

AI-Powered Mock Interview Platform using Computer Vision, Natural Language Processing and Generative AI

Tanishque Sharma

*Department of Computer Science and
Engineering
Sharda University
tanishque003@gmail.com*

Anmol Singh

*Department of Computer Science and
Engineering
Sharda University
Greater Noida, Uttar Pradesh
anmolsingh0914@gmail.com*

Sanjay Singh

*Department of Computer Science and
Engineering
Sharda University
Greater Noida, Uttar Pradesh
sanjay22020002@gmail.com*

Dr. Ganesh Gupta

*Department of Computer Science and
Engineering
Sharda University
Greater Noida, Uttar Pradesh
ganeshgupta81@gmail.com*

Abstract — In this paper, we propose an AI-based mock interview platform addressing effective interviewing techniques with real-time results and intelligent mock interviews for candidates. It is a combination of generative AI, computer vision, and natural language processing (NLP) that provides a simulated experience of real-time interviews. It still continues to use methods such as CNNs for Facial expressions analysis, emotion detection from facial expressions and Voice recognition and NLP for tone, fluency, and confidence detection of a candidate. The platform compares responses with relevant industry standards using keyword-based semantic analysis in order to evaluate domain knowledge. The main highlight of this work includes emotion and answer-based feedback, where it provides the candidate with a performance report and suggestions for improvement. This ensures that users receive dynamic, multimodal insights that are tailored to the user profile (job role and skill set) and go beyond the scope of traditional mock interviews, where all interview questions feel the same. Overall, the platform is designed to help minimize stress, enhance communication capabilities, and empower career progression, while contributing to the evolution of AI in talent development and acquisition.

Keywords—AI-driven Mock Interviews, Generative AI, Computer Vision, Natural Language Processing (NLP), Emotion Detection, Speech Recognition, Convolutional Neural Networks (CNNs), Interview Simulation, Personalized Feedback, Career Development.

I. INTRODUCTION

As the job competition is tough these days, getting an interview is the first and most important step for every job seeker out there. Not only technical skills, but communication, confidence and emotional intelligence are also considered by recruiters. But quite a few candidates go into interviews nervous, poorly prepared and without a framework for improvement. However, conventional methods for interview practice, such as pre-recorded video responses, career coaching, peer mock interviews, or self-practicing, come up short. These methods are useful, but generally miss

key elements like generating questions based on an individual user, feedback and grouping of non-verbal indicators. The reliance on humans is also expensive, biased, and inconsistent, restricting access to many applicants.

Artificial intelligence and machine learning are recent advancements that have transformed a number of sectors, such as health care, education, and recruitment. Interview preparation powered by AI provides an excellent solution to these challenges through the availability of automated scoring, scalability, and data-driven improvement processes. We propose an AI-based mock interview platform which brings together NLP, Computer Vision, Speech Recognition, and fine-tuned Generative AI (like Gemini) to enable a personalized and one-on-one interview experience.

Conventional mock interview techniques have several drawbacks, such as a lack of customization, an inability to evaluate nonverbal communication, exorbitant expenses, and no adaptive learning tools. The suggested AI-driven mock interview system incorporates real-time multi-modal analysis employing voice recognition and computer vision to assess verbal cues, body language, and facial expressions in order to overcome these problems.

In order to provide a personalized and engaging experience, it uses a refined Gemini fine-tuned model to dynamically produce personalized questions depending on the candidate's performance and history. It also uses natural language processing for automated scoring and semantic analysis, and it gives immediate feedback via an analytics dashboard that monitors engagement, confidence, and overall development.

II. RELATED WORK

With the latest developments in artificial intelligence technologies like machine learning, natural language processing, and emotion recognition, mock interviews have

become a game changer in interview preparation. Recent studies have centered around including behavioral and emotional assessments in interview appraisal.

