Operations Research Forum (2025) 6:8
https://doi.org/10.1007/s43069-024-00409-6

RESEARCH

Combining Autoencoders and Deep Learning for Effective Fraud Detection in Credit Card Transactions

Mohammed Tayebi[1] · Said El Kafhali[1]

Received: 17 November 2024 / Accepted: 19 December 2024

Abstract

The advancement of technologies and the proliferation of new payment services have improved our lives, offering limitless opportunities for individuals and companies in each country to develop their businesses through credit card transactions as a payment method. Consequently, continuous improvement is crucial for these systems, particularly in the classification of fraud transactions. Numerous studies are required in the realm of automated and real-time fraud detection. Due to their advantageous properties, recent studies have utilized different deep learning architectures to create well-fitting models to identify fraudulent transactions. Our proposed solution aims to exploit the robust capabilities of deep learning approaches to identify abnormal transactions. The solution can be presented as follows: To address the imbalanced data set issue, we applied an autoencoder combined with the support vector machine model (ASVM). For the classification phase, we utilize an attention-long short-term memory neural network as a weak learner for the gradient boosting algorithm (GB_ALSTM), comparing it with various techniques, including artificial neural networks (ANNs), convolutional neural networks (CNNs), long short-term memory neural networks (LSTMs), attention-long short-term memory neural networks (ALSTMs), and bidirectional long short-term memory neural networks (BLSTMs). We conducted several experiments on a real-world dataset, revealing promising results in detecting abnormal transactions and highlighting the dominance of our suggested solution over competing models.

Keywords Deep learning · Autoencoder · Fraud transactions detection · Attention-long short-term memory neural networks · Gradient boosting · Attention mechanism

✉ Said El Kafhali
said.elkafhali@uhp.ac.ma

Mohammed Tayebi
m.tayebi@uhp.ac.ma

[1] Computer, Networks, Modeling, and Mobility Laboratory (IR2M), Faculty of Sciences and Techniques, Hassan First University of Settat, Settat, Morocco

Published online: 04 January 2025

1 Introduction

Credit card fraud is the illegal use of credit cards to make transactions. Credit cards are a popular payment method that can be used at any time and in any location. Furthermore, the number of fraudulent transactions has recently increased [1]. According to the Nilson Report, approximately 28.65 billion and 35 billion global business losses were caused by credit card fraud in 2019 and 2020 [2]. Due to fraudsters' use of some technical means (such as Trojan horses), credit card-related organizations and financial institutions need to be proactive to detect credit card fraud efficiently and accurately. Similarly, financial institutions and banks try to improve their systems to catch illegal transactions using different approaches based on artificial intelligence such as deep learning, machine learning, and data mining [3]. Those approaches show better performance and more promising results.

Every day, the number of transactions performed is numerous. Banks and financial institutions handle a huge number of transactions that are made by fraudsters every day [4]. These systems use various technologies to secure these transactions [5]. Therefore, the crime related to credit card transactions is growing due to the advanced techniques used by hackers to steal credit card information, including sending fake SMS and calls and masquerading attacks [6]. Thus, highlighting the importance of developing a new solution to stop these criminals, as well as making these fraud detection systems more intelligent and effective in stopping abnormal transactions [7], the advancement of technology and the availability of credit card transaction datasets have made machine learning technologies a solution for detecting fraudulent transactions [8]. Several research papers [9] proposed various solutions to detect fraud transactions using supervised or unsupervised learning algorithms [13].

The detection of fraudulent transactions is a binary classification issue in which we classify samples into fraudulent or legitimate transactions by discovering hidden patterns in data set transactions using classification techniques [10]. Various classification techniques that use deep learning techniques, such as backpropagation neural networks [11], Bayesian networks [12], variational autoencoder [14], generative adversarial neural networks [15], artificial neural networks [16], long short-term memory neural networks [17], and convolutional neural networks [18], have been well developed and successfully applied to predict fraud transactions, and many other domains [19, 20].

However, the imbalanced issues in credit card transaction datasets have presented serious difficulty and a challenging problem with most classifier techniques [21] leading to poor generalization and performance in detecting fraud transactions. The imbalanced dataset is marked as having many more samples of certain classes than others [22]. In the literature review, many solutions have been proposed to solve this problem, including the synthetic minority sampling technique (SMOTE) [23]. In this work, we propose an oversampling technique based on an autoencoder combined with the support vector machine algorithm. First, an autoencoder was utilized to generate new fraud samples based on historical fraud transactions [24]. Second, support vector machine learning was trained on the randomly undersampled training dataset and then used to validate the generated samples.

This work is driven by the overarching goal of elevating the management of extensive and imbalanced credit card datasets, culled from diverse payment systems, through the application of deep learning technology. Central to our approach is the utilization of attention-long short-term memory (ALSTM) neural networks as the foundational learner for a gradient boosting algorithm. In conjunction with this, we introduce a novel oversampling technique leveraging an autoencoder and support vector machine classifier. The significance of our proposed solution for fraud transaction detection lies in its efficiency, which is derived from the robust characteristics of each constituent method. ALSTM proves to be a potent tool for modeling sequential data due to its intrinsic capacity as a strong learner. The gradient boosting technique enhances model strength, effectively transforming a weak model into a more robust counterpart. Moreover, the incorporation of an autoencoder, a specialized artificial neural network for unsupervised learning, complements the system by learning representations for feature reconstruction within a subset of the data. The synergy of these methodologies establishes an effective and resilient classifier tailored for the detection of fraudulent transactions, underscoring the pivotal importance of our holistic solution in advancing fraud detection capabilities within the dynamic realm of payment services.

The key contributions of this article are the following:

- An autoencoder architecture combined with the support vector machine algorithm was proposed to handle the imbalance issue in the training set.
- An attention-long short-term neural network was used as a base learner in the gradient boosting algorithm.
- Several experiments have been conducted in which the proposed system has been compared with various deep learning architectures.
- The evaluation in this work was carried out on the European credit card dataset, using the widely used evaluation metrics in the credit card fraud detection problem.
- The experiments conducted show the dominance of the proposed technique in handling the imbalanced credit card dataset and in classifying fraudulent transactions.
- The model implementation with different tools that were used as well as the code can be found in GitHub [25].

This paper has the following structure: Section 2 examines the literature work. Section 3 describes the techniques utilized to construct our proposed solution for fraud transaction classification. However, in Section 4, a description of the model and the methodology used are given. Section 5 presents the results of the experiments conducted on a famous European real-world credit card transaction dataset. Section 6 concludes our paper with a brief description of our feature work.

2 Related Work

Research on detecting abnormal transactions has increased due to their potential social and economic significance. The purpose of this section is to discuss several effective studies on detecting credit card fraud transactions. For example, in their paper [26], the authors developed an innovative idea to classify credit card fraud transactions. The idea is to extract transactional behaviors from cardholders in order to learn representations of new transaction behaviors in order to detect credit card fraud. A recurrent neural network is used to learn long- and short-term transactional habits from users and to capture changes in behavior caused by different time intervals between transactions. Using a time-aware-attention module, their consecutive historical transactions with time intervals are used to extract behavioral information, which allows the proposed model to capture behavioral motives and periodicities within their historical transactional behaviors, as well as their cumulative transactional history. An interaction module is designed to learn more comprehensive and rational representations. While this can be considered an innovative approach, offering a promising avenue for preventing illegal transactions through the utilization of recurrent neural networks and time-aware-attention modules, certain limitations warrant consideration. Key questions regarding the proposed model's real-world application include its potential lack of generalization to various user actions, sensitivity to data quality, interpretability issues in learned representations, and the trade-off between complexity and explainability. Addressing these concerns will be crucial for enhancing the model's efficiency and ensuring its robust performance across a broader spectrum of situations.

Similarly, [27] solves the problem of credit card fraud transactions using the combination of various machine learning algorithms to obtain accurate results. Their solution aims at exploiting the efficiency of autoencoder techniques for extracting the predictive features of the high-dimensionality dataset from the low-dimensionality dataset. After that, they implemented a probabilistic random forest ensemble learning model for classifying fraudulent transactions. For accurate results, they applied three resampling methods, which are the synthetic minority oversampling technique, the adaptive synthetic approach, and the Tomek link technique. Based on many experiments conducted, it appears that the performance of their solution does not vary much whether resampling schemes are applied to the dataset or not. As a result, their solution is efficient for an imbalanced credit card dataset and will stop illegal and fraudulent transactions. Additionally, the presented credit card fraud detection solution using machine learning algorithms faces limitations in interpretability and potential information loss with autoencoder techniques. The marginal impact of resampling methods on performance prompts questions about their necessity, and the model's generalizability to varied fraud scenarios remains uncertain. Addressing these concerns is essential for enhancing the solution's adaptability and reliability in practical applications.

Moreover, the paper [28] proposes a new approach to identify fraudulent transactions based on deep learning and the advanced feature engineering technique of homogeneity-oriented analysis. To evaluate their solution, they conducted many experiments on a real credit card dataset. Those experiments declare that the proposed solution is efficient and superior to other methods for classifying illegal transactions. Although this framework may require specific details, comparative analyses, and comprehensive validation metrics, it raises concerns about the method's robustness and reported superiority over alternatives. Strengthening credibility requires further clarification and a more rigorous evaluation for broader applicability. Similarly, in paper [29], the authors propose a combination of supervised and unsupervised techniques for developing a well-fitting model for credit card fraud detection. Their solution was evaluated through several experiments, and they got better results. However, there are some limitations stemming from a lack of specific details and comparative analyses. The claim of better results lacks comprehensive metrics and comparisons with existing models, raising concerns about the proposed approach's robustness and generalizability. Detailed explanations and a thorough evaluation are necessary to establish the efficacy and limitations of their model in practical fraud detection scenarios.

