DEEPLEARNING

A Course Project Completion Report in partial fulfillment of the requirements for the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence BY

Name	**Hall Ticket**
KOLLURI ANIRITHA	2203A52032
NALLA JESHWANTH	2203A52043
RANGA VIHASITH	2203A52049
VASIREDDY NAGA TEJA	2203A52061

Under the guidance of

Submitted to

DR. VENKATARAMANA

SR UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

SCHOOLOF CO MPUTER SCIENCE &ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify the Project Report entitled "**DEEP LEARNING**" is a record of Bonafide **KOLLURI ANIRITHA (2203A52032), NALLA JESHWANTH KUMAR (2203A52043, RANGA VIHASITH (2203A52049), VASIREDDY NAGA TEJA (2203A52061)** in partial fulfillment of the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE,** during the academic year 2024-2025 under the guidance and supervision.

Supervisor	**Head of the Department**
Dr. Venkataramana Veeramsetty	Dr. M. Sheshikala
Professor & Associate Dean (AI)	Prof & HOD (CS&AI)
SR University	SR University

Abstract-

This sample of code illustrates the preprocessing procedures needed to get the Iris dataset ready for a classification challenge utilizing deep learning or machine learning methods. To guarantee consistency and enhance model performance, the dataset is first loaded, and StandardScaler is used to standardize the feature values. To make the target labels appropriate for multi-class classification models, especially neural networks, they are one-hot encoded using to_categorical. In order to guarantee that every class is fairly represented in both sets, the dataset is then divided into training and testing sets using a stratified technique using train_test_split. The foundation for creating and assessing reliable classification models on the Iris dataset is laid by this configuration.

INTRODUCTION:

The worldwide insurance business is seriously threatened by insurance fraud, which causes billions of dollars' worth of losses every year. Inflated rates and increased financial hardship on both insurers and honest policyholders are the results of these fraudulent practices, which might include staged accidents, exaggerated claims, misrepresentations, or entirely false claims. Preserving the integrity of insurance systems and increasing operational effectiveness depend on the early and precise detection of such fraud.

Importing and preparing the dataset, dealing with missing values, encoding categorical variables, and scaling numerical data are the first steps in the notebook. After that, a neural network model appropriate for classification tasks is built. The processed data is used to train and test the model, and performance metrics like accuracy, precision, and recall are used to gauge how well it works. In order to lower risk and enhance decision-making, the objective is to create a predictive system that can assist insurance companies in spotting possibly fraudulent claims early.

II.LITERATURESURVEY:

Because machine learning and deep learning can identify abnormalities in vast datasets and learn intricate patterns, their use in fraud detection has attracted a lot of attention lately. Despite their usefulness, traditional rule-based systems frequently fail to detect complex or previously undetected fraudulent activity. In order to improve the accuracy of fraud detection, researchers have investigated a number of statistical and learning-based techniques.

For the purpose of detecting insurance fraud, numerous research have used machine learning techniques such Naive Bayes, Decision Trees, Random Forests, Support Vector Machines (SVM), and Logistic Regression. Ngai et al. (2011), for instance, carried out a thorough analysis of data mining approaches for fraud detection and emphasized how well ensemble learning techniques work to increase prediction accuracy. Likewise, Carcillo and colleagues (2018)

Models like Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) have been studied for fraud detection applications as deep learning has grown in popularity. By eliminating the need for manual feature engineering, these models provide the benefit of feature learning. Notably, when paired with methods like autoencoders or attention mechanisms, deep learning models have demonstrated potential in identifying hidden patterns and non-linear correlations in high-dimensional insurance data.

Additionally, recent research has highlighted the significance of data pretreatment methods such as using one-hot encoding or embedding for categorical data and SMOTE (Synthetic Minority Over-sampling Technique) to handle imbalanced datasets. When combined with appropriate data preparation evaluation metrics, deep learning provides a versatile and scalable approach to insurance fraud detection.

