The Journal of Supercomputing (2024) 80:14824–14847
https://doi.org/10.1007/s11227-024-06030-y

An intelligent sequential fraud detection model based on deep learning

Georgios Zioviris[1] · Kostas Kolomvatsos[2] · George Stamoulis[1]

Accepted: 26 February 2024 / Published online: 27 March 2024

Abstract

Fraud detection and prevention has received a lot of attention from the research community due to its high impact on financial institutions' revenues and reputation. The increased use of the web and the provision of online services open up the pathway for exposing these systems to numerous threats and jeopardizing their effective functioning. Naturally, financial frauds are increased in number and form imposing various requirements for their efficient and immediate detection. These requirements are related to the performance of the adopted models as well as the timely response of the decision-making mechanism. Machine learning and data mining are two research domains that can provide a number of techniques/algorithms for fraud detection and setup the road for mitigation actions. However, these methods still need to be improved with respect to the detection of unknown fraud patterns and the incorporation of big data processing mechanisms. This paper presents our attempt to build a hybrid system, i.e., a sequential scheme for combining two deep learning models and efficiently detecting potential financial frauds. We elaborate on the combination of an autoencoder and a Long Short-Term Memory Recurrent Neural Network trained upon datasets which are processed through the use of an oversampling technique. Oversampling is adopted to handle heavily imbalanced datasets which is the 'natural' scenario due to the limited number of frauds compared to the humongous volumes of transactions. The proposed approach tends to capture much more fraud events in comparison with other conventional ML techniques. Our experimental evaluation exposes that our model exhibits a good performance in terms of recall and precision.

Keywords Fraud detection · Autoencoder · Variational Autoencoder · Dimensionality reduction · Oversampling · Recurrent Neural Networks · Long Short-Term Memory networks

Extended author information available on the last page of the article

1 Introduction

1.1 Motivation

Contemporary studies reveal that the fraud detection and prevention market is estimated to be worth $19.5 billion.[1] In parallel to the Consumer Sentinel Network of the USA, of the 3.2 million identity theft and fraud reports that were received in 2019, 1.7 million were fraud-related.

In this total of 1.7 million fraud cases, 23% reported that money was lost exposing the financial damage to institutes and individuals.[2] Fraudulent events ought to be recognized within the minimum possible time upon the reception of streams that convey the relevant financial information of a transaction. Clearly, humongous datasets can be formulated within the administration of a financial institution upon the discussed streams. Datasets will be also characterized by a high complexity due to the multiple features recorded by transactions.

Financial institutions face the critical task of swiftly and effectively identifying and isolating fraudulent transactions while ensuring a seamless customer experience. The term 'swiftly' emphasizes the need for a detection model that operates with minimal delay, safeguarding both customers and institutions from potential issues. Simultaneously, the term 'effectively' underscores the importance of accurate fraud detection, as false alerts could result in unnecessary resource allocation.

Traditionally, the methods adopted for fraud detection are related to (i) manual intervention or (ii) rule-based models with limited success [1]. The manual detection suffers by an increased time required to conclude the final outcome, while rule-based approaches deal with complex rules that should be fired and evaluated before a transaction is characterized as suspicious. In both scenarios (i.e., manual detection and rule-based systems), an increased effort is required to initialize the conditions upon which a transaction may be characterized as fraudulent. In any case, both approaches cannot efficiently detect new, unknown, and complicated fraud patterns.

As financial institutions grapple with the formidable task of swiftly identifying and meticulously distinguishing fraudulent transactions from legitimate ones, they recognize the pivotal role of artificial intelligence (AI) in this critical endeavor. AI-powered fraud detection systems offer unparalleled speed, efficiency, and adaptability. Now, let us delve into the intricate world of machine learning techniques that empower banks to proactively combat fraud before it even occurs.

Machine Learning (ML) can provide techniques toward the development of models that can solve the aforementioned problems. New trends in ML consist of Deep ML (DML) schemes that are capable of identifying more complex patterns upon huge volumes of data.

The application of ML/DML in fraud detection may allow financial institutions to discern genuine transactions from fraudulent events in real-time and through an

[1] https://www.statista.com/statistics/786778/worldwide-fraud-detection-and-prevention-market-size.
[2] https://www.ftc.gov/reports/consumer-sentinel-network-data-book-2019.

automated manner. Apart from the temporal aspect, ML/DML can assist in the definition of models that exhibit higher accuracy than other schemes [2].

Ensemble and sequential models, a powerful class of machine learning techniques, have revolutionized fraud detection in the financial domain. These models combine multiple individual algorithms, leveraging their collective intelligence to achieve superior performance. Ensemble and sequential models aggregate predictions from diverse base models, such as Decision Trees, Neural Networks, or Support Vector Machines. By combining their strengths and compensating for individual weaknesses, models significantly enhance detection accuracy. This translates to identifying more fraudulent transactions while minimizing False Positives. The primary goal of fraud detection is to prevent financial losses. Ensemble models play a pivotal role here. By capturing subtle patterns and anomalies that individual models might miss, they reduce the chances of undetected fraud slipping through the cracks. As a result, financial institutions can proactively mitigate losses by blocking fraudulent transactions promptly. Financial fraud evolves rapidly, with fraudsters devising new tactics to exploit vulnerabilities. Ensemble and sequential models excel in adaptability. When faced with novel fraud patterns, they adjust dynamically, ensuring continued effectiveness. Their robustness stems from diversity—each base model contributes a unique perspective, making the ensemble resilient to changing attack vectors. Financial institutions operate in a tightly regulated environment. Ensemble and sequential models aid in risk management by providing accurate risk assessments. Whether it is assessing credit risk, detecting insider trading, or identifying money laundering, ensemble models offer a holistic view. Regulators appreciate their transparency and ability to quantify uncertainty, aligning with compliance requirements. Efficient fraud detection involves allocating resources judiciously. Ensemble and sequential models strike a balance. They optimize computational resources by distributing the workload across base models. Moreover, their ability to handle large-scale data ensures scalability, crucial for real-time fraud prevention. While preventing fraud is paramount, maintaining a positive customer experience is equally crucial. Ensemble and sequential models strike this delicate balance. By minimizing False Positives, they avoid inconveniencing legitimate customers. This seamless experience fosters trust and loyalty, benefiting financial institutions in the long run. In summary, ensemble and sequential models are not just theoretical constructs; they directly impact the bottom line. Their practical benefits extend beyond accuracy—they safeguard financial stability, enhance risk management, and fortify regulatory compliance. As the financial landscape evolves, embracing ensemble models becomes imperative for staying ahead in the battle against fraud.

An interesting approach is to mix multiple types of ML models (e.g., supervised and unsupervised methods) to create ensemble schemes. It is proven that ensemble models can result in increasing the ‘detection’ capability of the system and identify hidden aspects of data distribution, leading to the recognition of new patterns in fraud management [2]. Evidently, any efficient learning methodology that will reveal the hidden characteristics of the collected data will be able to lead to the recognition of new patterns in fraud management. The need for the discussed models, both simple and ensemble schemes, is imperative due to the financial impact of frauds. ML/DML schemes can alleviate users from the manual detection of frauds

requiring less intervention by financial institutions employees. This automated approach will increase the speed of processing and, consequently, boost the throughput of the monitoring and detection mechanisms making financial institutions that will be capable of dealing with the exponentially growing domain of fraud detection and the relevant data. DML has been widely adopted in many research domains like image processing, speech recognition, and Natural Language Processing (NLP) [3]. Multiple DML models have been proposed with different targets and goals. Some example schemes are autoencoders [4], Convolutional Neural Networks (CNNs) [5], and Recurrent Neural Networks (RNNs) [6]. All of them incorporate mechanism that could be 'aligned' with the training data and learn their distribution to be ready to be adopted in real scenarios. In recent years, DML attracts tremendous growth and attention as the provision of more powerful hardware can facilitate the definition and execution of advanced processing mechanisms, the management of larger datasets, and the support of fast training activities [7].

