

Fire Detection With Erroneous Input Correction

ECGR 4105/5105 - Introduction to Machine Learning

1st Jaskin Kabir
ECGR Student
UNC Charlotte
Charlotte, NC
jkabir@charlotte.edu

2nd Axel Leon Vasquez
ECGR Student
UNC Charlotte
Charlotte, NC
aleonvas@charlotte.edu

3rd Matthew Anderson
ECGR Student
UNC Charlotte
Charlotte, NC
mande137@charlotte.edu

Abstract—The accuracy of typical commercially available fire alarm systems is often limited due to their reliance on single sensors with constrained detection capabilities. Moreover, a sensor failure renders the entire system ineffective, posing a critical risk. While AI-based fire detectors have been shown to massively outperform traditional systems, little work has addressed handling sensor failures. To address this challenge, we propose AlarmNet, a robust fire detection system that incorporates erroneous sensor data during training.

Errors were randomly into the dataset and imputation was applied to replace missing or faulty sensor values with the median of corresponding features from the training data. When faced with a dataset with 24.2% of the measurements missing, our 3-layer Artificial Neural Network was able to achieve a remarkable accuracy of 98.38% and correctly detected 99.73% of fires. Compared to a model trained on the full dataset, this marks a reduction in fire detection rate, or recall score, of only 0.25% but a whopping 41.51% improvement over a simulated commercial-grade fire alarm system.

Index Terms—machine learning (ML), algorithm, model. Artificial Neural Network, Imputation, Sensor fusion, Fire Alarms, Fire Detection

I. GITHUB REPOSITORY:

https://github.com/jaskinkabir/Intro_ML_Project

II. INTRODUCTION

According to a 2013 study, the Honeywell FS90, a widely used commercial fire detection system, only achieves an accuracy of 87.5% [1]. This contributes to deaths, injuries, and damage due to fires in two ways: Firstly when a fire occurs and the fire detection system fails to recognize the fire, the occupants of the building can only react to the fire

once they are close enough to it to be able to observe it, which often means it is too late to evacuate. Secondly, like the villagers who did not believe the boy who cried wolf, a 1995 report stated that fewer than 25% of residents of mid-rise residential buildings interpreted the sound of the fire alarm as a potential indication of a real emergency [2]. In recent years, machine learning has proven to be a robust solution to various data problems. Machine Learning is particularly apt for analyzing large datasets and creating predictive models using advanced statistical models. These statistical models allow the program to determine which factors or data points are most relevant to a certain outcome

and make a prediction based on the data it has been trained on.

There are three main approaches to Machine Learning: supervised learning, unsupervised learning, and reinforcement learning. Each has its strengths and weaknesses. Our project uses supervised learning using the Sklearn library in Python. Supervised learning is defined by presenting example inputs and desired outputs. The goal is to teach the model a general rule that maps these inputs and outputs and allows for reliable prediction.

However, more importantly, our project uses a Neural Network to make predictions.

III. EASE OF USE

A. Maintaining the Integrity of the Specifications

In machine learning, models are used to solve robust problems with a variety of datasets combined with the topic 'Internet of Things', the development of a fire detection model offers significant ease of integrating existing fire alarms without any extensive hardware modifications. Sensor data is commonly available in modern times, such as stores, schools, offices, and industrial facilities. These erroneous input models can minimize the setup costs. Furthermore, the approach for continuous improvement as additional data becomes available ensures the adaptability to various environments in fire scenarios. The design model ensures both residential and industrial users benefit from fire detection without compromising reliability and simplicity.

IV. MOTIVATION

Sensors are one of the most crucial aspects of modern-day technology which helps improve daily lives most particularly fire detection. It plays a critical role in public safety such as industrial, residential, and public places. Traditional systems during effect, often suffer from false positives and false negatives which leads to unnecessary evacuations or worse, failure to alert real danger. These unnecessary issues underscore the need for in-depth research of advanced, accurate, and reliable solutions. This project builds a bridge by developing a machine-learning model capable of predicting fire events based on air condition data. Stefan Blattmann's Smoke Detection Dataset contains diverse

environmental fire events which intend to create a system that improves accuracy but also handles sensor errors. The potential impact of the model can extend beyond the immediate technical systems. Enhanced fire detection systems can save lives, reduce damages, and prevent disruptions caused by false alarms. This design applicability from household safety to the industrial system level highlights the value of safety technology.