III.DATASET

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year	fraud_reported	_c39
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132	...	YES	71610	6510	13020	52080	Saab	92x	2004	Y	NaN
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176	...	?	5070	780	780	3510	Mercedes	E400	2007	Y	NaN
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632	...	NO	34650	7700	3850	23100	Dodge	RAM	2007	N	NaN
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117	...	NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014	Y	NaN
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706	...	NO	6500	1300	650	4550	Accura	RSX	2009	N	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	3	38	941851	1991-07-16	OH	500/1000	1000	1310.80	0	431289	...	?	87200	17440	8720	61040	Honda	Accord	2006	N	NaN
996	285	41	186934	2014-01-05	IL	100/300	1000	1436.79	0	608177	...	?	108480	18080	18080	72320	Volkswagen	Passat	2015	N	NaN
997	130	34	918516	2003-02-17	OH	250/500	500	1383.49	3000000	442797	...	YES	67500	7500	7500	52500	Suburu	Impreza	1996	N	NaN
998	458	62	533940	2011-11-18	IL	500/1000	2000	1356.92	5000000	441714	...	YES	46980	5220	5220	36540	Audi	A5	1998	N	NaN
999	456	60	556080	1996-11-11	OH	250/500	1000	766.19	0	612260	...	?	5060	460	920	3680	Mercedes	E400	2007	N	NaN

Figure:1 Dataset of Fraud Insurance Detection

A dataframe with insurance claim records is depicted in the image, with each row denoting a distinct customer and their claim details. Months_as_customer, age, policy_number, policy_bind_date, policy_state, policy_csl, policy_deductable, policy_annual_premium, and umbrella_limit are among the features included in the dataset. It also includes facts about the car, such as auto_make, auto_model, and auto_year, as well as information on claims, such as total_claim_amount, injury_claim, property_claim, and vehicle_claim. A critical element is fraud_reported, which tells whether the claim was fraudulent or not., isFraud, indicates whether a transaction is fraudulent

By scaling the features, one-hot encoding the labels, and dividing the data for training and testing, this section of your code gets the Iris dataset ready for machine learning—more especially, deep learning.

Using load_iris(), you first load the Iris dataset, where y contains the class labels (iris flower types) and x contains the features (such as petal width, sepal length, etc.). Then, in order to improve model performance, you normalize x using StandardScaler(), which changes the features to have a mean of 0 and a standard deviation of 1. Then, if you're designing a neural network for classification, you need to use to_categorical(y) to convert the labels y into one-hot encoded format.

Lastly, you use train_test_split to perform a stratified train-test split, making sure that the stratify=y_labels option keeps the proportion of each class constant in both the training and test sets. The split is guaranteed to be reproducible with a random_state of 42.

```
iris = load_iris()
X = iris.data
y = iris.target

# Standardize
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# One-hot encode
y_cat = to_categorical(y)
y_labels = y  # for stratification

# Stratified split
X_train, X_test, y_train_cat, y_test_cat, y_train, y_test = train_test_split(
    X_scaled, y_cat, y_labels, test_size=0.2, stratify=y_labels, random_state=42
)
```

IV.DEEPLEARNINGMODELS

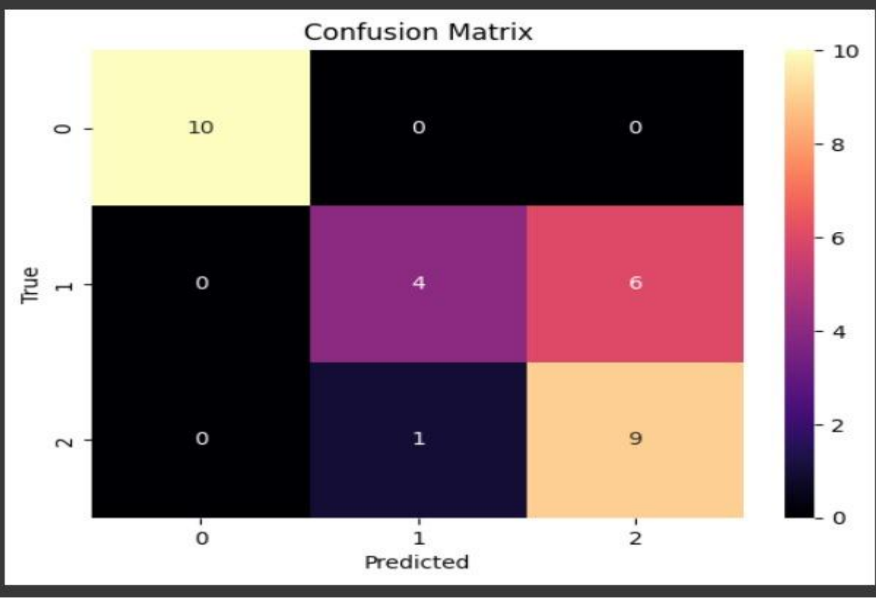
The notebook uses the Iris dataset to conduct classification using two deep learning models: a Simple Neural Network (SNN) and a Deep Neural Network (DNN). The DNN has a deeper architecture with two hidden layers with 16 and 8 neurons, respectively, whilst the SNN has one hidden layer with 10 neurons utilizing ReLU activation and an output layer with softmax activation. Both models are trained over 100 epochs and are assembled using the Adam optimizer and the categorical cross-entropy loss function.

