Journal of Theoretical and Applied Information Technology
31st December 2023. Vol.101. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EXPLORING THE CAPABILITIES OF DEEP LEARNING FOR ADVANCING CREDIT CARD FRAUD DETECTION: A REVOLUTIONARY APPROACH

ABDERRAHMANE DAIF[1*], SOUMAYA OUNACER[1], SOUFIANE ARDCHIR[2], MOHAMED GHAZOUANI[1], MOHAMED AZZOUAZI[1]

[1]Laboratory Of Information Technologies And Modeling, Ben M'sick Faculty of Science University Hassan II Casablanca, Morocco

[2]National School of Commerce And Management University Hassan II, Casablanca, Morocco.

E-mail: *daif.abdou@gmail.com,

ABSTRACT

The use of credit cards for both online and in-person purchases has become increasingly prevalent in our daily lives. However, this convenience also exposes users to the risks associated with credit card fraud. Credit card fraud presents a significant challenge for banks, merchants, and consumers, emphasizing the crucial need for the swift and accurate detection of such fraudulent activities. In response to this challenge, recent research has delved into the application of deep learning techniques for credit card fraud detection. This article presents a study that combines a Bi-LSTM (Bidirectional Long Short-Term Memory) with an attention layer to identify fraudulent transaction patterns and achieve a balanced classification of data. The results of this study demonstrate the method's high accuracy, surpassing the performance of other fraud detection approaches. Notably, this innovative approach efficiently identifies critical transactions within input sequences, significantly improving the prediction accuracy for fraudulent transactions. This research provides a unique perspective on the use of deep learning to enhance security in credit card transactions.

Keywords: *Deep Learning, Machine Learning, Credit Card Fraud, Bi-LSTM, Attention Layer, SMOTE.*

1. INTRODUCTION

Credit cards have become one of the most common payment methods for both online and in-person transactions. However, the use of credit cards also presents fraud risks, which have increased in recent years due to the growth of online commerce. Fraudsters are continually developing new and sophisticated techniques to perpetrate fraud, which can cause significant damages to customers, merchants, and banks.

According to a Nilson Report study[1], losses due to credit card fraud reached $27.85 billion worldwide in 2018, representing an increase of 16.2% from the previous year (Figure 1). In the United States, losses due to credit card fraud reached $8.14 billion in 2019, according to data from the Federal Reserve Bank of Kansas City. These losses can include refunds for fraudulent transactions, costs associated with fraud management and payment system security, as well as customer trust losses.

To mitigate these losses, it is essential to detect credit card fraud quickly and accurately. Traditional fraud detection methods rely on machine learning algorithms that are often insufficient in detecting the most sophisticated frauds. They do not account for variations and trends in consumer spending behavior, such as fluctuations during holiday periods or across different geographical regions.

In response to these challenges, there is an imperative need to establish a more sophisticated and adaptable fraud detection system. This is where Bidirectional Long Short-Term Memory (Bi-LSTM) networks, coupled with attention mechanisms (Attention Layer), come into play. This innovative approach allows for precise prediction of fraudulent credit card transaction behavior by considering transaction history and emphasizing the most relevant aspects.

Bi-LSTM models the temporal sequence of transactions, while the Attention Layer highlights key elements within each transaction, thereby illuminating potential fraud patterns. This combination of techniques significantly enhances credit card fraud detection.

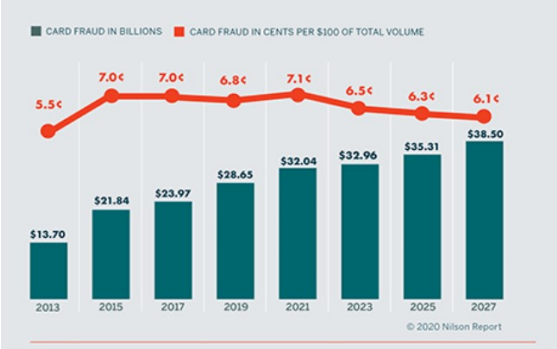

Figure 1: Card Fraud Worldwide.

We conducted experiments to validate the effectiveness of this approach. Preliminary results indicate that the Bi-LSTM model with the Attention Layer offers significantly improved fraud detection performance compared to traditional machine learning models. Our Bi-LSTM model, trained on a real transaction dataset with balanced classes, achieved a fraud detection accuracy of over 99%, surpassing other fraud detection methods.

The organization of this paper is as follows. Section 1 provides an introduction to the problem and the research questions addressed in this study. In Section 2, a literature review is presented to shed light on the various issues and challenges related to credit card fraud. Section 3 delves into the proposed approach for detecting fraudulent transactions, based on Bilstm and an attention layer. Section 4 provides a detailed description of the dataset used in this study, the resampling techniques employed, the selected methods, and the results obtained from the analysis. In conclusion, Section 5 summarizes the entire article, emphasizing the effectiveness of the suggested approach in the context of credit card transactions.

