A Comparative Study in Predicting Smoke Detection Through Multilayer Perceptrons and Support Vector Machines

Nicholas Tsioras

Nicholas.Tsioras@city.ac.uk

Abstract

This paper aims to present a critical comparison, and evaluation of two NECO methods in a binary classification task. The two methods being compared are Multilayer Perceptron (MLP) and Support Vector Machines and the task is to classify instances of fire alarms going on through smoke detection. Various models were trained and validated with different hyperparameters in a search grid manner and finally, the best models of each method were further tested. The tested results of the best models (one for MLP and one for SVM) were evaluated and compared with the use of Confusion Matrices and Receiver Operation Curves (ROC).

1. Introduction – Motivation of the Problem

The spread of fire is a serious issue that can cause a lot of harm to people and damage local residences. The National Fire Protection Association (NFPA) published a report in 2021, in which they mention that in the US the average death rates per million population ranges from 4.7 to 23.7 with the average being around 10 [6]. Also, the current trend from 2014/15 in the number of incidents attended by fire in the UK has been one of increase. The number of staff working in the fire services has been significantly reduced during the 2010s, with the total number of workers falling by around 10,000 between 2008 and 2018. Also, the International Association of Fire and Rescue Services (CTIF) [1] published a report in 2022, in which it mentioned that the fire brigades served almost 70 million missions, from which the 4 million were on fires, across 48 countries of the world where 3.3 billion people live. These measures and statistics show that fire is a common, serious issue across the globe which can lead to many deaths and injuries. In such cases, being able to use technology to sensor fires at an early stage can be crucial for several reasons related to saving lives, minimizing damage and fire spread, and ensuring quick reactions.

2. Summary of the Two NECO Methods

2.1 Multilayer Perceptron (MLP)

The Multilayer Perceptron, also known as Artificial Neural Network (ANN) is one of the networks that represents artificial human neural networks. According to [2], *«ANN is a system that can solve problems by changing the structure owned by external and internal information adaptively»*. For this project, the Backpropagation Neural Network is being used, which is a feedforward neural network with its main characteristics being forward signal transmission and error backpropagation. In the forward signal transmission, the input features are being processed layer by layer from the input layer, to each one of the multiple hidden layers and finally to the output layer, with a non-linear activation function. Backpropagation is then used if the output layer does not have the desired output. During backpropagation, the network adjusts its weights and thresholds according to the loss error of the predictions [16]. In this way, the predicted output of the model is improving and getting closer to the desired output.

2.2 Support Vector Machines (SVM)

Support Vector Machines are mainly used for pattern classification and non-linear regression. The main principle of SVM is structural risk minimization and the main idea is to establish a classification hyperplane that can maximize the isolated edge between positive and negative instances [16]. In other words, the concept of SVM is considered *«an attempt to find the best line (hyperplane) contained in the input space and serves to separate the two classes»* [2]. There have been many studies that have used SVM to classify objects into one of the known categories (classes) of the target variable that we want to predict. So, when new data is being classified, the SVM checks on which side of the hyperplane separator every instance belongs to and hence to which target class it should be assigned to.

Sometimes though, the input features are not linearly separable, meaning that there is no straight line (hyperplane) that can perfectly separate the data belonging to different classes. To address this issue, different kernel functions can be used to transform and map the input space of features into higher-dimensional feature spaces where the data can be easily separated [8].

2.3 Advantages and Disadvantages of Both NECO Methods and Hypothesis Statements

According to [9], the main advantage of MLPs is that they are extremely flexible and support more different data types and larger datasets. Also, they are great at learning through training which of the input features are more important and drive predictions. On the other hand, [9] also mentions that SVMs have the advantage of needing fewer hyperparameters as they require less grid-searching for getting accurate models. To have a better understanding of how MLPs and SVMs compare, an SVM without a kernel tends to perform as well as a single MLP. The advantage SVMs have over MLPs is mainly due to the kernel function, especially the non-linear (RBF) kernel which makes it comparable with a two or three hidden layer MLP [9]. In most studies where the two methods are compared for binary classification tasks, SVMs usually perform better than MLPs with the use of the RBF function [16].

