International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Exploration of Deep Learning Algorithm for Fraud Detection using Spark Platform

Mrs. Srilatha Komakula[1*], Dr. M. Jagadeeshwar[2]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: Fraudulent activities pose an important threat in several areas, requiring robust and efficient mechanisms for detection. It is critical to halt fraudulent transactions since they have a long-term influence on financial circumstances. Anomaly detection has several essential applications for detecting fraud. This paper presents a novel fraud detection method using deep learning algorithms, combining Convolutional Neural Networks (CNNs) for feature extraction from transaction data and Long Short-Term Memory (LSTM) networks for capturing temporal dependencies in financial transactions, thereby enhancing robust and efficient detection mechanisms. This paper proposes a new framework that combines Spark with a deep learning technique. However, it compares the performance of deep learning approaches for credit card fraud detection with other machine learning algorithms, including CNN-LSTM, on three distinct financial datasets. This paper also employs several machine learning algorithms for fraud detection, such as random forest, SVM, and KNN. Various parameters are used in comparative analysis. Both the training and testing datasets achieved more than 96% accuracy. The text outlines the creation of a high-performance deep learning model for detecting credit card fraud. The paper proposes hybrid attention to integrate current time output with unit state, determining its weight, and optimizes accuracy using Adam optimization. It uses various machine learning methods for a comparative study using the proposed deep architecture.

Index Terms: *Fraud detection, deep learning, machine learning, online fraud, credit card frauds*

1. Introduction

In the digital age, fraud detection has become a significant concern. Artificial intelligence, particularly deep learning algorithms, has become a key tool in this fight. Apache Spark, a distributed computing platform, offers scalability and speed for analysing massive datasets and identifying fraudulent activities in real-time. Deep learning algorithms, including neural nets, autoencoders, recurrent neural networks, and convolutional neural networks, are used to detect complex patterns and anomalies in transaction data. Spark can parallelize the training and deployment of deep learning models on massive datasets, ensuring no fraudulent transactions are overlooked. Its real-time streaming capabilities keep the detection engine running, analysing transactions as they occur and preventing losses before they materialize. Spark integrates with popular deep learning frameworks like TensorFlow and PyTorch, making it an ideal choice for detecting and preventing fraud in the digital age. Responsibility involves assessing a person's responsibility for their actions. The rise of new technology has increased the opportunity for fraud, particularly in credit card use. Financial losses from fraud affect institutions, individual customers, and banks' reputations. Non-financial losses, which are difficult to quantify in the short term but become more apparent in the long run, are also a concern [1]. The customer will no longer be able to rely on his bank and will switch to a more reliable competitor. Since the introduction of credit card payments, fraud prevention has worked to prevent and detect fraudulent transactions. The Address Verification System, Card Verification Method, and Personal Identification Number are all fraud-prevention technologies. In contemporary banking, several fraud detection algorithms and machine learning technologies are applied. Using classification algorithms, each transaction is allocated to a risk category. Artificial intelligence algorithms create the model on the basis of databases.

The use of learning methods in fraud detection is a useful tool since it allows for the finding of patterns in large-scale datasets with many variables. The model is often a parametric function that predicts the likelihood of transaction fraud. Furthermore, fraudulent transactions often coincide in both time and location. Similarly, [2] investigated credit card fraud detection via individual. Fraudulent transactions often coincide in time and location, and credit card fraud detection can be improved by adding “statistical features” resulting from a feature. Machine learning (ML) is an AI subfield that trains processers to predict future outcomes based on known patterns in existing data [3], requiring attention to factors like quick reaction time, “cost sensitivity," and “feature pre-processing.”.Many studies have used ML models to solve a range of issues. Deep learning (DL) algorithms have been utilised in a variety of applications, such as data centres and mobile

[1]*Research Scholar, Department of Computer Science, Chaitanya (Deemed to be University), Warangal(Urban), Telangana, India*
[2]*Professor, Department of Computer Science. Chaitanya (Deemed to be University), Warangal(Urban), Telangana, India*
Corresponding Author Email: srilatha.kom@gmail.com

communications. Deep learning approaches have gained interest due to their promising outcomes in computer vision, NLP, and audio, but a small number of works have explored their use in identifying credit card fraud (CCF) [4]. It identifies CCF using a number of deep-learning techniques. Figure 1 depicts a payment card authorization mechanism. Biometric authentication is classified into three types: physiological, behavioural, and combination authentication [5], [6].