2. RELATED WORK

Detecting fraudulent activities in credit card transactions is a complex task that requires the fraud detection system to accurately distinguish between normal and fraudulent transactions while detecting fraud quickly. To choose the appropriate fraud detection technique, a detailed analysis of various research studies has been conducted. In [2], authors used a credit card transaction dataset to train and test their anomaly model, which consists of an isolation forest algorithm. The algorithm can detect anomalies in transaction data by isolating them in smaller subsets of data and measuring their distance from other transactions. The study's findings showed that the isolation forest algorithm was effective in detecting credit card fraud, with high accuracy and a low false positive rate. [3] provides a comprehensive review of recent research on credit card fraud detection using machine learning. The authors discuss various aspects of credit card fraud, such as its types and characteristics, and review the state-of-the-art techniques and algorithms for detecting fraud. They also provide a detailed analysis of the strengths and limitations of each technique and summarize the datasets commonly used for evaluating the algorithms. In this study, the authors devised a solution aimed at enhancing the accuracy of credit card fraud detection. They introduced the use of a hybrid model incorporating an Artificial Neural Network (ANN), which not only improves the precision of detection but also ensures data confidentiality. This approach represents a significant advancement in securing financial transactions by effectively balancing heightened accuracy with the protection of sensitive user information.

In [4], authors discuss the application of three unsupervised methods for credit card fraud detection, namely SVM, autoencoder, and Mahalanobis. While SVM and autoencoder rely on labeled data for training, Mahalanobis method only requires a minimum covariance determinant matrix to identify anomalies. authors do not compare the performance and training of the three models but evaluates their performance using available labels. They suggest that future studies should consider cardholders' behavior and historical transaction data to achieve higher accuracy, and both global and local outliers should be considered.

The work [5] addresses the issue of imbalanced datasets in credit card fraud detection. The authors propose a methodology that uses a combination of oversampling, undersampling, and cost-sensitive learning to tackle this problem. The authors experiment with different sampling methods and show that the combination of SMOTE oversampling, Tomek links undersampling, and cost-sensitive learning leads to the best results. In [6] authors proposes a methodology for credit card

fraud detection using a combination of pipeline and ensemble learning techniques. The proposed approach involves feature engineering, which includes feature scaling, feature selection, and feature creation. The authors use several machine learning algorithms, such as logistic regression, random forest, gradient boosting, and support vector machine, in an ensemble approach to improve the accuracy of the model. they also provide an experimental evaluation of the proposed approach using a real-world dataset, showing that the ensemble approach outperforms the individual algorithms. The study concludes that the proposed approach can be used as a reliable tool for detecting credit card fraud with high accuracy. In a separate research study that utilized the identical dataset, conducted by a group of researchers[7]. Authors compared four machine learning algorithms in terms of accuracy, AUC score, and false positive rate to determine the best performing method for detecting credit card fraud. The isolation forest algorithm achieved better results than the other approaches. The isolation forest was more precise, with a score of 99.74%, while decision tree achieved 95.52%, KNN 96.91%, and SVM 97.11%. Additionally, SVM had a false positive rate of 32.67% and decision tree had a false positive rate of 21.78%. While KNN's false positive rate reached 20.78%, the isolation forest rate did not exceed 10.99%. Therefore, it proved to be the best approach to use.

The mentioned research has several limitations. They are sensitive to evolving fraud techniques and schemes, making it challenging to adapt to emerging fraudulent activities. Moreover, these methods fail to consider the diverse range of customer behaviors, which is crucial in fraud detection. They also overlook issues related to imbalanced class distribution in data and false alarms, leading to significant losses for financial institutions and merchants. These limitations have a substantial impact: they can result in time losses when addressing genuine fraud cases, cause customer dissatisfaction, and generate high costs, both in terms of customer service and resources used to address false alarms. Consequently, it's crucial to address these shortcomings by designing more robust and adaptable models capable of adjusting to new forms of fraud, better understanding customer behaviors, and reducing false alarms. This approach would minimize financial losses and enhance the overall effectiveness of fraud detection systems in the financial sector.

3. PROPOSED APPROACH FOR FRAUDULENT TRANSACTION DETECTION

The aim of the Bi-LSTM model is to automate the assessment of customers' credit card behavior and trigger early alerts in the event of credit card payment defaults. Figure 2 illustrates the framework of the proposed model. This workflow allows for a comprehensive exploration of the model's performance to derive reliable conclusions.

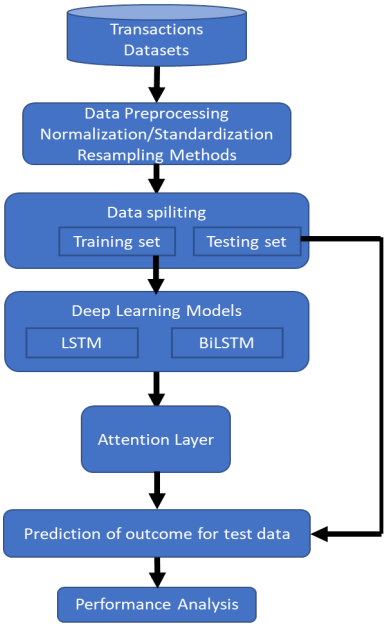

Figure 2: Proposed workflow

The proposed framework consists of multiple stages. Firstly, the dataset undergoes preprocessing and formatting, followed by data balancing using the SMOTE technique. Additionally, to enhance the quality of the input data, normalization (or standardization) is applied to adjust the value distribution, facilitating the model's learning process. Normalization aims to align the entire value distribution within the range of [0, 1], which is particularly important for deep learning models. Subsequently, the data is split into training and testing sets, before being fed into the model for use by the Bi-LSTM classifier. To identify multicollinearity within a given transaction, an attention layer is employed. This layer functions akin to our brain, focusing on relevant elements while disregarding others, thereby facilitating learning and yielding robust results. The combination of the attention layer with Bi-LSTM produces a high-quality output. Finally, a five-fold validation technique is used to obtain predictions for all clients in the dataset. The model's performance is then assessed by calculating various

performance metrics. The results are compared to benchmark models using diverse performance metrics, enabling the determination of the model's effectiveness in credit card fraud detection.