So based on the structures and the pros and cons of each method mentioned, the following hypothesis statements were made before training, validating, and testing the models:

- For this project an MLP with three hidden layers was created. It is expected to perform comparably with the RBF SVM, as suggested by the literature, but can also outperform it due to the ability of MLPs with multiple hidden layers to capture deeper non-linear patterns and interactions between the input features using activation functions.

- RBF SVMs are expected to have higher accuracy than Linear SVMs due to the ability they have to separate the data classes in higher dimensional spaces with multiple input features.
- For large-scale datasets, training time for SVMs with the RBF kernel may become significantly longer than for MLPs, due to the complexity of the calculations for the RBF kernel, potentially making MLPs a more time-efficient method for larger datasets.

3. Initial Analysis of the Dataset and Exploratory Data Analysis

The dataset used to perform an analysis and comparison of MLPs and SVMs is the smoke detection dataset which was obtained from Kaggle [12]. The dataset contains a total of 62630 rows and 15 attributes. The target variable that we aim to predict is binary and it is named 'Fire Alarm' which indicates that the fire alarm goes on for a value equal to 1 and the fire alarm stays off for value equal to 0.

As for the target variable that we want to predict, we observe in Figure 1 that the classes are significantly imbalanced. This suggests that SMOTE should be applied to generate synthetic samples of the minority class. This will make the dataset more balanced, and it ensures that models are not biased and don't overfit to the majority class [3].

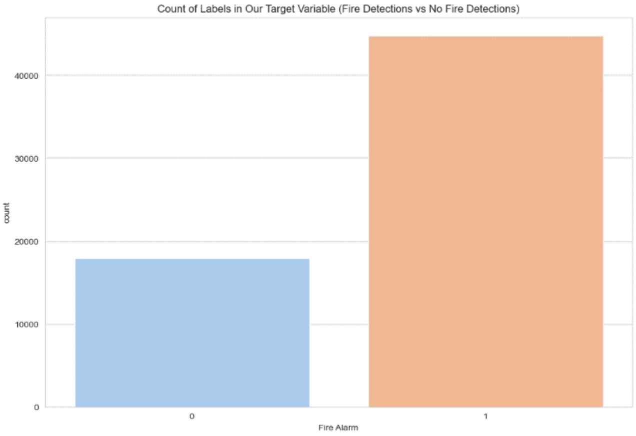


Figure 1. Smoke Detection vs No Smoke Detection Instances

4. Methods

4.1 Training and Evaluating Methodology

The methodology consists of splitting the data into separate sets for training (80%), validating (10%) and testing (10%) the models. The training data was used to train all of the models, the validation data was used to compare the training and validation loss to make sure the models are not overfitting to the training data, and finally the 2 best models (one for MLP and one for SVM) were further tested on the test set to see how well they generalize and make predictions to new, unseen data. The training and validation sets were used in a simple grid search, which is based on searching all possible combinations of different hyperparameters and evaluating them on the validation set after being trained [3]. Before tuning the parameters, SMOTE was applied on the train set to have an equal representation of both binary outcomes. The results of SVM and MLP were consistently good, hence this is the reason that this approach was also implemented in this paper. Also in [10], the accuracy was compared between the hyperparameter tuning (grid search) method and the k-fold cross-validation, where the first approach clearly outperformed the second for SVMs.

In the case of MLP, early stopping was applied to monitor the performance of each model. According to [5], early stopping can be used to specify a large amount of training epochs without worrying about overfitting, since the model stops training once it stops improving performance after a certain number of epochs. Early stopping is very useful when using a hold out validation grid search approach. Also, momentum was used to accelerate the convergence of gradient descent alongside different learning rates. Finally, weight decay was also applied to penalize the loss function and maintain smaller weights [14].