Anomaly detection and intrusion detection systems (IDS) have been given a lot of focus with the emergence of advanced cyber threats. Early research like Agrawal & Agrawal [1] and Mishra et al. [4] gave detailed summaries of data mining and machine learning methods suitable for IDS. Various studies conducted classical classifiers, such as ANN [2], decision trees [3], and classifier combination approaches for better precision. Semi-supervised approaches with fuzziness have also been suggested to deal with uncertain information in IDS [1]. Deep learning has dramatically progressed intrusion detection abilities, with algorithms like Deep Belief Networks (DBNs) [5], autoencoders [6][9][10], and hybrid methods that integrate autoencoders and statistical techniques showing high performance in feature extraction and anomaly classification. Javaid et al. [7] and Sommer & Paxson [8] critically analyzed the application of deep learning to IDS, both its potentialities and its limitations. Moreover, new outlier detection techniques like MSD-Kmeans have been proposed to effectively detect global and local anomalies [13][14], while statistical bases such as the Three Sigma Rule [15] and exploratory data analysis methods [12] continue to aid in model development. The ease of use of popular machine learning libraries like Scikit-learn [16] allows for common experimentation and deployment of various algorithms, thus opening up advanced anomaly detection to researchers and practitioners alike.

III. METHODOLOGY

This study aims to develop a job-mock interview platform powered by Artificial Intelligent tools based on Natural Language Processing (NLP), Computer Vision, Speech recognition, and Generative AI (e.g. Gemini). The system evaluates candidate confidence and performance based on verbal, nonverbal and emotional expressions and facial expressions. The platform aims to provide personalized advice and insights into the user experience to facilitate learning through adaptive feedback around the interview questions based on the user's profile and job role.

A. Dataset Used

- The facial emotion recognition model is trained on the FER-2013 and FER-Plus datasets in this study. FER-2013 consists of 35,887 grayscale images of human faces with labels for seven basic emotions, but it suffers from mis-training labels. FER-Plus, an upgraded version, offers more precise annotations using crowd-sourcing and introduces a new emotion category, contempt.

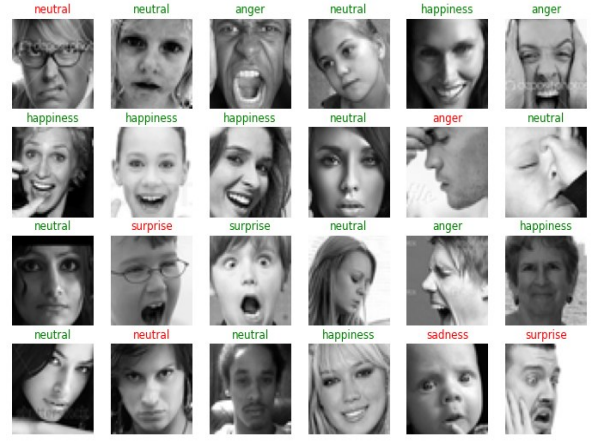


Fig. 1. Show the Facial Expression[28] dataset Used (Kaggle)

- For natural language processing, the platform uses React Hook libraries (e.g., react-speech-recognition) to convert speech to text. These libraries work along with browser-based APIs such as the Web Speech API which will convert spoken words into text in real-time. React Hook libraries do not depend on custom datasets, so they use pre-trained models that are derived from browsers or third-party APIs trained on large audio datasets to facilitate accurate speech recognition.
- The Interview Question Generator dataset(Kaggle) used for dynamically generating interview questions on the platform. With addition of more than 45,000 questions asked across 192 job roles, skills and experience levels, this data allows the platform to build tailored interview simulations.

B. Equations

1. Cross-Entropy Loss Function: cross-Entropy Loss when we deal with multi-class classification problems and our model provides us with a probability of each class. The formula is:

$$L = - \sum_{i=1}^N y_i \log(\hat{y}_i)$$

Final class label is y_i , predicted probability is \hat{y}_i and the number of classes is N . The loss function is perfect for training facial emotion recognition models for soft targets.

2. Accuracy: Accuracy is a simple metric to evaluate a model's performance and is calculated as:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} \times 100$$

Accuracy is thus utilized to measure model performance when using validation and test sets. The standard deviation of reported results is also recorded to address variability across runs.