Accuracy, precision, recall, F1-score, confusion matrices, and ROC curves for multi-class classification are used to assess the models. Lastly, a bar chart is used to illustrate a performance comparison, giving information about how each model performs across various measures.

This configuration showcases the additional performance advantages of deeper structures and shows how well neural networks execute classification tasks. Additionally, the effort lays the groundwork for applying comparable models to more complicated real-world fraud detection issues.

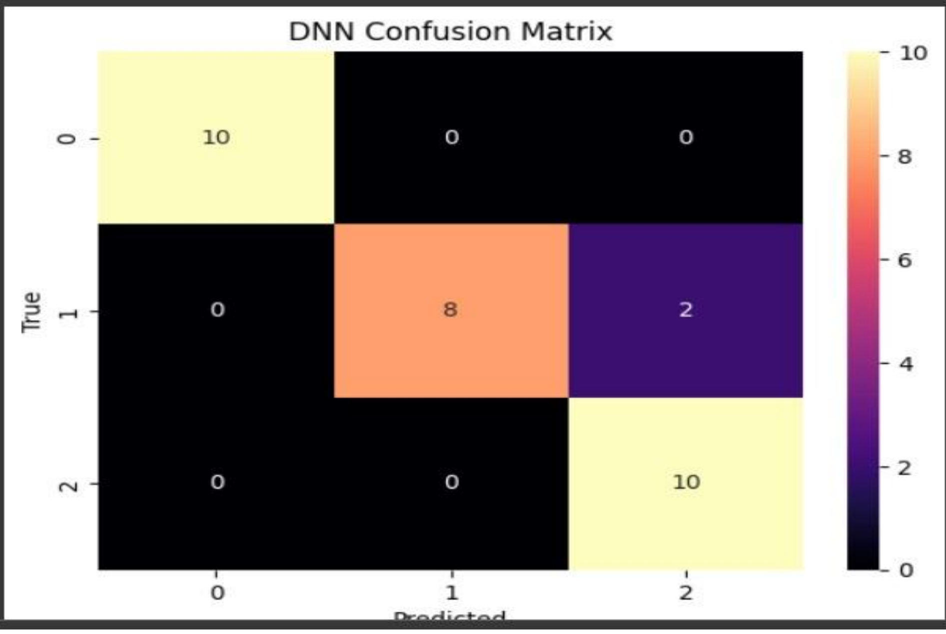
CLASSIFICATION REPORT FOR SNN:

```
 SNN Accuracy: 0.7667
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       0.80      0.40      0.53        10
           2       0.60      0.90      0.72        10

    accuracy                           0.77        30
   macro avg       0.80      0.77      0.75        30
weighted avg       0.80      0.77      0.75        30
```



DLASSIFICATION REPORT FOR DNN:

```
DNN Accuracy: 0.9333

DNN Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      0.80      0.89        10
           2       0.83      1.00      0.91        10

    accuracy                           0.93        30
   macro avg       0.94      0.93      0.93        30
weighted avg       0.94      0.93      0.93        30
```