4.2 Architecture and Parameters Used for Both NECO Methods

For the MLP, a *''Backpropagation Neural Network (BPNN)''* was used to adjust the network weights and thresholds according to the loss error of predictions [16]. The binary cross entropy with logits loss was used alongside the Adam optimizer to minimize the loss function during backpropagation, which according to [7] are a good combination for binary classification tasks. The sigmoid activation function was used in the outcome to transform it into probabilities of belonging to each class, making it useful for binary classification problems [15]. A threshold of 0.5 was set so probabilities over 0.5 belong to class 1 and below 0.5 to class 0. In terms of specifying the parameters and hyperparameters for grid search, the patience for early stopping was 10 meaning that if the validation loss did not improve over 10 consecutive epochs, the model should stop training to avoid overfitting to the training data. For momentum, a value of 0.9 was selected and for weight decay, a low value of 0.0001, since according to [13], high values of momentum between 0.9-0.99 are appropriate for the acceleration of the convergence process and low values of weight decay are better to prevent overfitting.

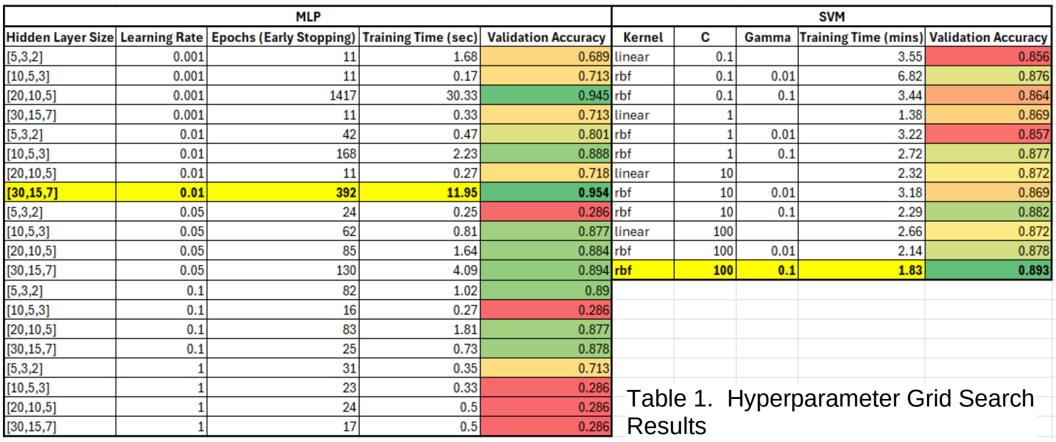
For the hyperparameter tuning, various learning rates were tested between 0.001 and 1 but also different numbers of neurons in each of the 3 hidden layers. The choice of the total neurons for each hidden layer was not arbitrary as we made sure they were not far off from the 7 input features that were used. These hyperparameters were tuned alongside the other parameters mentioned with grid search to find the optimal ones.

For the SVM, the Linear and RBF kernels were compared. For the Linear kernel, the tuned hyperparameter was the penalty weight C. Different C values were validated between 0.1 and 100 to test different penalize values on misclassifications [8]. For the RBF kernel, the Sigma parameter was tuned jointly with the C parameter. This method is applied, for instance, by Patil et al. [10] to optimize performance.

5. Analysis and Critical Evaluation of the Results

5.1 Selecting the Best Models

To make the evaluations comparable, both models were tested on the same test set. Table 1 shows the hyperparameter grid search process and finding the best models for both MLP and SVM. The best models are highlighted in yellow. What stands out is that the validation accuracy for the MLPs ranged largely between 0.28-0.95. On the other hand, the differences between the Linear and RBF models were not major, but overall RBF kernels performed better due to their ability to separate data classes in higher dimensional spaces, as indicated by the second hypothesis statement. Another major difference between MLPs and SVMs is the training time. The total training time for all MLP models was only around 1 minute and for all SVM models was around an hour. This is mainly because of the early stopping criteria in MLPs which was applied after a few epochs in most models. Also, the third hypothesis statement seems to be true due to the complexity of calculations in larger datasets, especially for RBF SVMs. Overall, some MLPs have higher accuracy than the SVMs, due to the ability they have to capture deeper non-linear patterns with the use of activation functions. This confirms that the first hypothesis statement is also true.



