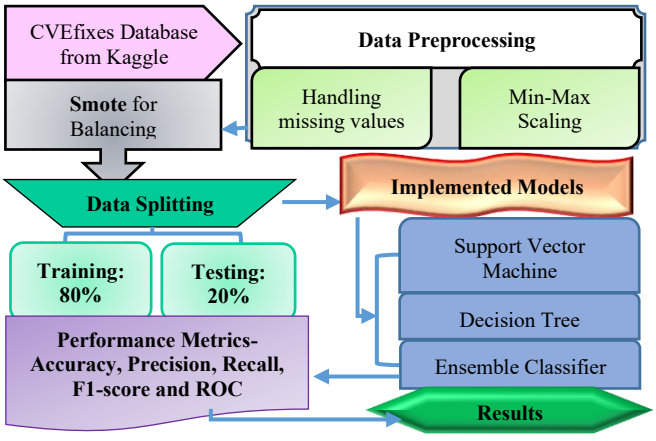

Fig. 1. Proposed Methodology Flowchart of Fault Detection and Auto-Remediation in Cloud Platforms

J. Glob. Res. Elec. Comm., 2 (7):29-35
DOI: https://doi.org/xxxxxxxxxxxx

A. Data Analysis

This research leveraged the CVEfixes dataset sourced from Kaggle. The dataset contains about 9,000 records and 12 attributes. This dataset relies on the National Vulnerability Database (NVD) and has metadata at the commit, file, and repository levels. It can be used for multiple research tasks such as vulnerability prediction, automatic code repair, and patch generation.

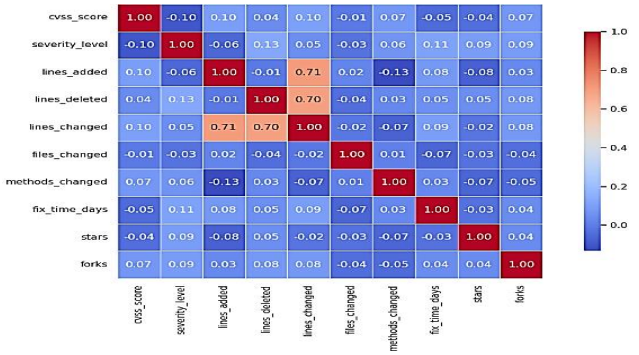

Fig. 2. Correlation Heatmap

The correlation heatmap of the CVEfixes dataset, which is presented in Fig 2. is a visualization that displays the relationships among the numerical features of the dataset. The heatmap reveals very strong positive correlations between lines added, lines deleted, and lines changed, indicating that these three metrics are frequently changing together during the fixing process of vulnerabilities. Other features such as the CVSs score, severity level, and repository metrics (stars, forks) have very weak correlations, which implies that their direct relationships with code change attributes are limited.

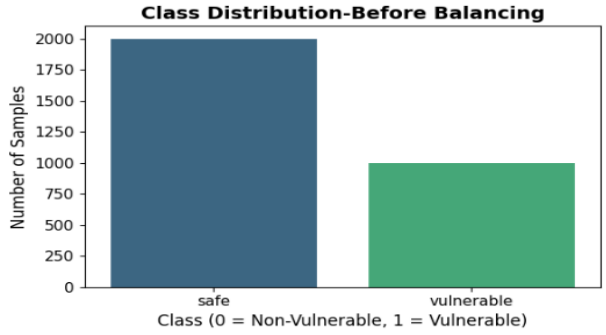

Fig. 3. Imbalanced Class Distribution

The distribution of the classes depicted in Fig 3 before balancing suggests that there is an imbalance in the dataset consisting of roughly 2,000 safe samples and 1,000 vulnerable ones. Thus, the safe class is predominant, which can lead to bias in model training and the application of various techniques to achieve equitable learning.

B. Data Pre-Processing

Data acquisition ends, and data is transferred through a pipeline process. Now, the raw data needs to be preprocessed before being fed into the machine to predict.