3. Model Parameters Calculation: The parameter count is calculated by summing up contributions from each layer of the neural network, including convolutional, batch normalization, and fully connected layers. The formula is:

$$\text{Parameters} = \sum (\text{kernel height} \times \text{kernel width} \times \text{input channels} \times \text{output channels}) + \text{bias terms}$$

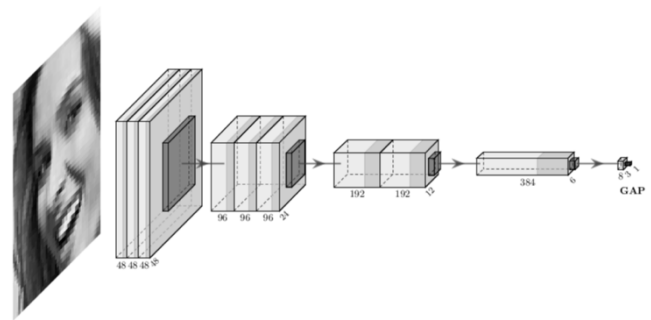


Fig. 2. Show the CNN Architecture with Different layers

C. Platform Architecture & Model Workflow

An AI powered mock interview platform architecture that can elevate the interview experience to a notch higher is depicted below. We use Next.js to build our front-end web application. js, using its server-side rendering features to achieve better performance and SEO. The user interface uses several JavaScript components and UI components like ShadCN to develop a modern and interactive user interface. When the user books an interview, they are asked to fill out some important information such as job profile, years of experience, and skills required. This is sent to a large language model (LLM) which uses this data stream to generate five unique interview questions tailored to the user's profile.

After generating questions, there is a section where the user can note their responses, and this is evaluated using Natural Language Processing (NLP) that processes the responses in real time and provides an analysis of their content. Moreover, the system incorporates computer vision to analyze nonverbal cues, like facial expressions and gestures, which can help in analyzing confidence and communication skills. Once the interview is finished, it provides tailored feedback about both the verbal and non-verbal aspects of the user's performance.

The CNN Architecture in Fig. 2 used for Facial Expression Recognition. The image shows a CNN Architecture for Facial Expression Recognition. This will take an input image of a face in grayscale, and run it through a succession of convolutional and pooling layers. These layers capture spatial features like edges, textures, and facial landmarks. As the network goes deeper, the depth and dimensionality gradually reduce. Following the final convolutional layer, the output is flattened into a 1D vector and fed through one or more fully connected (or dense) layers. The last part is a softmax classifier that outputs probabilities for all emotion classes (e.g. happy, sad, angry etc).

Layer (type)	Output Shape	Param #
separable_conv2d_20 (Separab	(None, 48, 48, 48)	105
batch_normalization_18 (Batac	(None, 48, 48, 48)	192
leaky_re_lu_18 (LeakyReLU)	(None, 48, 48, 48)	0
separable_conv2d_21 (Separab	(None, 48, 48, 48)	2784
batch_normalization_19 (Batac	(None, 48, 48, 48)	192
leaky_re_lu_19 (LeakyReLU)	(None, 48, 48, 48)	0
separable_conv2d_22 (Separab	(None, 48, 48, 48)	2784
batch_normalization_20 (Batac	(None, 48, 48, 48)	192
max_pooling2d_8 (MaxPooling2	(None, 24, 24, 48)	0
spatial_dropout2d_8 (Spatial	(None, 24, 24, 48)	0
leaky_re_lu_20 (LeakyReLU)	(None, 24, 24, 48)	0
...		
Total params:	173,537	
Trainable params:	171,137	
Non-trainable params:	2,400	

Fig. 3. Show the Model Summary and Architecture (CNN)

It also contains a feedback generation module that evaluates user responses against ideal answers and assesses factors such as fluency, confidence, and emotional cues. Then, it generates a comprehensive feedback report summarizing the user's performance and recommendations for improvement. Such feedback is given to the user through a user-friendly UI that provides ease of polishing users' interview skills. All processes are done in a cloud with real-time data running, providing an uninterrupted flow of events that cover all aspects of an interview.

IV. RESULT

With a seamless integration of its components, the AI-enabled mock interview platform performs exceptionally well to deliver an authentic and revealing interview experience. Using the user's job role, experience, and skillset, a large language model dynamically generates context-aware questions. They use speech recognition to capture user responses, and NLP to rate the fluency, coherence, and relevance of user responses. (events) An auxiliary CNN model tailored to the FER+ dataset employs cross-entropy loss and softmax to classify facial expressions for eight different types of events. The performance metrics such as training loss, training accuracy, and confusion matrix validate the reliability of our model, particularly in human emotions detection: happiness and surprise. The result is a post-interview feedback report that forms an evaluation on verbal

answers and also provide insight on emotions through visualizations that help users to analyze on both their content delivery and non-verbal cues.