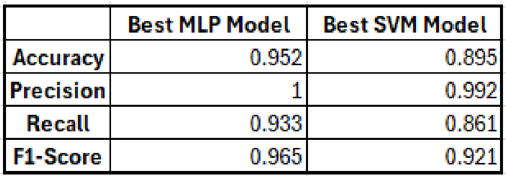
MLP					SVM				
Hidden Layer Size	**Learning Rate**	**Epochs (Early Stopping)**	**Training Time (sec)**	**Validation Accuracy**	**Kernel**	**C**	**Gamma**	**Training Time (mins)**	**Validation Accuracy**
[5,3,2]	0.001	11	1.68	0.689	linear	0.1		3.55	0.856
[10,5,3]	0.001	11	0.17	0.713	rbf	0.1	0.01	6.82	0.876
[20,10,5]	0.001	1417	30.33	0.945	rbf	0.1	0.1	3.44	0.864
[30,15,7]	0.001	11	0.33	0.713	linear	1		1.38	0.869
[5,3,2]	0.01	42	0.47	0.801	rbf	1	0.01	3.22	0.857
[10,5,3]	0.01	168	2.23	0.888	rbf	1	0.1	2.72	0.877
[20,10,5]	0.01	11	0.27	0.718	linear	10		2.32	0.872
[30,15,7]	**0.01**	**392**	**11.95**	**0.954**	rbf	10	0.01	3.18	0.869
[5,3,2]	0.05	24	0.25	0.286	rbf	10	0.1	2.29	0.882
[10,5,3]	0.05	62	0.81	0.877	linear	100		2.66	0.872
[20,10,5]	0.05	85	1.64	0.884	rbf	100	0.01	2.14	0.878
[30,15,7]	0.05	130	4.09	0.894	**rbf**	**100**	**0.1**	**1.83**	**0.893**
[5,3,2]	0.1	82	1.02	0.89					
[10,5,3]	0.1	16	0.27	0.286					
[20,10,5]	0.1	83	1.81	0.877					
[30,15,7]	0.1	25	0.73	0.878					
[5,3,2]	1	31	0.35	0.713					
[10,5,3]	1	23	0.33	0.286					
[20,10,5]	1	24	0.5	0.286					
[30,15,7]	1	17	0.5	0.286					

Table 1. Hyperparameter Grid Search Results

5.2 Comparing the Best MLP and SVM Models

Figures 2, 3, and Table 2 indicate that the MLP model outperforms the SVM model across all metrics. Although both models are precise in predicting a positive class, the MLP model is more accurate, captures a higher proportion of the relevant cases, and has a better balance between recall and precision. The testing and validation accuracy are very similar for both models (Around 95% for MLP and 89% for SVM), which shows consistency in generating predictions from new, unseen data. This shows that both models are not overfitting on the training data.

Figures 4 and 5 show the comparison of the AUC scores and ROC curves of both models. Although both models perform well, the MLP model seems to demonstrate slightly superior performance in the ability to classify correctly and balance the sensitivity and specificity [4]. Although the AUC scores for both models are relatively the same, there should be one which is preferable. In the context of detecting smoke in an area, we aim for a model that minimizes the risk of not detecting smoke when there is. In



	Best MLP Model	Best SVM Model
Accuracy	0.952	0.895
Precision	1	0.992
Recall	0.933	0.861
F1-Score	0.965	0.921

Table 2. Performance Metrics of the Two Best Models

other words, we should prefer a model that has the lowest false negative rate or the highest true positive rate. Examining the Confusion Matrices in Figures 2 and 3, and the ROC curves in Figures 4 and 5, it is evident that the MLP has fewer false negative predictions and a higher true positive rate compared to the SVM. Therefore this indicates that the MLP is a better method to use for this kind of problem.