The work presented in the paper [30] proposes a gradient boosting tree classifier for the real-time identification of credit card fraud transactions on the streaming Card-Not-Present network. Transactions are investigated by using various attributes of card transactions. Handcrafted numerical, categorical, and textual attributes are combined to form a feature vector to be used as a training instance. This work also includes vectors from a character embedding model trained on merchant names to determine categorical values and includes vectors from an aggregation of transaction categories. Furthermore, the authors propose a new strategy for the generation of training datasets based on a sliding window approach over a specified period of time to adapt to changes in the trend of fraudulent transactions. Experiments on real credit card transactions evaluate the feature engineering strategy and the methodology for automatically generating training sets. In addition, the lack of explicit discussion on model interpretability and potential bias in feature engineering suggests a need for transparency and fairness assessments. Furthermore, the paper could benefit from a thorough comparison with existing methods to establish the novelty and superiority of the proposed approach.

Jiang et al. [31] used a long short-term memory neural network as a weak learner for AdaBoost machine learning algorithms to classify non-authorized transactions. The suggested solution was applied to predict fraud in transactions in the European real-world credit card dataset. To fix the imbalanced dataset issue, they implemented an oversampling technique called the synthetic minority oversampling algorithm combined with the edit nearest neighbor technique. The experiments conducted against many machine learning models, including decision trees, multilayer perceptrons, support vector machines, AdaBoost, and long short-term memory neural networks, demonstrated the superiority of the proposed solution, although the absence of detailed performance metrics and comparisons with contemporary methods raises concerns about the generalizability and true efficacy of the proposed approach. Hence, the paper [33] proposes a new solution for identifying non-authorized transactions in the European dataset. They implemented an attention-long short-term memory neural network and the SMOTE technique for balancing the dataset. For feature selection, they applied a method called the uniform manifold approximation and projection technique (UMAP). To conclude, the proposed solution showed promising results in the experiments. Furthermore, the lack of detailed performance metrics and comparisons with alternative methods raises concerns about the solution's robustness and generalizability. Future work should include a more comprehensive evaluation and comparative analysis to establish the true effectiveness and potential limitations of the proposed approach in diverse transaction scenarios.

Recently, we have found, in the literature, many other innovative methods to deal with the imbalanced credit card transaction dataset issue by using deep learning techniques. For example, in paper [34], the authors used variational autoencoder technology to generate new fraud samples to balance the dataset used for evaluation. The experiments carried out highlight the strong ability of the proposed method against the synthetic minority oversampling method and adversarial generative neural networks. The method's limitations may include potential biases introduced during the generation process. Additionally, the paper lacks a comprehensive exploration of real-world applicability and the impact of varying degrees of fraud complexity. Further research is needed to assess the robustness of the proposed approach in diverse scenarios and to understand its limitations in handling evolving fraud patterns. Likewise, the paper [35] proposes a novel approach to handle imbalanced issues in their credit card dataset by implementing generative adversarial neural networks (GAN) for enhancing the classification of illegal transactions. This approach is described as follows, training a generative adversarial neural networks model to mimic minority class samples for generating new fraud samples, which merged with the training dataset. The conducted experiments prove the importance of using a GAN model as an oversampling technique. The classifiers trained on the oversampled dataset outperform those trained on the original dataset. Additionally, potential limitations may arise from the synthetic samples' fidelity compared to real-world fraud patterns. The study focuses on improved classification performance but lacks a thorough examination of the GAN-generated samples' generalization to diverse fraud scenarios and potential biases introduced during the oversampling process. Further investigation is essential for a comprehensive understanding of the proposed approach's applicability and limitations in practical settings. Furthermore, the proposed solution for detecting fraudulent transactions in this paper [36] aims to exploit the strong abilities of two techniques: the sparse autoencoder for reconstructing a new representation of the dataset and the support vector machine classifier for classifying fraud transactions based on the sparse autoencoder's features. To demonstrate the superiority of their proposed solution, they conducted numerous experiments comparing it to the following classifiers: J48, naive Bayes, random forest, and SVM, showcasing the proposed approach's ability to detect illegal transactions. The model's dependence on pre-defined features and its varying performance across diverse datasets could pose limitations. Although the study showcases effectiveness against specific classifiers, it lacks a thorough analysis of adaptability to changing fraud patterns and real-world challenges. To comprehensively understand the proposed approach's strengths and weaknesses in dynamic and complex fraud scenarios, additional exploration is required.

Raghavan et al. [37] proposed a new deep learning-based solution for detecting illegal transactions. They combined the following methods to construct a well-fitting classifier. BiLSTM, MaxPooling, and BiGRUMaxPooling, which are based on bidirectional long short-term memory (BiLSTM). They conducted several experiments to illustrate the outperformance of the proposed solution against traditional models. The study may lack a thorough assessment of the model's adaptability to diverse fraud patterns and real-world scenarios, potentially limiting its generalization. Additionally, there's a need to explore biases introduced during training and improve the classifier's interpretability for practical reliability. Further research is crucial to enhance the proposed approach's robustness in dynamic fraud detection environments.

The authors in [28] employed a deep learning model combined with an intelligent feature engineering process using the method called homogeneity-oriented behavior analysis technique (HOBA). Several experiments carried out on a real-world credit card transaction dataset illustrated that the proposed solution is efficient at distinguishing between legal and illegal transactions. The study's limitations may stem from a potential lack of exploration into the model's generalization across diverse datasets and the interpretability of HOBA-derived features. Further research is essential to comprehensively assess the proposed solution's adaptability and reliability in varying real-world scenarios and potential biases introduced during training. In the paper [38], a long short-term memory neural network technique was applied for the identification of fraud transactions. For evaluation, a credit card fraud transaction is used, and the results are compared with an autoencoder architecture, a random forest classifier, a support vector machine algorithm, and a logistic regression classifier. Those comparisons show the out-performance of the proposed technique in distinguishing abnormal transactions in terms of accuracy. However, the study may lack a comprehensive analysis of the model's generalization across diverse datasets and real-world scenarios, warranting further research for a thorough understanding of its adaptability and potential limitations. Another study, cited in [39], used artificial neural networks for detecting fraud transactions in a real-world dataset. The results of the experiments show that this model performs equally well on the training data. Also, it reveals the superiority of artificial neural networks over the logistic regression classifier on the test data. However, the study's limitations may include a lack of exploration into the model's generalization across diverse datasets and potential biases, necessitating further investigation for a comprehensive assessment of its adaptability and robustness in practical scenarios. Deepthi Sehrawat et al. [40] focus on using an autoencoder in combination with Gated Recurrent Unit (GRU) and long short-term memory (LSTM) models for detecting fraudulent transactions. Their method involves initially passing data through the autoencoder without labels and then feeding the autoencoder's output into the LSTM model with labels to detect fraud. While this approach shows promise, a notable limitation is the potential for overfitting due to the complexity of deep learning models, which may impact the system's ability to generalize across different datasets and adapt to new fraudulent techniques.

J. Karthika et al. [41] address the significant issue of credit card fraud (CCF) in the context of the growing reliance on online payments within the financial industry. The increase in internet usage has led to higher instances of fraud and a loss of trust in online banking, causing substantial financial losses for institutions and merchants due to unauthorized transactions. The study highlights challenges in CCF detection, including the availability of public data, high false alarms, data imbalance, and the evolving nature of fraud. While traditional machine learning (ML) techniques have shown limited efficiency, the paper proposes a solution using deep learning (DL). Specifically, it introduces a one-dimensional dilated convolutional neural network (DCNN) designed to address CCF detection issues by learning both spatial and temporal features. The DCNN model incorporates a dilated convolutional layer (DCL) to enhance performance. Data imbalance is managed through under-sampling and oversampling techniques. Experiments on three datasets demonstrate that the proposed DCNN model, with sampling techniques, achieved an accuracy of 97.39% on a small card database, outperforming the existing CNN model, which achieved 94.44% accuracy on the same dataset. However, the interpretation of the obtained results can be challenging due to the complex techniques used. Table 1 presents a summary of recent works addressing credit card fraud detection, detailing the methods, key findings, and limitations of each approach. These studies explore a variety of machine learning and deep learning techniques, such as recurrent neural networks with time-aware attention, autoencoders, and generative adversarial networks, applied to manage imbalanced datasets and improve fraud detection accuracy. Key findings demonstrate the effectiveness of these models in extracting behavioral patterns, balancing datasets, and enhancing classification through innovative oversampling and feature engineering techniques. However, each study also presents unique limitations. Common challenges include interpretability issues, generalizability across diverse

Table 1 Summary of recent studies in fraud detection

Study	Method	Key findings	Limitations
[26]	Recurrent neural network (RNN) with time-aware attention	Extracts behavioral patterns for detecting fraud; captures long- and short-term transactional habits	Issues with generalization, sensitivity to data quality, interpretability, and complexity
[27]	Autoencoder + Probabilistic random forest	Efficient for imbalanced datasets; uses SMOTE, ADASYN, and Tomek Link resampling techniques	Limited impact of resampling on performance, interpretability issues, potential information loss with autoencoders
[28]	Deep learning with homogeneity-oriented analysis	Effective and superior for identifying fraudulent transactions	Lacks robustness verification and requires comparative analysis
[29]	Combination of supervised and unsupervised learning	Achieved high accuracy in experiments	Limited details on evaluation metrics, requires comparative analysis for robustness
[30]	Gradient boosting with feature engineering	Combines various attributes; includes training set generation with sliding window approach	Needs transparency in feature engineering and interpretability; lacks comparison with other methods
[31]	LSTM with AdaBoost	Applied SMOTE and edit nearest neighbor techniques; performs well against traditional models	Limited performance metrics; requires validation of generalizability
[33]	Attention-LSTM with SMOTE and UMAP	Shows promising results in detecting non-authorized transactions	Requires comprehensive evaluation and comparison with alternative methods
[34]	Variational autoencoder (VAE) for oversampling	Effective for balancing datasets; outperforms SMOTE and GANs	Potential biases in synthetic data generation; lacks real-world applicability testing
[35]	Generative adversarial network (GAN)	Improves classification by generating new fraud samples	Fidelity of synthetic samples questionable; limited exploration of generalization to diverse scenarios