A. Dataset and Preprocessing

The dataset used in this project is the Smoke Detection Dataset provided by Stefan Blattmann. The dataset consists of approximately 62,629 sensor readings encompassing both normal and fire-related scenarios, providing diverse training and evaluating fire detection models.

B. Key Features

- **Data Features:** Environmental features such as temperature, humidity, and pressure, with particulate matter readings such as PM1.0, PM2.5, NC0.5, NC1.0, and NC2.5 measuring air quality sensors.
- **Gas Sensor Outputs:** TVOC (Total Volatile Organic Compounds) indicates the presence of volatile organic compounds in combustion. eCO2 contains high CO2 concentration often indicating fire activities. Raw H2 and Raw Ethanol are outputs from chemical sensors detecting hydrogen and ethanol which are released during certain types of fires.
- **Handling Sensor Errors:** Simulates sensor malfunctions to evaluate error-handling capabilities. Including thresholds to identify invalid readings and imputed missing values within the median or mean imputation.
- **Correlation:** Analysis to rank features by importance reduction from 12 features to 4 key features, humidity, Raw Ethanol, Pressure, and TVOC. Additionally, PM0.5 for sensor redundancy, resulting in a 5-feature model.

C. Feature Selection

The smoke detection system was crucial in developing an efficient and robust model. Starting with 15 initial features, only 12 were usable features, after removing non-informative columns such as UTC Timestamps, CNT, and Unnamed :0. Optimizing the model's performance and reducing computational complexity, tests were conducted such as correlation analysis, ranking features based on their absolute correlation with 'Fire Alarm' target variables. This test led to a significant reduction in the feature set in a compact yet highly effective 5-feature model that maintained excellent performance within sensor redundancy and avoided overfitting.

- **Correlation Analysis Methodology:** it utilized absolute correlation values with 'Fire Alarm' as the target variable. It uses ranking features based on the strength to get rid of the non-use features, such as TVOC, PM2.5, and NC1.0 show strong correlations with the presence of fire.

- **Dimensionality Reduction:** The feature set from 12 to 4 key features significantly lowers model complexity. It also demonstrated the trade-off between performance and simplicity, by removing 8 features with minimal impact.
- **Sensor Redundancy:** Adding the PM0.5 as the fifth feature represented all four sensors which explains the importance of maintaining sensor diversity for robustness thus approaching the creation of a balance model capturing key patterns while avoiding overfitting.

D. Model Architecture

The model implemented a custom deep neural network (AlarmNet) for fire detection. The dataset's multivariate sensor data balance complexity and efficiency. Input layer size depends on the number of features such as 12 features as the initial full-feature model. It was reduced to 4 features via a correlation ranking system. Hidden layers were added with (62,32) for a 12-feature model and a 5-feature model.

The 4-feature system uses (64,64) to provide additional capacity. All these models activated functions with ReLU(Rectified Linear Unit) introducing non-linearity and mitigating the vanishing gradient problem. The training model algorithm implemented the Adam Optimizer technique for its adaptive learning rate abilities with an initial learning rate of 1e-3 for most models and 1e-5 for the imputed error model. Parameters are updated using gradient descent with momentum adjustment for faster convergence. The learning rate is based on validation loss improvements, configured with a factor of 0.5 to halve the learning rate when performed. Having a patience of 4000 iterations without improvement and a threshold of 1e-3 for significant improvements. In the training configuration, 3000 epoch models were extended to 16,000 for the imputed error model to ensure convergence. Using epoch will improve based on the device GPU accelerated training on CUDA when available, otherwise CPU.