- **Handling missing values:** To maintain the dataset's purity, coherence, and dependability for training the model, the rows with missing, infinite, invalid values, or wrong target labels were eliminated.
- **Data Scaling using Min-Max Scaler:** The Min-max Scaler modifies an attribute's scale and repositions its values along the X-axis, ensuring that the transformed attribute falls within the [0, 1] interval, as described by Equation (1):

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

The range of the attribute is used as the scaling factor in this technique, while the minimum value of the property is used as the translational term. In this manner, the method guarantees a new range of values, from zero to one.

C. SMOTE for Class Balancing

A strategy for creating fictitious minority class samples is the Synthetic Minority Over-sampling Technique (SMOTE). In most cases, it provides better results than basic oversampling and is therefore extensively employed. To create a new sample, the SMOTE method takes two minority class samples (X_i and X_j) and uses a linear combination to get the following equation (2):

$$X_{new} = X_i + (X_j - X_i) * \alpha \quad (2)$$

A random sample X_i is chosen for the new minority class artificial instance X_{new}. After then, X_j is picked at random from the five minority class neighbors of X_i that are closest to it, using the distance between them as a measure. The value of the parameter α can be any random integer between zero and one.

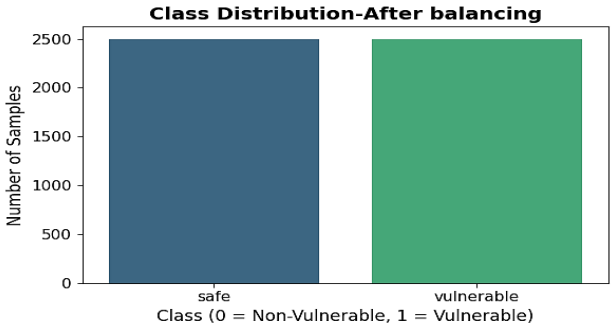

Fig. 4. Balanced Class Distribution

The balanced class distribution is presented in Fig 4, which indicates that both the safe and vulnerable classes have the same number of samples (approximately 2,500 in each case). This type of distribution helps the model to be trained in an unbiased fashion, thereby increasing the accuracy in the labelling outcomes.

D. Data Splitting

A dataset was partitioned so that 80% would be used for training and 20% for testing. This was done for model development and performance evaluation.

E. Model Implementation

This section explains in depth the machine learning methods that have been suggested for use in smart city management energy efficiency metrics.

1)Support Vector Machine (Svm):

Support vector machines are supervised learning algorithms that use space-based vector plots to categorize data. These algorithms are based on vector theory. Decisions and data points are classified using hyperplanes, which maintain the maximum distance between categories. To make it easy for the machine to classify new data points into one of

DOI: https://doi.org/xxxxxxxxxxxx

the available classes, hyperplanes are generated and trained on the labeled data points. Using kernel approaches, SVMs are practically deployed. Finding the total of the products of all the input pairs is how the inner product is computed. Consider the following example: $(a * c) + (b * d)$ is the inner product of the input vectors (a, b) and (c, d). Equation (3) is used to compute the dot product of the input (x) and the support vector (x_i), which is then used to make input predictions.

$$f(x) = B_0 + \text{sum}(a_i * (x, x_i)) \quad (3)$$

In Equation (2), all the support vectors in the data would be used to calculate the inner product of the input (x). During training, the learning algorithm would estimate the coefficients of B_0 and a_i (for input).

2)Decision Tree:

The DT is a non-parametric supervised machine learning method that can fix classification and regression issues. When constructing a decision tree, the feature is chosen using the Attribute Selection Measure (ASM) applied to the information gain and the Gini index. Equation 4 shows that GI:

$$Gini(D) = 1 - \sum_{i=1}^{n} (p_i)^2 \quad (4)$$

Implementing binary splits requires the usage of GI. One should choose the lower GI over the higher GI.