1.2 Contribution and novelty

Fraudulent events are crucial as they can affect the reputation of the institution while spending resources without any reason. We propose a sequential scheme connecting the hidden layer of an autoencoder (Simple or Variational) with a LSTM Recurrent Neural Network taking advantage of the dimensionality reduction that the autoencoder has performed. In that way, the input of the LSTM model is cleaned from the redundant features, not to mention reduced in data space. We adopt the Synthetic Minority Oversampling Technique (SMOTE) and its variants utilizing the k-Nearest Neighbors (kNN) algorithm, in order to identify minority classes in the training dataset and learn their features. SMOTE can provide a dataset that is balanced concerning the desired classes ensuring that data fed to our ML/DML models will be more resistant to overfitting [8]. Nevertheless, to expose the performance and a comparative assessment of multiple oversampling techniques, we incorporate into our scheme the following models: (i) SMOTE; (ii) Borderline-SMOTE (minority examples near the borderline are oversampled) [9]; (iii) Support Vector Machine (SVM) SMOTE (new minority class instances are generated near borderlines with the use of SVM to assist establishing boundary between classes) [10]; (iv) k-Means SMOTE (minority class instances are generated in safe and critical areas of the input space) [11]; and (v) Adaptive Synthetic (ADASYN) that adopts a weighted distribution for different minority class instances based on their level of difficulty in the learning process [12].

The differences of our approach when compared with our past efforts [13] are (i) first of all, in our experiments, we use both Variational Autoencoder and Simple Autoencoder. In our previous work, our experiments have included only a Simple Autoencoder; (ii) secondly, we test several oversampling techniques such as ADASYN, Borderline-SMOTE, K-Means SMOTE, and not only SMOTE; (iii) lastly, we use a LSTM Recurrent Neural Network instead of a Convolutional Neural Network, as its performance is much better.

The novelty of our approach when compared with the relevant research efforts in the domain is presented by the following list:

- We propose a sequential model that combines (i) an autoencoder (experiments have been done using a Deep Autoencoder and a Variational Autoencoder separately) for performing dimensionality reduction and the identification of the most significant features in the collected dataset and (ii) a LSTM RNN that is responsible for the final classification process. The aim is to deal with scenarios where a high number of dimensions are present. The autoencoder efficiently learns the representation of data under consideration and generates the reduced encoding as a representation as close as possible to the original input. The LSTM is used in order to perform the final classification of the task.
- We provide a decision-making mechanism for the detection of frauds applied upon the outcome of the aforementioned autoencoder and the LSTM. The proposed LSTM adopts connectivity patterns of the involved neurons that can learn the attributes of the distribution of data and, finally, detect potential frauds. Additionally, the aforementioned connectivity patterns incorporate data overlaps to learn the connections between features.
- We provide an extensive experimental assessment to reveal the pros and cons of the proposed scheme. Our evaluation can be considered as a comparative study upon the use of multiple DML models and oversampling techniques while performing their combination.
- We present an extensive comparison among five (5) oversampling techniques, while in use with our core model
- Finally, we compare our model with a set of recently proposed schemes found in the respective literature. We adopt the same datasets and performance metrics to secure the fairness of the comparative assessment.

Our work is organized as follows: Section 2 reports on the related work, Sect. 3 discusses the proposed approach, and Sect. 4 analytically presents the proposed model. In Sect. 5, we present the envisioned experimental evaluation of the proposed approach, while in Sect. 8, we conclude our paper by exposing our future research plans.

2 Related work

Many techniques have been applied to maximize the detection rate of fraudulent events through the adoption of ML/DML techniques.

In the groundbreaking study by Sumanth et al. [14], the primary objective centers around the effective detection of credit card fraud instances. As the surge in Valentine's Day scams coincides with the increased use of debit cards for both in-person and online transactions, the urgency to fortify fraud detection mechanisms becomes paramount. To achieve this, the researchers meticulously construct an extensive credit card dataset for rigorous testing and training. Upon this robust foundation, a Deep Neural Network takes the center stage. However, what sets this approach

apart is its holistic integration of multiple techniques. Alongside the Neural Network, Support Vector Machine (SVM), Deep Neural Network (DNN), and Naive Bayes collaboratively contribute to a comprehensive system. This ensemble method, as underscored by the existing literature, yields remarkable precision in identifying credit card fraud. In parallel, the research conducted by Alarfaj et al. [15] delves into the pervasive issue of credit card fraud within the realm of online transactions. Driven by the ease and popularity of digital payments, this study aims to enhance detection accuracy while minimizing losses due to fraud. The arsenal of machine learning techniques deployed includes XGBoost, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, and Extreme Learning Method. However, the true innovation lies in the study's unwavering focus on deep learning algorithms. By meticulously exploring various Convolutional Neural Network architectures, the researchers dissect layering effects and model configurations. The results are nothing short of remarkable: accuracy, F1 score, precision, and AUC meticulously tuned at 99.9, 85.71, 93, and 98%, respectively. Notably, this suggested model outperforms existing machine learning and deep learning algorithms in credit card fraud detection, as empirically validated.

In the study conducted by Esenogho and colleagues [16], the upswing in both traditional and online purchases driven by the recognition of Valentine's Day is attributed to the expanding domain of electronic commerce and communication systems. However, this trend has also resulted in a rise in credit card fraud, leading to significant annual financial losses for banks. In response to this challenge, the research focuses on the development of precise fraud detection algorithms, a task made intricate by biases inherent in credit card datasets and the dynamic nature of user purchasing behaviors. To surmount these challenges, the authors propose an innovative approach that combines a neural network ensemble classifier with a hybrid data resampling strategy. The suggested method involves the use of adaptive boosting (AdaBoost) in conjunction with a Long Short-Term Memory (LSTM) neural network as the foundational elements for the ensemble classifier. This study constitutes a valuable contribution to credit card fraud detection, utilizing advanced techniques to address the complexities associated with evolving purchase patterns and biased datasets.

The interested reader can find a relevant survey in [17]. ML models involve Neural Networks (NNs), Decision Trees, and genetic algorithms, while outlier detection techniques can be also adopted for the identification of frauds as exposed by [17]. The adoption of the aforementioned schemes requires the modeling of the environment and the solution space as well as a training phase (it is the common scenario for the majority of models). In [18], the authors present an experimental comparison of various classification algorithms such as Random Forests and gradient boosting classifiers for unbalanced scoring datasets. The presented research depicts that Random Forests and gradient boosting algorithms outperform the remaining models involved in the comparison (e.g., C4.5, quadratic discriminant and k-Nearest Neighbors—kNNs). However, the complexity of these approaches may jeopardize the 'visibility' of the internal processes and lead to consider them as 'black boxes'. In [19], the authors conclude that Support Vector Machines (SVMs) improve the accuracy of events detection compared to Logistic Regression, linear discrimination

