Identification of Fraud in Accidental & Healthcare Insurance using Local Outlier Factor

Jyoti Lele[1], Vaidehi Deshmukh[2], Abhinav Chandra[3], Radhika Desai[4]

[1,2,3,4] *Department of Electrical and Electronics Engineering, Dr. Vishwanath Karad MIT World Peace University Pune, India*

E-mail address: jyoti.lele@mitwpu.edu.in, vaidehi.deshmukh@mitwpu.edu.in, ethicallyabhinav@gmail.com, ridhs2001@gmail.com
Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: An unsupervised machine learning model that uses the mechanism of the Local Outlier Factor to flag and detect ambiguous and potentially fraudulent claims in Accidental and Healthcare insurance is proposed. The ethos of this model is to comprehensively automate and expedite the claim investigation process using certain parameters to aid the claim appraiser's workload of going through straightforward claims and saving their time to investigate more critical and complex claims. The model flags claims which are anomalous which when compared to the model's threshold and input parameters are generated as alerts. These alerts generated are then investigated for fraud based on the parameters stated. The model can classify these claims and the cost of billable associated with these claims by reporting an accuracy of 99.5% for the Local Outlier Factor model in comparison with other implemented techniques of Isolation Forest which had an accuracy of only 78.37%. The clusters were visualised with DBSCAN using Plotly whereas the outliers were seen using TSNE. Our model has been tested and validated on real-world data and is showing promising results. Being able to identify and flag potentially fraudulent claims before they are paid out can save insurance companies a lot of money and resources. The model can classify the claims based on risk levels and associated costs. This will help the insurance company prioritise which claims to investigate first and allocate their resources accordingly. Our model has been tested and validated on real-world data and is showing promising results. Being able to identify and flag potentially fraudulent claims before they are paid out can save insurance companies a lot of money and resources. The model can classify the claims based on risk levels and associated costs. This will help the insurance company prioritise which claims to investigate first and allocate their resources accordingly.

Keywords: A&H, LOF, DBSCAN, TSNE, BOW, KPIs, Healthcare Insurance

1. INTRODUCTION

Health insurance fraud is a complex and pressing issue, causing substantial financial losses that reverberate throughout the industry. In the quest to address this challenge, a pivotal focus lies on creating a robust model that can effectively identify potential instances of fraudulent activities. The proposed methodology takes a two-fold approach: it amalgamates well-established statistical methods with contemporary machine learning techniques, thereby enhancing the model's accuracy and practicality [1].

The crux of the problem revolves around the inherent ambiguity and resemblance between fraudulent claims and legitimate ones. This likeness necessitates a localized approach for outlier detection, one that can discern anomalies that a global perspective might miss. It is within this context that the Local Outlier Factor (LOF) algorithm finds its relevance.

Motivating this endeavor is the aspiration to usher in automation for detecting fraudulent claims. The driving force behind this automation is the potential to eliminate human intervention and associated errors. By doing so, the proposed model strives not only to enhance the precision of fraud detection but also to yield substantial savings in terms of time, resources, and capital.

The central objective of this paper is to present a comprehensive and effective solution to combat health insurance fraud. By ingeniously blending traditional statistical methods with cutting-edge machine learning techniques, the model is poised to become a stalwart defense against fraudulent activities. The motivation to curtail significant financial losses attributed to health insurance fraud, coupled with the desire to preserve the integrity of the insurance ecosystem, propels this research forward.

In the existing landscape of research, the complexity of health insurance fraud detection has been acknowledged, yet comprehensive solutions remain limited. The gap lies in the integration of diverse methodologies into a unified framework that not only identifies anomalies but also does so with a reduced reliance on human intervention.

The novelty of this conducted research lies in the fusion of localized outlier detection using LOF with a comprehensive machine learning approach. The seamless integration of these two elements empowers the

E-mail: vaidehi.deshmukh@mitwpu.edu.in

model to not only detect fraud effectively but also minimize the chances of false positives and negatives. This dual-edged approach encapsulates the essence of this paper's innovation.

2. LITERATURE REVIEW

Healthcare fraud has emerged as a daunting challenge, causing substantial financial setbacks and impacting patient well-being [1]. Addressing this complex issue calls for innovative strategies within the intricate framework of the US healthcare system. The ultimate objective is to introduce automation into fraud detection, a move that holds the potential to curb human errors and save valuable resources [2]. However, the endeavor is not without hurdles, as detecting healthcare fraud and abuse through traditional methods remains an uphill battle. This underscores the pressing need for automated solutions capable of navigating the complexity of the healthcare landscape [3].

The insurance sector grapples with a persistent problem, as insurance fraud continues to siphon off significant sums of money. Recognizing that static approaches are insufficient, the industry has embraced dynamic technologies to proactively identify fraud patterns [4]. Amidst this evolution, data analytics and machine learning stand out as the pillars of modernizing the insurance market. Yet, the journey is not without challenges, as insurers face a dearth of analytical models and algorithms that can truly support their objectives. It's clear that machine learning holds the key to unlocking deeper insights and efficiencies within the sector [5].

The expansion of insurance clientele has propelled the importance of thorough claim analysis. This analysis, enabled by exploratory data examination and feature selection, empowers insurance companies to distinguish between valid and fraudulent claims [6]. Parallelly, the healthcare landscape witnesses its battle against fraud. Data mining techniques offer a ray of hope, fueling efforts to expose fraudulent claims within the healthcare insurance domain. A novel hybrid approach, melding supervised and unsupervised learning, is poised to elevate fraud detection capabilities [7].

The healthcare sector's pivotal role in people's lives is juxtaposed with the challenges posed by fraud [8]. The misuse of medical insurance adds a layer of complexity to an already intricate field. Machine learning and data mining enter the scene as potential saviors, offering tools to identify and combat healthcare fraud. The call for advanced techniques and data sources is apparent, suggesting a path to affordability and fraud mitigation. However, the road ahead involves strategic maneuvering through these advanced methodologies.

A systematic review of healthcare insurance fraud detection techniques underscores the industry's pursuit of effective solutions. The quest to uncover ideal application solutions is a testament to the ongoing efforts against fraud [9]. Against this backdrop, a hybrid model combining classification and clustering steps forward to differentiate legitimate claims from fraudulent ones. The impact of this approach echoes on a larger scale, potentially uplifting economies by curbing healthcare fraud [10]. A systematic review of healthcare insurance fraud detection techniques underscores the industry's pursuit of effective solutions. The quest to uncover ideal application solutions is a testament to the ongoing efforts against fraud [9]. Against this backdrop, a hybrid model combining classification and clustering steps forward to differentiate legitimate claims from fraudulent ones. The impact of this approach echoes on a larger scale, potentially uplifting economies by curbing healthcare fraud [10].