E. Training and Evaluation

- **Data Splitting:** The dataset is split into 80/20 percent for training/testing to evaluate model performance.
- **Hardware:** A code was implemented to perform on GPU(CUDA) for acceleration; otherwise CPU is available.
- **Training Configuration:** Adam optimizer was implemented to configure with learning rates of 1e-3 for most models and 1e-5 for the imputed error model. Learning rate by a factor of 0.5, the patience of 4000 epochs, and threshold 1e-3. In most models, 3000 epochs were implemented while 16,000 epochs were for the imputed error model to ensure proper convergence on data.
- **Precision:** Proportion of correctly identified fire alarms

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

- Recall: Measures the proportion of actual fire cases detected by the model to minimize false negatives

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

- F1-Score and Accuracy: Implements harmonic mean of precision and recall with overall correctness of the model.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

- FPR: False Positive Rate measures the actual negatives that are incorrectly classified as positives

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (4)$$

- Confusion Matrix: This model visualizes true positive/negative and false positive/negatives

F. Error Handling and Robustness

In the real world sensor data, is often due to hardware communication limitations and errors. It can lead to incomplete or noisy environmental factors. The reliability of the model can go in such conditions in error handling using robust strategies to incorporate into the model and training process. These strategies allowed the model to maintain high accuracy and recall during the sensor error process. In simulating real-world scenarios, errors were intentionally introduced into the dataset. In the model run, 20 percent was applied across all features, with particulate matter (PM) sensors likely to encounter errors due to insufficient redundancy. The dataset showed four sensors in their categories, temperature and Humidity, Pressure, (TVOC, CO₂, Ethanol, H₂), and Particulate Matter (NC0.5, NC1.0, NC2.5, PM1.0, PM2.5). erroneous values were represented as missing data (NaN) or extreme outliers to emulate hardware failures or any communication drops. Missing values were handled using median imputation, which replaces erroneous values with the median of the corresponding feature calculated from the training dataset. The median was chosen against outliers suitable for skew distributions in sensor data.

Imputation implementation ensures the dataset remains intact, allowing it to learn effectively. The feature was further improved by identifying the most correlated features with fire detection. Key features such as TVOC, Pressure, Raw Ethanol, and Humidity were retained. The training process specialized adjustments for handling noisy and incomplete data. During the deep ANN error handling architecture of [(256,256,256)] for better resolution. Extending training epochs were used, enabling the model to generalize effectively. In the error handling stage, errors were introduced to replicate real-world conditions challenging the robustness of the architecture. The implementation strategy introduced erroneous or missing values were replaced with the median of each feature to maintain data integrity. The imputation approach was particularly effective for skewed feature distributions which enchanted the model [(256,256,256)] and demonstrated resilience to noisy and

incomplete data. The imputed error model showcased its ability to handle a 20 percent sensor error rate with minimal performance degradation and showcasing minimized false negatives under challenging conditions. Performance metrics were involved in all models to evaluate using recall, precision, F1-Score, and confusion matrices. The model was benchmarked against the Honeywell FS90 fire detection system. Comparison revealed superior recall and resilience in the proposed model, under simulated noisy conditions, such as histograms of missing values and bar charts comparing recall and recall without errors showing the model's robustness. Furthermore, the size and fast inference of the final model exported in ONNX format ensure the deployment in real-time IoT scenarios of fire detection systems. These error-handling strategies demonstrate the model's ability to adapt to real-world challenges for a reliable solution in fire detection in different environments. Simulating error handling, imputation, and targeted features in correlation highlights the model's resilience and practical applications.