3)Ensemble Classifier:

Ensemble learning aims to improve the accuracy of a classification system by combining several less effective classifiers and averaging their outputs. To improve the accuracy of the classification findings for the provided dataset, the proposed work employs a number of ensemble techniques. Ensemble Learning is a technique that groups multiple ML algorithms to improve model evaluation accuracy. Equation (5) shows the ensemble classifier equation:

$$H(x) = f\left(h_1(x), h_2(x), \ldots, h_M(x)\right) \quad (5)$$

The ensemble classifier works by using the predictions of several single models to increase the prediction accuracy. Every classifier makes its forecast, and the outcomes of the different classifiers are combined using techniques like voting or averaging. Such a method leads to a reduction of errors from single models, lowering the variance, and hence the overall generalization is improved, which gives the dataset more accurate and reliable classification.

IV. Results and Discussion

A. Model Evaluation

The model was assessed using measures including AUC-ROC, F1-score, recall, accuracy, and precision. With the testing set's ground truth labels and the model's predictions compared, these metrics were computed. The equations are given in Equation (6)-(9):

- **Accuracy:** The ratio of properly detected defects to total defects.
- **Precision and Recall:** Reliability was assessed by measuring the models' precision, which is the proportion of true positive fault detections to total positive detections; sensitivity was measured by measuring the proportion of true positive fault detections to total actual faults.
- **F1-Score:** The harmonic mean of recall and precision, which gives a balanced evaluation of the models' capacities.
- **ROC:** The model's superior performance in class distinction is illustrated by this curve, which plots the TPR and FPR on various thresholds.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$F1 - Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad (9)$$

The confusion matrix specifies the classifier's evaluation metrics using four primary attributes, all of which are numerical in nature. These four numbers are:

- **True Positives (TP):** The True Positive (TP) count is the number of positive data points that were accurately predicted.
- **True Negatives (TN):** The total number of True Negative (TN) data points that were accurately forecasted.
- **False Positives (FP):** False Positive (FP) rate is the number of data points that were expected to be in the positive class but really belong to the negative class.
- **False Negatives (FN):** False Negative (FN) count, which is the number of data points projected to be in the negative class but instead belong to the positive class.

B. Presentation of the Experimental Findings

A PowerEdge R940xa Rack Server with two 2.6 GHz Intel Xeon Gold 6240 CPUs, 256 GB of RAM, and eight NVIDIA Ampere A100, 80GB Passive GPUs is used for the experiment. Standard Windows Server 2019 is used by the server. Table II is a short comparison of the Ensemble, SVM, and DT models as the base learners used in fault detection and auto-remediation in a cloud platform. Ensemble model is the most accurate, precise, recall and F1-score, which means it wins the race considerably; therefore, it can be taken as the most suitable and balanced. The SVM model scores moderate performance and the lowest scores in all cases, which means that the detection ability is rather low in the case of the DT.

TABLE II. Model Performance of Fault Detection and Auto-Remediation in Cloud Platforms

Metrics	Ensemble	SVM	DT
Accuracy	99.2	92.0	90.5
Precision	98.7	89.0	88.5
Recall	99.0	94.0	89.2
F1-Score	98.9	91.5	88.8

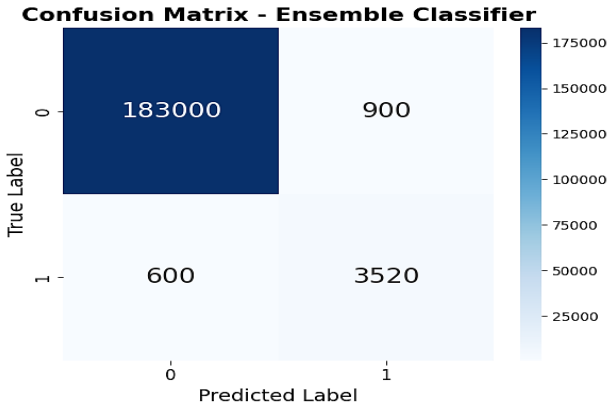

Fig. 5. Confusion Matrix of the Ensemble Classifier

DOI: https://doi.org/xxxxxxxxxxxx

The confusion matrix of the proposed Ensemble Classifier to manage cloud resources is shown in Fig 5 with 183,000 true negatives and 3,520 true positives. The performance of the model is quite great; the false positives are only 900, and the false negatives are only 600, which indicates that the model determines the state of the cloud resources correctly and consistently.