A. Question Generation and Analysis

It generates dynamic interview questions using a large language model (LLM), tailored to the user's job profile, years of experience, and skills. In general it generates rich and role-relevant questions allowing diversified interviews for the user.

```
Full stack Development React, Nextjs node js 2 page-a767e1b5a05517ca.js:1
Parsed JSON Response:
  (5) [(-), (-), (-), (-), (-)] 1
    0: {question: 'Describe your experience building full-stack appls of projects and
    1: {question: 'Explain your understanding of server-side rendering_a you leveraged $
    2: {question: 'How do you handle data fetching in a Next.js appl_xplain when you v
    3: {question: 'Describe your experience with databases in the con_database interacti
    4: {question: 'How do you approach testing in a full-stack develo_sting, end-to-enc
      length: 5
    [[Prototype]]: Array(0)
```

Fig. 4. Show the Response for Questions Generated in JSON format

B. Computer Vision-Based Non-Verbal Analysis

This study utilized two prominent datasets for training and evaluating the facial emotion recognition model: **FER-2013** and **FER-Plus**. To ensure efficiency as well as the accuracy of emotion recognition, we have created a custom CNN model. Uses several convolutional layers of neural network. BatchNorm2d for training regularization and speed. Instead of normal ReLU the model employs Leaky ReLU activation preserving the gradient for the negative input as well.

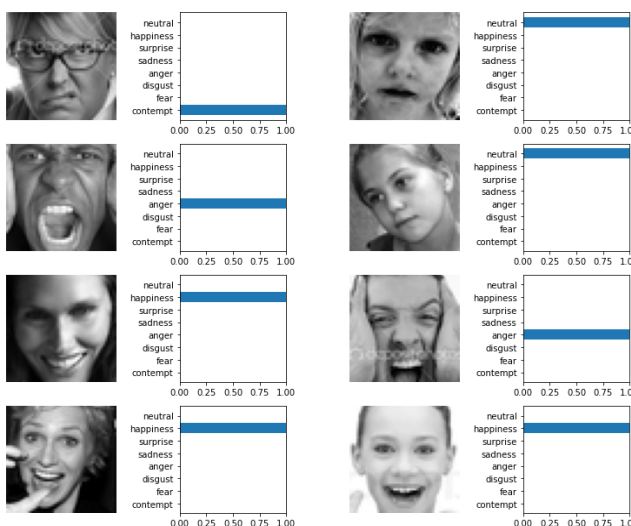


Fig. 5. Show the Different Facial Expression Captured by Model

Through this, prediction is used for larger data sets. Supplementing the model's pursuit for fewer params without diminishing performance, by emulating from ultra-light models such as MobileNet and Mini-Xception, by introducing separable convolutions and global average pooling. Finally, fully connected (dense) layers are included for the classification task along with a soft-max output layer for predicting one of the eight emotion classes.