G. Comparative Analysis

In a comparative analysis, SVM(Support Vector Machine) and NN (Neural Network) were both considered for fire detection within the 12-feature test bench. Neural networks are ultimately chosen for their superior flexibility and robustness. SVMs are known for their simplicity and effectiveness within smaller datasets due to their ability to model linear and non-linear relationships using kernel functions. However, in this dataset, the project had a large size of 62,000 samples which limited the SVM training model as memory requirement scale quadratically with the number of samples. On the other hand neural network demonstrates superior performance and adaptability. They efficiently scaled with large datasets by leveraging batch size gradient descent and GPU acceleration with mixed precision and gradient scaling. It can learn hierarchical features representing them as particularity effective for multi-sensor data its architecture model is tailored to handle noisy and missing data, as seen in the imputed error model under results., It can maintain high recall and precision under a 20 percent error rate. The full feature model utilizes all 12 -features that achieved high precision and recall but its computational complexity made it less efficient. After using a correlation technique ranking them by target variable it reduced to 4 and 5 feature sets retaining only the top four features (Humidity, Raw Ethanol, Pressure, and TVOC), achieved performance to the full feature model with minimal precision loss of 0.1 percent. It significantly improved making it more suitable for different environments. The redundant feature model added the NC0.5 enchanting redundancy and improved robustness maintaining efficiency. The imputed error model designed to handle noisy and missing data shows a resilience with 20 percent error rate. Thus outperforming the Honeywell FS90 system. In addition to accuracy and recall, time was evaluated to determine real-time applicability. The proposed models achieve an

average time of 2e-2 seconds per sample making them viable for deployment in fire detection systems. Evaluation metrics showed visual comparison including confusion matrices, and bar charts illustrating the superior performance. The reduced features and imputed error and aggregation demonstrated the ability to balance accuracy, efficiency, and robustness for real-world fire detection applications. The Federated model trains the local copy of a global model using local data. The training sends model weights (weights of the neurons in the neural network) to the global mode. The federate and AlarmNet achieve near identical accuracy. These bar graphs show the comparison of recall with and without error and precision with 20 percent error and without. With Machine Learning algorithms, it was able to provide excellent validation against Honeywell FS90 highlighting the potential ANN that can provide for enchanting better real-world safety protocols (fig 18 and 19).

H. Conclusion

This machine learning model successfully demonstrated the development an efficient fire detection system using sensor data. By leveraging a custom neural network architecture, the system achieved high precision and recall even with real-world scenarios such as noisy or missing sensor data. Feature selections and target preparation, including median imputation and error handling, ensure the model’s exploration. The reduced feature model maintained comparable performance to the 12-feature model significantly reducing computational requirements and making it good for constrained environments. Comparative analysis highlights the proposed neural network over the traditional such as SVM. The neural network can handle large datasets, capture complex features tolerate sensor errors underscores the effects in IoT-base fire detection systems. Optimizing time and compact model size ensure viability for read-time devices. The project object results emphasize the importance of designing models that are balanced, efficient, accurate, and robust for safety and critical protocol applications. Future work can be explored further optimizing the learning, advancing imputation, and improving the federated model weights between three clients to enhance system performance. The project demonstrates the potential for machine learning to revolutionize fire detection technology to challenge high-end companies such as Honeywell to ensure the power of ANN to contribute to improving safety and reliability in diverse environments.

I. Contributions

Matthew Anderson: Worked on Federated Learn Class

Jaskin Kabir: Worked on AlarmNet class that contains Neural Network Model

Axel Leon Vasquez: Wrote Presentation/Report, worked on debugging and visualization

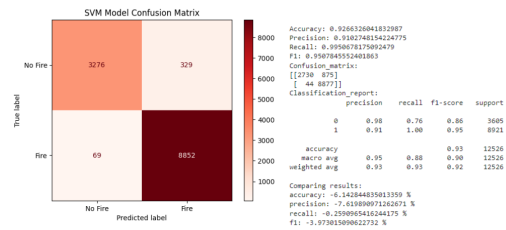


Fig. 1. SVM Metric

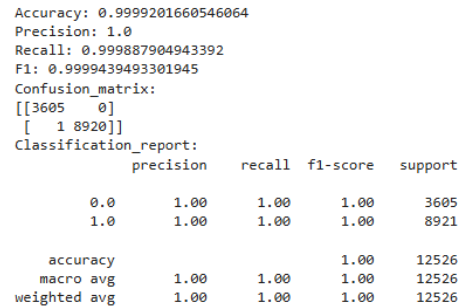


Fig. 2. 12 - Feature Metrics

J. References

REFERENCES

- [1] N. k. Fong, "Investigation of the performance and improvement of optical smoke detectors," *Procedia Engineering*, vol. 62, 12 2013.
- [2] G. Proulx, J. C. Latour, J. W. MacLaurin, J. Pineau, L. E. Hoffman, and C. Laroche, "Housing evacuation of mixed abilities occupants in highrise buildings," National Research Council of Canada. Institute for Research in Construction, Tech. Rep., Aug 1995. [Online]. Available: <https://doi.org/10.4224/20375582>