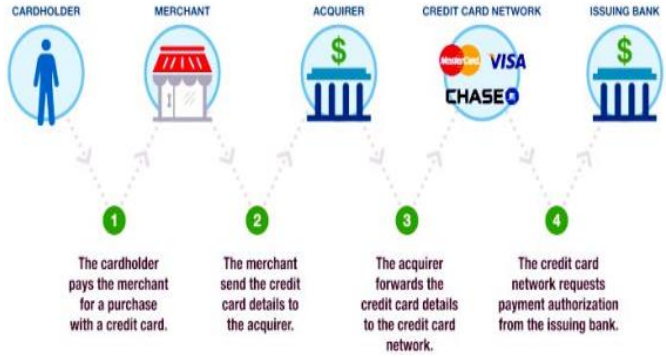

Fig 1: Payment card authorization process

This paper focuses on the performance of deep learning techniques like CNN and LSTM for credit card fraud classification. Classic machine learning techniques like SVM, Decision Trees, and Logistic Regression are suitable for small datasets. Deep learning models excel at detecting complex patterns and anomalies in large datasets, leading to higher accuracy in identifying fraudulent transactions compared to rule-based or shallow learning algorithms. The pre-built algorithms and tools in Spark MLlib facilitate deploying deep learning models on Spark clusters, simplifying the development and deployment process. The paper presents a novel deep learning framework for financial fraud detection using Spark, compares different machine learning architectures, and presents a unique stacked-based strategy for feature selection.

2. Background and Related Work

Financial institutions can navigate the digital landscape effectively by utilizing deep learning's ability to understand complex patterns and Spark's data processing and real-time analysis skills. These algorithms ensure transaction validity and secure the money. Convolutional neural networks, popularised by AlexNet's ImageNet Challenge win, have become increasingly used in computer vision [7, 8], with more sophisticated deep networks like VGGNet improving classification and recognition [9]. Deep residual networks (ResNets) have transformed the area of deep learning, notably for computer vision applications such as image recognition and object identification. Their significance arises from their capacity to solve a significant challenge in training deep neural networks: the vanishing gradient issue [10]. Similarly, in [11], the author said that in today's technological age, particularly in online business and banking, Mastercard transactions are fast rising. Mastercard has evolved into a very useful piece of equipment for Internet shopping. This increased usage creates significant harm as well as fraud charges. It is critical to halt fraudulent transactions since they have a long-term influence on financial circumstances. Anomaly detection has several essential applications for detecting fraud. Detecting fraud concerning online transactions may be the most difficult task for financial institutions and banks. As a result, financial institutions and banks must have highly effective fraud detection procedures to minimise their losses due to charge card fraud transactions. Many researchers have discovered various strategies to spot these scams while also taking them down. Following the study of the dataset, the dependability is 97% for LOF and 76% for IF. The author mentioned in [12] that in today's technological age, particularly in Internet commerce and banking, Mastercard transactions are fast rising. Mastercard has evolved into a very useful piece of equipment for Internet shopping. This increased usage creates significant harm as well as fraud charges. It is critical to halt fraudulent transactions since they have a long-term influence on financial circumstances. Anomaly detection has several essential applications for detecting fraud. This platform identifies fraud relatively rapidly, resulting in less loss and danger.

3. Proposed Modelling

This paper proposes a revolutionary framework that combines Spark with a deep learning technique. Figure 2 illustrates the suggested framework. This paper also employs several machine learning algorithms for fraud detection, such as random forest, SVM, and KNN.

Dataset Description

Credit card transaction datasets are crucial for machine learning models to identify fraudulent activity. They track customer spending habits, merchant categories, and geographic locations, helping understand customer preferences and segmenting. Lenders use these datasets to assess default risk and determine loan terms, while borrowers' credit history and income are also considered. The present paper aims to evaluate the effectiveness of classification models in detecting CCF using three distinct datasets: "European Card Data (ECD), Small Card Data (SCD), and Tall Card Data (TCD)." The datasets are skewed, with few fraud incidents compared to ordinary transactions. The datasets are labeled with '0' indicating no fraud and '1' indicating fraud.