analysis, and kNNs. A survey on SVMs introduces the application of the technology and the techniques adopted to predict frauds using broad and narrow definitions [20]. In any case, SVMs are not suitable for the management of large datasets and do not perform well when noise is present in data (e.g., overlapping classes). Another effort presented in [21] tries to evaluate ML models (SVMs, bagging, boosting, and Random Forests) to predict bankruptcies one year prior to the event. The authors also compare the performance of the adopted algorithms with results retrieved by discriminant analysis, Logistic Regression, and NNs. The aforementioned attempt evaluates the strength of ensemble models over single classifiers focusing on the accuracy of the outcomes. However, bagging may suffer from high biases if it is not modeled properly leading to underfitting while becoming computationally expensive when large-scale data are the case. Boosting cannot, usually, be implemented in real-time due to its increased complexity and may result in multiple parameters having direct effects on the behavior of the model. In [22], the authors proposed the PrecisonRank and the total detection cost as the appropriate metrics for measuring the detection performance in credit datasets. In an additional effort presented in [23], the authors focus on an effective learning strategy for addressing the verification latency and the alert–feedback interaction problem, while they propose a formalization of the fraud detection problem that realistically describes the operating conditions of FDSs that analyze massive streams of credit card transactions everyday. A denoising autoencoder for credit risk analysis has been introduced to remove the noise from the dataset [24]. Denoising autoencoders often yield better representations when trained on corrupted versions of a dataset; thus, they can capture information and filter out noise more effectively than traditional methods [24]. A Deep Autoencoder and a Restricted Boltzmann Machine (RBM) that can reconstruct normal transactions to finally find anomalies have been applied to a credit card dataset for fraud detection [3]. The authors conclude that the combination of the autoencoder with the RBM outperforms other techniques when the training dataset is large enough to train them efficiently. Sparse autoencoders and Generative Adversarial Networks (GANs) have been also adopted to detect potential frauds [25]. The discussed models can achieve higher performance than other state-of-the-art one-class methods such as one-class Gaussian Process (GP) and Support Vector Data Description (SVDD). In general, autoencoders may be somehow limited in the processes that can perform. One potential use may be the pre-training of a model to get the dataset latent representation and isolate the most significant features. This means that for concluding a classification process, autoencoders should be combined with other schemes. In [26], the authors introduce a hybrid 'Relief–CNN' model, i.e., a combination of a CNN and the Relief algorithm. The Relief algorithm is a filter-method approach to feature selection that is notably sensitive to feature interactions. This algorithm calculates a score for each feature which can then be applied to rank and select top-scoring features for the final selection. The utilization of the Relief algorithm can efficiently reduce the size of an image pixel matrix, which can reduce the computational burden of the CNN. The authors in [27] expand the labeled data through their social relations to get the unlabeled data and propose a semisupervised attentive Graph Neural Network, named SemiGNN, to utilize the multiview labeled and unlabeled data for fraud detection. Moreover, they propose a

hierarchical attention mechanism to better correlate different neighbors and different views. Lastly, in [28], the authors propose a method to combine label propagation and Transductive Support Vector Machine (TSVM) with Dempster-Shafer theory for accurate default prediction of social lending using unlabeled data. In order to train a lot of data effectively, they ensemble semisupervised learning methods with different characteristics. Label propagation is performed so that data having similar features are assigned to the same class and TSVM makes moving-away data have different features. The authors of [29] use various ML algorithms, with and without the usage of the AdaBoost and majority voting algorithms in order to detect fraudulent transactions. A recent approach includes the implementation of a LSTM as presented by [30]. Another ensemble method has been used in [13] that involves a hybrid deep learning scheme that uses an autoencoder and a CNN model, using an oversampling technique in order to overcome the problem of the unbalanced datasets in the training process. The same authors in [31] have published a paper in which they elaborate a multistage deep learning model that targets to efficiently manage the incoming streams of transactions and detect the fraudulent ones. They propose the use of two autoencoders to perform feature selection and learn the latent data space representation based on a nonlinear optimization model. On the delivered significant features, they subsequently apply a Deep Convolutional Neural Network to detect frauds, thus combining two different processing blocks. In [32], the authors suggest to exploit the framework of rough sets for detecting outliers. They propose a novel definition of outliers—RMF (Rough Membership Function)-based outliers, by virtue of the notion of Rough Membership Function in rough set theory. In [33], the authors present the basic concepts of rough set theory, and its possible applications are briefly discussed. Further research problems conclude the paper. Lastly, the authors of [34] aim to investigate and present a thorough review of the most popular and effective anomaly detection techniques applied to detect financial fraud, with a focus on highlighting the recent advancements in the areas of semisupervised and unsupervised learning.

3 The proposed approach

3.1 High-level architecture

Trying to support a high quality solution with increased efficiency, we propose the combination of an autoencoder and a LSTM Recurrent Neural Network to conclude a system that performs fraud detection. The aforementioned models are connected in a 'sequential' manner in order to create a scheme. At first, our data become the subject of processing by the autoencoder to realize the envisioned dimensionality reduction and make our system capable of managing huge volumes of streaming data. Through the adoption of the autoencoder, we are able to reveal the statistical information of data and expose strong correlations between features. The output of the hidden layer of the autoencoder (i.e., the compressed representation of the initial dataset) is transferred to the LSTM. A second advanced data management is performed before the final classification to classes C (for normal transactions), and

$\overline{C}$ (for fraudulent transactions) takes place. Our approach performs much better that any other published method in terms of recall, precision, and F1 score. The combination of the two DML models targets to improve the classification recall which depicts whether our approach can eliminate False-Negative events (i.e., the proportion of fraudulent transactions that are not detected). As we show later in this paper, the proposed model also exhibits a good performance related to the precision which is mainly affected by False-Positive events, i.e., normal transactions that are classified to class *C*. For financial institutions, recall is more important than precision under the condition that precision is below a certain percentage of transactions to avoid a negative impact on customers/users experience. Hence, the elimination of False-Negative events becomes the motivation for the adoption of the sequential DML model that is capable of processing complex datasets as well. False Negatives can be avoided if we enhance the learning process of the hidden aspects of the adopted datasets. Compared with other efforts in the respective literature, we go a step forward and identify the 'hidden' aspects of fraud detection. We can discern the features that exhibit strong correlations with the remaining ones. For instance, we propose a more efficient dimensionality reduction technique via the Deep Autoencoder that could lead to more accurate classification results. PCA (one of the most famous dimensionality reduction methods) is not an appropriate technique to use in our case, because it is essentially a linear transformation. Autoencoders, on the other hand, are capable of modeling complex nonlinear functions, which in our case are better to use for dimensionality reduction.

Following [8], we create new instances of the minority class, while, in parallel, we reduce the instances of the majority class (i.e., $\overline{C}$) by implementing oversampling techniques (see next sub-section for more details). The reduction of the majority instances is done automatically by the SMOTE algorithm, simply by removing the extra instances, until the point that the two classes reach the 50% of the newly created dataset. In general, oversampling focuses on the enhancement of the minority class (i.e., *C*) to adjust the class distribution of our dataset, thus, to achieve better performance. The newly created dataset is adopted to train the autoencoder. For the experiments regarding the Simple Autoencoder, we adopt three (3) hidden layers, and gradually, we reach up to ten (10) nodes (features) from the initial thirty (30) features of the dataset. For the experiments that use a Variational Autoencoder, we use two (2) hidden layers, and we gradually reach up to ten (10) nodes (features) from the initial thirty (30) features of the dataset. The point of using a Variational Autoencoder is to perform the dimensionality reduction using the distribution of the original training set of the dataset. In some problems, a Variational Autoencoder seems to perform better that any other kind of autoencoder. Finally, we have to notice that the model loss is calculated with the binary cross-entropy loss function. After the end of the training process, we get the new feature-reduced dataset from the hidden layer and eliminate redundancies in the feature representation space. For instance, in a dataset with thirty (30) features, we can conclude with only ten (10), significantly reducing the data space upon which we deliver the final classification outcome. The above approach increases the speed of processing with positive impact in our model as we target to support a streaming environment