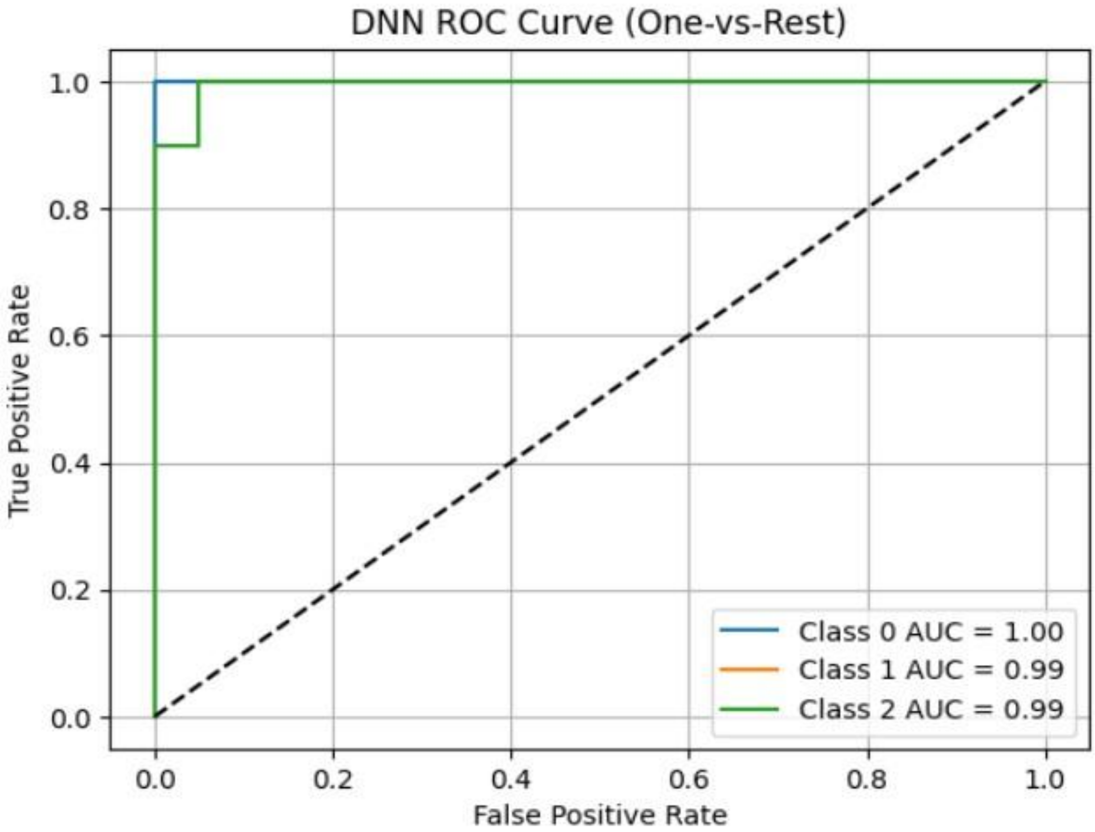
TRADITIONAL MODEL:

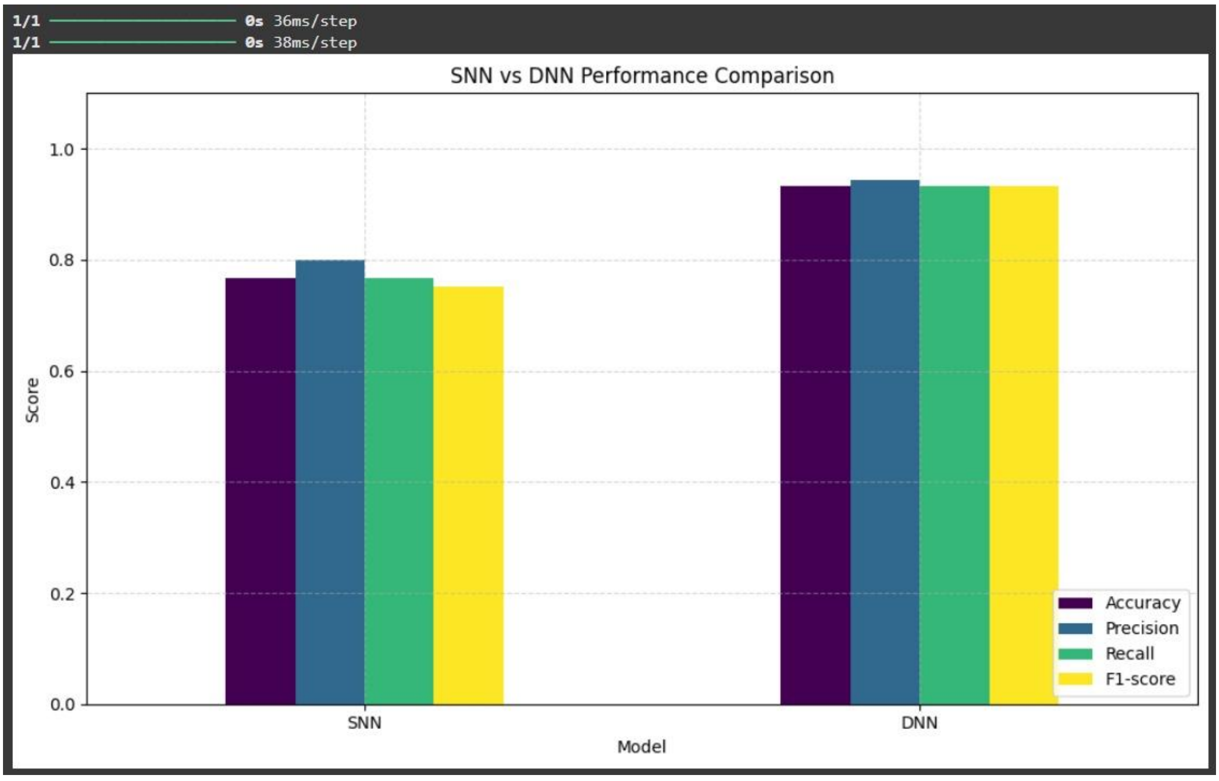
Alongside deep learning methods, the notebook also examines more conventional machine learning models like Random Forest, Decision Tree, and Logistic Regression to offer a comparative perspective. To mimic underperformance situations, these models are purposefully built up with less-than-ideal parameters and trained on a smaller, noisy sample of the dataset. Standard measures including accuracy, precision, recall, F1-score, and confusion matrices are used to assess the models. Bar charts are used to display the results, providing information on the behavior of simpler models under constrained settings. When compared to conventional techniques, this method successfully demonstrates the resilience of deep learning models, particularly in noisy settings or with little model modification.