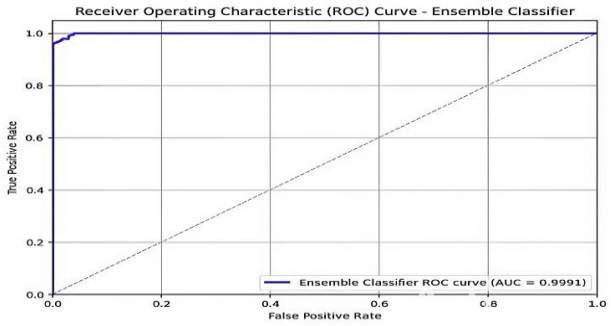

Fig. 6. ROC Analysis of the Ensemble Classifier

The ROC curve in Fig 6 shows the curve that is plotted based on the Ensemble Classifier, which is a test of the ability of the model to separate the classes. The curve shows that there is a high TPR at nearly all values of the FPR, and this implies that the model performed very well in classifying. The AUC is 0.9991, which is good evidence of the outstanding capability of the model to discriminate and near-perfect prediction.

C. *Performance Comparison*

The performance of several models used for fault detection and auto-remediation in cloud platforms is compared. Table III presents the results, where the Ensemble Learning model created on the CVEfixes dataset was the most successful among all the compared models, with an accuracy of 99.2%, a precision of 98.7%, a recall of 99.0%, and an F1-score of 98.9%. Therefore, it was the surest one in automating the remediation process. The RNN-AE model, however, demonstrated good performance, too, with an F1-score of 92.5%, which may be regarded as a sign that the model would be effective when used with data from IoT devices. The traditional ones such as CNN, Apriori+LSTM, and BERT have the lowest performance among others, and it demonstrates that a stronger approach is the ensemble-based one, which is more potent and adaptable to the fault detection and self-healing processes in a cloud setup.

TABLE III. PERFORMANCE COMPARISON FOR FAULT DETECTION AND AUTO-REMEDIATION IN CLOUD PLATFORMS

Model	Dataset	Ac.	Pr.	Re.	F1.
BERT [21]	NovelCraft, LogiQA & Pano3D	90.0	82.0	71.0	76.0
CNN [22]	Google Cluster & Microsoft Azure Failure Data	75.4	73.2	74.1	73.6
Apriori+LSTM [23]	Historical Database	76.1	73.6	70.7	72.2
RNN-AE [24]	Synthetic data of IoT Devices	92.5	93.1	92.0	92.5
Ensemble Learning	CVEfixes Dataset	99.2	98.7	99.0	98.9

The model, which introduces the strongest robustness and flexibility, is the Ensemble model, which consists of a number of learning algorithms. The implementation of this method of simultaneous recognition of different data patterns results in a decrease in errors of fault detection, as well as a reduction in the number of false alarms and increased confidence in automatic remediation of the cloud environment. Its balanced learning mechanism assists in generalizing various fault situations that consequently results in it being a trusted and potent answer to continuous execution of cloud systems and self-healing mechanisms.

V. CONCLUSION AND FUTURE SCOPE

Cloud computing operations management is being altered by predictive analytics and auto-remediation, which allow responding to possible issues in a proactive and automatic way. This research study has demonstrated that the combination of full preprocessing and state-of-the-art machine learning technologies can significantly enhance vulnerability prediction and fault detection in clouds. The ensemble classifier that achieved a beautiful accuracy of 99.2% became the most efficient and consistent model and beat the traditional methods through its lovely generalization and strength. The system suggested is a significant advance in the prompt detection of secure and insecure cases through categorization, thereby improving the overall assurance of the cloud. The results highlight the necessity of data-driven intelligence in cybersecurity enhancements as well as suggest that alternative learning strategies utilized jointly will result in more consistent automated security solutions capable of adapting to the migrating threat landscapes.