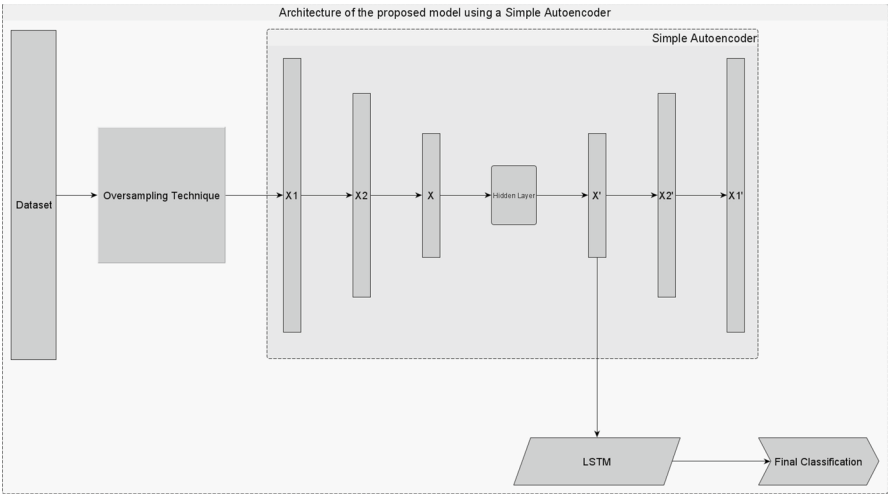

Fig. 1 The high-level architecture of the proposed approach using a Deep Autoencoder

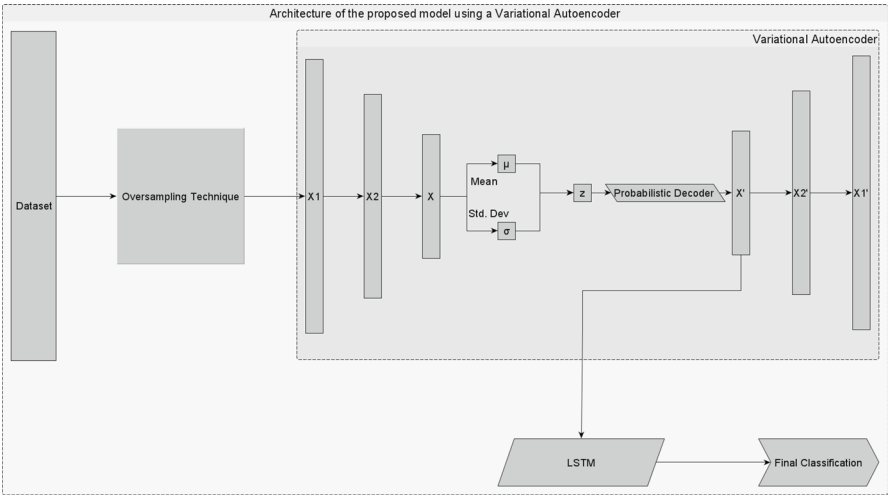

Fig. 2 The high-level architecture of the proposed approach using a Deep Variational Autoencoder

where numerous transactions are collected. After the discussed step, we create a new encoded (and reduced) feature dataset which is fed to the LSTM, which have an input layer, two (2) hidden layers, and an output layer. In the first two (2) layers, we adopt the ReLU activation function, and for the output layer, the activation process is performed by a sigmoid function. Our decision for adopting the specific activation functions is concluded through an extensive experimentation that reveals their performance for the specific problem. The model loss is calculated with the binary cross-entropy loss function (cross-entropy minimization is frequently used

in optimization and rare-event probability estimation). The LSTM is evaluated with the assistance of a test set to evaluate its performance. In the following figures, we present the architecture of the proposed models, categorized according to the use of a Simple Deep Autoencoder or a Deep Variational Autoencoder (Figs. 1 and 2).

3.2 Oversampling the minority class

An unbalanced dataset could be a common problem when applying ML/DML algorithms. The reason is that training a ML/DML algorithm with such a dataset often results in a particular bias toward the majority class. The purpose of using an oversampling method is to create new instances for the minority class in order to have more data to train your model. If a model is trained with few instances for one class, it is likely for the model not to be able to detect new data which belong to the particular class. To tackle the problem of imbalance in the training dataset, the authors of [8] have introduced SMOTE which is one of the most popular oversampling techniques. SMOTE is based on a kNNs upon the Euclidean distance between data points in the feature space. For every instance that belongs to the minority class, k of Nearest Neighbors is detected, such that they belong to the same class where C is oversampled. We take each sample in C and introduce synthetic instances along the line segments joining any/all of the k minority class Nearest Neighbors [8]. Depending on the required number of oversamples, instances from the k-Nearest Neighbors are randomly chosen. We incorporate into our decision-making multiple oversampling techniques to reveal their performance when combined with the proposed autoencoder and the LSTM. Hence, apart from the SMOTE technique, we also study the adoption of additional schemes for the management of the unbalanced datasets. Another method for oversampling is the K-Means SMOTE technique. This technique avoids the generation of noise and effectively overcomes imbalances between and within classes, by employing the K-Means clustering algorithm in combination with SMOTE oversampling [11]. In Borderline-SMOTE technique, for every minority instance, its k-Nearest Neighbors of the same class are extracted, and some of them are randomly selected based on the oversampling rate [9]. In SVM-SMOTE technique, the method first pre-processes the data by oversampling the minority instances in the feature space, and then, the pre-images of the synthetic samples are found based on a distance relation between feature and input spaces. Finally, these pre-images are appended to the original dataset [10]. The last oversampling technique that is tested in this paper is the ADASYN [12]. ADASYN is based on the idea of using a weighted distribution of the instances in C according to their level of difficulty in learning. Synthetic data are generated for C being harder to learn than to C instances that are easier to learn. ADASYN improves the learning ability with respect to data distributions and reduces the biases introduced by the class imbalance problem. The final target is to adaptively shift the classification decision boundary toward the space of the 'difficult' instances. Before the use of the oversampling technique, our dataset had 594.643 observations, with the training dataset constitutes 70% of them that are used for oversampling. The 587.443 of them are normal payments and 7.200 fraudulent transactions. After the oversampling

technique the fraudulent transactions and the normal ones, constitutes 50% of the newly created dataset, halved with 587.443 in each category. It is clarified that the dataset that has been through oversampling is used solely for training purposes and is not added to the testing set. The testing set remains original, consisting of data that have not been manipulated or oversampled.