The intricate web of healthcare insurance brings its own set of challenges. The proposed theoretical model for medical insurance fraud identification takes a holistic approach, exploring dimensions of time, quantity, and expenses. This approach, validated through real-world medical records, sheds light on distinctive behavioral characteristics that can drive AI and machine learning technologies for fraud detection [11].

Machine learning's potential to revolutionize fraud detection is tangible. Decision Trees, Bagging, Random Forests, and Boosting algorithms all play a part in this transformation. The efficacy of these algorithms comes to light through rigorous evaluation metrics. While challenges persist, the promise of machine learning in tackling the costly menace of insurance fraud remains undiminished.

In the proposed work, the use of machine learning algorithms takes center stage. The fusion of these techniques aims to categorize statements as true or false, a fundamental aspect of fraud detection. This approach is grounded in real-world datasets, highlighting the relevance of accurate data in detecting fraud. The meticulous structure of the paper, encompassing dataset description, system details, methodology, results, discussions, and conclusions, underscores the rigor of research in this domain [12]. The journey to reduce fraud across sectors demands continued exploration, innovation, and collaboration. Amidst challenges and complexities, the collective efforts of researchers, industries, and systems forge ahead to safeguard financial systems and patient welfare alike.

3. SYSTEM DESCRIPTION

Before In the proposed work we have used the Accident and Health Insurance dataset under the name India A&H Fraud for Liberty Mutual Insurance Group open source Bitbucket repository and Kaggle's open source dataset on Credit Card fraud analysis.

The breakdown of some relevant parameters i.e. feature set in the dataset is as follows; It includes a 'claim number', 'TPAClaimNumber', which is used by a third party authorized to sell insurance, 'Policy Number' and 'Policy Inception Date' that detail the insurance policy, and a 'Hospital Code', which identifies registered hospitals. There is also related data about the location of

the hospitals, whether they are part of an affiliate network, the 'type of hospital', and individuals' 'Claimed Amount'. The 'ICD Code' gives information about the disease, lab tests done, and overall symptoms. Other data fields include the 'Type of Hospitalization', 'DOA_MOD' (Date of Authorization of the policy), and the 'Product Name'.

Additional data fields in the report include 'Business Type', which refers to the renewal process of the insurance policy, 'Claim Registration Date', which indicates when the claim was registered in the system, and 'Channel Name'. Information about the 'Treatment Type' and 'Intermediary Category' is also captured, along with 'Age', 'Doctor's Charges', 'Diagnosis Text', and 'Doctor's Fee'. Fields that capture different time durations include 'Days to Report', 'Inception to Loss', 'Loss to Exp', and 'Start to Loss'. The dataset also accounts for 'Blacklisted Hospitals', 'Median Claimed' in relation to 'ICD Codes', and 'Claim Difference'. Other factors noted in fraud detection include 'ICD Freq', 'ICD Weight of Evidence', 'ICD Zero Fraud', 'Frequency Pin of Hospital', and 'Pin_of_Hospita_WOE'.

In the insurance industry, health insurance fraud is a growing issue that costs billions of dollars annually. It's crucial to create a precise and trustworthy model to recognize potential fraud cases to lessen this issue. The suggested model analyses claims data to find potential fraud cases using a combination of conventional statistical methods and machine learning techniques.

The claims data are first pre-processed by the model, which includes data cleaning, imputation, and normalization. The next step is to find outliers and anomalies in the data using conventional statistical techniques. This includes methods like hypothesis testing, regression analysis, and clustering. Following the identification of the outliers, the data is fed into machine learning algorithms like decision trees, balanced random forests, and ANN. The claims data is used to train these algorithms to find trends and connections that point to fraud. After that, predictions about the likelihood of a claim being fraudulent are made using the algorithms.

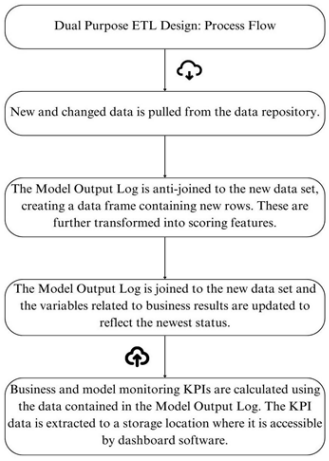


Figure 1. Explaining the Purpose of ETL Used In Our Model

To uncover groups of claims with similar features and to identify claims that deviate noticeably from the rest of the data, the model also uses unsupervised learning techniques like clustering and anomaly detection. As a result, it is simpler to identify claims that might be a component of a broader fraud operation. The model is validated using a holdout sample of claims data that has been set aside for this purpose. The model is evaluated based on its F1 score, recall, accuracy, and precision. This assessment is visualized via a dashboard that helps in tracking the model's performance on a timely basis. We can conclude that the model offers a solid and trustworthy solution to the issue of health insurance fraud by combining conventional statistical techniques and machine learning algorithms.

As shown in figure, ETL stands for Extract, Transform, Load, and it refers to the process of extracting data from various sources, transforming that data into a usable format, and then loading it into a target system, such as a database or data warehouse [13], [14]. Here's a more detailed breakdown of how ETL works:

Extract: The first step is data extraction from multiple sources, such as databases, APIs, online services, or flat files, which is the initial stage. From these sources, the information is gathered and copied into a staging area where it may be processed. In our case, the data is extracted from a backend Excel sheet, but the model proposes to extract data from the Jump Box. Jump Box.

Transform: After the data has been extracted, it must be converted into a format that can be used. Prepare the data for analysis, this includes cleaning, eliminating duplicates, and reformatting the data. In this step, the data may also be enhanced by the addition of new fields, such as computed fields or geolocation information or geolocation information.

Load: The modified data must then be loaded into a target system, like a database or data warehouse, as the last

step. Data must be mapped to the target schema for it to comply with the standards of the target system.

Overall, the ETL process is critical for organizations that need to integrate data from multiple sources and make it available for analysis and decision-making. It ensures that data is accurate, consistent, and reliable, and can help organizations gain insights that can drive business success.