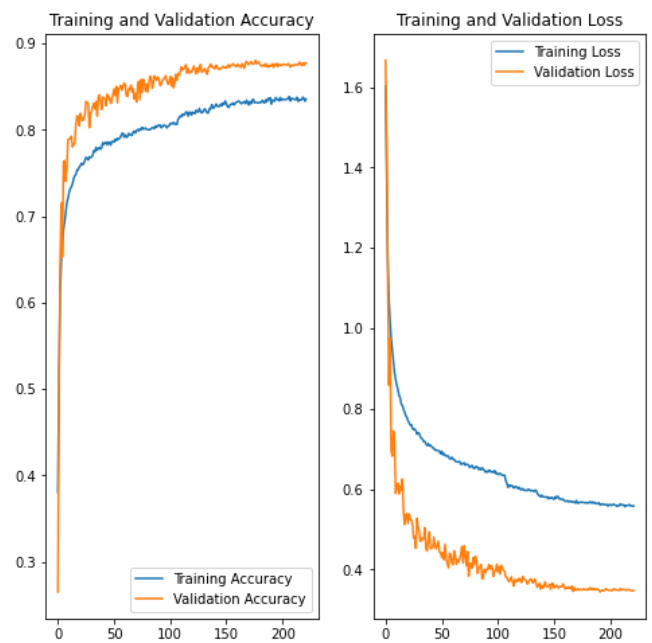


Fig. 6. Show the Training and Validation Accuracy and Loss

The Fig.6 shows the training and validation performance of a CNN model over time. On the left, the accuracy plot demonstrates that both training and validation accuracy steadily increase and eventually level off, indicating effective learning and good generalization. On the right, the loss plot shows a consistent decrease in both training and validation loss, with no major gaps between them, suggesting that the model is not overfitting. Overall, the graphs reflect a well-trained CNN that performs reliably on both the training and validation datasets.

The tested converted model shows that our trained model resulted in an overall test accuracy of 87.66% (which shows good generalization in predicting facial expressions on the dataset). This confirms the power of the lightweight separable convolution-based architecture in conjunction with regularization techniques like batch normalization and spatial dropout. The model demonstrated strong accuracy alongside a relatively low parameter count (if we have ~173k on total parameters) which make it very suitable for real-time or edge deployments for emotion recognition tasks.

C. Feedback System Overview

Once the interview session is done, the platform creates a detailed feedback report for the user in an automated manner. The analysis of this report comprises two critical constituents verbal response assessment and visual emotion examination. Based on certain natural language processing techniques and speech quality, the verbal feedback captures the user's recorded answers' strengths and areas of improvement. At the same time, the visual feedback conveys to you information regarding the non-verbal cues of the user—such as facial expressions and emotional state consistency—based on computer vision analysis. When combined, these elements provide a comprehensive, personalized review to help users better performance in interviews going forward.

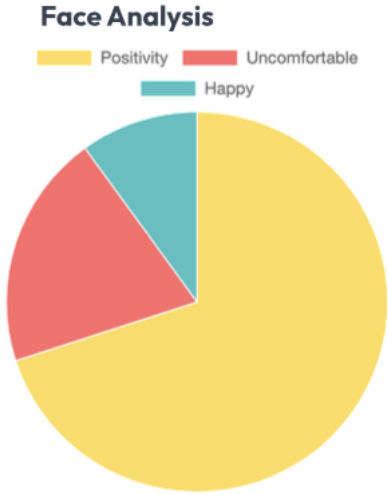


Fig. 7. Show the Feedback Response of non- Verbal Interview



Fig. 8. Show the Feedback Response of Verbal Interview

D. Figures and Tables

Here we see a confusion matrix in Fig.7 of how the model is performing in recognizing different facial expressions. It really knows its common emotions such as neutral, happiness, and surprise, where the vast majority of the predictions are correct. For instance, it correctly identified what is "neutral" more than 1000 times. But it's less accurate with emotions like sadness, anger and especially fear, disgust and contempt — which tend to be misclassified or confused with another emotion. Many "sad" and "angry" images are misclassified as "neutral" while some "fear" expressions are confused for "surprise". Contempt is the thing the model struggles with most, which makes sense because there are

fewer instances of this in the data or it's more difficult to differentiate.

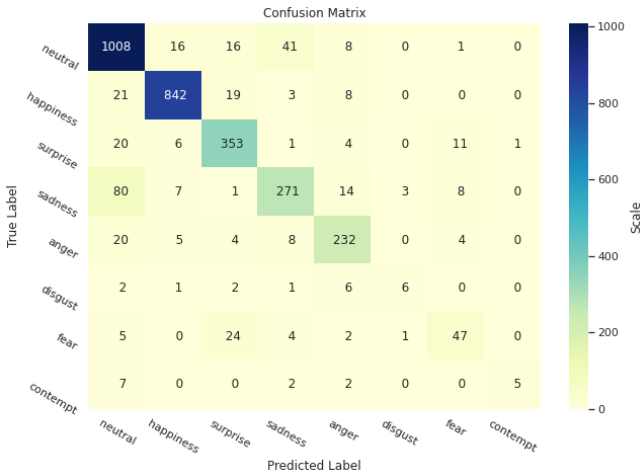


Fig. 9. Show Confusion Matrix of Emotion Classification Model

Overall, the model shows strong performance on the more common emotions but needs improvement on the subtler or less frequent ones.