4 Intelligent fraud detection

4.1 Feature selection for dimensionality reduction

Autoencoders are Neural Networks that are, traditionally, used for dimensionality reduction. Autoencoders are trained to be capable of reproducing their inputs to outputs [7]. They adopt a hidden layer $h(x')$ trained to depict the provided input. An autoencoder may be viewed as a model containing two parts: (i) the encoder function $f(x)$ (x is the input into the autoencoder) and (ii) a decoder scheme that produces a reconstruction of the initial input $g(h)$. The following equations hold true:

$$f(x) \rightarrow h(x') \tag{1}$$

$$h(x') \rightarrow g(h) = x \tag{2}$$

$$f(x), h(x') = \underset{f(x),h(x')}{\operatorname{argmin}} \|x - (f(x)\ o\ h(x'))x\|^2 \tag{3}$$

The encoder function, denoted by $f(x)$, maps the original data x to a latent space $h(x')$ present at the hidden layer before they are reproduced by the decoder. The decoder function, denoted by $g(h)$, maps the latent space $h(x')$ at the hidden layer to the output which is the same as the input. The discussed encoding network can be represented by a standard NN function transferred through the activation function, where l is the latent dimension, i.e., W and b are the weights and biases of the layers.

$$l = \sigma(Wx + b) \tag{4}$$

Similarly, the decoder can be depicted in the same way, however, with different weights, biases, and potentially activation functions. The decoding phase can be represented by the following equation ($W^{'}$ and $b^{'}$ are the weights and biases of the hidden layer):

$$x^{'} = \sigma^{'}(W^{'}l + b^{'}) \tag{5}$$

In the proposed autoencoder, we adopt a loss function written in terms of the aforementioned functions. The loss function is utilized to affect the training of the NN through the backpropagation algorithm, i.e., our model is continuously seeking to

limit the error between the calculated values and target outputs. This forces the hidden layer to 'perform' dimensionality reduction and eliminate noise while reconstructing inputs, especially when the number of neurons in the hidden layer is low. The following equation holds true:

$$L(x, x^{'}) = \|x - x^{'}\|^2 = \|x - \sigma^{'}(W^{'}(\sigma(Wx + b)) + b^{'})\|^2 \tag{6}$$

We implement our autoencoder adopting three (3) hidden layers and the Exponential Linear Unit (ELU) as the activation function, and gradually, we reach up to ten (10) nodes (features) from the initial thirty (30) features of the dataset. We consider ELU as it performs better than other activation functions like the Rectified Linear Unit (ReLU), the sigmoid, or the hyperbolic tangent function (tanh). For concluding the activation function, we rely on a high number of experiments to reveal their performance. The activation function ELU tends to converge the cost to zero more quickly (it can derive negative values allowing the network to push the mean activation closer to zero) than other functions while being capable of producing more accurate results. In general, the ELU activation function decreases the gap between the normal gradient and the unit natural gradient and, thereby, speeds up the learning process [35]. The dataset that is used to train the autoencoder constitutes the 70% of the entire dataset, without using a batch size, while the model is trained in 100 epochs.

4.2 A Variational Autoencoder for feature selection

Unlike typical autoencoders, Variational Autoencoders (VAEs) are generative models that exhibit different mathematical formulations if compared with autoencoders. VAEs focus on probabilistic graphical models with posterior probabilities being approximated by a NN, thus formulating the architecture of an autoencoder [7]. VAEs try to emulate how data are generated to reveal the underlying causal relations. This approach differs with discriminating models that aim to learn a predictor-given specific observations. VAEs rely on strong assumptions for the distribution of latent features using a variational approach. This approach results in additional loss components and a specific estimator for training purposes, i.e., the Stochastic Gradient Variational Bayes (SGVB) estimator. The assumption is that data are generated by a directed graphical model, i.e., $p_\theta(\mathbf{x}|\mathbf{h})$, and that the encoder learns the following approximation $q_\phi(\mathbf{h}|\mathbf{x})$ to the posterior distribution $p_\theta(\mathbf{h}|\mathbf{x})$ where $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ denote the parameters of the encoder and decoder, respectively. The probability distribution of the latent vector of a VAE typically matches that of the training data much closer than a standard autoencoder. The loss function of VAE has the following form:

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{x}) = D_{\mathrm{KL}}(q_\phi(\mathbf{h}|\mathbf{x})\|p_\theta(\mathbf{h})) - \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x})}\big(\log p_\theta(\mathbf{x}|\mathbf{h})\big) \tag{7}$$

In the above equation, D_{KL} depicts the Kullback–Leibler divergence. The prior over the latent features is usually set to be the centered isotropic multivariate Gaussian, i.e.,

$$p_\theta(\mathbf{h}) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8}$$

Commonly, the shape of the variational and the likelihood distributions are chosen such that they are factorized Gaussian distributions:

$$q_\phi(\mathbf{h}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\omega}^2(\mathbf{x})\mathbf{I}) \tag{9}$$

$$p_\theta(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{h}), \boldsymbol{\sigma}^2(\mathbf{h})\mathbf{I}) \tag{10}$$

where $\boldsymbol{\rho}(\mathbf{x})$ and $\boldsymbol{\omega}^2(\mathbf{x})$ are the encoder outputs, while $\boldsymbol{\mu}(\mathbf{h})$ and $\boldsymbol{\sigma}^2(\mathbf{h})$ are the decoder outputs. These formulations are justified by the rationale of simplifying the final outcomes in the evaluation process of both the Kullback–Leibler divergence and the likelihood term in the variational objective defined above. We implement the proposed VAE adopting two (2) hidden layers, and we gradually reach up to ten (10) nodes (features) from the initial thirty (30) features of the dataset. As the activation function, we adopt ELU in every layer relying on a set of simulations to reveal the performance of multiple activation functions and choose the best one. Finally, we have to notice that the model loss is calculated with the binary cross-entropy loss function. The dataset that is used to train the autoencoder constitutes the 70% of the entire dataset, without using a batch size, while the model is trained in 100 epochs.

4.3 The proposed LSTM for detecting fraudulent events

As of this writing, the most effective sequence models used in practical applications are called gated RNNs. These include the LSTM network and networks based on the gated recurrent unit [7]. The idea of introducing self-loops to produce paths where the gradient can flow for long durations is a core contribution of the initial LSTM model [36]. A crucial addition has been to make the weight on this self-loop conditioned on the context, rather than fixed [37]. By making the weight of this self-loop gated (controlled by another hidden unit), the time scale of integration can be changed dynamically. In this case, we mean that even for an LSTM with fixed parameters, the time scale of integration can change based on the input sequence, because the time constants are outputs of the model itself. Instead of a unit that simply applies an element-wise nonlinearity to the transformation of inputs and recurrent units, LSTM recurrent networks have 'LSTM cells' that have an internal recurrence (a self-loop), in addition to the outer recurrence of the RNN. Each cell has the same inputs and outputs as an ordinary recurrent network but also has more parameters and a system of gating units that controls the flow of information. The most important component is the state unit $s_i^{(t)}$ which has a linear self-loop. Here, the self-loop weight (or the associated time constant) is controlled by a forget gate unit

$f_i^{(t)}$ (for time step t and cell i), which sets this weight to a value in the unity interval through the adoption of a sigmoid unit:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \tag{11}$$

where $x^{(t)}$ is the current input vector, $h^{(t)}$ is the current hidden layer vector containing the outputs of all the LSTM cells, and $\mathbf{b}^f$, $\mathbf{U}^f$, and $\mathbf{W}^f$ are, respectively, biases, input weights, and recurrent weights for the forget gates. The LSTM cell's internal state is updated as follows, but with a conditional self-loop weight f_i^t:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^t \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \tag{12}$$

where **b**, **U,** and **W** denote the biases, input weights, and recurrent weights into the LSTM cell, respectively. The external input gate unit $g_i^{(t)}$ is computed similarly to the forget gate (with a sigmoid unit to obtain a gating value in the unity interval), but with its own parameters:

$$g_i^t = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \tag{13}$$

The output $h_i^{(t)}$ of the LSTM cell can also be eliminated through $q_i^{(t)}$ which also uses a sigmoid unit for gating:

$$h_i^{(t)} = tanh\left(s_i^{(t)}\right) q_i^{(t)}, \tag{14}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \tag{15}$$

which has parameters b^o, U^o, and W^o for its biases, input weights, and recurrent weights, respectively. Among the variants, one can choose to use $s_i^{(t)}$ as an extra input (with its weight) into the three gates of the i-th unit.