The results of this deep learning-based credit card fraud detection workflow are discussed in the results section of the report. This includes an evaluation of the impact of normalization (or standardization) on the model's performance. These insights are critically valuable for banking institutions and financial service providers seeking to enhance their fraud detection systems.

4. METHODOLOGY FOR CREDIT CARD FRAUD DETECTION

4.1. Dataset Description

The dataset used in the experiment for fraud detection was obtained from Kaggle.com and consisted of online credit card transactions made by European citizens over two days in September 2013 [8]. Out of a total of 284,807 transactions, only 492 were fraudulent, which demonstrates a significant imbalance in the dataset. The majority of the features from V1 to V28 were transformed by PCA, leaving only the time and amount variables untransformed. The time variable represents the duration between the first and second fraudulent transactions. The class characteristic is binary, with 0 representing no fraud and 1 representing fraud. Due to the highly imbalanced distribution of classes, the classification model's performance may decline, as machine learning and deep learning algorithms struggle to learn about both classes when the data is highly imbalanced. The algorithm may be biased toward genuine transactions, neglecting the importance of the fraudulent class.

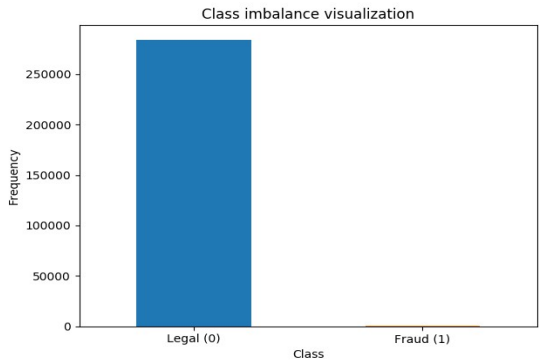

Figure 3:The distribution of transactions by class.

4.2. Data preprocessing

In the context of this issue, most datasets exhibit a pronounced imbalance, characterized by a significantly higher number of normal transactions compared to fraudulent ones. This is naturally due to the rarity of fraud cases. To overcome this challenge, which can negatively impact our model's performance due to the predominance of the majority class, we employ sampling techniques. Several resampling methods are available, including undersampling and oversampling, each with its specific advantages and disadvantages.

Undersampling involves reducing the quantity of data from the majority class to balance the dataset. This approach can be useful when data is abundant, but the collection of additional data is costly. However, it can lead to the potential loss of valuable information. On the other hand, oversampling entails increasing the number of minority class examples by duplicating or generating synthetic data. This approach can enhance the model's ability to detect fraud, provided that the synthetic data is representative of reality, while minimizing the risk of overfitting.

We will test two methods, namely SMOTE and ADASYN, to explore the benefits of each of these techniques to determine which one is best suited for our specific case. SMOTE, which was developed by Chawla and Bowyer [9], is widely acknowledged and extensively utilized within the machine learning community. Its primary aim is to generate fresh instances for the minority class in the form of synthetic examples. In contrast to simply duplicating existing instances, SMOTE operates by interpolating between the nearest neighbors of the minority class instances. This approach effectively addresses the data imbalance issue while also mitigating the risk of the model overfitting to the training data. Additionally, the extent of oversampling required can be fine-tuned by randomly selecting the nearest neighbors of minority class instances, providing a high degree of adaptability based on specific requirements. Additionally, we use ADASYN, an adaptation of the SMOTE technique, called "Adaptive Synthetic Sampling." ADASYN is designed to generate synthetic examples of the minority class in a more adaptive manner. This approach creates instances that are more challenging to classify, thereby strengthening the model's ability to detect fraud.

Feature normalization is a commonly employed method during the data preprocessing

phase due to its significant effects on the predictive capability of models [10]. Its primary objective is to standardize the numerical features of a dataset, bringing them to a common scale while preserving inherent variations. There are two types of normalization:

Min-Max Normalization: This method uses the minimum and maximum values to bring all values into the [0,1] interval while preserving the distance ratios between values. To perform this normalization, the following formula is used:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

Standard Normalization: Also known as standardization, this method involves subtracting the mean of the values and dividing by the standard deviation. This allows each value to represent its distance from the mean in terms of standard deviation units.

$$X_{stand} - \frac{X \quad X_{moyen}}{Ecart\_type} \qquad (2)$$

In the context of this study, our dataset consists of 30 features, each with a different scale. For instance, the "Time" feature ranges from 0 to 175,000 seconds, while the "Amount" feature extends from 0 to 25,000 dollars. At first glance, it may appear that the "Time" feature is more important and has a more significant impact on the model's performance. However, this impression is primarily due to the differences in feature scales. Therefore, we will explore the two normalization methods to determine which one yields the best results.

For normalization, we will focus on the "Time" and "Amount" variables. We exclude the "Class" variable from this process as it is binary and represents our target variable. As for the other 28 variables, we have chosen not to normalize them. This decision is based on several considerations. Firstly, we observe that the values of these variables are already relatively small and fall within well-defined ranges, including both positive and negative values. Additionally, these variables have already undergone transformations during the initial data preparation. Therefore, there is no need to normalize them again. Ultimately, this approach will allow us to preserve the essential characteristics of the data while improving the comparability of the "Time" and "Amount" variables, contributing to more effective modeling and more accurate results.