European Card Data: The “machine Learning Group of Université Libre de Bruxelles” received a dataset from Kaggle [14], containing 284,807 European cardholder transaction data from September 2013. Only 492 of the samples were fraudulent, accounting for 0.172% of the total.

Due to transactional details' sensitivity and client information privacy. The dataset pertains to electronic commerce, which involves the purchase and sale of electronic items.

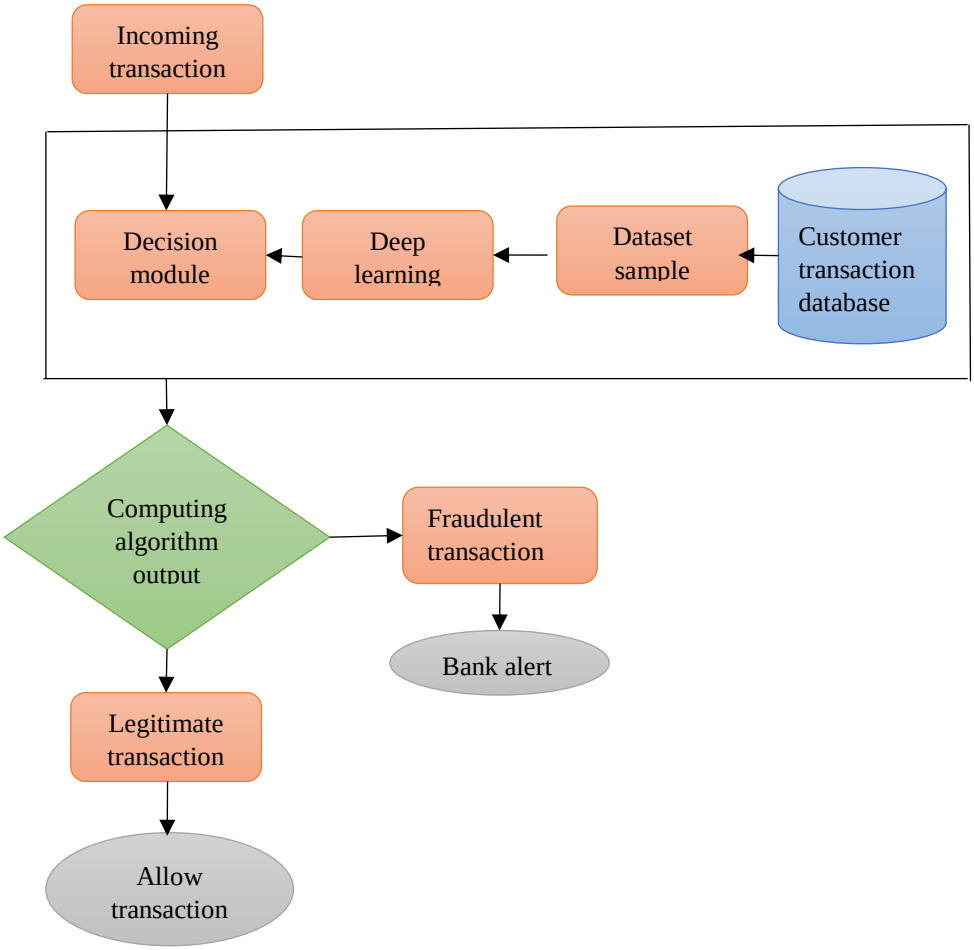

Fig 1: Proposed framework

Small Card Data: Electronic commerce refers to the sale of electronic goods. A dataset from Kaggle [15] contains 3075 samples and 12 features, including cash and fraud status. The dataset, known as SCD, has fewer rows and columns.

Tall Card Data: The paper uses Tall Card Data (TCD), an online database with 10 million samples and nine features. The dataset includes customer ID, gender, state, card balance, transactions, international transactions, credit line, and fraud risk [16]. Due to limited computational capacity and longer training durations. The "class imbalance" is maintained, with 28,000 fraud instances out of 500,000 total samples. The study aims to analyze the data using a small amount due to restricted computational capacity and longer training durations associated with classifiers.