A. Implementation of the algorithm

The proposed model considers several variables, including the claim amount, the kind of service rendered, the patient's medical background, and previously submitted claims. Natural language processing (NLP) techniques are also incorporated into the proposed model to analyze the claims' text data and find discrepancies and uncommon language usage. The process of EDA, and implementation of all algorithms has been done using Python 3.4.

Exploratory data analysis was performed by resolving missing values, feature engineering, target variable enhancement, and moving to feature selection and feature validation using Boruta's algorithm which will enable us to select features based on which features would impact our model most to least.

Boruta's algorithm's primary objective is to meticulously navigate through an array of features and pinpoint those of paramount importance. It trains on real and simulated data, assigning scores based on impact. Features with higher scores in real data are kept, while others are discarded. This iterative process continues until confidence in chosen features is high. These culled features build an optimized model, enhancing predictions. Essentially, Boruta accelerates discovery in data, boosting predictive abilities.

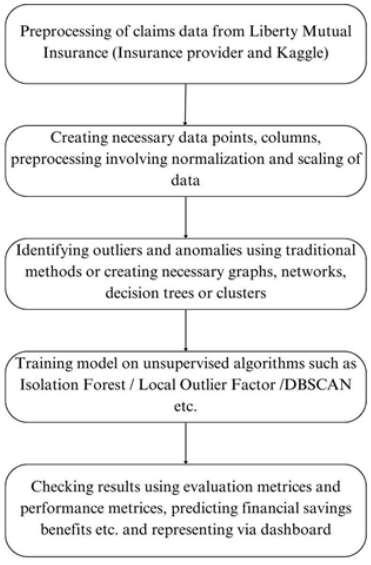


Figure 2: Flowchart demonstrating the model

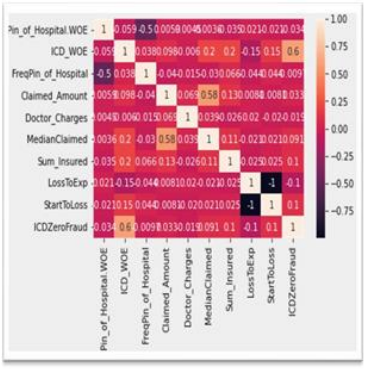


Figure 3: Heat Map Showing Feature Dependencies

Figure 3 shows the features that have been selected using Boruta's algorithm, i.e., the subset are 'Pin_of_Hospital.WOE', 'ICD_WOE', 'FreqPin_of_Hospital', ' Claimed_Amount', 'Doctor_Charges', 'MedianClaimed', 'Sum _Insured', 'LossToExp', 'StartToLoss', 'ICDZeroFraud'.

The heat map reflects the dependencies of the features of the model, i.e., the input parameters are their correlation. We have made sure that the features that have been used as input parameters do not have a correlational value above 0.6.

To detect outliers, the process of anomaly detection is implemented which is a subcategory of unsupervised machine learning that identifies cases that are probable statistical outliers and overall categorizes the data into clusters in which these outliers are present. The reason we are using anomaly detection to detect outliers is that it will help us classify cases that are ambiguous in nature, maybe one of their kind in terms of their uniqueness, or also spot claim cases prone to be illegitimate or false i.e. fraudulent. These ambiguous cases can be potentially fraudulent as a fraud case is unique and cannot be differentiated easily as it resembles a lot of similarities from legitimate cases.

The implementation of the local outlier factor (LOF) is shown in Figure 4. It produces an anomaly score that identifies the outlier data points in the data set. The local density deviation of a given data point in relation to neighboring data points is calculated to achieve this. TSNE was used to visualize the results and plot outliers. Using DBSCAN, the clusters were made visible.

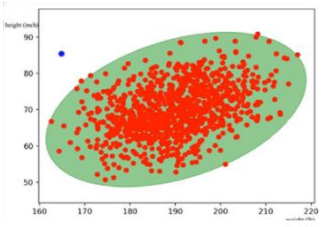


Figure 4: Anomaly Detection Example using LOF

We implemented the Isolation Forest algorithm for

comparison. This unsupervised machine learning technique is based on the principle of isolating anomalies rather than the general practice of profiling good data points. The algorithm randomly selects features from the dataset and randomly selects a split value between the maximum and minimum values of the selected features [15]. Our results showed an accuracy of 78.37% which was considerably less when compared to LOF.

A column 'Diagnosis Text' in the dataset consisted of unstructured data that could be converted and additional features could be taken into consideration from the column.

Specific techniques from Natural Language Processing (NLP) were explored that are significant for text analysis. One of these techniques is Term Frequency-Inverse Document Frequency (TF-IDF), which yields vocabulary-based calculations [16]. These computations accentuate the weight of words in the text, capturing their importance effectively. This process aids in constructing features that have the potential to enhance model performance [17].

GloVe, a methodology grounded in word co-occurrence was looked at next. This approach involves creating a matrix that captures the frequency of words appearing together in texts. Through dimensionality reduction, these co-occurrence scores are transformed into meaningful insights, revealing the frequency of word collaborations with other terms [18], [19].

Word2Vec was considered, an algorithm that generates a distributed semantic representation of words in the text. This sophisticated technique generates word embeddings that encapsulate contextual meanings and relationships. This empowers a deeper comprehension of nuanced meanings embedded in the vocabulary of the text [20], [21].

Each phrase's context might be used to train the model, yielding numerical representations of related terms. However, using the Bag of Words paradigm, we settled on sparse vector representations.

This model counts the number of times a word appears in a document, hence the text content will be converted to numerical feature vectors by vectorization. We can use those word counts to compare documents and determine how similar they are for applications like topic modeling, document classification, and search.