In our sequential model, we use a LSTM, which has an input layer, two (2) hidden layers, and an output layer. In the first two (2) layers, we adopt the ReLU activation function, and for the output layer, the activation process is performed by a sigmoid function. Our decision for adopting the specific activation functions is concluded through an extensive experimentation that reveals their performance for the specific problem. The model loss is calculated with the binary cross-entropy loss function (cross-entropy minimization is frequently used in optimization and rare-event probability estimation). The dataset that is used to train the autoencoder constitutes the 70% of the entire dataset, without using a batch size, while the model is trained in 100 epochs.

5 Experimental evaluation

5.1 Experimental setup and performance metrics

We report on the evaluation of the proposed model upon a real dataset. This dataset has been generated using BankSim, a bank simulator for a Spanish bank. BankSim is an agent-based simulator of bank payments based on a sample of aggregated transactional data provided by a bank in Spain. The main purpose of BankSim is the generation of synthetic data that can be used for fraud detection research. Statistical and a Social Network Analysis (SNA) of relations between merchants and customers were used to develop and calibrate the model. Our ultimate goal is for BankSim to be usable to model relevant scenarios that combine normal payments and injected known fraud signatures. The datasets generated by BankSim contain no personal information or disclosure of legal and private customer transactions. BankSim was run for 180 steps (approximately six months), several times, and calibrated the parameters in order to obtain a distribution that get close enough to be reliable for testing. There were collected several log files and selected the most accurate. There were simulated thieves that aim to steal an average of three cards per step and perform about two fraudulent transactions per day producing 594.643 records in total where 587.443 are normal payments and 7.200 fraudulent transactions. Since this is a randomized simulation, the values are, of course, not identical to original data. Therefore, it can be shared by academia, and others, to develop and reason about fraud detection methods. Synthetic data have the added benefit of being easier to acquire, faster, and at less cost, for experimentation even for those that have access to their own data. We argue that BankSim generates data that usefully approximate the relevant aspects of the real data [38]. In our experiments, we used Python in a Jupyter Notebook, in a laptop that has a processor of 2.40 GHz and a RAM memory of 12.0 GB. The detailed model described in this paper is readily available for exploration and implementation through its corresponding GitHub repository. Researchers, developers, and enthusiasts alike can access the complete source code, documentation, and related resources by visiting the provided link https://github.com/ziovi ris/Credit-card-fraud-detection-using-a-deep-learning-multistage-model.

Six performance metrics are adopted to evaluate our model, i.e., precision (ϵ), recall (ζ), the F1 score (δ), the Area Under Curve or simply (AUC), the Mean Squared Error or MSE (κ), and the Mean Absolute Error or MAE (λ). ϵ is the fraction of true events (i.e., frauds) among all samples which are classified as frauds, while ζ is the fraction of frauds which have been classified correctly over the total amount of frauds. δ is a performance metric that combines both, ϵ and ζ.

AUC provides an aggregate measure of performance across all possible classification thresholds. The Area Under the Curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). It is also common to calculate the Area Under the ROC convex as any point on the line segment between two prediction results

Table 1 Group of experiments implementing a Simple Autoencoder

Models	ζ (%)	ϵ (%)	δ (%)	κ	λ	AUC (%)
SMOTE - AE - LSTM	99.83	98.85	**99.39**	0.0029	**0.0054**	99.54
Borderline-SMOTE - AE - LSTM	99.86	98.17	99.01	0.0031	0.0062	99.58
SVM-SMOTE - AE - LSTM	**99.93**	98.59	99.26	0.0033	0.0062	99.60
ADASYN - AE - LSTM	99.72	98.16	98.93	0.0033	0.0065	98.94
K-Means SMOTE - AE - LSTM	99.89	**98.94**	99.14	**0.0028**	0.0060	**99.61**

Table 2 Group of experiments implementing a Variational Autoencoder

Models	ζ (%)	ϵ (%)	δ (%)	κ	λ	AUC (%)
SMOTE - VAE - LSTM	99.69	98.75	99.22	0.0027	0.0058	99.22
Borderline-SMOTE - VAE - LSTM	99.71	97.64	98.66	**0.0026**	**0.0055**	98.68
SVM-SMOTE - VAE - LSTM	**99.76**	98.55	99.15	**0.0026**	0.0057	99.16
ADASYN - VAE - LSTM	98.70	98.35	98.53	0.0032	0.0061	98.53
K-Means SMOTE - VAE - LSTM	99.47	**99.00**	**99.23**	0.0033	0.0059	**99.24**

that can be achieved by randomly using one or the other system with probabilities proportional to the relative length of the opposite component of the segment. The following equations hold true:

$$\epsilon = \frac{TP}{TP + FP} \tag{16}$$

$$\zeta = \frac{TP}{TP + FN} \tag{17}$$

$$\delta = 2 \cdot \frac{\epsilon \cdot \zeta}{\epsilon + \zeta} \tag{18}$$

$$\kappa = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - y_n^{'}\right)^2 \tag{19}$$

$$\lambda = \frac{1}{N} \sum_{t=1}^{N} |(y_t - y_t^{'}| \tag{20}$$

In the above equations, TP (True Positive) is the number of frauds which have been classified correctly. FP (False Positive) is the number of normal transactions which have been classified as frauds. FN (False Negatives) is the number of frauds which

Table 3 Group of experiments implementing conventional machine learning algorithms

Models	ζ (%)	ϵ (%)	δ (%)	κ	λ	AUC (%)
SVM	95.56	78.51	88.45	0.0065	0.0087	86.73
XGBoost	91.39	95.78	94.31	0.0055	0.0083	94.94
Random Forest	**97.44**	**96.34**	**96.33**	**0.0037**	**0.0065**	**95.32**

have been classified as normal ones. TN (True Negatives) is the number of normal transactions that have been classified as normal.

5.2 Performance assessment

Our set of experiments involve the implementation of a stratified validation technique with five folds to ensure that the separations of the adopted dataset are relatively unbiased. Our results are shown in Tables 1, 2, and 3, respectively.

In any case, we can clearly state that our proposed models perform better than the traditional machine learning algorithms that we tested.

In the following figures, we present our experimental evaluation outcomes of the main aforementioned models, i.e., SMOTE - AE - LSTM, SMOTE - VAE - LSTM.

In Fig. 3, we observe the performance of the SMOTE - AE - LSTM, while in Fig. 4, we observe the performance of the SMOTE - VAE - CNN model.