Data Pre-Processing

Data preprocessing is crucial for developing effective machine learning models, especially for detecting credit card fraud. In this study, three datasets (ECD, SCD, and TCD) are inspected manually and statistically to provide updated input for classifiers. The goal is to generate the best possible output, which can be impacted by missing data, categorical features, variable scale, and high dimensionality. For ECD, all features were numeric, no missing values were found, and no features were removed. SCD's categorical features were transformed to numbers and the 'Transaction Date' feature was eliminated due to the large number of missing values. TCD was used in a smaller proportion of the dataset, with all variables numeric and the 'custID' feature removed. Correlation is found during data exploration to determine the dependability of variables. This helps eliminate data features with similar behavior, reducing data dimensions and allowing for quicker training and improved classification outcomes.

The SCD dataset shows no negative correlation between features and no consistent pattern as shown in Figure 3. Larger transaction amounts are associated with a higher average daily transaction number. The number of chargebacks per day, amount, and frequency are highly connected, making the high-risk country feature crucial for fraud classification. Due to the dataset's small size, no dimensionality reduction is performed.

Table 1: Data exploration

ECD	
Number of Rows	284807
Number of Columns	31
Feature Type	Numeric
Missing Values	None
Dropped Features	None
Categorical to Numeric	None
Smaller Sample Used	No
SCD	
Number of Rows	3075
Number of Columns	12
Feature Type	Numeric + Categorical
Missing Values	3075
Dropped Features	'Transaction date'
Categorical to Numeric	'Merchant_id', 'Is declined', 'isForeignTransaction', 'isHighRiskCountry', 'isFradulent'
Smaller Sample Used	No
TCD	
Number of Rows	10000000
Number of Columns	9
Feature Type	Numeric
Missing Values	None
Dropped Features	'custID'
Categorical to Numeric	None
Smaller Sample Used	Yes

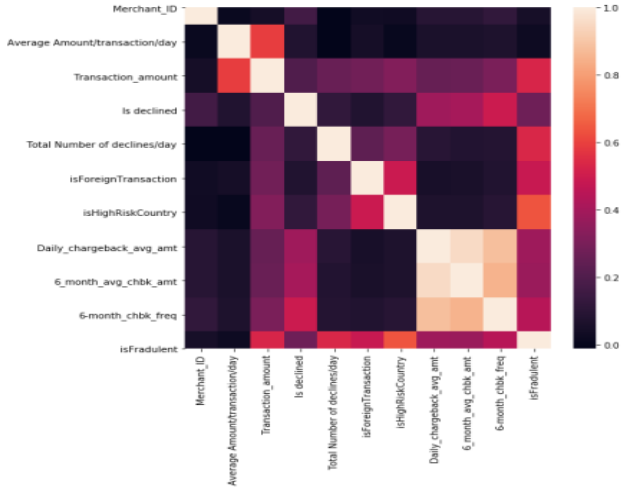

Fig 3: Correlation Heatmap of the SCD dataset

Data Scaling and Standardization

Machine learning involves crucial steps such as scaling and standardization to prepare data for efficient model training. These steps are essential when dealing with varied sizes of features in datasets. In the ECD dataset, the features 'Amount' and 'V1' have mean values. To make the input more intelligible, approaches like scaling and standardization aggregate features to nearly the same scale. In this paper, the "StandardScaler" and Python's "sklearn" are employed to convert each feature into a mean and standard deviation.

Test, Train and Validation Split

In machine learning, the test, train, and validation split is a fundamental technique for evaluating the performance of a model. It involves dividing the data into three sets: Electronic commerce refers to the sale of electronic goods. For training 60% data is used, Test data is a randomly chosen portion of 20%. Validation of 20% used to validate the classifier, preventing overfitting and enhancing model performance. This technique is crucial for any machine learning project to build accurate and generalizable models.