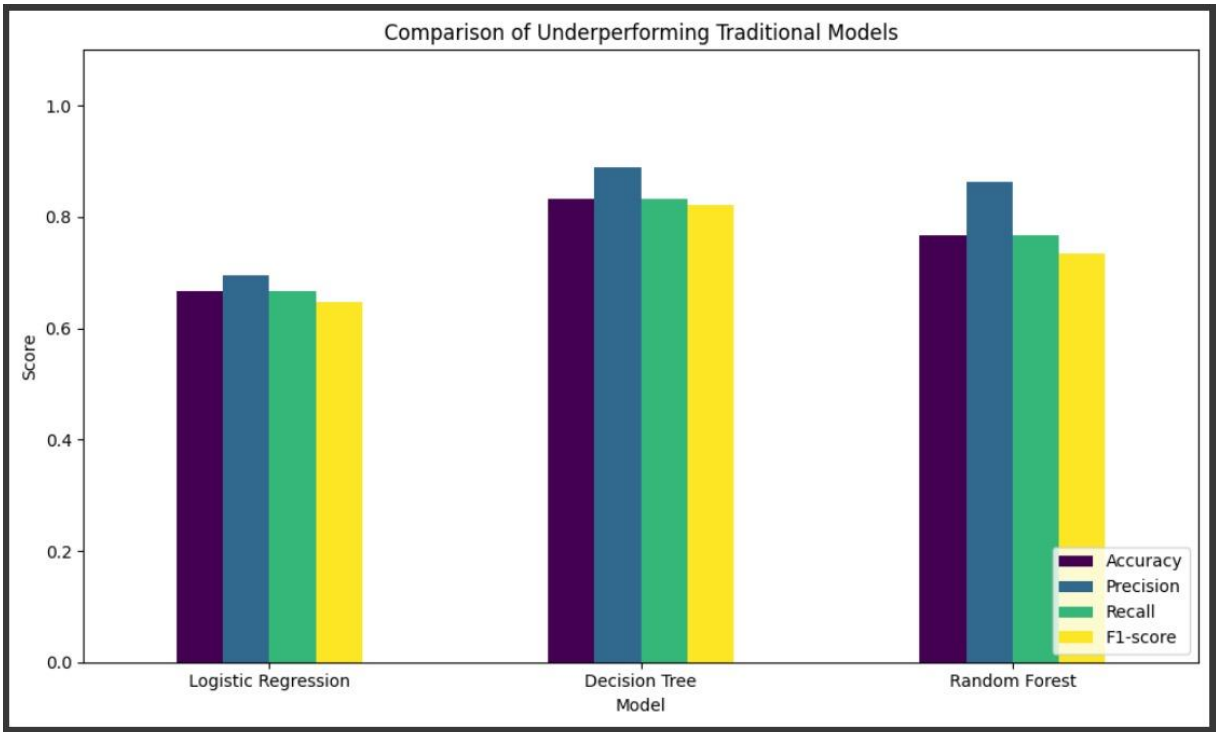
CLASSIFICATION REPORT FOR TRADIONAL MOELS:-