There were a lot of challenges involved in terms of modifying the dataset to be suitable for model training. They included data acquisition challenges, the feature selection process being difficult due to an imbalance in data, and unidentified fraudulent claims in the data. This challenge was aced using specific data sources such as using specific data sources like AWS Redshift. The challenge for unidentified fraud was aced using LOF anomaly detection. We used BRF for data imbalance and deployed a Jump Box for deployment. The challenges

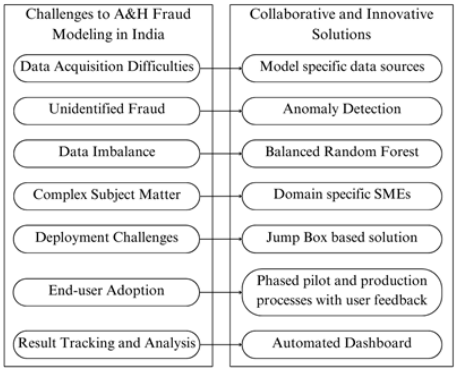
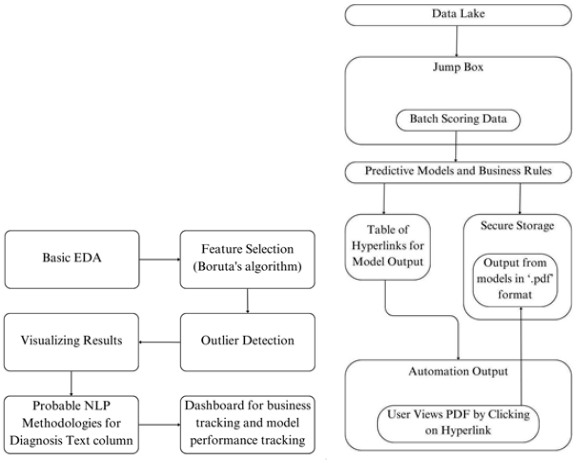


Figure 5: Challenges and Complexities Involved and their Solutions

were countered through exploratory data analysis where SMOTE was used, model-specific data selection was performed, and feature selection was enhanced by applying NLP methodologies to relevant data in the dataset. Representation and visualization of outliers were done via DBSCAN with Plotly and TSNE. dataset. Representation and visualization of outliers were done via DBSCAN with Plotly and TSNE.

B. System block diagram

Figure 6 explains a theoretical overview of a system block diagram, and an explanation of an entire pipeline-based solution using a Jump Box (also known as a Bastion Host) in a Data Lake architecture.



(a) (b)

Figure 6: System Block Diagram (a) Theoretical Overview, (b)Entire pipeline-based solution using Jump box, i.e. Data Lake

A system block diagram is a graphical depiction of a system that demonstrates the main parts or subsystems of the system as well as the communication channels between them. It is a broad overview of a system that aids in visualizing and comprehending its general functions. Each

component is shown as a block in a system block diagram, and the relationships between them are shown as lines or arrows.

Large amounts of organized and unstructured data can be stored and analyzed at any scale using a data lake, which is a centralized repository. The storage layer, the processing layer, and the analytics layer are common layers in a data lake architecture. A pipeline-based solution in this sense refers to a collection of connected data processing procedures that convert raw data into actionable insights. Data ingestion, data cleaning, data transformation, and data analysis are common stages in a pipeline.

A Jump Box, or Bastion Host, is a server that is used to securely access and manage other servers or devices within a network. In a data lake architecture, a Jump Box can be used to securely access and manage the various components of the system, including the storage and processing layers. An entire pipeline-based solution using a Jump Box in a Data Lake architecture would involve several steps. First, raw data would be ingested into the data lake through various sources such as API calls, log files, and batch uploads. The data would then be cleaned, transformed, and prepared for analysis using various tools and techniques.

The data would then be saved in the storage layer of the data lake, which might use a variety of tools including the Hadoop Distributed File System (HDFS), Amazon S3, or Azure Blob Storage. Following that, the data would be processed and analyzed using programs like Apache Spark or Apache Hadoop, which would run on the data lake's processing layer.

Finally, the insights gained from the analysis would be presented to the end-users through various visualization tools, such as Tableau or Power BI, which would run on the analytics layer of the data lake. Throughout this entire process, the Jump Box would be used to securely manage and access the various components of the data lake architecture, ensuring that the data is stored, processed, and analyzed securely and efficiently.

4. RESULTS

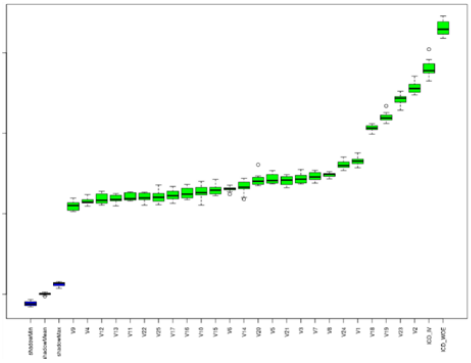


Figure 7: EDA and Feature Selection

The aforementioned figure illustrates how our model's exploratory data analysis (EDA) method works. EDA generally involves studying and comprehending data sets to identify patterns, trends, and correlations within them as well as to identify their key properties.

As seen in Figure 7, EDA is a crucial stage in any data analysis project because it gives data scientists new perspectives on the data they are using, which can guide the study's subsequent steps. The process of choosing a subset of the most important features (sometimes referred to as variables or predictors) from a data collection to utilize in a model or analysis is known as feature selection. By removing unnecessary or redundant characteristics that could impair the performance or interpretability of the model, feature selection aims to lower the dimensionality of the data collection.

EDA and feature selection are closely related, as EDA can help data scientists identify which features are most relevant to the outcome they are trying to predict. EDA can also reveal any relationships or correlations between features, which can help data scientists decide which features to include or exclude from their model. In other words, EDA can inform the feature selection process by providing insights into which features are most important and informative for the analysis at hand.

In conclusion, feature selection and exploratory data analysis are both crucial processes in the data analysis process, and they can operate in tandem to assist data scientists in better understanding the data they are working with and deciding which features to include in their analyses.

The popular algorithm t-SNE (t-distributed Stochastic Neighbour Embedding), which is used to visualize high-dimensional datasets [22], is demonstrated in Figure 8. In the illustration provided here An approach for finding clusters in a dataset is called DBSCAN (Density-Based Spatial Clustering of Applications with Noise). A Python module called Plotly is used to provide interactive data visualizations.

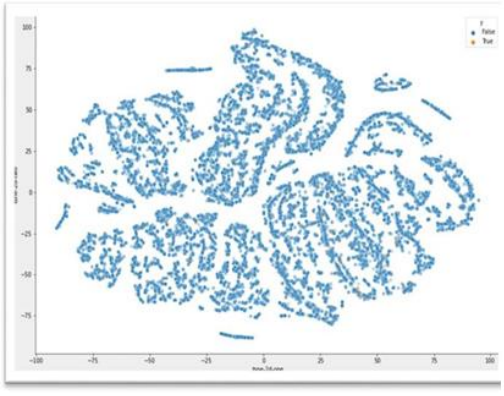


Figure 8: Outliers Using TSNE

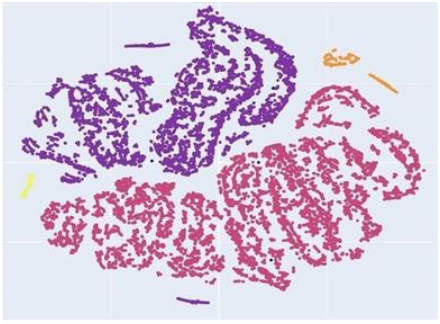


Figure 9: DBSCAN with Plotly

In Figure 9, DBSCAN groups data points that are closely packed together and identifies outliers among the data points that are not closely packed.