Table 1 continued

Study	Method	Key findings	Limitations
[36]	Sparse autoencoder + SVM	Detects fraud effectively in comparison with traditional classifiers	Depends on pre-defined features; may struggle with adaptability to evolving fraud patterns
[37]	BiLSTM with MaxPooling and BiGRUMaxPooling	Outperforms traditional models in experiments	Limited assessment of adaptability and potential biases in training
[28]	Deep learning with homogeneity-oriented behavior analysis	Efficient for distinguishing legal and illegal transactions	Potential limitations in model interpretability and generalization across datasets
[38]	LSTM for fraud detection	Outperforms autoencoder, RF, SVM, and LR in accuracy	Lacks comprehensive analysis on adaptability and generalization to real-world scenarios
[39]	Artificial neural network (ANN)	Performs well on training data; superior to logistic regression on test data	Needs further exploration on generalization across datasets and bias issues
[40]	Autoencoder + GRU + LSTM	Promising for fraud detection	Risk of overfitting, which may limit generalization
[41]	Dilated convolutional neural network (DCNN)	Designed to learn spatial and temporal features for fraud detection	Limited information on robustness across varied fraud patterns

real-world scenarios, and potential biases introduced by synthetic data generation. Additionally, while some methods achieve high accuracy, there are concerns about their robustness, adaptability to evolving fraud patterns, and real-world applicability. This overview emphasizes the need for further research to address these limitations and improve the reliability of fraud detection systems in practical applications. In contrast to prior research efforts, our paper introduces an advanced fraud detection system utilizing cutting-edge deep learning capabilities. Our innovative approach integrates an autoencoder with a support vector machine (ASVM) and an attention-long short-term memory neural network for gradient boosting (GB_ALSTM), effectively addressing challenges posed by imbalanced datasets and surpassing limitations present in existing models. Unlike some referenced works relying solely on individual deep learning methods, our ASVM technique not only tackles these limitations but also enhances adaptability, providing a comprehensive solution to augment the classifier's ability in uncovering hidden patterns associated with fraudulent transactions. This method is based on two advanced techniques: an autoencoder, which serves as a superior architecture for simulating fraud schemas, and SVM, a robust classifier to distinguish and control the generated fake fraudulent transactions identified by the autoencoder. The primary goal is to eliminate those not conforming to the fraudulent scheme. Additionally, for classification purposes, we implement a gradient boosting algorithm combined with the ALSTM deep learning architecture, known for its efficiency in handling sequential data. Through rigorous experimentation on a real-world dataset, our proposed solution consistently demonstrates superior performance, particularly excelling in abnormal transaction detection.

3 Background

3.1 Autoencoder Technique

An autoencoder is a type of neural network architecture that is trained to reconstruct its input. It consists of two main parts: an encoder and a decoder. The encoder compresses the input into a lower-dimensional representation. The decoder then takes this compressed representation and reconstructs the original input. This approach can be used for a variety of tasks, such as dimensionality reduction, generative modeling, and anomaly detection. The training process for an autoencoder involves feeding the network input data and then adjusting the weights of the network to minimize the reconstruction error between the input and the output. The encoder and decoder can have different architectures, such as fully connected layers or convolutional layers [42]. There are different variations of autoencoders, including the following:

- Variational autoencoder (VAE), which adds a probabilistic interpretation to the bottleneck, allowing the autoencoder to generate new samples from the learned distribution.
- Denoising autoencoder (DAE), which is trained to reconstruct the original input from a corrupted version of it.
- Convolutional autoencoder (CAE), which uses convolutional layers instead of fully connected layers in the encoder and decoder.

As illustrated in Fig. 1, it has two parts. One is an encoder, and the other is a decoder. These two components can be thought of as two stochastic functions $D : \mathbf{R}^n \rightarrow \mathbf{R}^d$ and $E : \mathbf{R}^d \rightarrow \mathbf{R}^n$ where $d \leq n$. The D maps the data point from data space n to feature space d, and D produces a new reconstruction of the data point. This process can be formulated mathematically as follows: let $X = \left\{x_i / i \in \mathbf{R}^n\right\}$ be the non-authorized transactions in the training dataset, and we have the following equations:

$$Z = D(w, X) \tag{1}$$

$$\hat{X} = E(\hat{w}, Z) \tag{2}$$

where D and E are encoder and decoder functions, respectively, and w and $\hat{w}$ are the encoder and decoder parameters. Training an autoencoder is to minimize a pre-defined

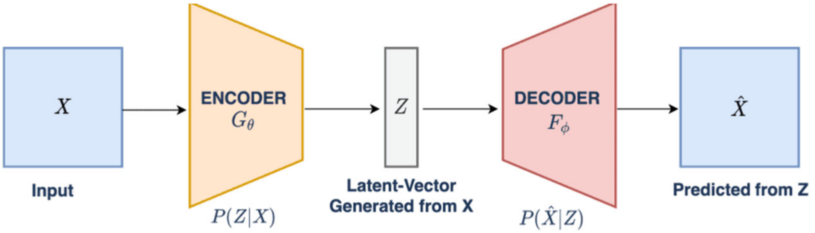


Fig. 1 Architecture of the autoencoder

Table 2 Autoencoder architecture

Encoder	Decoder
Datasets (31 features)	8 layers fully connected
23 layers fully connected	17 layers fully connected
Dropout layer (0.1)	Dropout layer (0.2)
19 layers fully connected	19 layers fully connected
Dropout layer (0.2)	Dropout layer (0.1)
17 layers fully connected	23 layers fully connected
8 layers fully connected	31 layers fully connected

loss function using the gradient descent or stochastic gradient descent algorithms. The function to be minimized is the following:

$$J(w, \hat{w}) = \min \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - \hat{x}_i \right\|_2^2 \tag{3}$$

where $\hat{X} = \left\{ \hat{x}_i / i \in R^n \right\}$ are the new fraudulent transactions generated.

Table 2 describes the autoencoder architecture used in this paper to fix the imbalanced issue in the European dataset. From this table, it is clear that the proposed autoencoder has parts: the encoder, which maps the training fraud transaction dataset from 31 features into 8 features. In the second phase, the decoder reconstructs the training fraud transactions dataset from 8 features into 31 features. Algorithm 1 summarizes the main steps of oversampling techniques using autoencoder.

Algorithm 1 Autoencoder workflow for data reconstruction.

```
Input: Training dataset $D_{\text{train}}$ consisting of $n$ samples, each represented by $\mathbf{x}_i \in \mathbb{R}^d$
Initialization:
  Initialize the encoder network $f(\mathbf{x}; \theta_e)$ and decoder network $g(\mathbf{z}; \theta_d)$ with random parameters $\theta_e$ and $\theta_d$
  Define the latent space size $m$ and the learning rate $\eta$
Preprocessing: Normalize or standardize the input data $D_{\text{train}}$ for better model performance
Data Encoding and Decoding:
for each sample $\mathbf{x}_i$ in $D_{\text{train}}$ do
    Compute the latent representation $\mathbf{z}_i = f(\mathbf{x}_i; \theta_e)$ using the encoder
    Reconstruct the input as $\hat{\mathbf{x}}_i = g(\mathbf{z}_i; \theta_d)$ using the decoder
end for
Reconstruction Loss Calculation:
  Compute the reconstruction loss $\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ as the mean squared error between the original and reconstructed data
Backpropagation:
  Update the parameters $\theta_e$ and $\theta_d$ using backpropagation and gradient descent to minimize the reconstruction loss
Output: Trained encoder $f(\cdot; \theta_e)$ and decoder $g(\cdot; \theta_d)$ networks
```

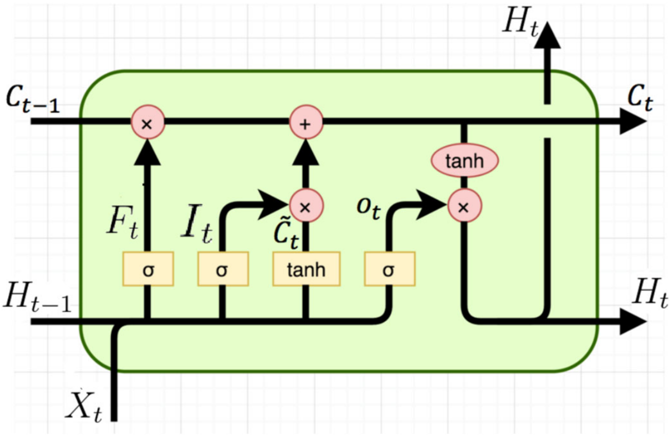


Fig. 2 Long- and short-term memory (LSTM) diagram

3.2 Long Short-Term Memory Networks

Recurrent neural networks (RNNs) with long-term dependencies, such as long short-term memory (LSTM), are designed to handle sequential data and make predictions, as shown in Fig. 2. The LSTMs are well suited to tasks such as speech recognition, language translation, and captioning of images. A traditional RNN uses a single layer of neurons to process sequential data, but LSTM introduces a new architecture that includes memory cells and their updates $C, \widetilde{C} \in R^d$, input gates $I \in R^d$, forget gates $F \in R^d$, and output gates $O \in R^d$. Memory cells are responsible for storing information over a prolonged period of time, while gates control the flow of information into and out of memory cells. The input gate controls how much new information is added to the memory cell, the forget gate controls how much of the previous state is retained, and the output gate controls what information is passed on to the next layer of the network. This allows LSTMs to selectively retain or discard information as they process the sequential data, which helps prevent the "vanishing gradients" problem that plagues traditional RNNs. LSTMs have been extremely successful in various NLP tasks, such as language translation, speech recognition, and text summarization. They are also used in image captioning and video analysis tasks. The LSTM architecture has been further evolved to create new architectures such as the Gated Recurrent Unit (GRU), which is similar to the LSTM but with fewer parameters and gates [43, 44].