The skewness coefficient is a useful measure for understanding the symmetry of a data distribution. It can take various values, indicating the degree of asymmetry in the distribution [11]:

• If the skewness coefficient falls between -0.5 and 0.5, it suggests that the data is nearly symmetric.

• When the coefficient is between -1 and -0.5 (negative skewness) or between 0.5 and 1 (positive skewness), it means that the data is slightly asymmetric.

• If the coefficient is less than -1 (negative skewness) or greater than 1 (positive skewness), it indicates that the data is highly asymmetric.

To assess the data's asymmetry, it is common to plot distribution curves and calculate the skewness coefficient to identify variables that exhibit significant asymmetry. A concrete example of this approach is the "Amount" variable, which initially displayed extreme asymmetry as depicted in Figure 4. It can be observed that the skewness coefficient value is very high, indicating that the distribution of this variable is highly asymmetric. Various methods are available for addressing data asymmetry. In our work, we opted to apply a logarithmic transformation, specifically the "log" method. This approach has proven effective in making the data distribution more balanced, hence its adoption [12].

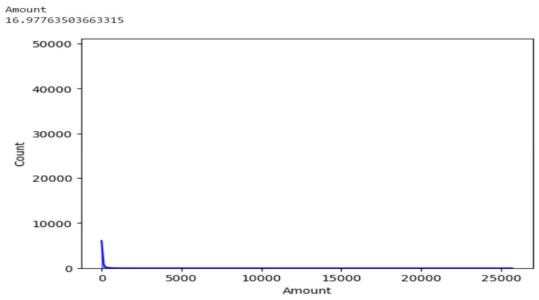

Figure 4:Distribution of the "Amount" variable

After applying a logarithmic transformation, we were able to make its distribution more symmetrical, as illustrated in Figure 5.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

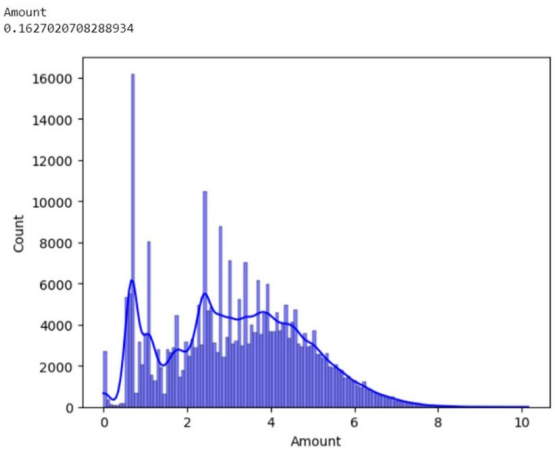

Figure 5: Distribution of the "Amount" variable after the logarithmic transformation

4.3. Deep learning classifiers

4.3.1. LSTM

A Recurrent Neural Network (RNN) is a type of artificial neural network designed to process sequential or temporal data. It differs from traditional neural networks by having a recurrent loop structure, which allows it to consider previous data while processing each element of the sequence, taking into account past information in the computation. This dynamic handling of sequential data makes RNNs suitable for various applications. However, traditional RNNs suffer from the "vanishing gradient" problem during training on long sequences, making it difficult to retain long-term information and limiting their ability to model long-term dependencies in sequential data. This is where Long Short-Term Memory (LSTM) networks come into play. LSTMs are a variant of RNNs specifically designed to address the vanishing gradient problem. They are equipped with gate mechanisms that proficiently regulate information flow through network cells, enhancing the efficient retention and forgetting of long-term information.

LSTMs introduce a long-term memory structure within the network, empowering them to effectively tackle tasks demanding a comprehension of long-term relationships in data. Beyond their applications in text generation, time series prediction, and automatic translation, LSTMs have exhibited remarkable performance in financial fraud detection. The LSTM sustains vital information by utilizing several key components, as illustrated in Figure 6:

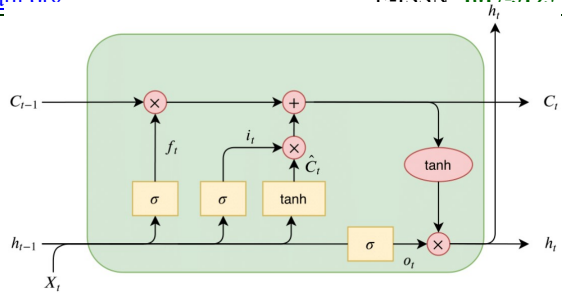

Figure 6: LSTM Cell

Memory Cell: This component stores information over extended periods, which is critical for detecting subtle fraud patterns that may span multiple transactions.

Input Gate: This gate assesses the importance of incoming information, aiding the model in assigning significance to essential data. It performs this assessment by using a combination of sigmoid and hyperbolic tangent activation functions to generate an update vector, which is then added to the cell state.

Forget Gate: This gate plays a crucial role in determining which pieces of information from the previous time step should be discarded or "forgotten," and which should be passed on to the current time step. To accomplish this, it employs a sigmoid activation function to produce values ranging between 0 and 1 for each element of the hidden state. A value of 0 signifies "forget," while a value of 1 indicates "remember."