At first the import of the DBSCAN module from the sklearn.cluster library is done before you can use DBSCAN. Then, specifying the eps (epsilon) and min_samples arguments when creating an instance of the DBSCAN class is done [23]. Eps indicates the radius of the neighborhood surrounding a data point, while min_samples specifies the minimal number of data points needed to build a dense zone.

Table 1: Bag of Words

Variable Name	Mean Variable Importance	Associated Terms
V2	51.26916	FEVER
V23	48.43494	UTI
V19	43.98489	DENGUE WITH TCP
V18	41.28179	DENGUE
V1	33.09415	(Entry is blank)
V24	32.22399	PNEUMONIA
V8	29.66566	DENGUE FEVER
V7	29.23064	COVID 19

The outcomes of Bag of Words (BoW) are displayed in the above table. It is a method that is frequently used in Natural Language Processing (NLP) to extract features from text. As seen in Table 1, the fundamental concept of BoW is to represent a text document as a "bag" of its words while accounting for their frequency rather than their order [24], [25].

We first generate a vocabulary out of all the distinct terms in the corpus in order to create a BoW representation of a document. The frequency of each word in the lexicon is then created as a vector for each page. Several machine learning algorithms can use this vector as input for tasks including sentiment analysis, topic modeling, and document classification.

Figure 10: Business KPIs Tracking

To summarize, we constructed vectors for multiple ICD codes that represent similar diagnoses or related diseases, such as fever combined with a cold or pneumonia, fractures accompanied by fever, or merely fever. Since fever is the recurring element, we generated vectors with the same diagnosis or ailment. We employed the Bag of Words model for analyzing textual diagnoses. Support Vector Machine (SVM) methods were used for ICD code analysis, and the Bag of Words model was applied to the textual diagnosis.

One important limitation of the BoW model is that it does not consider the order of words or their semantic meaning. Therefore, it can result in a loss of important information, particularly in tasks where the context of words is crucial, such as language translation and sentiment analysis. To overcome this limitation, more advanced techniques such as word embeddings and neural networks have been developed.

Table 2: Impact and Pending Amount

	April	May	June	July	Aug
Non-impacted	2.98M	4.25M	2.85M	4.7M	2.78M
Pending	0.15M	4.35M	0.42M	2.79M	0.54M
Impacted	1.07M	1.16M	1.03M	0.17M	0.27M

As shown in Figure 10, and represented in Table 2, the impacted amount, non-impacted amount, and pending amount are shown, i.e., an account of fraudulent claims impacted by the business, not impacted by the business, and amount accounting for the claims that have been categorized as pending to be classified as fraudulent or not the business and amount accounting for the claims that have been categorized as pending to be classified as fraudulent or not.

As shown in Figure 10, and represented in Table 3, we see the number of alerts classified as impact, i.e., fraudulent claims, non-impacted, i.e., legitimate claims, and pending claims, that are generated based upon the threshold that is given to the model.

Table 3: Segregation of Alerts

	April	May	June	July	Aug
Impact	9	20	9	4	----
No-impact	19	25	18	44	5
Pending	1	5	4	34	7

Table 4: Disposition of Alerts

	April	May	June	July	Aug
Alert generated	30	50	31	82	12
No alert generated	43	115	94	745	140

Table 4 shows the disposition of alerts generated. The above Figure 10 shows the business outcomes of the model and its significance. These are essentially the KPIs that comprehensively track business results and their disposition at any given amount of time. It directly reflects the cost savings of the model at any given pointing time at any given point of time.

A. KPI NOMENCLATURE

Count of Claims Scored - This KPI effectively measures how many claims have been processed via the model and compared to a predetermined threshold to determine whether they are fraudulent**.**

Alerts generated – This KPI evaluates the number of fraudulent claims that have been generated in comparison to the model's threshold. If we give the claim a score that is higher than the threshold, we declare it to be fraudulent.

Total No of Claims Non-Impacted- This KPI measures the overall number of claims on which the company had no bearing**.**

Total No of Claims Pending- This KPI measures the total number of claims that are still being processed, such as those that are being assessed, reimbursed, or subject to more scrutiny**.**

Scored Claimed Amount - This KPI provides the total amount value of all fraudulent claims that the model has assessed. Its formula is as follows:

$$\sum(Claimed\ Amount) \quad (1)$$

Amount Impacted - This KPI provides a total amount accounting of all fraudulent claims that have been impacted by the business. Its formula is as follows:

$$\sum(Impacted\ Amount) \quad (2)$$

Pending Amount - This KPI provides a total amount accounting of all claims that have been scored by the model under the disposition of pending. Its formula is as follows:

$$\sum(Pending\ Amount) \quad (3)$$

Alert Rate - The model's capacity to differentiate between claims it scores as fraudulent and the overall number of claims it has validated is described by the alert rate KPI. Its formula is as follows:

$$\sum(Alerts\ generated)/\sum(claims) \quad (4)$$

Impact Rate - The success rate KPI describes the model's capacity to separate the claims that are impacted from the total claims that are both unaffected and impacted. Its formula is as follows:

$$\frac{\sum(claims\ impacted)}{\sum(impacted+non\_impacted)} \quad (5)$$

Model Output Rate - The model ratios to the total number of claims impacted to the total number of alerts created are what is referred to as the model output rate KPI. Its formula is as follows:

$$\frac{\sum(claims\ impacted)}{\sum(impacted+on-impacted+pending)} \quad (6)$$

Disposition of Claims Scored- This KPI displays the distribution of the total number of claims that the model passed in comparison to the monthly alerts that were generated.

Disposition of Alerts- This KPI shows the model's classification of all warnings as fraudulent into three categories: impact, no-impact, and pending claims. In a stacked bar chart, these claims are shown as monthly distributions.

Impact and Pending - This KPI shows how the model divides the impacted amount and pending amount against the impacted claims and pending claims.

Table 5, gives an ad-hoc analysis of the claim appraisers and stakeholders of business every week. It shows the implementation of the KPI nomenclature resulting in a Power BI dashboard.