Formally, the "forget gate" is the first layer in the LSTM architecture, and its goal is to control the information that can be transformed into memory by removing or keeping that information. Its inputs are the previous state H_{t-1} and an input vector X_t. The outcoming from this cell is a number between 0 and 1 for each sample in the memory vector C_{t-1}.

$$F_t = \sigma(W_F.\left[H_{t-1}, X_t\right] + B_t) \tag{4}$$

A strict 1 would leave the memory unchanged, whereas a strict 0 would set the corresponding component in the memory vector to zero. The second layer is called the "input gate" and consists of two neural network layers. Its objective was to decide which information was going to be stored and where it would be stored in the memory. $\widetilde{C}_t$ denotes the new memory vector, and I_t its vector consists of the proportion in which the current data inside the memory will be overwritten by the new information $\widetilde{C}_t$

$$I_t = \sigma(W_i.\left[H_{t-1}, X_t\right] + B_i) \tag{5}$$

$$\hat{C}_t = than(W_c.\left[H_{t-1}, X_t\right] + B_c) \tag{6}$$

Once the outputs of the gates are generated, then those outputs are used to manipulate the current memory

$$C_t = F_t * C_{t-1} + I_t * \hat{C}_t \tag{7}$$

Finally, the output gate generates the output using Eqs. 8 and 9:

$$O_t = \sigma(W_o\left[H_{t-1}, X_t\right] + B_o) \tag{8}$$

$$H_t = O_t * than(C_t) \tag{9}$$

where $X \in R^d$ is an input variable, and it can have a variety of dimensions (in our study, it denotes the training set with m = 30 features). $W_i, W_o, W_c \in R^{d\times(d+m)}$ are parameters matrices, and the bias vectors $B_f, B_i, B_c, B_o \in R^d$ constitute the parameters of the LSTM model. Furthermore, the notation [,] refers to the concatenation of two vectors.

3.3 Attention Mechanism

Attention is a powerful mechanism for enhancing the performance of encoder-decoder algorithms. The idea behind this technique was to allow the encoder to extract the most important information from the encoded dataset to construct a new representation of this dataset [45], by a weighted combination of all of the encoded input features, with the most important vectors being attributed the highest weights.

The mechanism of this technique can be described as follows. First, we encode the input of the long short-term memory network into a set of hidden states. These hidden states represent many aspects or features of the input. After that, the attention mechanism assigns a weight to each hidden state, which indicates the importance of the state for predicting fraudulent transactions. The weights are typically calculated using a dot product between the hidden states and a set of learnable parameters called attention weights. Finally, the output of the attention mechanism is a weighted sum of the hidden states, where the weights are determined by the attention mechanism. This enhances the model by allowing it to focus on certain parts of the input that are relevant to the current task.

In our paper, the LSTMs used generate a sequence of annotations $(h_1, h_2, ..., h_n)$ for each input vector. The context vector C_i for the output value is generated using the weighted sum of the annotations.

$$C_i = \sum_{j=1}^{n} a_{ij} h_j \tag{10}$$

where a_{ij} denote the weights, and it is calculated by the following formula:

$$a_{ij} = \frac{e^{\alpha(S_{i-1},h_j)}}{\sum_{k=1}^{n} e^{\alpha(S_{i-1},h_k)}} \tag{11}$$

$\alpha(S_{i-1}, h_j)$ is an alignment model. Their objective was to characterize the matching capacity between the inputs near position j and the outputs at position i.

Figure 3 describes the LSTM classifier for the identification of fraud transactions. This figure can be formulated as follows; first, let $X = \left\{(x_i, y_i)_{i=1}^{n}\right\}$, be the training dataset. This dataset is passed through six LSTM layers to generate nine annotation vectors $h_{i\in\{1,\ldots,6\}}$. After that, an attention layer is used to generate the context vector C_i (Eq. 10). In the next step, we used a softmax layer to predict the y vector, which consists of 1 if the transaction is fraudulent, otherwise 0 if the transaction is non-fraudulent based on the context vector.

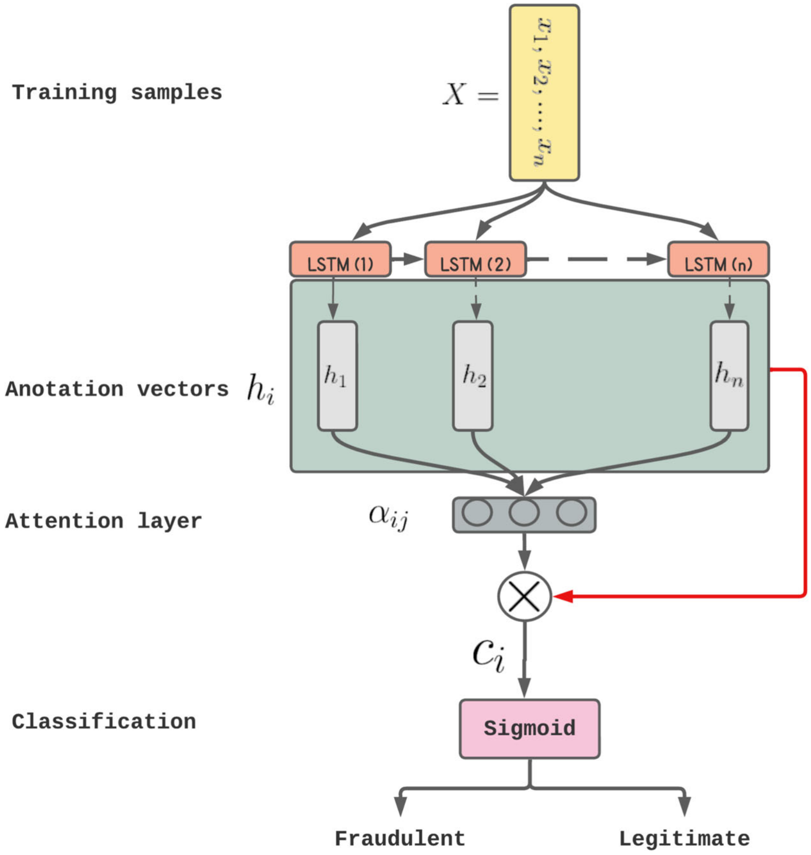


Fig. 3 The proposed attention-long short-term memory network structure

3.4 Gradient Boosting

Gradient boosting is a robust ensemble learning technique designed to combine multiple weak learners into a strong predictive model. Unlike single-model approaches that focus on optimizing performance individually, gradient boosting iteratively improves prediction accuracy by addressing the weaknesses of its predecessors. This methodology enhances the overall prediction capability of machine learning models. The process involves training a sequence of models, where each subsequent model focuses on minimizing the residual errors of the prior model. The iterative refinement ensures that the ensemble converges towards a well-fitted predictive function. Specifically, in the context of fraud detection, we employ attention-based long short-term memory (ALSTM) networks as base learners within the gradient boosting framework.

Figure 4 illustrates the process of constructing a Gradient Boosted ALSTM (GB-ALSTM) ensemble for fraud detection. The ensemble consists of N ALSTM models. Initially, the first model ($ALSTM_1$) is trained using the feature dataset X and the

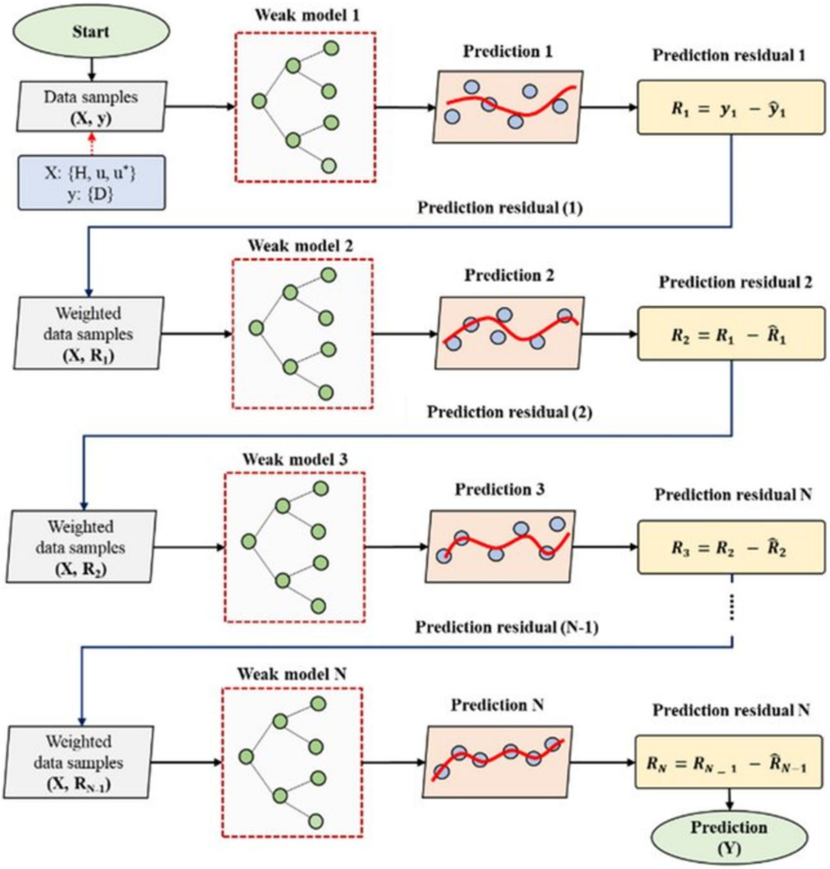


Fig. 4 Gradient Boosted ALSTM framework for fraud detection

target variable y. The predictions ($\hat{y}_1$) are then used to compute residual errors ($r_1 = y - \hat{y}_1$), which become the target for the next model. The second model ($ALSTM_2$) is subsequently trained on the same feature dataset X but with r_1 as the target variable. This iterative procedure continues until all N models are trained, with each model progressively refining the predictions of its predecessors.

Algorithm 2 presents the detailed steps of the GB-ALSTM approach, following the principles introduced by Friedman (2001) to enhance the performance of ALSTM networks.