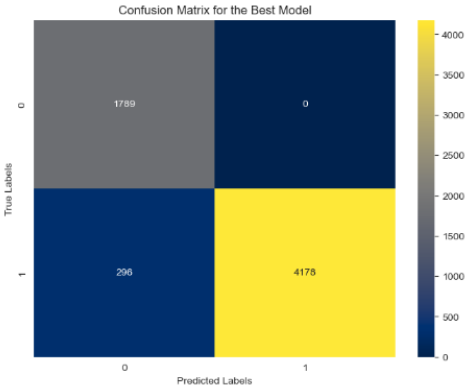


Figure 2. Confusion Matrix for Best MLP Model

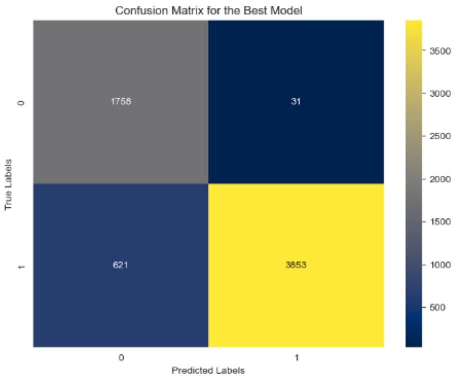


Figure 3. Confusion Matrix for Best SVM Model

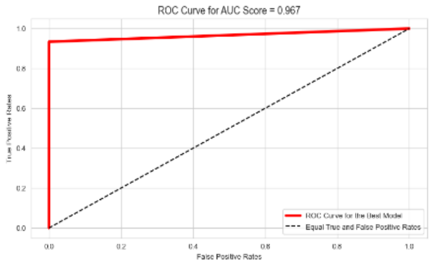


Figure 4. ROC Curve for Best MLP Model

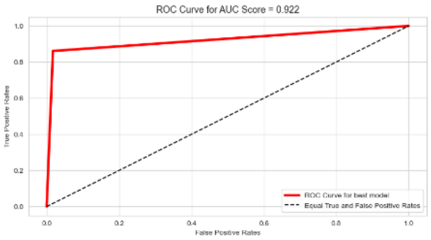


Figure 5. ROC Curve for Best SVM Model

6. Conclusions, Lessons Learned and Future Work

The study reviews how accurately two pre-trained models, the best MLP and best SVM models can predict whether an unseen data is able to detect smoke so the fire alarm goes either on or off. Overall, both models seem to be competent, but given the main metrics, Confusion Matrices, and ROC Curves, if we were to choose between the two models based on the ability to distinguish between the classes, the MLP would be the better choice. Both models seem to perform consistently well in the validation and test sets, showing that they are not overfitting to the training data.

We learned that adjusting the hyperparameters provides insights into model training and fine-tuning. We also discovered that MLPs are more sensitive to hyperparameter adjustments and even minor changes can significantly affect model performance. This underlines the importance of meticulously choosing the hyperparameters for the optimal results in model performance. We also learned that in the process of developing SVM models, it’s important to select the appropriate kernel functions to handle high-dimensional data, but also select the appropriate regularization parameters. Regularization plays a crucial role in controlling the trade-off between achieving a low training error and at the same time generalizing well to new, unseen data.

For future work, we believe that tuning more hyperparameters such as momentum could improve model performance. Instead of specifying a high value for achieving faster convergence, we could try lower values as well to find a better balance that encourages stable learning [Fang]. We could also combine grid search with cross-validation on both MLP and SVM models instead of just splitting the dataset into three parts (train, validation, and test sets). This will provide a more reliable estimate of the model's performance as it assesses the model's ability to generalize to new data [11].

7. References

[1] CTIF, ''CTIF World Fire Statistics Report No. 27'', CTIF, [online]. Available: https://ctif.org/news/ctif-world-fire-statistics-report-no-27-now-available-download . [Accessed: 07-04-2024].

[2] D. A. Anggoro and D. Novitaningrum, ''COMPARISON OF ACCURACY LEVEL OF SUPPORT VECTOR MACHINE (SVM) AND ARTIFICIAL NEURAL NETWORK (ANN) ALGORITHMS IN PREDICTING DIABETES MELLITUS DISEASE'', International Congress on Innovation in Engineering and Technology, Available: http://www.icicel.org/ell/contents/2021/1/el-15-01-02.pdf . [Accessed: 08-04-2024].

[3] E. Sara, C. Laila, and I. Ali, 'The Impact of SMOTE and Grid Search on Maintainability Prediction Models', in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, United Arab Emirates: IEEE, Nov. 2019, pp. 1–8. Doi: 10.1109/AICCSA47632.2019.9035342. [Accessed: 22-03-2024].

[4] F. S. Nahm, ''Receiver operating characteristic curve: overview and practical use for clinicians'', *in Korean J Anesthesiol*, vol. 75, no. 1, pp. 25-36, Feb. 2022, doi: 10.4097/kja.21209. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8831439/ . [Accessed: 14-04-2024].