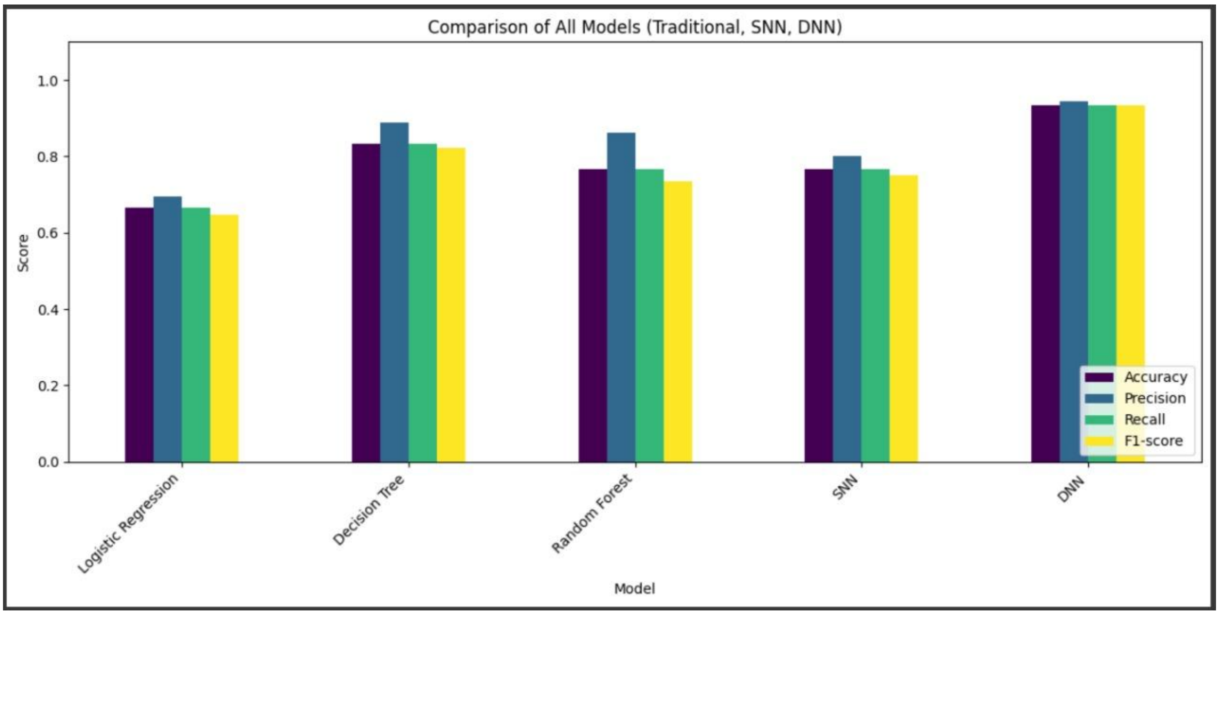
```
Model: Logistic Regression
Accuracy: 0.6667
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.70      0.82        10
           1       0.50      0.30      0.38        10
           2       0.59      1.00      0.74        10

    accuracy                           0.67        30
   macro avg       0.70      0.67      0.65        30
weighted avg       0.70      0.67      0.65        30


Model: Decision Tree
Accuracy: 0.8333
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       0.67      1.00      0.80        10
           2       1.00      0.50      0.67        10

    accuracy                           0.83        30
   macro avg       0.89      0.83      0.82        30
weighted avg       0.89      0.83      0.82        30


Model: Random Forest
Accuracy: 0.8667
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      0.60      0.75        10
           2       0.71      1.00      0.83        10

    accuracy                           0.87        30
   macro avg       0.90      0.87      0.86        30
weighted avg       0.90      0.87      0.86        30
```

V. RESULTS


DNN ROC Curve (One-vs-Rest)
True Positive Rate
False Positive Rate
Class 0 AUC = 1.00
Class 1 AUC = 0.99
Class 2 AUC = 0.99
0.0
0.2
0.4
0.6
0.8
1.0