Output Gate: this gate is responsible for determining which information from the cell state is relevant in generating the hidden state for the current time step. It operates similarly to the other gates, utilizing a sigmoid activation, followed by the application of the tanh function, to produce a filtered output from the cell state. With its three control gates and memory cell, LSTM can easily retain, read, reset, and update information over long periods. In the context of fraudulent transaction detection, the use of LSTM is particularly relevant due to its ability to identify complex and evolving fraud patterns. Its capacity to retain information over long temporal sequences makes it a valuable tool for financial institutions seeking to prevent fraudulent activities.

4.3.2. BILSTM

The BiLSTM, as described in [13], builds upon the power of LSTM with a unique capability: it processes data sequences in both forward and backward directions, considering both past and

future information. This feature is particularly valuable in fraudulent transaction detection, where accounting for both past and future information is essential. In contrast to traditional LSTM, which only considers past information in a data sequence, the BiLSTM examines both past and future data. This means it can better anticipate evolving fraud patterns over time, as it takes the entire transactional sequence into account. Here's how the BiLSTM works in fraud detection[14]:

Capturing Temporal Dependencies: The BiLSTM can capture temporal dependencies in the data. It can identify transactions that may appear normal at first but become suspicious when followed by unusual transactions.

Bidirectional Analysis: By processing data in both directions, the BiLSTM can detect fraud patterns that propagate in both temporal directions. For example, it can identify a series of fraudulent transactions followed by attempts to conceal them in the past.

Preventing False Negatives: Fraudulent transactions are becoming increasingly sophisticated, making it essential to anticipate new tactics. The BiLSTM excels in this task by considering emerging trends.

Reducing False Positives: With its deep understanding of data sequences, the BiLSTM minimizes classification errors, reducing the burden of false positive investigations for financial institutions.

Figure 7 shows the architecture of BiLSTM:

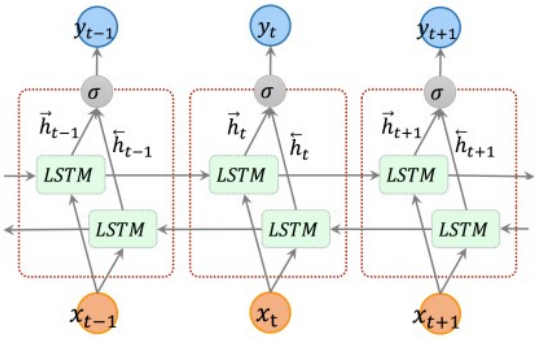

Figure 7:BiLSTM Architecture

Within the LSTM network, there are several essential components: $c_t$represents the memory unit, $i_t$stands for the input gate, o_t denotes the output gate, h_t signifies the hidden unit, and f_t represents the forget gate. The following presents the state of the LSTM network.

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (3)$$
$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \quad (4)$$
$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c[h_{t-1}, X_{t-1}] + b_c) \quad (5)$$
$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (6)$$
$$h_t = o_t * \tanh(C_t) \quad (7)$$
$$\sigma(\cdot) = \frac{1}{1+e^{-(\cdot)}} \quad (8)$$

$$\vec{h}_t = f(\vec{W} \cdot x_t + \vec{W} \cdot \vec{h}_{t-1} + \vec{b}) \quad (9)$$
$$\overleftarrow{h_t} = f(\overleftarrow{W} \cdot x_t + \overleftarrow{W} \cdot \overleftarrow{h_{t-1}} + \overleftarrow{b}) \quad (10)$$
$$y_t = g(U \cdot [\vec{h}; \overleftarrow{h}] + c) \quad (11)$$

The weights Wc, Wi, Wo, and Wf are associated with the memory cell, input gate, output gate, and forget gate, respectively. These are the weight matrices that connect the input of the hidden layer to the three gates and the input cell state. bc, bf, bi, and bo represent the biases of the LSTM cell during training. By using BiLSTM, a non-linear transformation and high-level abstraction of the collected fault data can be performed to provide more precise calculations, as illustrated in equations (3), (4), and (5). The vectors $\vec{W}$and $\overleftarrow{W}$represent the parameters of the network's hidden layer, xt is the input data, $\vec{h_t}$ and $\overleftarrow{h_t}$ are the outputs of the two LSTM layers, $\vec{b}$ and $\overleftarrow{b}$ are the bias values, and yt is the output of the BiLSTM.

The BiLSTM represents a significant advancement in fraudulent transaction detection compared to classical LSTM. It is an exceptionally effective tool for combating ever-evolving financial fraud.

4.3.3. ATTENTION LAYER

In modern deep learning research, particularly in fields such as credit card fraud detection, attention mechanisms have proven to be an effective tool for improving model accuracy by emphasizing crucial information. These mechanisms allow for selective focus on critical data elements, which is essential for accurate fraud detection [15]. The attention layer is a mechanism used to concentrate on specific elements of input data when making predictions or generating output sequences [16]. The fundamental concept behind an attention layer is to assign distinct attention weights to different components of the data. This weight assignment enables the model to prioritize more

important information while giving less priority to less crucial data. By employing the attention method, the model can dynamically adjust its focus based on the context, thereby enhancing its ability to process data effectively.

This addition is crucial because not all variables used for fraud detection are of equal importance. The attention layer assigns variable weights to the variables, highlighting those with the most influence.

4.4. Experiments & Results

The main objective of the machine learning model is to learn from previous experiences and utilize this capability to generate new instances. We evaluated various deep learning methods, including LSTM, BiLSTM, LSTM with an attention layer, and BiLSTM with an attention layer, using the complete set of dataset features after a preprocessing step. Figure 6 illustrates the balanced dataset using the SMOTE technique, which proved to be effective in the context of credit card fraud detection.