[5] J. Brownlee, ''How to Stop Training Deep Neural Networks at the Right Time Using Early Stopping'', Machine Learning Mastery, [Online]. Available: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/ . [Accessed: 09-04-2024].

[6] National Fire Protection Association, ''Fire Deaths By State'', NFPA, [Online]. Available: https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/fire-deaths-by-state . [Accessed: 07-04-2024].

[7] N. Vishwakarma, ''What is Adam Optimizer?'', Analytics Vidhya, [Online]. https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/ . [Accessed: 12-04-2024].

[8] P. Gaspar, J. Carbonell, and J. L. Oliveira, 'On the parameter optimization of Support Vector Machines for binary classification', *Journal of Integrative Bioinformatics*, vol. 9, no. 3, pp. 33–43, Dec. 2012, doi: 10.1515/jib-2012-201. [Accessed: 22-03-2024].

[9] P. Naraei, A. Abhari, and A. Sadeghian, 'Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data', in *2016 Future Technologies Conference (FTC)*, San Francisco, CA, USA: IEEE, Dec. 2016, pp. 848–852. Doi: 10.1109/FTC.2016.7821702. [Accessed: 23-03-2024].

[10] R. Patil, J. Pawar, K. Shah, D. Shetty, A. Ajith, and S. Jadhav, ‘Machine Learning based Forest Fire Prediction: A Comparative Approach’, *Int. Res. J. multidiscip. Technovation*, pp. 32–39, Jan. 2024, doi: 10.54392/irjmt2413. [Accessed: 23-03-2024].

[11] R. Shaikh, ‘’Cross-validation Explained: Evaluating Estimator Performance’’, Towards Data Science, [Online]. Available: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85 . [Accessed: 15-04-2024].

[12] ‘’Smoke Detection Dataset’’, https://www.kaggle.com/ . [Online]. https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset [Accessed: 23-03-2024].

[13] S. Fang, ‘’Hyper-parameters Tuning Practices: Learning Rate, Batch Size, Momentum, and Weight Decay’’, Medium, [Online]. Available: https://medium.com/analytics-vidhya/hyper-parameters-tuning-practices-learning-rate-batch-size-momentum-and-weight-decay-4b30f3c19ae8 . [Accessed: 10-04-2024].

[14] S. Mudadla, ‘’Weight Decay in Deep Learning’’, Medium, [Online]. Available: https://medium.com/@sujathamudadla1213/weight-decay-in-deep-learning-8fb8b5dd825c . [Accessed: 11-04-2024].

[15] S. Vishwakarma, ‘’Why is Sigmoid Function Important in Artificial Neural Networks?’’, Analytics Vidhya, [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/why-is-sigmoid-function-important-in-artificial-neural-networks/ . [Accessed: 08-04-2024].

[16] Y. Li, Z. Feng, S. Chen, Z. Zhao, and F. Wang, ‘Application of the Artificial Neural Network and Support Vector Machines in Forest Fire Prediction in the Guangxi Autonomous Region, China’, *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–14, Apr. 2020, doi: 10.1155/2020/5612650. [Accessed: 24-03-2024].

Appendix 1- Glossary

- **Perceptron:** It is a type of artificial neuron which is part of a neural network and it is foundational to the field of machine learning [9].
- **SMOTE:** Synthetic Minority Over-sampling Technique. It is a statistical technique used to increase the number of instances in a dataset in a balanced way [3].
- **Confusion Matrix:** A confusion matrix is a type of table which is used to evaluate the performance of a classification model on a set of test data which the actual values are known. It compares the model predictions to the actual values.
- **ROC Curve:** Receiver Operating Characteristic Curve. It is a graphical plot which is used to show the diagnostic ability of a binary classifier system as its discrimination threshold is varied [4].
- **Sigmoid:** The sigmoid function is an activation function and a fundamental component of the Artificial Neural Network and Machine Learning [15].