K. Results

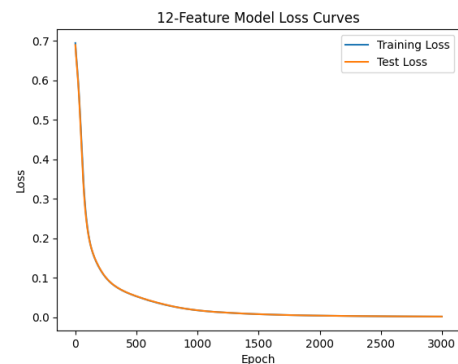


Fig. 3. 12 - Feature Loss Curve

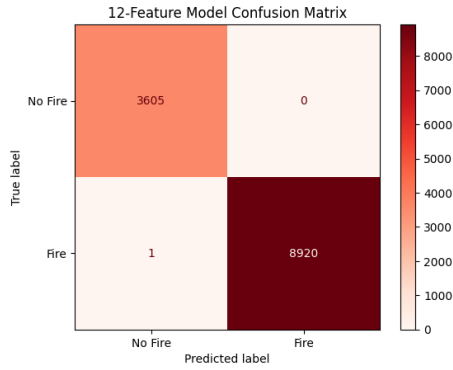


Fig. 4. 12 - Feature Confusion Matrix

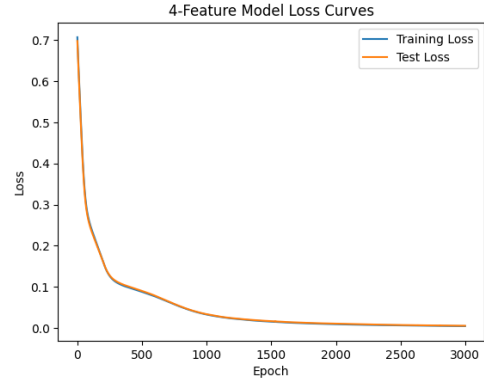


Fig. 7. 4 - Feature Loss Curves

```
Humidity[%]      0.399846
Raw Ethanol      0.340652
Pressure[hPa]    0.249797
TVOC[ppb]       0.214743
Temperature[C]   0.163902
NC0.5            0.128118
PM1.0            0.110552
Raw H2           0.107007
eCO2[ppm]       0.097006
PM2.5           0.084916
NC1.0            0.082828
NC2.5           0.057707
Name: Fire Alarm, dtype: float64
```

Fig. 5. Features

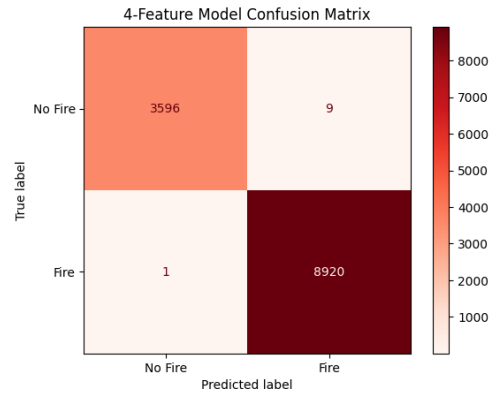


Fig. 8. 4 - Feature Confusion Matrix

```
Accuracy: 0.9992016605460642
Precision: 0.9989920483816777
Recall: 0.999887904943392
F1: 0.9994397759103641
Confusion_matrix:
[[3596  9]
 [ 1 8920]]
Classification_report:
      precision    recall  f1-score   support

     0.0         0.98      0.93      0.95        3605
     1.0         0.97      0.99      0.98        8921

   accuracy          0.97          0.97          0.97        12526
  macro avg          0.97          0.96          0.97        12526
 weighted avg          0.97          0.97          0.97        12526

Comparing results:
accuracy: -0.0719079578139993 %
precision: -0.10089686098654951 %
recall: 0.0 %
f1: -0.05044560282495212 %
```