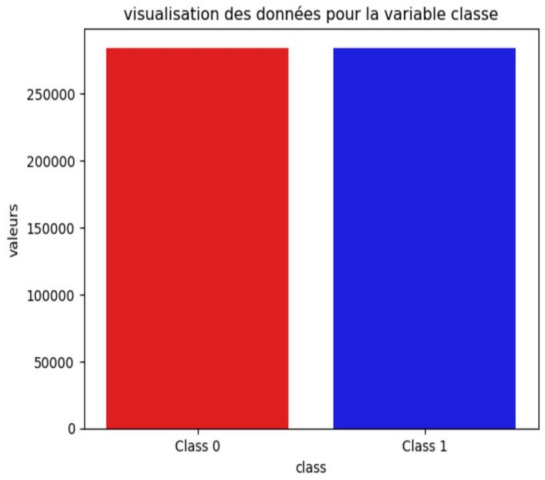

Figure 8: Data visualization for each class after applying SMOTE

To assess the model's performance, we followed a common procedure, which involves splitting the dataset into two parts: 70% for model training and 30% for performance evaluation. Model parameters, often referred to as hyperparameters, are determined during model training. These hyperparameters also played a role in finding the best model fit for a machine learning model. In this study, hyperparameter tuning, such as the number of neurons, optimizer type, number of epochs, and batch size, was carried out on the best model to achieve the highest possible accuracy while also ensuring overfitting prevention through 5-fold cross-validation during grid search. In the k-fold cross-validation technique, the training dataset is randomly divided into k distinct subsets without replacement. Out of these k subsets, k-1 are used to train the model, while one is reserved for testing. The model's performance is then evaluated by calculating the average performance across the different subsets. This provides an estimate of the overall model performance that is less influenced by potential biases due to inadequate learning from the training data.

The Python function "GridSearchCV" was used throughout the hyperparameter tuning process. The final hyperparameter values are summarized in Table 1 after completing the tuning with GridSearch.

Table 1: Hyperparameter Selection

Model	Hyperparameters	Value
LSTM/BiLSTM	Activation Function	Sigmoid / Relu
	Dropout	0.5
	Optimizer	Adam
	Epochs	50
	Batch Size	2000
	Number of Layers	10
	Cost Function	Binary Cross-Entropy

To demonstrate the reliability of the results obtained on the test set and to make the outcomes of the bidirectional LSTM significant, various measures need to be assessed, each reflecting different aspects of the model's performance:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (12)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

$$\text{F1-Score} = \frac{2*(\text{Precision}*\text{Recall})}{\text{Precision}+\text{Recall}} \quad (15)$$

$$MCC = \frac{TN * TP - FN * FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (16)$$

Accuracy: This metric refers to the number of correct predictions compared to all predictions. It gives us a general idea of the model's performance. In our case, it means the number of correctly predicted transactions compared to all transactions.

Precision: It refers to the number of correct positive predictions compared to all positive predictions. In our case, it corresponds to the number of correctly identified fraudulent transactions by the model compared to all transactions classified as fraudulent by the model.

Recall: It refers to the number of correct positive predictions compared to all truly positive observations. In our case, it corresponds to the number of correctly identified fraudulent transactions by the model compared to all truly fraudulent transactions.

F1 score: This metric plays a crucial role in balancing precision and recall. It is used to evaluate a model's performance by considering both the ability to correctly identify true positives and minimize false positives.

MCC (Matthews Correlation Coefficient): The Matthews Correlation Coefficient, widely used in binary classification, plays a crucial role in model evaluation, especially when dealing with highly imbalanced datasets.

The performance results presented in the table 2 summarize the outcomes achieved using various algorithms and data preprocessing techniques throughout our study. These performance values represent the average results obtained through 5-fold cross-validation, a method employed for training, testing, and evaluating our models.

According to the table, the importance of using a sampling technique is clearly evident. Without this technique, our models' performance is significantly reduced because they become biased towards the majority class, hindering their ability to handle or recognize minority class data effectively. Among the sampling techniques, it's interesting to note that SMOTE yielded significantly better results than ADASYN in this specific context. Furthermore, the importance of normalization is highlighted, especially in the deep learning domain. Standardization generated better results than Min-Max normalization, although the difference is not significant. Regarding neural network architectures, BiLSTM showed better performance than LSTM, even though the difference is not very pronounced. This underscores the importance of processing data sequences bidirectionally rather than unidirectionally, which can be extremely beneficial in fraud detection with real data. The table also emphasizes the importance of using multiple evaluation metrics. For example, accuracy does not provide an adequate overview in this type of problem, where the data is imbalanced, and fraud detection (true negatives) is crucial.

In the figure 9, we will examine the confusion matrix results for the proposed model. We used SMOTE, standardization, and BiLSTM during the 5-fold cross-validation. Based on the results presented in table, it can be observed that the majority of predictions made by our model are correct. Furthermore, most of the incorrect predictions are false positives rather than false negatives. This indicates that our model tends to accurately predict fraudulent transactions, which is crucial. It is better to classify a transaction as fraudulent, even if it is not, rather than classifying it as non-fraudulent when it is actually fraudulent.