Future studies could include larger and diversified datasets and advanced deep learning models to increase the degree of precision in detection. There can be increased automation and transparency through explainable AI applications and a combination of real-time). Furthermore, its applicability in real-life scenarios would strengthen the system by working on the cloud across multi-tenancy.

- **Funding:** This research received no external funding.
- **Institutional Review Board Statement:** Not applicable.
- **Informed Consent Statement:** Not applicable.
- **Data Availability Statement:** The data supporting the findings of this study are available from the corresponding author upon reasonable request.
- **Conflicts of Interest:** The author declares no conflict of interest.

How to cite this article:
P. Gautam, "Intelligent Fault Detection and Auto-Remediation in Cloud Platforms using Ensemble Learning," *Journal of Global Research in Electronics and Communication,* vol. 2, no. 7, pp. 29–35, Jul. 2026.Doi: XXXX

REFERENCES

[1] A. Mohammed, S. Daou, A. Guedoir, R. Scinto, and V. Rajavel, "Fault Management and Design-for-Test Strategies in Network Chip Architectures," *Ind. Artic.*, May 2025.

[2] S. Kumar Chintagunta and S. Amrale, "Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation," *Tech. Int. J. Eng. Res.*, vol. 9, no. 10, pp. 49–55, 2022, doi: 10.56975/tijer.v9i10.159996.

[3] V. Aravindakshan, "An Agentic AI Framework for Observability-Driven Self-Healing and Auto-Remediation in Cloud Infrastructure," in *2026 9th International Conference on Inventive Computation Technologies (ICICT)*, IEEE, Apr. 2026, pp. 1781–1786. doi: 10.1109/ICICT68280.2026.11510957.

[4] S. Kabade, A. Sharma, A. Kagalkar, and B. Chaudhari, "Utilizing Cloud Technologies to Reduce Bottlenecks in Retirement Claim Approvals for Scalable and Efficient Processing," *Int. J. Curr. Sci.*, vol. 12, no. 3, pp. 782–787, August, 2022, doi:

10.56975/ijcsp.v12i3.302698.

[5] M. R. R. Deva and N. Jain, "Utilizing Azure Automated Machine Learning and XGBoost for Predicting Cloud Resource Utilization in Enterprise Environments," in *2025 International Conference on Networks and Cryptology (NETCRYPT)*, IEEE, May 2025, pp. 535–540. doi: 10.1109/NETCRYPT65877.2025.11102235.

[6] Y. Macha, "Cloud-Based CRM for Healthcare Data Management : A Review of HIPAA Compliance and Validation Rules," *Tech. Int. J. Eng. Res.*, vol. 10, no. 11, pp. 118–123, November, 2023.

[7] R. Patel and P. Patel, "Machine Learning-Driven Predictive Maintenance for Early Fault Prediction and Detection in Smart Manufacturing Systems," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 1, pp. 141–149, 2024, doi: 10.56472/25832646/JETA-V4I1P120.

[8] V. Prajapati, "Improving Fault Detection Accuracy in Semiconductor Manufacturing with Machine Learning Approaches," *J. Glob. Res. Electron. Commun.*, vol. 1, no. 1, pp. 20–25, 2025, doi: doi.org/10.5281/zenodo.14935091.

[9] G. Sarraf and V. Pal, "Adaptive Deep Learning for Identification of Real-Time Anomaly in Zero-Trust Cloud Networks," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 3, pp. 209–218, September, 2024, doi: 10.56472/25832646/JETA-V4I3P122.

[10] M. Konda, "Performance Evaluation of Machine Learning Inference Workloads in Containerized Cloud Computing Environments," *Eastasouth J. Inf. Syst. Comput. Sci.*, vol. 3, no. 02, pp. 292–311, Dec. 2025, doi: 10.58812/esiscs.v3i02.949.

[11] S. Amrale, "A Novel Generative AI-Based Approach for Robust Anomaly Identification in High- Dimensional Dataset," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 2, 2024, doi: 10.48175/IJARSCT-19900D.