Apache Spark3

Apache Spark is known for its distributed computing capabilities, allowing it to process large volumes of data in parallel across a cluster of machines. This can significantly improve the speed of data processing, enabling faster fraud detection. Spark 3 introduces enhancements to Spark SQL and DataFrames, making it easier to express complex data manipulations and queries. This can simplify the preprocessing and analysis of credit card transaction data, leading to more effective fraud detection algorithms. MLlib, Spark's machine learning library, continues to evolve with each Spark release. Spark 3 may include improvements to existing machine learning algorithms, making it easier to develop and deploy more accurate models. Adaptive query execution, introduced in Spark 3, can dynamically optimize query plans based on runtime statistics. This can lead to more efficient processing of complex analytical queries, which is beneficial for tasks like feature engineering in fraud detection.

K-Nearest Neighbor (KNN)

KNN is used as classification for storing training data points and classifies new points based on their similarity to these points. It calculates the distance to all training points using a distance metric, assigns the most frequent class among its nearest neighbors, and finds the K training points. KNN is known for its ease of use, adaptability to non-linear problems, and sensitivity to local data structure. However, it's crucial to consider its computational cost, sensitivity to irrelevant features, and the right K value for specific tasks.

Support Vector Machine (SVM)

SVM is a classification algorithm that learns from labeled data to classify new unseen data points. It aims to find the decision boundary that maximizes separation between classes, leading to better generalization and robustness. Training support vector machines (SVMs) may be computationally costly, particularly for large data sets, despite their effective performance in many circumstances. Also, getting the parameters just right is key, particularly for picking the kernel and the regularisation parameter, and effectiveness in non-linear problems. However, it's crucial to consider computational cost and interpretability challenges when selecting an algorithm for a specific task.

CNN Model

CNN is a type of network that comprises Convolution, activation, and pooling layer.

Convolutional Layer. This is the fundamental process which comprises numerous convolution units, the parameters of which are improved via a backpropagation

method. Convolution operations are used to extract distinct characteristics from the input. Increasingly network layers may repeatedly extract more sophisticated information from low-level data. It's a small matrix of numbers that's used to perform convolution, a mathematical operation that balances or transforms an image based on the values in the kernel.

$$Z^{l+1}(i,j) = [Z^l \otimes w^l](i,j) + \sum_{k=1}^{K_l} \sum_{x=1}^{f} \sum_{y=1}^{f} Z_k^l . [(s_0 i + x, s_0 j + y) w_k^{l+1}(x,y)] + b, (i,j) \in \{0,1,\dots,L_{t+1}\} \quad (1)$$

$$L_{l+1} = \frac{L_l + 2P - f}{s_0} + 1$$

where Z^{l+1} Z (i, j) signifies "feature matrix"; f signifies "Kernel sixe".

Pooling Layer. A pooling layer is a crucial part of convolutional neural networks (CNNs) that reduces feature maps' spatial dimensions while maintaining crucial information. It is used to control overfitting and add spatial invariance. The output matrix is transferred to the pooling layer for feature extraction and filtering.

$$A_k^l(i,j) = \left[\sum_{x=}^{f} \sum_{y=1}^{f} A_k^l (s_0 i + x, s_0 j + y_-)^p \right]^{1/p} \quad (2)$$

LSTM stands for Recurrent Neural Network (RNN). Backpropagation based on gradients reduces loss in neural networks. Standard NNs cannot remember existing information and must begin learning from scratch. RNNs are neural networks with memory. As the gradient progresses back in RNN, the weight update decreases. RNN creates a short-term memory network, as the final layers lose learning capacity and recall early occurrences. LSTMs address this issue by providing a cell state as the network's memory and gates in each phase to preserve vital information while rejecting irrelevant information. There are forget, input, and output gates used. Forget, input, and output gates decide what to keep, add, and conceal. The hidden state recalls the model's previous inputs but has short-term memory, while the cell state remembers essential information from the beginning of the series. Figure 4 depicts the whole LSTM cell flow [13]. Each dotted box signifies a step. The following are the LSTM conversion functions:

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(W_i . [h_{t-1}, x_t] + b_f)$$

$$q_t = tanh(W_q . [h_{t-1}, x_t] + b_q) \quad (3)$$

$$o_t = \sigma(W_0 . [h_{t-1}, x_t] + b_0)$$

$$c_t = f_t . c_{t-1} + i_t . q_t$$

$$h_t = o_t . tanh(c_t)$$

where σ is in range of [0, 1], *tanh* signifies a "hyperbolic tangent function" in range of [−1, 1].