Algorithm 2 GB-ALSTM Algorithm.

```
1: **Input:** Training set $T = \{(x_i, y_i)\}_{i=1}^{m}$, ALSTM as base learner, number of estimators ($n\_estimators$)
2: Initialize $h^0(x)$ with a constant: $h^0(x) = \arg\min \sum_{i=1}^{n} L(y_i, 0)$
3: **for** $t = 1, \ldots, n\_estimators$ **do**
4:     Compute residuals: $r_i^t = -\frac{\partial L}{\partial h^{t-1}(x_i)}$
5:     Train $ALSTM_t(x)$ using the dataset $(x_i, r_i^t)$
6:     Calculate step length: $\alpha^t = \arg\min_{\alpha} \sum_{i=1}^{n} L(y_i, h^{t-1}(x_i) + \alpha h^t(x_i))$
7:     Update the model: $h(x) = h(x) + \alpha^t h_t(x)$
8: **end for**
9: **Output:** Final ensemble model $h(x)$
```

In this algorithm, α^t represents the weight of the t-th model, while $h^t(x)$ denotes the predictions of the corresponding ALSTM base learner. The loss function L is defined as follows:

$$L(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - h(x_i))^2 \tag{12}$$

The residuals, r_i^t, representing the negative gradient of the loss function, are computed as follows:

$$r_i^t = -\frac{\partial L(w)}{\partial h(x_i)} = -(y_i - h(x_i)) \tag{13}$$

Gradient boosting operates in the function space rather than the parameter space, enabling a flexible, iterative refinement of predictions. This approach can accommodate various loss functions, including squared error for continuous targets and binary cross-entropy for classification tasks. By utilizing ALSTM networks as base learners, the GB-ALSTM framework is particularly effective in capturing sequential dependencies in fraud detection scenarios, as demonstrated in this study.

4 Research Methodology

4.1 Credit Card Dataset

The dataset employed for our experiments is a publicly available and widely referenced credit card fraud detection dataset, originally introduced in [47]. This dataset

Table 3 Details of the dataset

Total transactions ($\|T\|$)	284,807
Legitimate transactions (T^+)	284,315
Fraudulent transactions (T^-)	492
Number of classes	2
Number of features	31

was created through a collaboration between Worldline, a major payment processing company, and the Université Libre de Bruxelles. It comprises over 280,000 European credit card transactions recorded between September 1st and September 30th, 2013, making it a unique resource as the only publicly available dataset that represents real-world credit card usage patterns. The dataset consists of 30 independent features, anonymized using principal component analysis (PCA). These features include "Amount," "Time," and "V1" through "V28," where the "V" features result from the PCA transformation applied to anonymize the data. The "Time" feature is excluded from our analysis due to prior findings that suggest it contributes to noise without offering meaningful predictive value. Transactions are labeled as either genuine or fraudulent, serving as ground truth for calculating the performance of supervised learning models. However, it is important to note that our method disregards these labels during the synthesis of new class labels. As is common in fraud detection datasets, this dataset is highly imbalanced, with genuine transactions vastly outnumbering fraudulent ones. The detailed breakdown of the dataset's characteristics is presented in Table 3, with 492 fraudulent and 284,315 genuine transactions, resulting in an overall count of 284,807 transactions. A significant challenge in the domain of fraud detection is obtaining accurate class labels for transactions. Privacy concerns necessitate the anonymization or removal of personally identifiable information, making the creation of publicly available datasets with real-world examples challenging.

Figure 5 illustrates the imbalance issue in our dataset, where the fraudulent class is underrepresented. This imbalance can cause the algorithm to pay less attention to fraudulent behavior, highlighting the necessity of applying a resampling technique to achieve accurate results.

Fig. 5 Dataset distribution

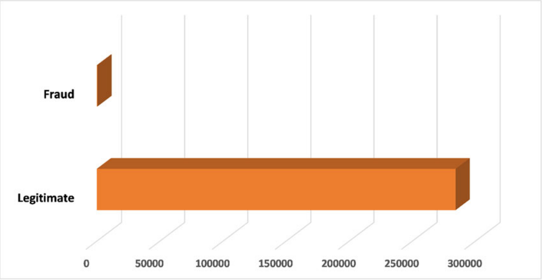


Table 4 Confusion matrix

	Legitimate	Fraud
Legitimate	(TN)	(FP)
Fraud	(FN)	(TP)

4.2 Evaluation Metrics

Every machine learning model needs to be evaluated; therefore, selecting the best evaluation metrics is important [48]. This section presents in detail the measurements which are utilized to evaluate the performance of our classifier. Those measurements are considered performance-evaluating metrics to demonstrate the prediction result related to fraud transactions.

Table 4 describes the confusion matrix evaluation metric, where

- (TP) shows how many fraud samples were successfully categorized.
- (FP) refers to the number of legitimate samples that were incorrectly categorized.
- (TN) describes how many legitimate samples have been accurately categorized.
- (FN) gives the number of misclassification fraud samples.

The following measurements, namely accuracy, precision, recall, specificity, and F1-score, are extracted from the confusion matrix and formulated as follows:

- Accuracy: This metric is a good choice for evaluating a model in a classification problem; it denotes the percentage of samples correctly classified out of all samples. However, the accuracy metric is not suitable when we have an imbalanced dataset. It is not a sufficient measurement to evaluate the performance of a model because it does not take into account the wrong predicted samples.

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

- Specificity: This is an important measurement in fraud transaction detection. This metric is the ratio of legitimate samples correctly classified out of all samples.

$$Specificity = \frac{(\mathrm{TP})}{(\mathrm{FN}) + (\mathrm{TP})} \tag{15}$$

transactions

- Precision is another metric that is used to evaluate the performance of a classification problem; this measurement identifies the ratio of fraud samples correctly classified out of all abnormal transactions.

$$Precision = \frac{(\mathrm{TP})}{(\mathrm{FP}) + (\mathrm{TP})} \tag{16}$$

- Recall is defined as the number of correctly classified fraud samples divided by the total number of relevant samples.

$$Recall = \frac{\text{(TN)}}{\text{(TN)} + \text{(FP)}} \tag{17}$$

- F-measure: This measurement is calculated on the basis of precision and recall; this metric describes the harmonic mean between the precision and recall scores.

$$F - measure = \frac{2 \times \text{(Recall)} \times \text{(Precision)}}{\text{(Recall)} + \text{(Precision)}} \tag{18}$$

4.3 Proposed Solution

In this subsection, we start with the description of our proposed solution for detecting fraudulent transactions. We divided this subsection into two parts: in the first part, we present our solution for handling the imbalanced dataset issue by using an autoencoder combined with the support vector machine classifier. In the second part, we present in detail our solution for classifying fraud transactions using an attention-long short-term memory network as a weak learner for gradient boosting classification.

Figure 6 describes the proposed solution for handling the imbalanced dataset issue. In this solution, we are using an autoencoder for generating new representations of fraudulent transactions. After that, those newly generated samples are fitted into the support vector machine algorithm trained on the randomly undersampled training dataset, to decide if we added the transaction generated to the fraud transactions data or not based on the prediction. This solution can be formulated as follows: let $X = \{x_i\}_{i=1}^{n}$ be the fraud transaction samples in the training dataset. These samples are passed through an autoencoder to generate new samples $\hat{X} = \left\{\hat{x}_i\right\}_{i=1}^{n}$; after that, if ASVM$(\hat{x}_i)$ is a fraudulent transaction, then the fraud transaction dataset is updated $X = X \cup \{\hat{x}_i\}$. This process keeps going until the number of fraud transactions is equal to the number of non-fraud transactions in the training dataset see Algorithm 3.

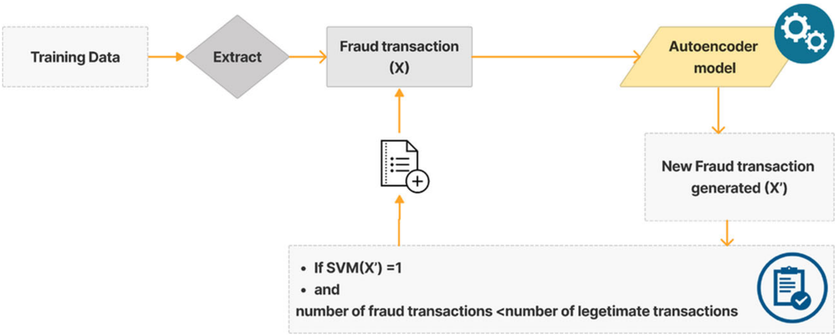

Training Data
Extract
Fraud transaction (X)
Autoencoder model
New Fraud transaction generated (X')
• If SVM(X') =1
• and
number of fraud transactions <number of legetimate transactions

Fig. 6 Architecture of the oversampling approaches proposed

Algorithm 3 ASVM algorithm.

```
Input: Training set $D = \{(x_i, y_i)\}_{i=1}^{m}$, SVM algorithm, Autoencoder model.
Train the SVM model using the undersampled training set $D$
Split the training set into the fraudulent set $D^F$ and the legitimate set $D^L$
while $card(D^F) < card(D^L)$ do
    Generate new fraudulent samples $x'' = Autoencoder(D^F)$
    if $SVM(x'') == 1$ then
        Update the fraudulent set $D^F = D^F \cup \{x''\}$
    end if
end while
Return the oversampled training set $D^F = D^F \cup D^L$.
```

Figure 7 describes the proposed solution for classification. Data preprocessing is the first step, in which we scaled the Time and Amount features using the Rebostscaler method. The data are then distributed in training data (70%) and testing data (30%). After that, we fit the training data into the ASVM model to oversample the training dataset. After that, we train the GB_ALSTM model using the oversampled training dataset. The next step is classifying the test samples and calculating the evaluation metrics. Algorithm 4 describes the entire process for detecting fraudulent transactions.