Table 2: Accuracy Measure

Training Accuracy	Training Test
0.9993493132616992	0.9963332922990346

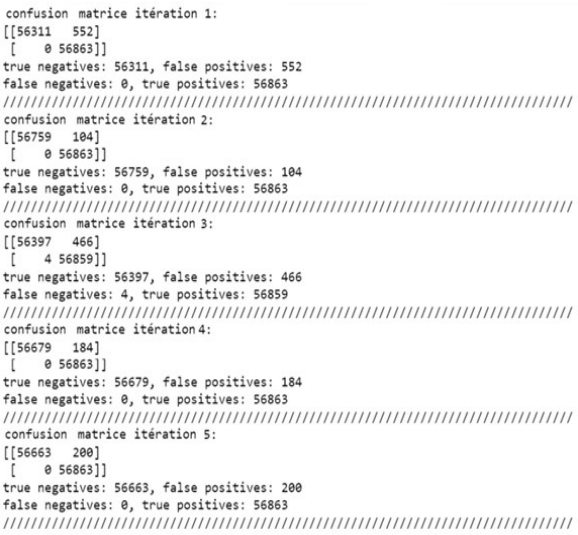

Figure 9: Confusion Matrix Results

In the graph below, we can observe the loss curves for the training and validation sets during the third fold, which is representative of the other k-folds. An analysis of this figure reveals an interesting trend: the loss, both for the training and validation data, steadily decreases until it stabilizes around epoch 20. This stabilization suggests that our model does not suffer from overfitting,

meaning it generalizes well beyond the training set and is capable of making accurate predictions on new data.

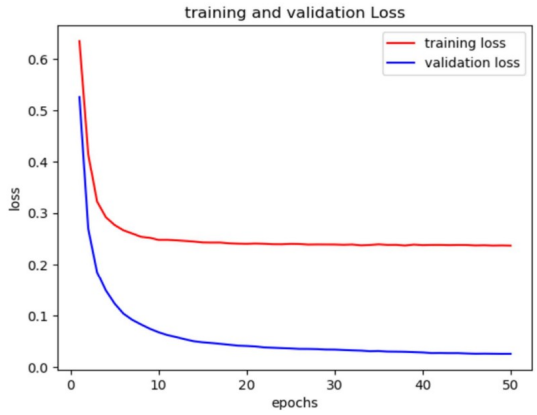

Figure 10: Loss curve

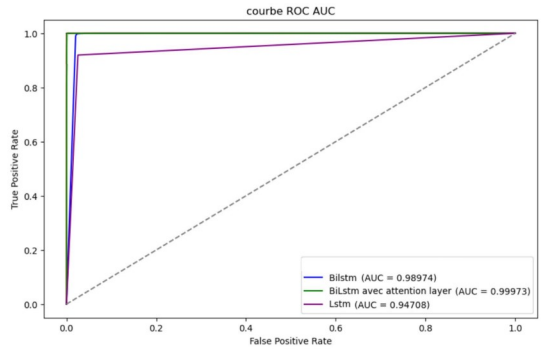

Figure 11: AUC Score

The curve below (Figure 11) represents the ROC-AUC curve. We evaluated the performance of several models used in our study, including LSTM, BiLSTM, and BiLSTM with an attention mechanism. It is evident that all of these models exhibit exceptional performance, with special mention to the BiLSTM with an attention layer, which stands out particularly.

5. DISCUSION

The detection of credit card fraud requires a robust model capable of accurately and in real-time detecting fraudulent transactions. Several research works [17][18][19][20] have explored fraudulent transaction detection, showcasing their efficacy, but they face significant challenges. These methods often overlook long-term transaction processing and inter-transaction dependencies. Additionally, they struggle to detect complex fraud patterns and incorporate consumer behavior variations, such as fluctuations during sales, promotions, or holiday periods. Our innovative approach has demonstrated exceptional real-time fraud detection efficiency. Our model showcased superior performance, achieving a precision rate of 99.99%. We addressed class imbalance by leveraging the SMOTE method, introducing synthetic samples that positively impacted our model. Furthermore, the application of logarithmic transformation, specifically the "log" method, effectively balanced data distribution. By combining the bilstm model with an attention layer, we achieved even more remarkable results. This approach better accounted for transaction dependencies, significantly enhancing the model's ability to detect complex fraud patterns. Our research marks a significant advancement in credit card fraud detection by developing a potent model capable of real-time transaction processing, overcoming data imbalances, and comprehending intricate fraud patterns.

6. CONCLUSION

Our study was dedicated to crafting a deep learning-based fraud detection model, involving a series of in-depth experiments. Our primary approach relied on BiLSTM augmented with an attention mechanism, complemented by data preprocessing techniques such as SMOTE for data balancing and variable standardization. This method showcased exceptional performance, evaluated across multiple metrics including accuracy, precision, recall, and MCC. It was rigorously compared against other techniques such as LSTM and traditional machine learning methods. The analysis results underscored the model's robustness, achieving an outstanding precision of 99%. Furthermore, the results confirmed that integrating an attention layer significantly enhanced the model's performance, accurately distinguishing between fraudulent and legitimate transactions. Among the limitations observed in our study, a notable challenge was posed by the presence of unlabeled data. While the BiLSTM model with an attention layer displayed proficiency in detecting credit card-related frauds, it did not directly address the underlying issue. Therefore, in pursuit of further enhancement, we are considering the development of a hybrid model. This hybrid approach aims to combine deep learning techniques (BiLSTM with an attention layer) with unsupervised learning methods, particularly the Isolation Forest algorithm. The objective is to create a hybrid model that could strengthen precision and real-time detection capabilities further.