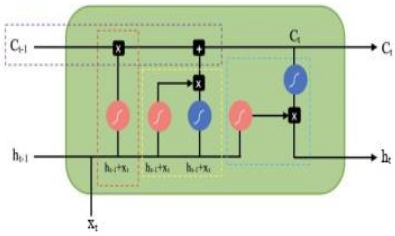

Figure 4: An LSTM Cell

CNN-LSTM.

The paper employs CNN and LSTM to address the challenge of learning information from variable-length sequence input. The improved CNN structure is comparable to ConvNets, with nine layers, six convolutional and three connected. RNNs, on the other hand, struggle with gradient issues as input sequence duration increases. LSTM calculates the need for new inputs and information loss from cell states using a gate mechanism in each unit. It also handles the retrieval of characteristics at word and character levels. The model is structured similarly for Char-Channel and Word-Channel channels, distinguished by embedding granularity. The convolution kernel and input sequence are sent to the CNN segment, which calculates the convolution result.

$$c = conv(X, K) + b \quad (4)$$

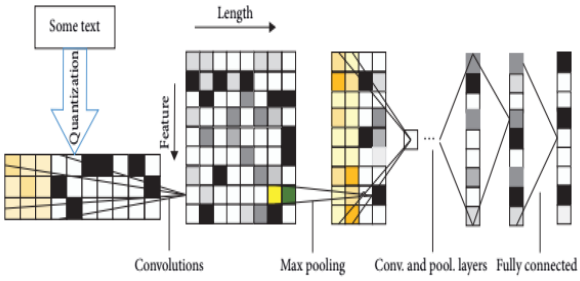

Fig 5: Structure of ConvNets model.

The LSTM technique has been consolidated into LSTM (x). Neural networks for convolution uses both short term and long term memory, can be structured in series or parallel. The series structure is commonly used, but it suffers from information loss due to data compression during convolution. The parallel system is selected in the place of series to capture all channels' structures simultaneously. This allows for the full use of LSTM's advantages in capturing time-series features in neural networks.

$$channel(x) = [conv(x) \oplus LSTM(x)] \quad (5)$$

$$C_{out} = channel_{emb=V_e}(x) \quad (6)$$

$$W_{out} = channel_{emb=V_g}(x) \quad (7)$$

V_e and V_g are word-level embedding vectors, and "two channel outputs" are combined to form a hidden layer output.

$$h = [c_{out} \oplus w_{out}] \quad (8)$$

The hidden layer's results are sent to the "fully connected layer", followed by the softmax.

$$y = softmax(dense(h)) \quad (9)$$

Hybrid Attention. *The "dynamic adaptive weight", a crucial component, is determined using Equations (10) and (11) to calculate the weight score ω.*

$$e_i = v_a^T tanh(W_a h_i + b) \quad (10)$$

$$h_i = \langle h_t' : c_t \rangle \quad (11)$$

where h_t' represents LSTM output, c_t *represents* LSTM status and *b* indicates bias. Subsequent, the score ω can be written as in

$$\omega = \frac{exp(e_i)}{\sum_{k=1}^{T_x} exp(e_i k)'} \quad (12)$$

where *x* indicates sequence length.

The dynamic adaptive weight is used to weight the output vector c_i, as illustrated in

$$c_i = \sum_{j=1}^{T_x} \omega . h_j \quad (13)$$

4. Results and Discussions

In this section, various machine learning techniques are tested for performance metrics on the dataset. A deep learning model is constructed with various ratios of training and testing sets. The study emphasises the importance of selecting appropriate grading techniques to assess the correctness of grading. The Confusion Matrix is used as a classification metric, displaying the generated data based on four possible combinations of expected and actual values: false positive (FP), true positive (TP), false negative (FN), and true negative (TN).