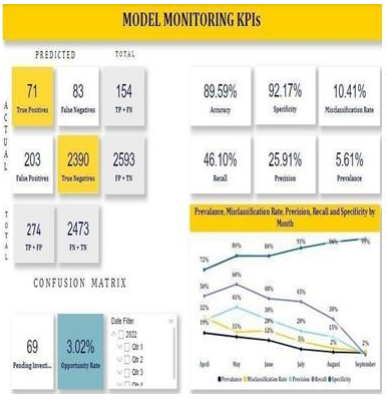


Figure 11: Model Monitoring Dashboard

Figure 11 above, is a model monitoring Dashboard, using Power BI, that corresponds to the results of the log table created at its backend. The log table for model monitoring is a week-by-week record of cases with their claim amount, generated alerts, true of cases with their claim amount, generated alerts, true negatives, false positives, pending investigations, etc., formulated to calculate

Table 5: Business Spreadsheet View in Tabular Format Replicated as per PowerBI Dashboard

Week	Scored Claim Count	Alert Generated	Alert Rate	Impact	No Impact	Pending	Impacted Amount	Pending Amount
18-04-2022	21	21	100%	5	14	1	₹ 7,85,023	₹ 1,49,738
25-04-2022	52	9	17%	4	5	0	₹ 2,91,343	₹ 0
02-05-2022	64	17	27%	2	8	2	₹ 4,66,840	₹ 10,30,731
09-05-2022	21	5	24%	7	3	0	₹ 96,394	₹ 0
16-05-2022	51	13	25%	6	7	0	₹ 3,88,027	₹ 0
23-05-2022	29	15	52%	5	7	3	₹ 2,17,975	₹ 33,27,892
06-06-2022	20	16	80%	5	8	3	₹ 3,25,740	₹ 1,51,941
13-06-2022	79	13	16%	4	8	1	₹ 7,13,510	₹ 2,72,685
20-06-2022	10	1	10%	0	1	0	₹ 0	₹ 0
27-06-2022	45	4	9%	0	2	2	₹ 0	₹ 1,34,366
04-07-2022	61	6	10%	0	2	4	₹ 0	₹ 3,32,777
11-07-2022	461	47	10%	1	35	11	₹ 63,798	₹ 12,77,990
18-07-2022	12	1	8%	0	1	0	₹ 0	₹ 0
25-07-2022	264	25	9%	3	5	17	₹ 1,15,500	₹ 10,51,339
01-08-2022	112	10	9%	0	5	5	₹ 0	₹ 3,07,698
08-08-2022	40	2	5%	0	0	2	₹ 0	₹ 2,41,130
Total	1342	205	15%	42	111	51	₹ 34,64,150	₹ 82,78,287

Table 6: Model Evaluation Matrices

	April	May	June	July	Aug	Sept
Accuracy	68%	80%	80%	89%	94%	97%
Prevalence	19%	15%	12%	5%	2%	2%
MFRate	32%	20%	20%	11%	6%	3%
Precision	29%	41%	30%	20%	15%	0%
Recall	50%	60%	48%	45%	30%	0%
Specificity	72%	84%	84%	91%	96%	99%

accuracy, specificity, precision, recall, and other matrices to track the model performance. Documentation for the creation and engineering of the A&H model on a weekly basis for proof of work was done.

Confusion matrices are a useful tool for solving classification problems, whether binary or multi-class. They show the expected and observed counts, with True Negative (TN) indicating correctly identified negative cases, True Positive (TP) indicating correctly identified positive cases, False Positive (FP) indicating negative cases mistakenly categorized as positive, and False Negative (FN) indicating positive cases mistakenly categorized as negative.

Four fundamental properties (numbers) make up the confusion matrix, which is used to provide the classifier's measurement parameters. These are the four numbers:

1. TP (True Positive): TP denotes the number of claims that the model has correctly identified as fraudulent
2. or otherwise impacted.
2. TN (True Negative): TN is a measure of how many accurately identified claims—those that are not fraud—are not scored by the model.
3. FP (False Positive): FP refers to the number of claims that the model flagged as fraudulent but that the appraisers later determined weren't. Another name for FP is a Type I mistake.
4. FN (False Negative): FN stands for the number of claims that the model incorrectly categorized as non-fraudulent but which were fraudulent. FN is another name for a Type II error.

B. Formulas for calculating evaluation metrics:

$$Accuracy = (TP + FP)/(TP + TN + FP + FN) \quad (7)$$
$$Precision = TP/(TP + FP) \quad (8)$$
$$Sensitivity = TP/(T + TP) \quad (9)$$
$$Specificity = TN/(TN + FP) \quad (10)$$
$$F1\ score = (2 * P * Sn)/(P + Sn) \quad (11)$$

The report may include information on the claims being appraised, such as its location, size, condition, and any unique features. The report may include an assessment of the claims value based on approaches to simplify the understanding of the alert levels.

Table 7: Output Format of the Results Viewed by Business and Claim Appraisers based on the alert level

Alert Date	**LGIL Claim Number**	**City**	**DOA**	**ICD**	**Claimed Amount**	**Model Score**	**Fraud Alert**
09-06-2022	44721X-XXXXXX-XXXXX	Garhi Harsaru	24-03-2022	K29	₹ 72,498	0.801195388	Very High

09-06-2022	44721X-XXXXXX-XXXXX	BAHRAICH	24-04-2022	K29	₹ 61,789	0.597087051	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Arjun Nagar	13-04-2022	R50	₹ 47,152	0.593037293	Medium
09-06-2022	44721X-XXXXXX-XXXXX	NEW DELHI	23-04-2022	A01	₹ 91,294	0.532344839	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Chorasi	09-04-2022	A01	₹ 46,692	0.530647463	Medium
09-06-2022	44721X-XXXXXX-XXXXX	BHIWANDI	26-04-2022	A75	₹ 93,254	0.512575094	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Daskroi	16-04-2022	A01	₹ 64,388	0.512534948	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Huzur	01-05-2022	A01	₹ 86,474	0.507008752	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Daskroi	08-04-2022	N20	₹ 1,11,145	0.506592672	Medium
09-06-2022	44721X-XXXXXX-XXXXX	Ghaziabad	06-06-2022	H65	₹ 55,455	0.503222135	Medium
10-06-2022	44722X-XXXXXX-XXXXX	Bangalore	25-05-2022	T84	₹ 2,27,254	0.572117505	Medium
10-06-2022	44722X-XXXXXX-XXXXX	Bangalore	25-05-2022	T84	₹ 74,930	0.560079825	Medium

Table 7: Output Format of the Results Viewed by Business and Claim Appraisers based on the alert level

	Precision	Recall	F1- score	Support	Accuracy	Macro Avg	Weighted Avg
False	0.99	0.99	0.99	21146	21262	0.50	0.99
True	0.00	0.00	0.00	116	0.99	21262	21262

LOF Score: 223	**LOF 0.9895118050982974**

Table 8: Model Performance Results

The output format of the results viewed by business and claim appraisers will depend on the specific tool or system being used. However, typically these professionals will be presented with a detailed report or summary of the appraisal results based on alert levels that have been used.