Fig. 6. 4 - Feature Metrics

```
Accuracy: 0.9998403321092129
Precision: 1.0
Recall: 0.9997758098867839
F1: 0.9998878923766816
Confusion_matrix:
[[3605  0]
 [ 2 8919]]
Classification_report:
      precision    recall  f1-score   support

     0.0         1.00      1.00      1.00        3605
     1.0         1.00      1.00      1.00        8921

   accuracy          1.00          1.00          1.00        12526
  macro avg          1.00          1.00          1.00        12526
 weighted avg          1.00          1.00          1.00        12526

Comparing results:
accuracy: 0.06387735547748555 %
precision: 0.10079516183223447 %
recall: -0.011212019284675844 %
f1: 0.044816670922210436 %
```

Fig. 9. 5 - Feature Metrics

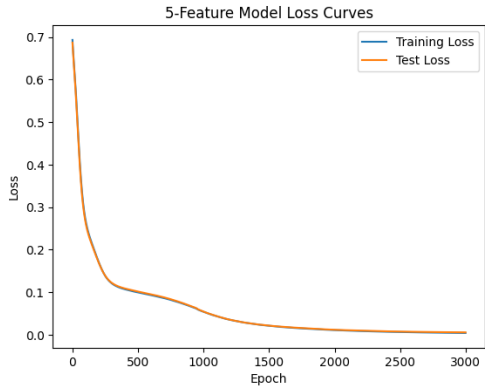


Fig. 10. 5 - Feature Loss Curves

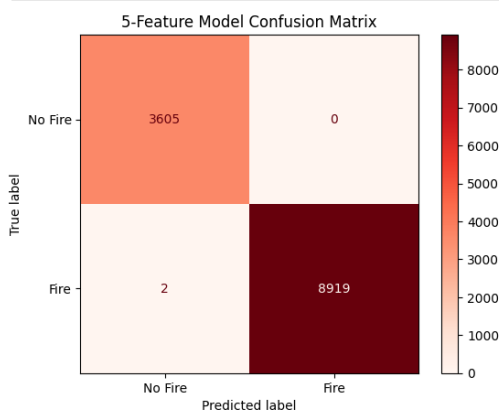


Fig. 11. 5 - Feature Confusion Matrix

```

Accuracy: 0.9835542072489223
Precision: 0.9796367638965328
Recall: 0.9976460038112319
F1: 0.9885593690991892
Confusion_matrix:
[[3420 185]
 [ 21 8900]]
Classification_report:
      precision    recall  f1-score   support

     0.0         0.98         0.97         0.98         3605
     1.0         0.99         0.99         0.99         8921

 accuracy         0.98         0.97         0.98         12526
 macro avg         0.99         0.97         0.98         12526
 weighted avg         0.98         0.98         0.98         12526

Comparing results:
accuracy: -1.6558441558441537 %
precision: -2.078651685393257 %
recall: -0.213483146067411 %
f1: -1.1459678659747208 %