Algorithm 4 Classification of Fraudulent Transactions with Attention-LSTM and Gradient Boosting.

```
Input: Oversampled training dataset $D_{\text{train}}$, Testing dataset $D_{\text{test}}$, ASVM model, Attention-LSTM
    model, Gradient Boosting model
Preprocessing: Scale the Time and Amount features in both $D_{\text{train}}$ and $D_{\text{test}}$
Oversampling: Use ASVM to oversample $D_{\text{train}}$
Train the Attention-LSTM model with the oversampled dataset $D_{\text{train}}$
Predict the class labels for the test dataset $D_{\text{test}}$ using the trained Attention-LSTM model
Evaluation: Calculate the performance metrics: Accuracy, Precision, Recall, F1-Score, etc.
Output: Evaluation metrics for the classification model
```

In addition, using Attention-LSTM and autoencoders for handling the imbalanced learning issue introduces computational costs and scalability challenges. Attention-LSTM, while effective at capturing temporal dependencies and emphasizing key

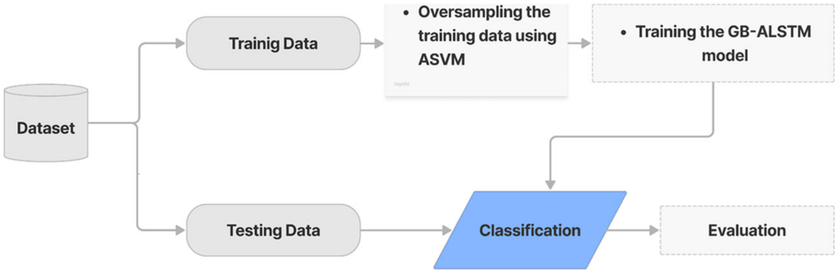

Dataset
Trainig Data
• Oversampling the training data using ASVM
• Training the GB-ALSTM model
Testing Data
Classification
Evaluation

Fig. 7 Architecture of the proposed classifier

features, requires substantial computational resources due to the complexity of its attention mechanisms, especially when applied to large-scale datasets. Similarly, autoencoders, which excel in learning latent representations and generating synthetic samples for balancing datasets, can be resource-intensive during training, particularly with high-dimensional data. These challenges highlight the need for scalable solutions. Future work could explore optimizing these methods through efficient attention mechanisms, parallelization strategies, or hybrid approaches that balance computational efficiency with robust model performance.

5 Results Analysis

In this section, we present the outcomes of our experiments, comparing the performance of our proposed solution with various deep learning techniques: LSTM, ALSTM, CNN, ANN, and BLSTM. While LSTM, ALSTM, and our proposed method are detailed in the background section, we provide brief definitions for the additional models to enhance clarity. CNN, specialized in processing grid-like data such as images through convolutional layers, is employed for its spatial feature extraction capabilities. ANN, a foundational type of neural network with interconnected layers, is widely utilized in diverse machine learning tasks for its versatility. In contrast, BLSTM, a bidirectional recurrent neural network, stands out for its unique ability to incorporate information from both past and future time steps, thereby enhancing its proficiency in capturing complex sequential dependencies. These models collectively serve as benchmarks, and their performance is rigorously compared to underscore the efficacy of our proposed solution in fraud detection within the context of sequential data analysis.

Table 5 presents obtained results of the proposed solution against the competitive models. In general, we noticed the superiority of our work in detecting fraudulent transactions in the European credit card dataset. Regarding the accuracy rating, all

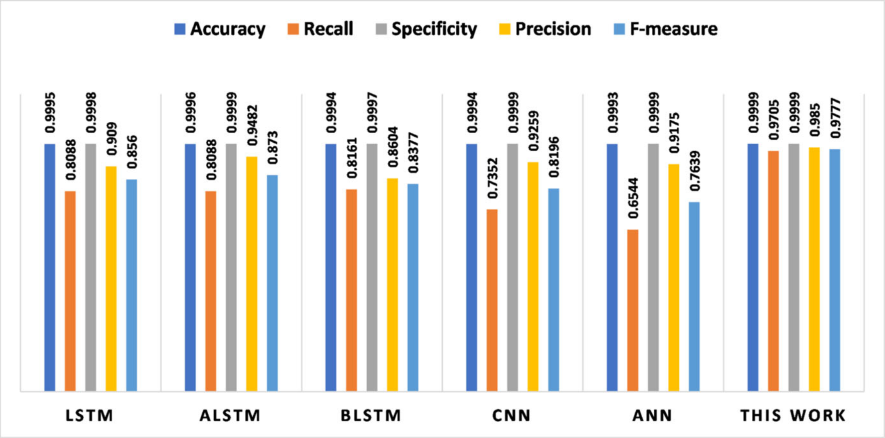


Fig. 8 Performance comparison of fraud detection models

Table 5 Performance evaluation of the proposed solution

Models	Accuracy	Recall	Specificity	Precision	F-measure
LSTM	0.9995	0.8088	0.9998	0.9090	0.8560
LSTM	0.9995	0.8088	0.9998	0.9090	0.8560
ALSTM	0.9996	0.8088	0.9999	0.9482	0.8730
BLSTM	0.9994	0.8161	0.9997	0.8604	0.8377
CNN	0.9994	0.7352	0.9999	0.9259	0.8196
ANN	0.9993	0.6544	0.9999	0.9175	0.7639
Our work	0.9999	0.9705	0.9999	0.985	0.9777

models achieved the same score, which is about 99% of the transactions are classified correctly. In terms of the precision score which describes how a model can classify fraud transactions, our work achieved the highest score, which is 98% of fraud transactions are identified correctly. This, emphasizes the higher performance of our proposed solution in discovering hidden patterns in the credit card dataset that categorizes fraud transactions. The lowest score is achieved by the BLSTM model which is 86% of fraud transactions are classified correctly. Likewise, the proposed model achieved a score of 99% of specificity, which means that the proposed solution also has a strong ability to correctly identify legitimate transactions in the credit card dataset. This score is the best score achieved in the experiments that were conducted. Otherwise, considering the recall score, the proposed solution reached the best rate which is 97%. Therefore, with respect to the F-measure, we obtained the highest score. These results demonstrated the strong ability of our model in stopping non-authorized transactions.

Figure 8 supports the results discussed in Table 5 and emphasizes the superiority of the proposed solution in distinguishing between normal and abnormal transactions in the European credit card transaction dataset.

Figure 9 shows a radar chart comparing the performance of different fraud detection models, including LSTM, ALSTM, BLSTM, CNN, ANN, and our proposed method. Each axis represents one of the evaluation metrics: accuracy, recall, specificity, precision, and F-measure. The radar chart visually highlights how each model performs across these metrics. Our proposed method, marked by a dashed black line, demonstrates superior performance, particularly in recall, precision, and F-measure, where it significantly outperforms the other models. In contrast, while models like LSTM and ALSTM show strong performance, particularly in accuracy and specificity, they do not match the high scores achieved by our method in the other metrics. This comparison underscores the effectiveness of our approach in addressing the class imbalance issue in fraudulent transaction detection and achieving a more balanced performance across multiple metrics.

Figures 10 and 11 display the confusion matrix that occurred in the experiments carried out. From these figures, it is clear that our work achieved the best results. Our proposed solution can classify 85,305 legitimate transactions correctly. In contrast, our model fields to classify 2 legitimate transactions correctly. In addition, the proposed model can correctly classify 132 fraud transactions. Therefore, it classifies 2

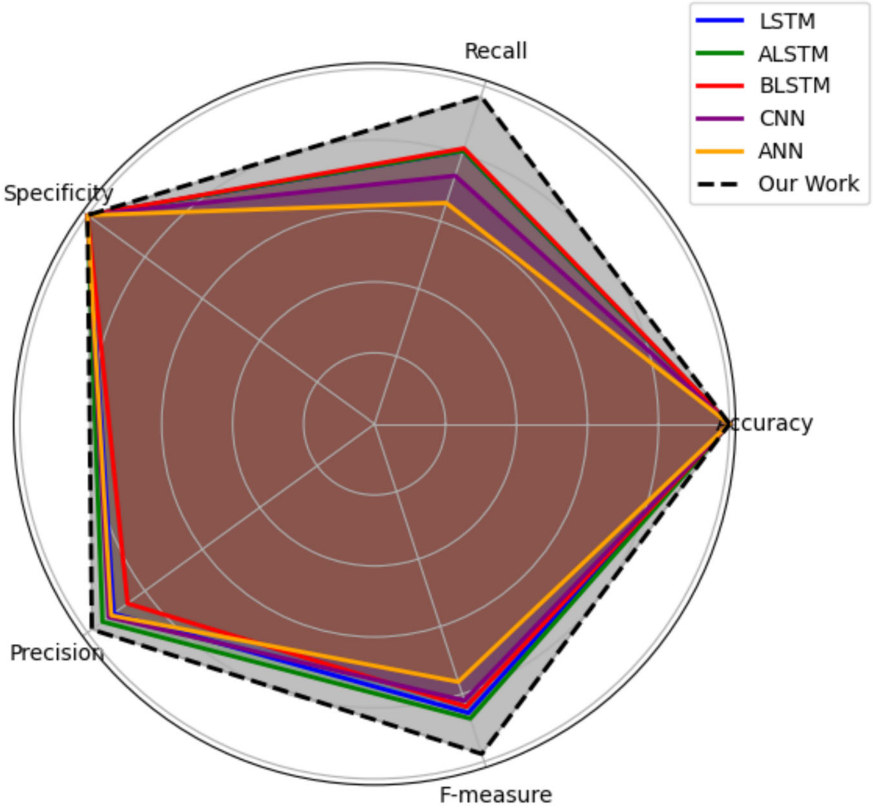
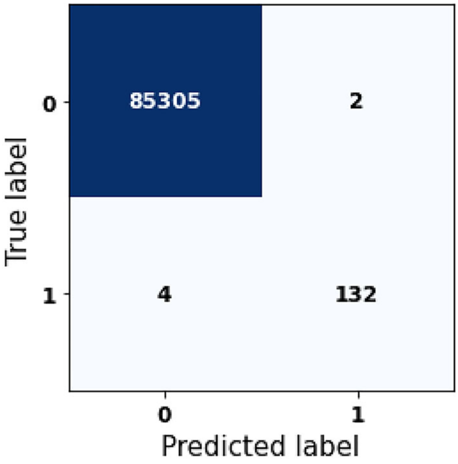


Fig. 9 Comparison of fraud detection models

Fig. 10 The confusion matrix obtained using the proposed solution model



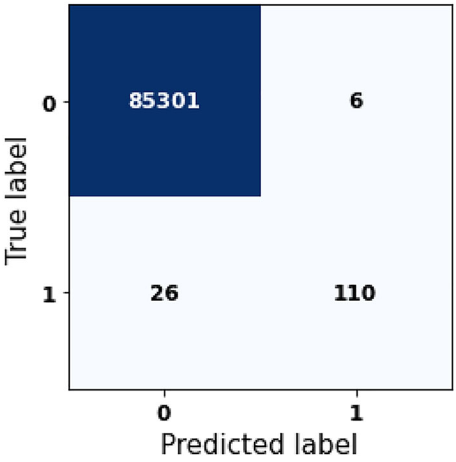


Fig. 11 The confusion matrix obtained using ALSTM model

fraud transactions incorrectly. ALSTM model classifies 85301 legitimate transactions correctly and 6 legitimate transactions wrongly. Likewise, it is able to identify 110 fraud transactions correctly and fields to identify 26 fraud transactions.