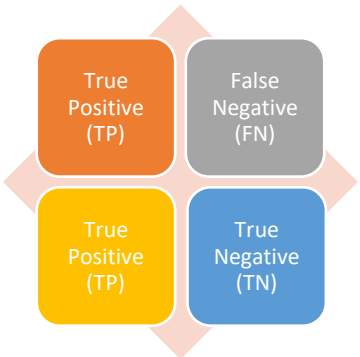

Fig 5: Confusion matrix. (Source: compiled by authors)

Accuracy measures the model's predictions, focusing on the fraction of transactions correctly classified as fraudulent or genuine. Precision measures the frequency of fraud detection, with high accuracy resulting in low false alarm rates. Recall measures the percentage of genuine fraud instances accurately classified as fraudulent, indicating the model's ability to detect true fraud cases. The F1 score, combining accuracy and recall, is beneficial for detecting fraud accurately and minimizing false alarms, ensuring a more comfortable experience for real customers.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

$$F1\ score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (17)$$

Area under the Curve (AUC): The area under the ROC curve is known as AUC, while the probability curve is known as ROC. To assess the model's outcomes, the worst AUC is 0, and the greatest AUC is near 1. Tables 2 and 3 reveal the results of using CNN-LSTM with varied training and testing split ratios. Figures 7 and 8 show the distribution of fraud scores for split ratios of 60-40 (Case 1), 70-30 (Case 2), and 80-20 (Case 3).

Table 2: Comparing different data splits on train dataset

Train data	Test data	Training accuracy	Time elapsed (sec)
(Case 1) 60	40	0.9774	14.21
(Case 2) 70	30	0.9687	16.57
(Case 3) 80	20	0.9537	18.54

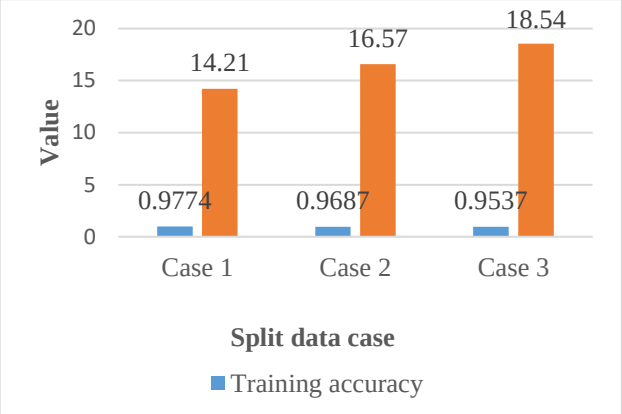

Figure 7: Fraud score distribution of training for split ratio

Table 3: Comparing different data splits on test dataset

Train data	Test data	Testing accuracy	Time elapsed (sec)
(Case 1) 60	40	0.9621	0.35
(Case 2) 70	30	0.9547	0.38
(Case 3) 80	20	0.9432	0.42

Figure 8: Fraud score distribution of testing for split ratio

Table 4: ML and Proposed method comparison table on the basis of performance measures

Classifiers	Accuracy	Specificity	Precision	F1-score
SVM	94.8	95.7	87.2	88.4
KNN	94.1	97.8	63.4	78.5
CNN-LSTM (Proposed)	98.5	98.7	98.9	99.1

Figure 9 shows that proposed CNN-LSTM is having highest accuracy of 98.5%, specificity of 98.7%, precision 98.9% and F1-score of 99.1% which is evident that the proposed CNN-LSTM outperformed as related to other two ML techniques.