In conclusion, while our study represents a significant advancement in fraud detection, future research endeavors targeting the creation of hybrid models integrating both supervised and unsupervised learning techniques seem promising. This approach could effectively address challenges associated with unlabeled data and further enhance the real-time detection of intricate frauds.

REFERENCES:

[1] D. Robertson, "The Nilson Report | News and Statistics for Card and Mobile Payment Executives," *The Nilson Report*. 2016. [Online]. Available: https://www.nilsonreport.com/

[2] S. Ounacer, H. A. El Bour, Y. Oubrahim, M. Y. Ghoumari, and M. Azzouazi, "Using Isolation Forest in anomaly detection: The case of credit card transactions," *Period. Eng. Nat. Sci.*, vol. 6, no. 2, pp. 394–400, 2018, doi: 10.21533/pen.v6i2.533.

[3] S. K. Shirgave, C. J. Awati, R. More, and S. S. Patil, "A review on credit card fraud detection using machine learning," *Int. J. Sci. Technol. Res.*, vol. 8, no. 10, pp. 1217–1220, 2019.

[4] M. Rezapour, "Anomaly detection using unsupervised methods: Credit card fraud case study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 1–8, 2019, doi: 10.14569/IJACSA.2019.0101101.

[5] S. Ounacer, H. Jihal, K. Bayoude, A. Daif, and M. Azzouazi, "Handling Imbalanced Datasets in the Case of Credit Card Fraud," 2022, pp. 666–678. doi: 10.1007/978-3-030-90633-7_56.

[6] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit Card Fraud Detection using Pipeling and Ensemble Learning," *Procedia Comput. Sci.*, vol. 173, no. 2019, pp. 104–112, 2020, doi: 10.1016/j.procs.2020.06.014.

[7] S. Ounacer, H. Jihal, S. Ardchir, and M. Azzouazi, "Anomaly Detection in Credit Card Transactions," 2020, pp. 132–140. doi: 10.1007/978-3-030-36674-2_14.

[8] "Credit Card Fraud _ Kaggle, Anonymized credit card transactions labeled as fraudulent or genuine." [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud

[9] N. V. K. W. B. Chawla and Lawrence O. Hall, "SMOTE: Synthetic Minority Over-sampling Technique Nitesh," *Ecol. Appl.*, vol. 30, no. 2, pp. 321–357, 2020, doi: 10.1002/eap.2043.

[10] S. G. K. Patro and K. K. sahu, "Normalization: A Preprocessing Stage," *Iarjset*, pp. 20–22, 2015, doi: 10.17148/iarjset.2015.2305.

[11] M. H. DeGroot and M. J. Schervish, *Probability and Statistics*, Fourth Edi. 2012.

[12] A. Charpentier, E. Flachaire, A. Charpentier, E. Flachaire, L. K. Density, and E. Flachaire, "Log-Transform Kernel Density Estimation of Income Distribution To cite this version : HAL Id : halshs-01115988 Working Papers / Documents de travail Log-Transform Kernel Density Estimation of Income Distribution," 2015.

[13] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction," pp. 1–11, 2018, doi: 10.48550/arXiv.1801.02143.

[14] J. Kim and N. Moon, "BiLSTM model based on multivariate time series data in multiple field for forecasting trading area," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2019, doi: 10.1007/s12652-019-01398-9.

[15] L. Li, Z. Liu, C. Chen, Y.-L. Zhang, J. Zhou, and X. Li, "A Time Attention based Fraud Transaction Detection Framework," 2019, [Online]. Available: http://arxiv.org/abs/1912.11760

[16] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio-temporal attention-based neural network for credit card fraud detection," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 362–369, 2020, doi: 10.1609/aaai.v34i01.5371.

[17] D. Shah and L. Kumar Sharma, "Credit Card Fraud Detection using Decision Tree and Random Forest," *ITM Web Conf.*, vol. 53, p. 02012, 2023, doi: 10.1051/itmconf/20235302012.

[18] *et al.*, "Fraud Detection of Credit Cards Using Supervised Machine Learning Techniques," *Pakistan J. Emerg. Sci. Technol.*, vol. 4, no. 3, pp. 38–51, 2023, doi: 10.58619/pjest.v4i3.114.

[19] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques," *Procedia Comput. Sci.*, vol. 218, pp. 2575–2584, 2022, doi: 10.1016/j.procs.2023.01.231.

[20] J. K. Afriyie *et al.*, "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions," *Decis. Anal. J.*, vol. 6, no. December 2022, p. 100163, 2023, doi: 10.1016/j.dajour.2023.100163.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Results and Performance of Different Models and Methods

Techniques Used	Resampling Methods	Normalization	Accuracy	Precision	Recall	F1 Score	MCC
Lstm	SMOTE	None	0.4969	0.4960	0.9966	0.6623	0.0279
BiLstm	SMOTE	None	0.4997	0.4989	0.9997	0.6658	0.0357
BiLstm	ADASYN	Standardization	0.8903	0.9959	0.7798	0.8729	0.7875
Lstm	SMOTE	Standardization	0.9947	0.9940	0.9950	0.9930	0.9882
BiLstm	SMOTE	Standardization	0.9975	0.9969	0.9981	0.9975	0.9960
BiLstm	SMOTE	Normalization Min-Max	0.9956	0.9914	0.9999	0.9978	0.9913
BiLstm + attention layer	SMOTE	Standardization	0.9985	0.9985	0.9998	0.9991	0.9986