Figures 12 and 13 describe the confusion matrix obtained using LSTM and BLSTM, respectively. From these figures, it is clear that the LSTM model can classify 85,296 legitimate transactions correctly and is unable to identify 11 of them correctly. In contrast, this model classifies 110 fraud transactions correctly and fields to classify 26 fraud transactions correctly. Likewise, the BLSTM model can identify 85,289 legitimate transactions correctly and fields to classify 18 legitimate transactions correctly. In contrast, it is able to identify 111 fraud transactions correctly, Hence, it failed to classify 25 fraud transactions as fraudulent transactions.

Figures 14 and 15 show the confusion matrix obtained using CNN and ANN models, respectively. From these figures, the CNN model is able to correctly classify 85,299

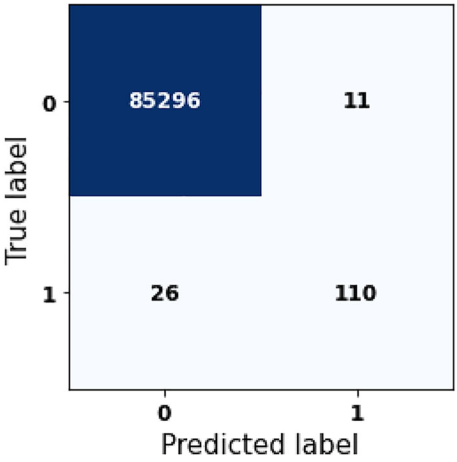
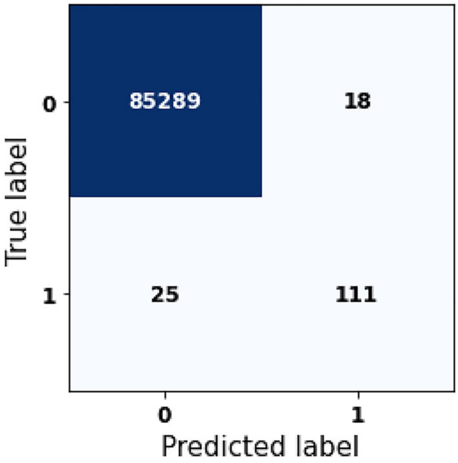


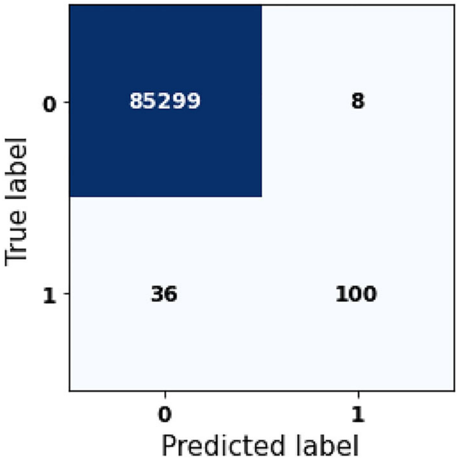
Fig. 12 The confusion matrix obtained using LSTM model

Fig. 13 The confusion matrix obtained using BLSTM model



legitimate transactions and 100 fraud transactions. In contrast, it is not able to correctly identify 8 legitimate transactions and 36 fraudulent transactions. Similarly, the ANN model correctly classifies 85,299 legitimate transactions and 89 fraud transactions. Besides, it classified 8 legitimate and 47 fraudulent transactions wrongly. Otherwise, the observed superiority of our proposed solution can be attributed to its capacity to balance learning from sequential patterns and complex interactions within the data. Traditional models, such as LSTM and BLSTM, demonstrate effectiveness in capturing temporal dependencies but face challenges in handling intricate feature interactions and the imbalanced nature of the dataset. Similarly, CNN, while excelling in spatial feature extraction, underperforms in sequential tasks due to its lack of temporal processing capabilities. ANN's relatively weaker performance stems from its reliance on static feature representation, which limits its ability to effectively leverage sequential information. Although ALSTM introduces attention mechanisms to enhance focus on critical temporal features, it still falls short compared to our approach. This limitation

Fig. 14 The confusion matrix obtained using CNN model



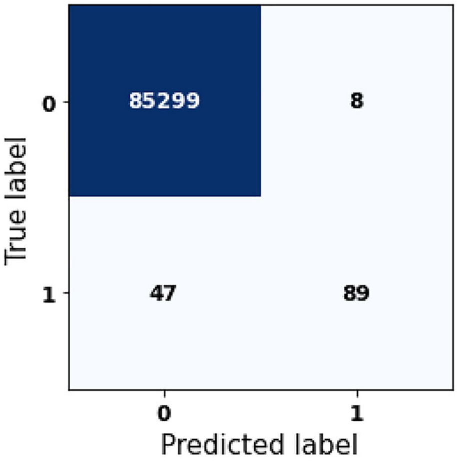


Fig. 15 The confusion matrix obtained using ANN model

arises from its less robust ability to manage extreme class imbalance and the complex fraud patterns inherent in the dataset. By contrast, our proposed solution integrates advanced architectures with enhancements specifically tailored to fraud detection, such as addressing class imbalance and extracting meaningful patterns. This targeted design enables our model to achieve superior recall and precision, effectively identifying fraudulent transactions while minimizing false negatives. However, the proposed solution is not without limitations. The model's complexity increases computational costs, making it less suitable for real-time applications without further optimization. Additionally, its performance heavily relies on the quality and quantity of the training data, which might limit its generalizability to datasets with significantly different characteristics. Future work will address these limitations by exploring strategies to reduce computational overhead and improve adaptability to diverse datasets. Furthermore, we will delve deeper into the qualitative aspects of the model, providing insights into their implications for designing robust fraud detection systems.

Table 6 shows an additional comparison carried out by comparing our work with some existing work. As illustrated in this table, our proposed solution achieved the highest recall (97.05%). The lowest recall is obtained with the KNN model (3.93%).

Table 6 Comparison with some existing methods

Method	Reference	Accuracy	Precision	Recall	Specificity
GRU	[49]	–	0.8626	0.7208	–
LSTM	[49]	–	0.8575	0.7408	–
SVM	[50]	0.9349	0.9743	0.8976	–
KNN	[50]	0.9982	0.7142	0.0393	–
ANN	[50]	0.9992	0.8115	0.7619	–
ALSTM	[33]	0.9672	0.9885	0.9191	–
GB_ALSTM	Our work	0.9999	0.985	0.9705	0.9999

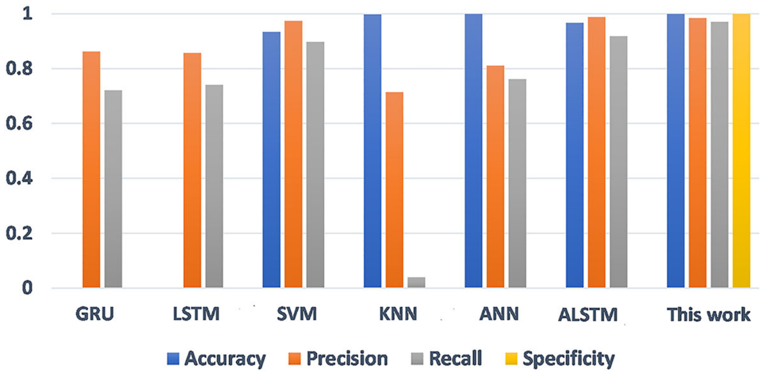


Fig. 16 Comparison of the proposed solution with some relevant works for fraud detection problem

Taking into account the accuracy score, our model got the best rating (99.99%), while the SVM achieved the lowest score, which is approximately 93.49%. In terms of precision, our model and the ALSTM model achieved the best score, which is approximately 98%. Figure 16 shows those results clearly and reveals that our suggested approach is efficient in comparison with other classifiers.

In summary, the evaluation of different deep learning models for fraud detection yields valuable insights. The LSTM model excels in correctly classifying legitimate transactions but faces challenges in identifying some fraud cases. Conversely, the BLSTM model demonstrates strong performance in identifying fraud transactions but encounters difficulties with certain legitimate ones. The CNN and ANN models exhibit proficiency in correctly classifying both legitimate and fraudulent transactions, albeit with limitations in specific instances. Comparing our proposed solution with existing work, our model stands out with the highest recall (97.05%) and accuracy score (99.99%). Notably, our model and the ALSTM model achieve the best precision scores at approximately 98%. These findings underscore the effectiveness of our proposed solution in achieving a balanced and superior performance in fraud detection, contributing to the advancement of robust and accurate systems for securing financial transactions.

6 Conclusion and Future Work

Credit card fraud transactions continue to be a significant challenge for banks and financial institutions. The improvement of algorithms for detecting fraudulent transactions has become a central research focus. Various approaches, including supervised, unsupervised, and semi-supervised techniques, have been explored to address this issue, demonstrating enhanced performance in distinguishing between authorized and unauthorized transactions. In this paper, we proposed an intelligent framework for fraud detection that leverages deep learning approaches, combining attention-long short-term memory (ALSTM) with gradient boosting algorithms and an autoencoder architecture with support vector machines (SVM) to address the class imbalance issue.