TABLE I. DIFFERENT PROCESS USED IN THE PLATFORM.

Component	Technology	Functionality
Speech Processing	CNN-RNN, MFCC	Extracts voice tone & hesitation
Text Processing	GPT-4, BERT	Analyzes coherence & correctness
Facial Recognition	OpenCV, ResNet50	Detects stress, engagement
Posture Tracking	PoseNet, MediaPipe	Tracks eye contact & gestures
Question Generation	GPT-4 fine-tuning	Adapts questions dynamically

The table 1. Explains how different technologies combined in a smart interview system can help in gaining a better understanding of a response of a person. And it takes in the way someone talks, CNN-RNN with MFCC, identifying details like tone and hesitation. It then uses language models, such as GPT-4 and BERT, to see if those answers make sense and are well-structured. It also monitors facial expressions using tools like OpenCV and ResNet50 to detect signs of stress or engagement. For body language it employs PoseNet and MediaPipe to monitor things like eye contact and gestures. Then finally, it's able to adapt its questions on the fly using a fine-tuned version of GPT-4, which makes the whole thing feel a lot more natural and responsive.

V. CONCLUSION & FUTURE SCOPE

It uses the latest in natural language processing, speech recognition, and computer vision-powered technologies to simulate real interview environments. The system helps users prepare by dynamically generating interview questions customized to each user's active resume and analysing their verbal and non-verbal reactions to provide comprehensive feedback. This combination of analysis and NLP-driven assessment ensures a well-tuned evaluation of the user's performance. Moreover, the user-friendly interface of the platform is created with Next.js and other modern UI components, ensuring seamless interaction and accessibility. In summary, the system shows a good potential for a scalable, intelligent, and interactive interview preparation solution, with possibilities for additional improvements to the system in the form of multilingual support and real-time coaching functionalities. This AI-driven mock interview platform has much scope in the future. Real-time coaching along with personalized recommendations for training sessions based on users' performance analytics are slated for the future. This update will help grow the platform for a larger audience and bring new users to the platform who are not fluent in English. It can also be integrated with large scale job portals and recruitment platforms for smooth hiring. This can be expanded by integrating more sophisticated computer vision tools that allow for more complex analysis of gestures, eye gaze, posture, etc. Domain-specific interview support (medical, legal, tech) and adoption of more advanced AI models will lead to better quality and relevance of the feedback. Lastly, mobile friendly and cloud-based deployment increases accessibility and experience across devices.

REFERENCES

1. S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Proc. Comput. Sci.*, vol. 60, pp. 708–713, Jan. 2015. [12] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.
2. B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.
3. J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 1051–1058, Dec. 2017.
4. P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.
5. T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 4153–4156.
6. M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *Proc. Int. Joint Conf. Neural Netw. (IJAUTOENCODERS)*, May 2017, pp. 3854–3861.
7. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *EAI Endorsed Trans. Secur. Saf.*, vol. 3, no. 9, p. e2, May 2016.
8. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 305–316.
9. C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, Apr. 2020.
10. K. Sadaf and J. Sultana, "Intrusion detection based on autoencoder and isolation forest in fog computing," *IEEE Access*, vol. 8, pp. 167059–167068, 2020.
11. G. S. Maddala and K. Lahiri, *Introduction to Econometrics*, vol. 2. New York, NY, USA: Macmillan, 1992.
12. J. W. Tukey, *Exploratory Data Analysis*, vol. 2. Reading, MA, USA: Addison-Wesley, 1977.
13. Y. Wei, J. Jang-Jaccard, F. Sabrina, and T. McIntosh, "MSD-Kmeans: A novel algorithm for efficient detection of global and local outliers," 2019, arXiv:1910.06588. [Online]. Available: <http://arxiv.org/abs/1910.06588>
14. Y. Wei, J. Jang-Jaccard, F. Sabrina, and H. Alavizadeh, "Large-scale outlier detection for low-cost PM10 sensors," *IEEE Access*, vol. 8, pp. 229033–229042, 2020.
15. F. Pukelsheim, "The three sigma rule," *Amer. Statist.*, vol. 48, no. 2, pp. 88–91, 1994.
16. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
17. <https://www.kaggle.com/datasets/msambare/fer2013>