Appendix 2 – Supplementary Material: Intermediate Results and Implementation Details

- **Exploratory Analysis and Data Pre-Processing Steps**

Not all features were used in this project as input features of the models. After a detailed correlation analysis between the variables, a total of 7 relevant features were chosen for this project. The features that were chosen were the ones that high, statistically significant correlation with the target variable (Fire Alarm). The features that were chosen are the Humidity, Total Volatile Organic Compounds (TVOC[ppb]), Raw Ethanol Gas, Air Pressure (Pressure[hPa]), Particulate Matter Size (PM1.0), the Co2 equivalent concentration (eCO2[ppm]) and the air temperature, all important factors in smoke detection. It was identified that the values of features had various ranges, so normalization was necessary, so the values of all columns are scaled from 0 to 1.

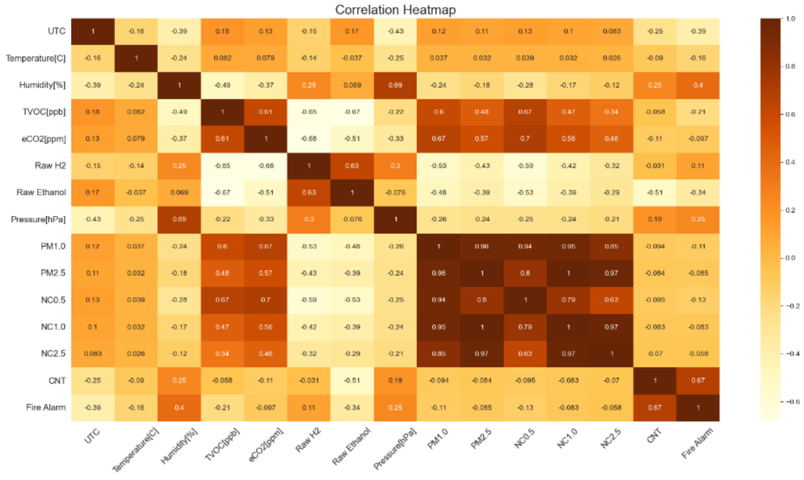


Figure 6. Correlation Heatmap of Variables

- **Changes Made Which Improved Results**

The first change that was made and improved results was normalizing the data (all features) with a minimum value of 0 and a maximum value of 1. This improved the average accuracy of the models in grid by more than 10-15%. The second change that was made which also improved results was applying SMOTE to the training data. Grid search was applied before and after SMOTE and the performance improvements were significant. The third change was feature selection. We first used all features of the dataset as inputs and then after correlation analysis and significance testing, only 7 variables were chosen. The results improved as we only selected the features that had a higher correlation with the target variable.

- **Intermediate Results**

While the best MLP model was trained and validated, we kept track of the training and validation loss over each epoch to see if the model was overfitting on the training data. We observe from Figure 7 that after around the 350th epoch, the validation loss is not decreasing anymore and has the tendency to increase. So early stopping was correctly applied in epoch 392 to stop the model from overfitting.

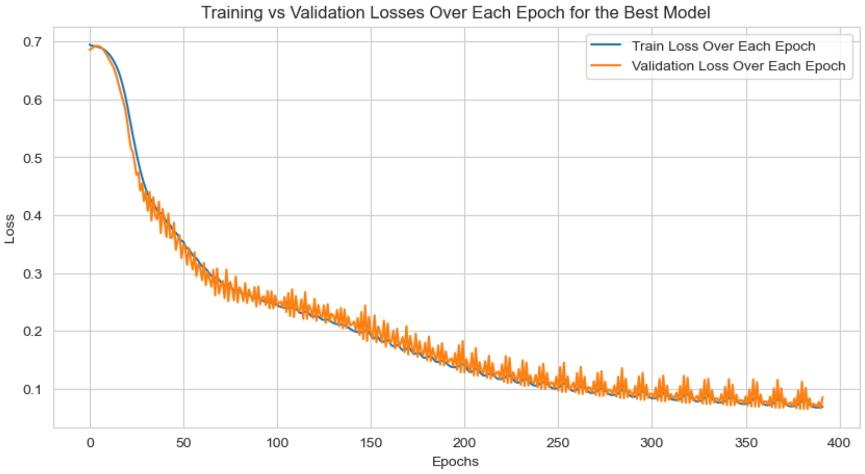


Figure 7. Train vs Validation Loss Over Epochs for the Best MLP Model