```

Fig. 12. Imputation Metrics

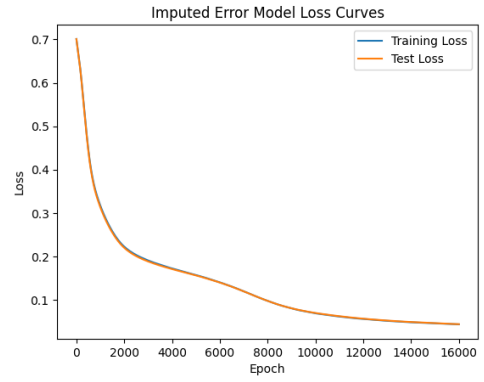


Fig. 13. Imputation Loss Curves

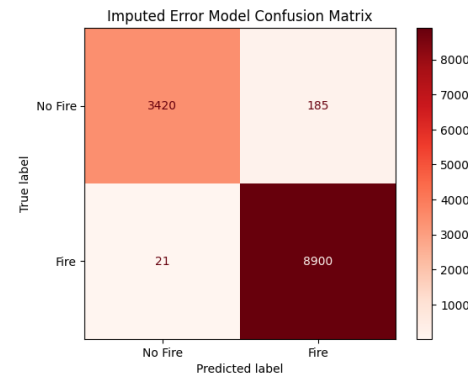


Fig. 14. Imputation Confusion Matrix

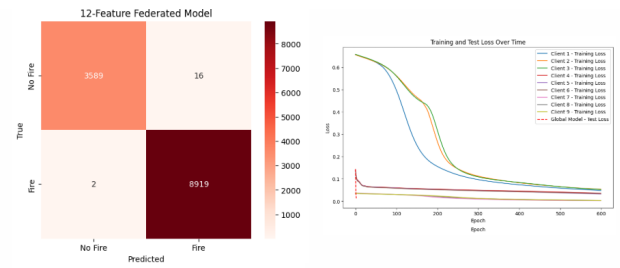


Fig. 15. Federated Learning: 12-Features

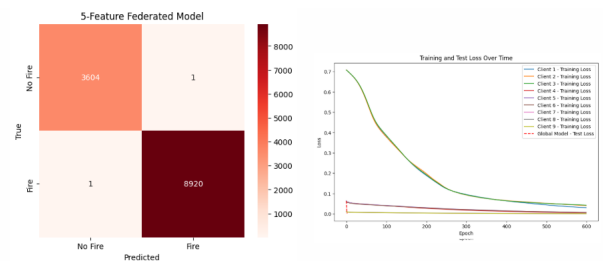


Fig. 16. Federated Learning: 5-Features

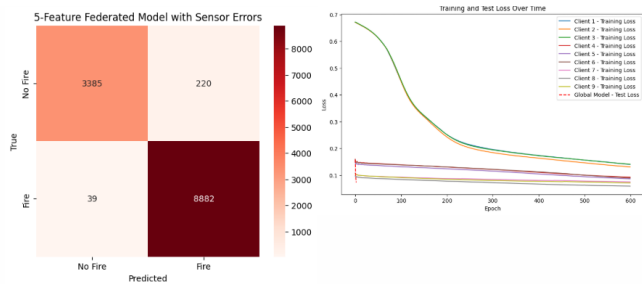


Fig. 17. Federated Learning: Error handling

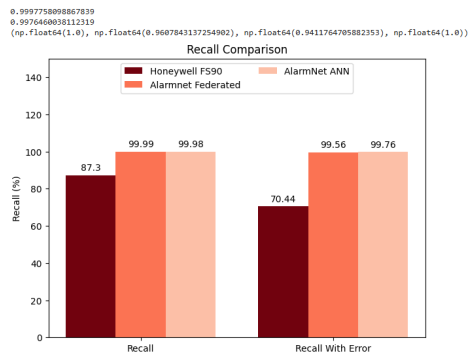


Fig. 18. Recall Comparison Honeywell FS90 Vs AlarmNet ANN Vs Federated

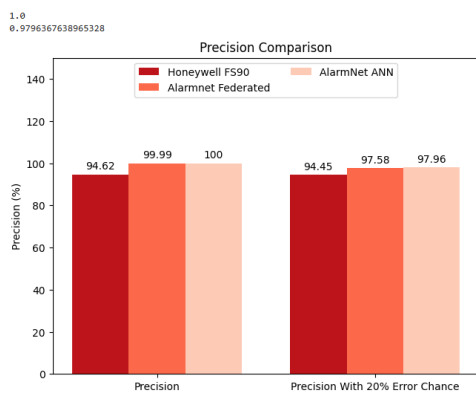


Fig. 19. Precision Comparison Honeywell FS90 Vs AlarmNet ANN Vs Federated