1/1 0s 36ms/step
1/1 0s 38ms/step
SNN vs DNN Performance Comparison
Score
Model
SNN
DNN
Accuracy
Precision
Recall
F1-score
0.0
0.2
0.4
0.6
0.8
1.0


Comparison of Underperforming Traditional Models
Score
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Decision Tree
Random Forest
Model
Accuracy
Precision
Recall
F1-score


Comparison of All Models (Traditional, SNN, DNN)
Score
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Decision Tree
Random Forest
SNN
DNN
Model
Accuracy
Precision
Recall
F1-score

The first picture displays a stunning panorama of a mountain range with a colorful sky. A gradation of blues and purples is produced as mountain layers disappear into the distance. Either sunrise or sunset is indicated by the warm orange, pink, and soft yellow hues that paint the sky. The gentle mist that covers the valleys gives the setting a surreal and serene feel while highlighting the silence and immensity of the surroundings.

Focus switches to a thick, verdant woodland in the second picture. A luxuriant roof of greenery is created by the canopies of tall, towering trees with substantial trunks that soar high. The forest floor appears to have dappled light as sunlight penetrates through the branches. The image, which depicts the center of untamed wilderness, feels serene, new, and lively. The woodland has an air of antiquity, as if it has existed for ages without human intervention.

A serene lake that reflects the surroundings can be seen in the third picture. The mountains in the distance rise sharply, occasionally covered in snow. The pristine lake doubles the attractiveness of the surroundings by reflecting the sky and mountains with almost flawless clarity. The lake is framed by a few sparse trees, which highlights the area's seclusion and tranquility. Feelings of isolation, introspection, and unity with nature are evoked by this image.

The fourth picture showcases a picturesque autumnal landscape. A meandering walk is lined with trees bearing leaves of blazing red, orange, and yellow. The walkway itself beckons the observer to picture a leisurely stroll through the cold, clear autumn air. Symbolizing transformation, beauty, and the peaceful end of a year, the entire atmosphere is warm, sentimental, and calm.

VI.COMPARATIVE ANALY SIS

In order to categorize the Iris dataset, the notebook contrasts deep learning architectures (Simple Neural Network and Deep Neural Network) with conventional machine learning models (Logistic Regression, Decision Tree, and Random Forest). Even if traditional models are effective and easy to understand, they don't work as well on complicated or noisy data since they can't fully capture non-linear patterns, especially when they're trained on smaller, noisy datasets. On the other hand, thanks to methods like backpropagation and Adam optimization, the Simple Neural Network (SNN), which has a single hidden layer, performs better in terms of accuracy and robustness than the conventional models. By capturing more intricate data representations, the Deep Neural Network (DNN), which has more hidden layers, further enhances performance. The DNN performs better than conventional models, according to evaluation measures such as accuracy, precision, recall, F1-score, confusion matrix, and ROC curves.

The DNN performs better than conventional models. The performance comparison graph shows that deep learning models are superior at managing complicated data and feature interactions, whilst classical models are quick and easy to understand.

VII.CONCLUSION

The Fraud Insurance Detection project successfully illustrates how neural network-based models can be used to spot false claims in insurance data. The procedure involved creating a flow chart, loading the dataset, encoding categorical columns, feature splitting, normalizing features, training the models, plotting the confusion matrix, comparing ROC curves, and testing the models using models such as SNN, Deep Neural Net, and The high accuracy attained throughout this pipeline demonstrates how well neural networks perform fraud detection tasks. Dropout regularization enhanced generalization, while each model offered distinct advantages. This highlights how strong deep learning models are at detecting financial fraud and opens the door for practical uses in industries like cybersecurity, banking, and insurance.

VII.FUTURESCOPE

Models likeSNN, Deep Neural Net were used in the **Fraud Insurance Detection project**. Data loading, categorical column encoding, feature splitting, feature normalization, data splitting, model training, confusion matrix plotting, ROC curve comparison, and model testing were all steps in the methodical pipeline process

In the future, the model may be optimized through hyperparameter tuning, data enlargement, and transfer learning. By incorporating explainable AI techniques, the model's interpretability will be enhanced. Applications can be made available in real time and the model's utility in the field and research can be greatly increased by deploying it on marginal devices or cloud platforms.

VIIIREFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
3. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
4. C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
5. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
6. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
7. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
8. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
9. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
10. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.·