We evaluated our proposed solution against other deep learning architectures, such as ALSTM, LSTM, ANN, CNN, and BLSTM. The results demonstrated that our proposed solution outperforms others based on key evaluation metrics, achieving 99.99% accuracy, 98% precision, 99.99% recall, and 97.77% F-measure. While these quantitative results highlight the success of our proposed solution, we recognize the importance of considering its limitations and practical applications. One such limitation is the computational cost of using deep learning models, which is significantly higher than traditional machine learning algorithms. The complexity increases exponentially with the size of the data, potentially posing challenges for real-time systems or large-scale deployments. Additionally, the interpretability of deep learning models remains a concern, particularly when deployed in critical applications like fraud detection, where understanding model decisions is crucial for trust and transparency.

To enhance the solution, we suggest future work focusing on optimization techniques such as hyperparameter tuning using metaheuristic algorithms and distributed platforms like Spark and Databricks to ensure efficient computation. This approach will help in fine-tuning the model and improving its performance in diverse scenarios. Moreover, further experiments with varied credit card datasets are essential to evaluate the generalizability and robustness of the proposed solution across different fraud characteristics. By addressing these aspects and incorporating fairness evaluations, sensitivity analysis, and interpretability improvements, we aim to develop a more comprehensive and reliable fraud detection system that can be practically deployed in real-world financial environments.

Author Contributions Mohammed Tayebi developed the proposed model, performed the analytic calculations, and performed the numerical and simulation results. Said El Kafhali contributed to the interpretation of the obtained results. Both Mohammed Tayebi and Said El Kafhali contributed to the final version of the manuscript. Said El Kafhali supervised the work of this article.

Data Availability The paper uses a European dataset to test the efficiency of the algorithms. This dataset is publicly available at https://www.kaggle.com/mlg-ulb/creditcardfraud and is free of charge.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Wu Y, Xu Y, Li J (2019) Feature construction for fraudulent credit card cash-out detection. Decis Support Syst 127:113155
2. Gianini G, Fossi LG, Mio C, Caelen O, Brunie L, Damiani E (2020) Managing a pool of rules for credit card fraud detection by a game theory based approach. Futur Gener Comput Syst 102:549–561
3. Arora B (2022) A review of credit card fraud detection techniques. Recent Innov Comput 1:485–496
4. Langevin A, Cody T, Adams S, Beling P (2022) Generative adversarial networks for data augmentation and transfer in credit card fraud detection. J Oper Res Soc 73(1):153–180
5. Adewumi AO, Akinyelu AA (2017) A survey of machine-learning and nature-inspired based credit card fraud detection techniques. Int J Syst Assur Eng Manag 8(2):937–953
6. Lakshmi SVSS, Kavilla SD (2018) Machine learning for credit card fraud detection system. Int J Appl Eng Res 13(24):16819–16824

7. Ryman-Tubb NF, Krause P, Garn W (2018) How artificial intelligence and machine learning research impacts payment card fraud detection: a survey and industry benchmark. Eng Appl Artif Intell 76:130–157
8. Tayebi M, El Kafhali S (2024) Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection. Evolutionary intelligence 17(2):921–939
9. Tayebi M, Kafhali SE (2021) Hyperparameter optimization using genetic algorithms to detect frauds transactions. In: The International conference on artificial intelligence and computer vision. Springer, Cham, pp 288-297
10. Patidar R, Sharma L (2011) Credit card fraud detection using neural network. Int J Soft Comput Eng (IJSCE) 1:32–38
11. Maes S, Tuyls K, Vanschoenwinkel B, Manderick B (2002) Credit card fraud detection using Bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies, Vol 261, p 270
12. Nguyen QP, Lim KW, Divakaran DM, Low KH, Chan MC (2019) Gee: a gradient-based explainable variational autoencoder for network anomaly detection. In: 2019 IEEE Conference on communications and network security (CNS) (pp. 91-99). IEEE
13. Rajan R, Chandrasekar A, Cao J (2014) Passivity and passification of memristor-based recurrent neural networks with additive time-varying delays. IEEE Trans Neural Netw Learn Syst 26(9):2043–2057
14. Zheng YJ, Zhou XH, Sheng WG, Xue Y, Chen SY (2018) Generative adversarial network based telecom fraud detection at the receiving bank. Neural Netw 102:78–86
15. Maes S, Tuyls K, Vanschoenwinkel B, Manderick B (2002) Credit card fraud detection using Bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies, Vol 261, p 270
16. Tayebi M, El Kafhali S (2022) Deep neural networks hyperparameter optimization using particle swarm optimization for detecting frauds transactions. In: Advances on smart and soft computing. Springer, Singapore, pp 507-516
17. Tingfei H, Guangquan C, Kuihua H (2020) Using variational auto encoding in credit card fraud detection. IEEE Access 8:149841–149853
18. Lin W, Sun L, Zhong Q, Liu C, Feng J, Ao X, Yang H (2021) Online credit payment fraud detection via structure-aware hierarchical recurrent neural network. In: Thirtieth international joint conference on artificial intelligence (IJCAI-21), pp 3670-3676
19. Chandrasekar A, Radhika T, Zhu Q (2022) Further results on input-to-state stability of stochastic Cohen-Grossberg BAM neural networks with probabilistic time-varying delays. Neural Process Lett 1-23
20. Radhika T et al (2023) Analysis of Markovian jump stochastic Cohen-Grossberg BAM neural networks with time delays for exponential input-to-state stability. Neural Process Lett 55(8):11055–11072
21. Singh A, Ranjan RK, Tiwari A (2022) Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. J Exp Theor Artif Intell 34(4):571–598
22. El Kafhali S, Tayebi M (2022) Generative adversarial neural networks based oversampling technique for imbalanced credit card dataset. In: 2022 6th SLAAI International conference on artificial intelligence (SLAAI-ICAI), IEEE (pp 1-5)
23. Tayebi M, El Kafhali S (2022) Credit card fraud detection based on hyperparameters optimization using the differential evolution. Int J Inf Secur Priv (IJISP) 16(1):1–21
24. Meng C, Zhou L, Liu B (2020) A case study in credit fraud detection with SMOTE and XGboost. In: Journal of physics: conference series (Vol. 1601, No. 5, p 052016). IOP Publishing
25. Tayebi M, El Kafhali S (2023) [Online]. Available at: https://github.com/tayebimed/Fraud-Transactions-Classification-Performance-Analysis-using-Deep-Learning-Approach
26. Xie Y, Liu G, Yan C, Jiang C, Zhou M (2022) Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors. IEEE Trans Comput Soc Syst
27. Lin TH, Jiang JR (2021) Credit card fraud detection with autoencoder and probabilistic random forest. Mathematics 9(21):2683
28. Zhang X, Han Y, Xu W, Wang Q (2021) HOBA: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture. Inf Sci 557:302–316
29. Carcillo F, Le Borgne YA, Caelen O, Kessaci Y, Oblé F, Bontempi G (2021) Combining unsupervised and supervised learning in credit card fraud detection. Inf Sci 557:317–331
30. Itoo F, Singh S (2021) Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. Int J Inf Technol 13(4):1503–1511

31. Jiang C, Song J, Liu G, Zheng L, Luan W (2018) Credit card fraud detection: a novel approach using aggregation strategy and feedback mechanism. IEEE Internet Things J 5(5):3637–3647
32. Esenogho E, Mienye ID, Swart TG, Aruleba K, Obaido G (2022) A neural network ensemble with feature engineering for improved credit card fraud detection. IEEE Access 10:16400–16407
33. Benchaji I, Douzi S, El Ouahidi B, Jaafari J (2021) Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. J Big Data 8(1):1–21
34. Tingfei H, Guangquan C, Kuihua H (2020) Using variational auto encoding in credit card fraud detection. IEEE Access 8:149841–149853
35. Fiore U, De Santis A, Perla F, Zanetti P, Palmieri F (2019) Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Inf Sci 479:448–455
36. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K (2018) Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access 6:52843–52856
37. Raghavan P, El Gayar N (2019) Fraud detection using machine learning and deep learning. In: 2019 international conference on computational intelligence and knowledge economy (ICCIKE). IEEE, (pp 334-339)
38. Alghofaili Y, Albattah A, Rassam MA (2020) A financial fraud detection model based on LSTM deep learning technique. J Appl Secur Res 15(4):498–516
39. Sahin Y, Duman E (2011) Detecting credit card fraud by ANN and logistic regression. In: 2011 international symposium on innovations in intelligent systems and applications. IEEE, (pp 315-319)
40. Sehrawat Deepthi, Singh Yudhvir (2023) Auto-encoder and LSTM-based credit card fraud detection. SN Comput Sci 4(5):557
41. Karthika J, Senthilselvi A (2023) Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique. Multimed Tools Appl 82(20):31691–31708
42. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Asari VK (2019) A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3):292
43. Alarfaj FK, Malik I, Khan HU, Almusallam N, Ramzan M, Ahmed M (2022) Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. IEEE Access 10:39700–39715
44. Abroyan N (2017) Convolutional and recurrent neural networks for real-time data classification. In: 2017 Seventh international conference on innovative computing technology (INTECH). IEEE, pp 42-45
45. Niu Z, Zhong G, Yu H (2021) A review on the attention mechanism of deep learning. Neurocomputing 452:48–62
46. Natekin A, Knoll A (2013) Gradient boosting machines, a tutorial. Front Neurorobot 7:21
47. Credit card fraud dataset. [Online]. Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud/data
48. Prakash A, Chandrasekar C (2015) An optimized multiple semi-hidden Markov model for credit card fraud detection. Ind J Sci Technol 8(2):176
49. Asha RB, KR SK (2021) Credit card fraud detection using artificial neural network. Global Transitions Proc 2(1):35–41
50. Forough J, Momtazi S (2021) Ensemble of deep sequential models for credit card fraud detection. Appl Soft Comput 99:106883

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.