6 Results

Our investigation into the performance of various sequential models that include oversampling technique, autoencoder architectures, specifically the Simple Autoencoder and the Variational Autoencoder (VAE), and an LSTM model has yielded valuable insights. We evaluated both autoencoder variants using various metrics. The Simple Autoencoder consistently outperformed the VAE in all three metrics. Its deterministic encoding and straightforward reconstruction contribute to this

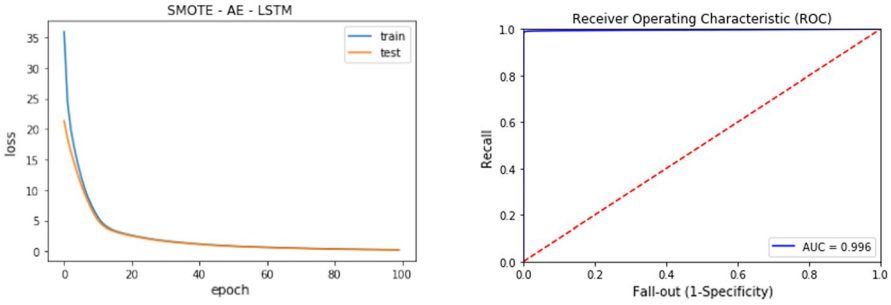

Fig. 3 The performance of SMOTE - AE - LSTM (left: SMOTE - Autoencoder - LSTM performance - right: ROC curve)

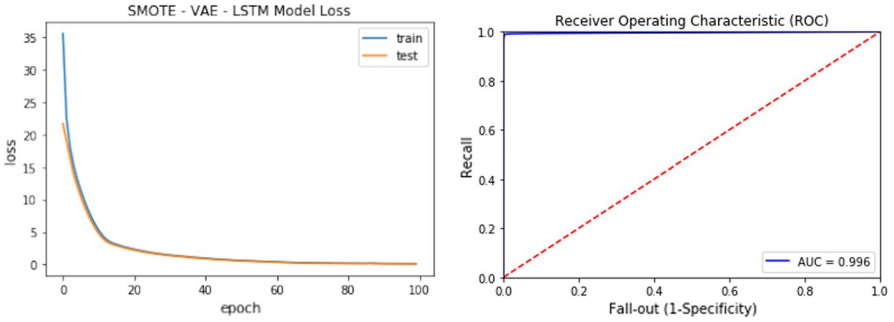

Fig. 4 The performance of SMOTE - VAE - CNN (left: SMOTE - Autoencoder - LSTM Performance - right: ROC curve)

advantage. By compressing input data into a lower-dimensional latent space and then faithfully reconstructing the original data, the Simple Autoencoder provides reliable representations for downstream tasks. The VAE introduces probabilistic modeling, assuming that the latent space follows a specific probability distribution (often Gaussian). While the VAE lags behind the Simple Autoencoder in precision and recall, its probabilistic nature provides valuable uncertainty estimates. These estimates can be particularly useful in scenarios where uncertainty quantification matters. Surprisingly, both autoencoder variants exhibited exceptional performance. This finding underscores the significance of incorporating any autoencoder architecture to enhance the predictive abilities of the LSTM. While the Simple Autoencoder excels in performance, the VAE's probabilistic nature introduces trade-offs. Practitioners must weigh the deterministic reliability of the Simple Autoencoder against the uncertainty-aware capabilities of the VAE. In summary, autoencoders play a pivotal role as pre-processing steps for sequence models like LSTMs, whether one opts for simplicity or embraces probabilistic modeling, integrating an autoencoder can elevate model performance. As researchers, let us continue exploring autoencoder variants tailored to specific problem domains.

7 Comparative assessment

In this section, we compare our models with the results that the authors of [30] present in their work, considering that the same dataset is used, so the comparison is easier to implement. The authors use a LSTM Recurrent Neural Network, in order to perform the classification of the transactions of the dataset as fraudulent or not. In this study, the authors set the LSTM memory cell to 15, with 100 epochs, while the loss function that is used is the cross-entropy, and the optimizer that the authors use is the Adam optimizer. They used one hidden layer with 15 neurons.

The only metrics that are available in the particular study are the Area Under Curve (AUC), the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE).

Table 4 Comparison between our models' results and the results from [30], i.e., LSTM

Models	AUC (%)	MSE	RMSE
SMOTE - AE - LSTM	99.54	0.0029	0.0054
Borderline-SMOTE - AE - LSTM	99.58	0.0031	0.0062
SVM-SMOTE - AE - LSTM	99.60	0.0033	0.0062
ADASYN - AE - LSTM	98.94	0.0033	0.0065
K-Means SMOTE - AE - LSTM	99.61	0.0028	0.0060
SMOTE - VAE - LSTM	99.22	0.0027	0.0058
Borderline-SMOTE - VAE - LSTM	98.68	0.0026	0.0055
SVM-SMOTE - VAE - LSTM	99.16	0.0026	0.0057
ADASYN - VAE - LSTM	98.53	0.0032	0.0061
K-Means SMOTE - VAE - LSTM	99.24	0.0033	0.0059
LSTM ([39])	99.56	0.0034	0.0063

7.1 Comparison between our models' results and the results from [30]

From this comparison, we can see that the majority of our models perform better than the model of the authors of [30]. The best of our models in terms of AUC's score performs a score of 99.64% with a MSE of 0.0028 and a RMSE of 0.0060. Once again, we state that this paper did not provide sufficient information about precision, recall, F1 score of any other metric besides AUC's score, MSE, and RMSE (Table 4).

From this comparison, we can see that the majority of our models with the Deep Autoencoder perform better than the models of the authors of [30]. The best of our models performs a score of 99.61% in terms of Area Under Curve (AUC), a score of 0.0028 in terms of MSE, and a score of 0.0060 in terms of RMSE, while the best performance in [30] has a score of 99.56% in AUC, a score of 0.0034 in terms of MSE, and a score of 0.0063 in terms of RMSE. Once again, we state that the authors of [30] did not provide sufficient information about precision, recall, F1 score, or any other metric.

8 Conclusions

In this study, we propose the combination of multiple deep learning technologies like autoencoders (AE and VAE) and LSTM Recurrent Neural Networks to predict fraud cases in financial interactions. The discussed autoencoder is adopted for dimensionality reduction, while the LSTM is utilized to perform the final classification of the type of each transaction (fraudulent or not). We also meet the challenges coming from highly unbalanced datasets when the training process of deep learning models is the case. We adopt various oversampling techniques to deal with the limited number of the positive class. In addition, the results of our models are compared with the results of algorithms that use the same dataset and are recently published.

The result of this comparison is that the majority of our models perform better than the proposed ones in that paper. This aspect gains our attention and becomes one of our targets for future research activities, i.e., the incorporation into our model of the temporal axis and the study of the seasonality detected in fraudulent events. While our comparative study sheds light on the performance of autoencoder architectures within the context of LSTM models, several avenues remain unexplored. Here are potential directions for future research:

Hybrid architectures Investigate hybrid approaches that combine the strengths of both the Simple Autoencoder and the VAE. Can we design an architecture that leverages deterministic encoding while incorporating probabilistic uncertainty estimates? Such hybrid models could strike a balance between reliability and adaptability.
Dynamic latent spaces Explore adaptive latent spaces that adjust dynamically based on input data characteristics. Can we design autoencoders that learn to adapt their latent representations during training? Dynamic latent spaces could enhance model robustness across varying contexts.
Task-specific autoencoders Tailor autoencoder architectures to specific problem domains. For instance, consider specialized autoencoders for natural language processing, image analysis, or time-series data. Customized architectures may yield better feature representations.
Regularization techniques Investigate novel regularization methods for autoencoders. Regularization can prevent overfitting and improve generalization. Techniques such as dropout, weight decay, or adversarial training could be adapted to autoencoder training.
Interpretable latent representations Develop techniques to interpret latent representations. Can we visualize what specific features or patterns each dimension of the latent space captures? Interpretable representations enhance model transparency.
Transfer learning with autoencoders Explore transfer learning scenarios where pre-trained autoencoders serve as feature extractors for downstream tasks. Fine-tuning the encoder on task-specific data could accelerate convergence.
Ensemble approaches Combine multiple autoencoders to form an ensemble. Ensemble methods often improve robustness and generalization. Investigate how ensemble autoencoders impact LSTM performance.
Hyperparameter optimization Systematically tune hyperparameters for autoencoder architectures. Grid search, Bayesian optimization, or evolutionary algorithms can help identify optimal settings.
Autoencoder regularization Extend the study to include other autoencoder variants, such as denoising autoencoders, contractive autoencoders, or sparse autoencoders. Each variant introduces unique regularization mechanisms.
Benchmarking on diverse datasets Our experiments focused on specific datasets. Future work should explore diverse data sources, including real-world applications. Robustness across different domains is essential.