Additionally, the report may include an assessment of the claims value based on various appraisal methods, such as the sales comparison approach, the cost approach, or the income approach as seen in Table 7. Matrices and Scores as seen from the code after model training are represented in Table 8**.**

5. DISCUSSIONS

The suggested model analysis of data claims to find potential fraud cases using a combination of conventional statistical methods and machine learning techniques. The

model first performs a pre-processing step that includes data cleaning, imputation, and normalization of the claims data. The following step is to use established statistical methods to identify outliers and anomalies in the data. The techniques covered here include clustering, regression analysis, and hypothesis testing. In addition, the model makes use of unsupervised learning techniques like clustering and anomaly detection to identify groups of claims with comparable traits and to identify claims that stand out from the rest of the data. As a result, it is simpler to identify claims that might be a component of a larger fraud scheme. The model's accuracy, precision, recall, and F1 score are evaluated. A dashboard used to track the model's performance during the assessment process helps to visualize the results. The benefits that are inculcated from the model developed are reduced referral cycle time, i.e. less time consumed in reviewing claims, identifying claims based on fraud alert severity, automated result tracking via the dashboard, no direct costs involved in deployment, and use case of a reusable framework.

Challenges included difficulties with data acquisition, a difficult time choosing features because of data imbalance and presenting unidentified fraudulent claims in the data. To resolve these challenges, SMOTE was used, model-specific data selection was carried out, and feature selection was improved by incorporating NLP methodologies into pertinent dataset data. Undiscovered incidents of fraud in the data were taken care of by anomaly detection. Approaching the problems by trial and error worked best with a dataset that came with certain challenges. It was important to get an understanding of the dataset, what it represents, the terminologies and calculations, and the workings of healthcare firms to ensure we relied on the correct features to achieve good accuracy for our model. These were some of the key takeaways. Using the same technical foundation, additional models can be added. To detect more fraud, sophisticated methods like deep learning and network analysis can be used. At various points during the claim

lifecycle, models may be scored, and learning and network analysis can be used. At various points during the claim lifecycle, models may be scored.

6. CONCLUSION

The proposed model offers a thorough method for identifying health insurance fraud claims, to conclude using an approach of anomaly detection LOF with a very high accuracy of 99.5% which in turn results in cost savings for any organization. Thus the model offers a solid and dependable answer to the issue of health insurance fraud by combining conventional statistical techniques and machine learning algorithms. The model has demonstrated promising results after being tested and validated on actual data, making it an important tool for insurance companies to lessen the effects of fraud. It can track down cases based on its fraud alert level which alerts the appraisers to give high priority to complex claims and ease their burden by appraising simple claims automatically. The comprehensive business dashboard elevates and tracks the business impact of the model actively and the model monitoring dashboard tracks that the model is performing well and that there are no data population changes in the model.

Suggestions for future work to expand upon the work done in this work so far are presented here. More potential scenarios need to be investigated in order to improve the results that have already been given.

Additional models with sophisticated techniques such as deep learning and network analysis can be leveraged to identify more fraud claims in the models that can be scored at different times throughout the claim lifecycle.

REFERENCES

[1] N. Kumaraswamy, M. K. Markey, T. Ekin, J. C. Barner, and K. Rascati, "Healthcare Fraud Data Mining Methods: A Look Back and Look Ahead," *Perspect Health Inf Manag*, vol. 19, no. 1, p. 1i, 2022.

[2] H. Joudaki *et al.*, "Using Data Mining to Detect Health Care Fraud and Abuse: A Review of Literature," *Glob J Health Sci*, vol. 7, no. 1, pp. 194–202, Aug. 2014, doi: 10.5539/gjhs.v7n1p194.

[3] Markovskaia N., "Detecting Insurance Fraud with Machine Learning," *Plug and Play Tech Center*, Jul. 09, 2020.

[4] R. D. Burri, R. Burri, R. R. Bojja, and S. R. Buruga, "Insurance Claim Analysis using Machine Learning Algorithms," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6S4, pp. 577–582, Jul. 2019, doi: 10.35940/ijitee.F1118.0486S419.

[5] S. Rawat, A. Rawat, D. Kumar, and A. S. Sabitha, "Application of machine learning and data visualization techniques for decision support in the insurance sector," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100012, Nov. 2021, doi: 10.1016/j.jjimei.2021.100012.

[6] V. Rawte and G. Anuradha, "Fraud detection in health insurance using data mining techniques," in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, IEEE, Jan. 2015, pp. 1–5. doi: 10.1109/ICCICT.2015.7045689.

[7] S. S. Waghade and A. M. Karandikar, "A Comprehensive Study of Healthcare Fraud Detection based on Machine Learning," *International Journal of Applied Engineering Research*, vol. 13, no. 6, pp. 4175–4178, 2018.

[8] J. K. Gill and S. Aghili, "HEALTH INSURANCE FRAUD DETECTION," Dec. 2020.

[9] Thotakura Lalithagayatri, Tawde Priyanka, and Aruna Pavate, "Fraud Detection in Health Insurance using Hybrid System," *International Journal of Engineering Research and Technology (IJERT)*, vol. 5, no. 1, pp. 1–3, 2017.

[10] J. Li, Q. Lan, E. Zhu, Y. Xu, and D. Zhu, "A Study of Health Insurance Fraud in China and Recommendations for Fraud Detection and Prevention," *Journal of Organizational and End User Computing*, vol. 34, no. 4, pp. 1–19, Apr. 2022, doi: 10.4018/JOEUC.301271.

[11] R. Kunickaitė, M. Zdanavičiute, and T. Krilavičius, "Fraud Detection in Health Insurance Using Ensemble Learning Method," *International Conference on Information Technology.*, pp. 70–77, 2020.