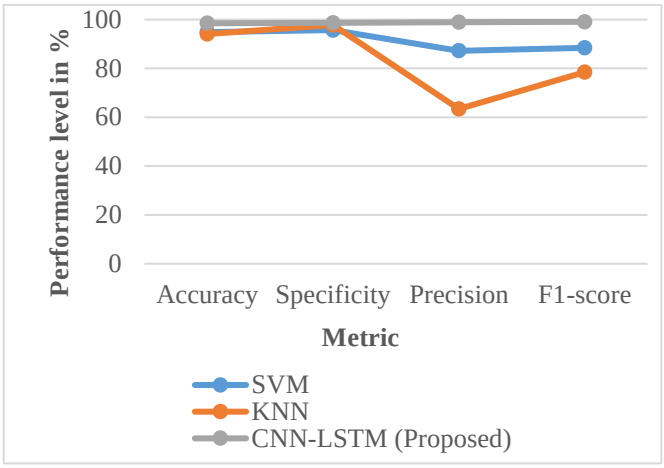

Fig 9: Performance comparison of SVM, KNN and CNN-LSTM proposed method

5. Conclusion

Deep learning algorithms and Spark's scalability are powerful tools for combating credit card fraud. However, challenges persist, and ongoing research and development efforts are improving their capabilities. The main objective of fraud detection systems is to accurately predict fraud instances and reduce false-positive cases. This paper examines various methods for detecting fraud in card transactions using an unbalanced dataset. The under-sampling approach balances the dataset during pre-processing, while CNN-LSTM measures fake observants and combines probabilities to find alerts. The model employs a ranking technique to prioritize alerts and correct class imbalances. CNN-LSTM outperforms all algorithms with an F1-Score of 99.1%. Future work should focus on techniques that enhance model interpretability and build trust in decision-making. Regular monitoring and updating of data and models are essential to maintain effectiveness. Combining deep learning with other machine learning techniques can enhance overall performance. By addressing these challenges and leveraging the strengths of deep learning and Spark, organizations can significantly improve their ability to detect and prevent credit card fraud, protecting both customers and financial interests.

References

[1] Burhanudin, Burhanudin. (2022). Examining the effect of service value and reputation on customer loyalty with trust and electronic word of mouth as mediation. Jurnal Aplikasi Manajemen. 20. 10.21776/ub.jam.2022.020.03.05.

[2] Gurav, Prof & Badhe, Mrs & Nagtilak, Mrs & Sonawane, Mr & Agarwal, Mr. (2022). Credit Card Fraud Detection. International Journal for Research in

Applied Science and Engineering Technology. 10. 2009-2012. 10.22214/ijraset.2022.41594.

[3] Y. Abakarim, M. Lahby, and A. Attioui, ‘‘An efficient real time model for credit card fraud detection based on deep learning,’’ in Proc. 12th Int. Conf. Intell. Systems: Theories Appl., Oct. 2018, pp. 1–7, doi: 10.1145/3289402.3289530.

[4] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, ‘‘Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence,’’ Mobile Inf. Syst., vol. 2020, pp. 1–13, Oct. 2020, doi: 10.1155/2020/8885269.

[5] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, ‘‘Performance analysis of feature selection methods in software defect prediction: A search method approach,’’ Appl. Sci., vol. 9, no. 13, p. 2764, Jul. 2019, doi: 10.3390/app9132764.

[6] B. Bandaranayake, ‘‘Fraud and corruption control at education system evel: A case study of the Victorian department of education and early childhood development in Australia,’’ J. Cases Educ. Leadership, vol. 17, no. 4, pp. 34–53, Dec. 2014, doi: 10.1177/1555458914549669.

[7] Krizhevsky A, Sutskever I, Hinton G E (2012) ImageNet classification with deep convolutional neural networks. In: Paper presented at the twenty-sixth annual conference on neural information processing systems, Lake Tahoe, Nevada, 3–6 December 2012, pp 1097–1105

[8] Szegedy C, Liu W, Jia Y, Sermanet, P, Reed S (2015) Going deeper with convolutions. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, 7–12 June 2015, pp 1–9.

[9] Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Paper presented at the international conference on learning representations, San Diego, California, 7–9 May 2015, pp 1–14

[10] Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko (2017) MobileNets: efficient convolutional neural

[11] H. John and S. Naaz, “Credit card fraud detection using local outlier factor & isolation forest,” International Journal of Computer Sciences and Engineering, vol. 7, no. 4, pp. 1060–1064, 2019.

[12] W. F. Yu and N. Wang, “Research on credit card fraud detection model based on distance sum,” in IJCAI international joint conference on artificial intelligence, pp. 353–356, Hainan, China, 2009.

[13] M. Phi, Illustrated Guide to LSTM’s and GRU’s: A step by step explanation, Towards Data Science, 2018. Accessed on: Nov. 23, 2020. [Online]. Available at: https://towardsdatascience.com/illustrated-guideto-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[14] J. Desjardins, How much data is generated each day?, World Economic Forum, April 17, 2019. Accessed on: Nov. 18, 2020. [Online]. Available: https://www.weforum.org/agenda/2019/04/how-much-data-is-generatedeach-day-cf4bddf29f/

[15] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques." International Journal of System Assurance Engineering and Management 8, no. 2 (2017): 937-953.

[16] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," arXiv preprint arXiv:1906.05799 (2019).