In summary, the field of autoencoders continues to evolve, and there is ample room for innovation. By addressing these research directions, we can enhance the synergy between autoencoders and sequence models, ultimately advancing the state-of-the art in deep learning.

Author contributions All authors contributed to the study conception and design. All authors read and approved the final manuscript.

Funding Open access funding provided by HEAL-Link Greece.

Data availability The datasets generated analyzed during the current study are available in the [https://www.kaggle.com/kartik2112/fraud-detection-banksim/data repository.

Declaration

Conflict of interest The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Ethical approval Authors confirm that the appropriate ethics review has been followed.

Informed consent Not applicable.

References

1. https://interceptd.com/how-is-machine-learning-used-in-fraud-detection
2. https://www.netguru.com/blog/fraud-detection-with-machine-learning-banking
3. Pumsirirat A, Yan L (2018) Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. Int J Adv Comput Sci Appl 9(1):18–25. https://doi.org/10.14569/IJACSA.2018.090103
4. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11(110):3371–3408 (http://jmlr.org/papers/v11/vincent10a.html)
5. Valueva MV, Nagornov NN, Lyakhov PA, Valuev GV, Chervyakov NI (2020) Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Math Comput Simul 177:232–243. https://doi.org/10.1016/j.matcom.2020.04.031
6. Dupond S (2019) A thorough review on the current advance of neural network structures. Annu Rev Control 14:200–230
7. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
8. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953

9. Han H, Wang W-Y, Mao B-H (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Adv Intell Comput 3644:878–887. https://doi.org/10.1007/1153805991
10. Zeng ZQ, Gao J (2009) Improving SVM classification with imbalance data set. In: Leung CS, Lee M, Chan JH (Eds) Neural information processing. ICONIP 2009. Lecture Notes in Computer Science. Vol 5863. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10677-444
11. Last F, Douzas G, Bação F (2017) Oversampling for imbalanced learning based on K-means and SMOTE
12. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the International Joint Conference on Neural Networks. pp 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969
13. Zioviris G, Kolomvatsos K, Stamoulis G (2021) On the use of a sequential deep learning scheme for financial fraud detection. In: Arai K (ed) Intelligent computing. Lecture notes in networks and systems. Springer, Cham
14. Sumanth CH, Kalyan PP, Ravi B, Balasubramani S (2022) Analysis of credit card fraud detection using machine learning techniques. In: 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 1140–1144. https://doi.org/10.1109/ICCES54183.2022.9835751
15. 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. https://doi.org/10.1109/ACCESS.2022.3166891
16. Ebiaredoh-Mienye SA, Swart TG, Esenogho E, Mienye ID (2022) A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease. Bioengineering. https://doi.org/10.3390/bioengineering9080350
17. Prasad NR, Almanza-Garcia S, Thomas TL (2009) Anomaly detection. Comput Mater Contin 14(1):1–22. https://doi.org/10.1145/1541880.1541882
18. Brown I, Mues C (2012) An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst Appl 39(3):3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033
19. Tony B, Crook J (2009) Support vector machines for credit scoring and discovery of significant features. Expert Syst Appl 36(2 PART 2):3302–3308. https://doi.org/10.1016/j.eswa.2008.01.005
20. Harris T (2013) Quantitative credit risk assessment using support vector machines: broad versus Narrow default definitions. Expert Syst Appl 40(11):4404–4413. https://doi.org/10.1016/j.eswa.2013.01.044
21. Barboza F, Kimura H, Altman E (2017) Machine learning models and bankruptcy prediction. Expert Syst Appl 83:405–417. https://doi.org/10.1016/j.eswa.2017.04.006
22. Dal Pozzolo A, Caelen O, Borgne YAL, Waterschoot S, Bontempi G (2014) Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 41(10):4915–4928. https://doi.org/10.1016/j.eswa.2014.02.026
23. Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G (2018) Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Trans Neural Netw Learn Syst 29(8):3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643
24. Fan Q, Yang J (2018) A denoising autoencoder approach for credit risk analysis. In: Proceedings of the 2018 International Conference on Computing and Artificial Intelligence. https://doi.org/10.1145/3194452.3194456
25. Chen J, Shen Y, Ali R (2019) Credit card fraud detection using sparse autoencoder and generative adversarial network. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, (May):1054–1059. https://doi.org/10.1109/IEMCON.2018.8614815
26. Zhu B, Yang W, Wang H, Yuan Y (2018) A hybrid deep learning model for consumer credit scoring. In: 2018 International Conference on Artificial Intelligence and Big Data, ICAIBD, (May):205–208, 2018. https://doi.org/10.1109/ICAIBD.2018.8396195
27. Wang D et al (2019) A semi-supervised graph attentive network for financial fraud detection. In: 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China. pp. 598–607, https://doi.org/10.1109/ICDM.2019.00070
28. Kim A, Cho S-B (2019) An ensemble semi-supervised learning method for predicting defaults in social lending. Eng Appl Artif Intell 81:193–199. https://doi.org/10.1016/j.engappai.2019.02.014

29. Randhawa K, Loo CK, Seera M, Lim CP, Nandi AK (2018) Credit card fraud detection using AdaBoost and majority voting. IEEE Access 6:14277–14284. https://doi.org/10.1109/ACCESS.2018.2806420
30. Benchaji I, Douzi S, El Ouahidi B (2021) Credit card fraud detection model based on LSTM recurrent neural networks. J Adv Inf Technol 12(2):113–118. https://doi.org/10.12720/jait.12.2.113-118
31. Zioviris G, Kolomvatsos K, Stamoulis G (2022) Credit card fraud detection using a deep learning multistage model. J Supercomput 78:14571–14596. https://doi.org/10.1007/s11227-022-04465-9
32. Jiang F et al (2008) A rough set approach to outlier detection. Int J Gen Syst 37:519–536
33. Pawlak Z (1997) Rough set approach to knowledge-based decision support. Eur J Oper Res 99:48
34. Hilal W, Gadsden S, Yawney J (2021) Financial fraud: a review of anomaly detection techniques and recent advances. Expert Syst Appl 193:116429. https://doi.org/10.1016/j.eswa.2021.116429
35. Clevert DA, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (ELUs). In: 4th International Conference on Learning Representations, ICLR 2016—Conference Track Proceedings. pp 1–14
36. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
37. Gers FA, Schmidhuber J (2000) Recurrent nets that time and count. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium. vol.3. pp. 189–194. https://doi.org/10.1109/IJCNN.2000.861302
38. Lopez-Rojas EA, Axelsson S (2014) BankSim: a bank payments simulator for fraud detection research
39. Loterman G, Brown I, Martens D, Mues C, Baesens B (2012) Benchmarking regression algorithms for loss given default modeling. Int J Forec 28(1):161–170. https://doi.org/10.1016/j.ijforecast.2011.01.006

Authors and Affiliations

Georgios Zioviris[1] · Kostas Kolomvatsos[2] · George Stamoulis[1]

✉ Georgios Zioviris
gzioviris@uth.gr

Kostas Kolomvatsos
kostasks@uth.gr

George Stamoulis
georges@uth.gr

[1] Department of Electrical and Computer Engineering, University of Thessaly, Glavani 37, 38221 Volos, Thessaly, Greece

[2] Department of Informatics and Telecommunications, University of Thessaly, Papasiopoulou 2-4, 35131 Lamia, Thessaly, Greece