[12] G. Baader and H. Krcmar, "Reducing false positives in fraud detection: Combining the red flag approach with process mining," *International Journal of Accounting Information Systems*, vol. 31, pp. 1–16, Dec. 2018, doi: 10.1016/j.accinf.2018.03.004.

[13] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. El Bastawissy, "A proposed model for data warehouse ETL processes," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, no. 2, pp. 91–104, Jul. 2011, doi: 10.1016/j.jksuci.2011.05.005.

[14] F. Ying-lan and H. Bing, "Design and Implementation of ETL Management Tool," in *2009 Second International Symposium on*

Knowledge Acquisition and Modeling, IEEE, 2009, pp. 446–449. doi: 10.1109/KAM.2009.105.

[15] X. Jiang, K. Lin, Y. Zeng, and F. Yang, "Medical Insurance Medication Anomaly Detection based on Isolated Forest Proximity Matrix," in *2021 16th International Conference on Computer Science & Education (ICCSE)*, IEEE, Aug. 2021, pp. 512–517. doi: 10.1109/ICCSE51940.2021.9569723.

[16] M. Wang, L. Xu, and L. Guo, "Anomaly Detection of System Logs Based on Natural Language Processing and Deep Learning," in *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, IEEE, Sep. 2018, pp. 140–144. doi: 10.1109/ICFSP.2018.8552075.

[17] W. Li, P. Ye, K. Yu, X. Min, and W. Xie, "An abnormal surgical record recognition model with keywords combination patterns based on TextRank for medical insurance fraud detection," *Multimed Tools Appl*, vol. 82, no. 20, pp. 30949–30963, Aug. 2023, doi: 10.1007/s11042-023-14529-4.

[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[19] J. M. Johnson and T. M. Khoshgoftaar, "Medical Provider Embeddings for Healthcare Fraud Detection," *SN Comput Sci*, vol. 2, no. 4, p. 276, Jul. 2021, doi: 10.1007/s42979-021-00656-y.

[20] J. M. Johnson and T. M. Khoshgoftaar, "Semantic Embeddings for Medical Providers and Fraud Detection," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, IEEE, Aug. 2020, pp. 224–230. doi: 10.1109/IRI49571.2020.00039.

[21] J. M. Johnson and T. M. Khoshgoftaar, "Encoding High-Dimensional Procedure Codes for Healthcare Fraud Detection," *SN Comput Sci*, vol. 3, no. 5, p. 362, Jul. 2022, doi: 10.1007/s42979-022-01252-4.

[22] F. Lacruz and J. Saniie, "Applications of Machine Learning in Fintech Credit Card Fraud Detection," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, May 2021, pp. 1–6. doi: 10.1109/EIT51626.2021.9491903.

[23] M. Amiruzzaman, R. Rahman, Md. R. Islam, and R. M. Nor, "Evaluation of DBSCAN algorithm on different programming languages: An exploratory study," in *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, IEEE, Nov. 2021, pp. 1–6. doi: 10.1109/ICEEICT53905.2021.9667925.

[24] M. Diaz-Granados, J. Diaz-Montes, and M. Parashar, "Investigating insurance fraud using social media," in *2015 IEEE International Conference on Big Data (Big Data)*, IEEE, Oct. 2015, pp. 1344–1349. doi: 10.1109/BigData.2015.7363893.

[25] Y. Wang and W. Xu, "Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud," *Decis Support Syst*, vol. 105, pp. 87–95, Jan. 2018, doi: 10.1016/j.dss.2017.11.001.

Prof. Jyoti Lele
Assistant Professor, MIT World Peace University, Pune

Jyoti Lele graduated in Electronics Engineering from Shivaji University in the year 2000 and received M. E. Electronics (Digital Systems) from Pune University in 2009. She is pursuing a Ph.D. from the Department of Technology, SPPU, Pune. She has a strong enthusiasm for research and is particularly interested in computer vision, speech and music synthesis, artificial intelligence, and fuzzy logic. She is currently employed as an Assistant Professor in the School of Professor in the School of Electronics and Communication Engineering at Dr. Vishwanath Karad of the MIT World Peace University in Pune. Jyoti Lele has a total of 20 years of teaching experience and 1 year of industrial experience. She has published more than twelve research papers in journals and conferences at both the national and international levels. She is proficient in a number of programming languages, including C, C++, VHDL, Matlab, Python, etc., and also has experience working with biometric-related initiatives like fingerprint and face recognition and speaker identification. Her study focuses on the organization of synthetic music using various signal-processing methods and machine learning algorithms.

Dr. Vaidehi Deshmukh
Assistant Professor, MIT World

Peace University, Pune

Dr. Vaidehi Deshmukh is an experienced Electronics and Communication engineering faculty involved in teaching various engineering subjects like Signals and systems, Machine Learning, Python, and Java Programming at Dr. Vishwanath Karad of the MIT World Peace University in Pune. Her doctoral research involved fusion of two images using deep learning models and development of an image processing-based algorithm for the same. She has published a book on Image Fusion. She has handled research projects in image fusion, emotion detection using Convolutional neural networks, disease detection, etc. She has proficiency in MS Excel, Python Programming, and MATLAB and has collaborated with the industry assisting them in solving their problems. She has also completed a certificate course in Data Science.

Abhinav Chandra
Student, MIT World Peace University, Pune

Abhinav Chandra is a final year undergraduate student pursuing his major in Bachelor of Technology from Maharashtra Institute of Technology with a concentration in Electronics and Communication Engineering and a focus in Machine Learning and AI. He has expertise in data science with a focus on deep learning, machine learning, and image processing. He enjoys identifying patterns, deciphering their meaning, connecting them, and using his intuition in this way so that he can write a beautiful story with his data that goes from the beginning to creating an excellent market insight. The author also addresses himself as an enthusiast who enjoys working on issues involving business and quantitative analytics. He is a challenging artificial intelligence and machine learning engineer looking to deliver cutting-edge projects with a transforming theme.

Radhika Desai
Student, MIT World Peace University, Pune

The fourth author of this paper is Radhika Desai, a final year B.Tech Electronics and Communications Engineering student specializing in Artificial Intelligence and Machine Learning pursuing her Bachelor's degree from Maharashtra Institute of Technology. She has a keen interest in data science, and in the data-driven world, hopes to pursue a career that would allow her to work with impactful applications of data science, participating in distinguished projects.