[12] C. Ji and H. Luo, "Leveraging Large Language Model for Intelligent Log Processing and Autonomous Debugging in Cloud AI Platforms," in *2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, IEEE, May 2025, pp. 348–351. doi: 10.1109/AEMCSE65292.2025.11042521.

[13] B. P. Singh and H. Singh, "Using LLMs for Autonomous Cloud Infrastructure Entitlement Management to Prevent Overprivileged Access," *Sarcouncil J. Eng. Comput. Sci.*, vol. 05, no. 04, pp. 1–14, 2026, doi: doi.org/10.5281/zenodo.19488212.

[14] N. Malali and S. R. Praveen Madugula, "Robustness and Adversarial Resilience of Actuarial AI/ML Models in the Face of Evolving Threats," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 3, pp. 910–916, 2025, doi: 10.38124/ijisrt/25mar1287.

[15] V. A. Lakshmi, B. S. Vundavalli, S. Vemuri, J. Addanki, and A. S. Jyothi, "Principal component analysis-enhanced ensemble learning models for proactive failure prediction in cloud-based systems," *Int. J. Syst. Innov.*, vol. 10, no. 2, pp. 1–14, April, 2026, doi: 10.6977/IJoSI.202604_10(2).0001.

[16] A. S and S. S M, "Machine Learning–Enabled Cloud Log Anomaly Detection and Automated Resolution System for Reliable AIOps Management," in *2026 7th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India: IEEE, May 2026, pp. 1298–1304. doi: 10.1109/ICICV68925.2026.11554712.

[17] R. Laheri, "Self-Healing Infrastructure: Leveraging Reinforcement Learning for Autonomous Cloud Recovery and Enhanced Resilience," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 49s, pp. 352–357, May 2025, doi: 10.52783/jisem.v10i49s.9888.

[18] A. D. Vibhute and V. Nakum, "Deep learning-based network anomaly detection and classification in an imbalanced cloud environment," *Procedia Comput. Sci.*, vol. 232, pp. 1636–1645, 2024, doi: 10.1016/j.procs.2024.01.161.

[19] Y.-Y. Chen, S.-Y. Jhong, S.-K. Tu, Y.-H. Lin, and Y.-C. Wu, "Autonomous Smart-Edge Fault Diagnostics via Edge-Cloud-Orchestrated Collaborative Computing for Infrared Electrical Equipment Images," *IEEE Sens. J.*, vol. 24, no. 15, pp. 24630–24648, 2024, doi: 10.1109/JSEN.2024.3415639.

[20] T. Luan and M. A. E. Damian, "Research on Improved Grey Wolf Optimization Algorithm Based on Random Walk in Task Scheduling of Cloud Computing Platform," in *Proceedings - 2023 7th International Symposium on Computer Science and Intelligent Control, ISCSIC 2023*, 2023. doi: 10.1109/ISCSIC60498.2023.00059.

[21] W. Wu, "Fault Detection and Prediction in Models: Optimizing Resource Usage in Cloud Infrastructure," in *Proceedings of the 2025 4th International Conference on Intelligent Systems, Communications and Computer Networks*, New York, NY, USA, NY, USA: ACM, Feb. 2025, pp. 131–138. doi: 10.1145/3732945.3732964.

[22] J. Sekar, "AI-powered fault detection and mitigation in cloud computing infrastructures," *World J. Adv. Res. Rev.*, vol. 18, no. 03, pp. 1600–1612, 2023, doi: 10.30574/wjarr.2023.18.3.0235.

[23] N. Chen, Q. Zhong, Y. Liu, W. Liu, L. Bai, and L. Hu, "Container cascade fault detection based on spatial–temporal correlation in cloud environment," *J. Cloud Comput.*, vol. 12, no. 1, p. 59, Apr. 2023, doi: 10.1186/s13677-023-00438-2.

[24] I. Ortiz-Garces, W. Villegas-Ch, and S. Luján-Mora, "Implementation of edge AI for early fault detection in IoT networks: evaluation of performance and scalability in complex applications," *Discov. Internet Things*, vol. 5, no. 1, p. 108, Oct. 2025, doi: 10.1007/s43926-025-00196-4.