IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 29, 2020, accepted September 20, 2020, date of publication September 23, 2020, date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026222

Evaluation of Deep Neural Networks for Reduction of Credit Card Fraud Alerts

RAFAEL SAN MIGUEL CARRASCO[1] **AND MIGUEL-ÁNGEL SICILIA-URBÁN**[2]
[1]Minsait (Indra), 28108 Madrid, Spain
[2]Department of Computer Science, University of Alcalá, 28801 Alcalá de Henares, Spain
Corresponding author: Miguel-Ángel Sicilia-Urbán (msicilia@uah.es)

ABSTRACT Fraud detection systems support advanced detection techniques based on complex rules, statistical modelling and machine learning. However, alerts triggered by these systems still require expert judgement to either confirm a fraud case or discard a false positive. Reducing the number of false positives that fraud analysts investigate, by automating their detection with computer-assisted techniques, can lead to significant cost efficiencies. Alert reduction has been achieved with different techniques in related fields like intrusion detection. Furthermore, deep learning has been used to accomplish this task in other fields. In our paper, a set of deep neural networks have been tested to measure their ability to detect false positives, by processing alerts triggered by a fraud detection system. The performance achieved by each neural network setting is presented and discussed. The optimal setting allowed to capture 91.79% of total fraud cases with 35.16% less alerts. Obtained alert reduction rate would entail a significant reduction in cost of human labor, because alerts classified as false positives by the neural network wouldn't require human inspection.

INDEX TERMS Neural networks, deep learning, fraud detection, alert reduction.

I. INTRODUCTION

As new payment methods become widely available and credit card customer base grows, the volume of incoming transactions to be processed by an FDS increases, and so does the number of alerts to be reviewed by fraud analysts. Hiring more fraud analysts can alleviate the problem in the short term, but it's neither desirable nor scalable. Instead, leveraging computer-assisted techniques to automatically discard false positives is a preferred approach, that let analysts focus on the most advanced fraud cases, or those for which no recognizable pattern exists yet.

Several techniques have been designed to deliver decision support and false positive minimization in the intrusion detection field, which is very similar in nature to fraud detection. These techniques include adaptive learning [1], similarity with verified alerts [2], greedy aggregation algorithm [3], neuro-fuzzy approach [4], alert enrichment framework [5], and outlier detection [6].

On the other hand, deep learning has captured significant attention in the research community as promising candidates to achieve this type of optimization. Deep learning has been used in decision support systems for fields like intrusion detection [7], loan application processing [8], managerial decision making [9], medical diagnosis and treatment prescription [10], and clinical imaging classification [11]. Another research suggests that decision making can be modeled for a wide range of fields by combining rule-based expert systems and neural networks [12].

In our research, several deep neural network architectures that have proven to be effective in other domains [13] have been used to process alerts of suspicious credit card transactions. Our aim was to assess if deep neural networks can accurately discriminate well-known false positives, thus reducing the number of alerts that analyst have to investigate, and, if so, to what extent and at which error rate. In our optimal setting, triggered alerts were reduced by 35.16%, with 91.79% of the fraud cases in the remaining alerts (8.21% misclassification rate).

The rest of this paper is structured as follows. In Section 2, previous research is documented. In Section 3, our research methodology is described. In Section 4, our experimental setting and obtained results are detailed, discussing what neural network architectures are more effective for alert reduction. Lastly, in Section 5, conclusions and outlook are included.

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Cusano.

II. BACKGROUND

A. ALERT REDUCTION

Alert reduction has been subject to extensive research in the field of intrusion detection.

Previous research [1] designed and built a working prototype for automated real-time alert classification. ALAC (Adaptive Learner for Alert Classification) implemented adaptive learning through a classifier that captured decision patterns from analyst's feedback. The classifier of choice was RIPPER [14], a rule learner whose main benefits are generalization accuracy and concise conditions [15]. The obtained results showed that false positives were reduced by approximately 30% when tested against DARPA 1999 dataset [16].

Clustering [2] has also been leveraged for alert reduction. Alerts exhibiting short distances to previously verified alerts were tagged as true positives. Cluster definition was based on alert relevance, criticality, and frequency, as well as other features extracted from supporting evidence and vulnerability assessment data. Tested against DARPA 1999 dataset, it achieved an average alert reduction rate of 78%.

Moreover, another clustering technique [3] based on a greedy aggregation algorithm achieved an average reduction rate of 83.2% when tested against 30 days of Snort IDS alerts. In this approach, alerts were grouped into meta-alerts, that contained common information. No feature engineering or external sources of information for enrichment purposes were required.

Jrip (Weka's implementation of RIPPER rule learner) and NEFCLASS [17] classifiers were also used to perform alert reduction against Snort IDS output from DARPA 1999 dataset. Results showed a detection rate of 88% with Jrip, and 84.63% with NEFCLASS.

On the other hand, an alert quality framework [5] implementing data enrichment achieved a reduction rate of 35.04%. The scoring criteria measured alert correctness, accuracy, reliability and sensitivity, as well as vulnerability information for operating system, network ports and applications.

Lastly, an improved variant of frequent pattern-based outlier detection [18] has also been effective in reducing the number of alerts to be investigated by security analysts [6]. In the proposed approach, reduction rate ranged from 86% to 92% when tested against a dataset with lab network traffic generated by 10 hosts.

B. DEEP LEARNNG FOR DECISION-MAKING AUTOMATION

Deep learning is a popular technique for capturing and automating decision-making processes.

Researchers designed and built [7] a deep neural network classifier that automated initial triage of security alerts generated by an IDS. The proposed system was able to classify these alerts into meaningful categories, applying the same heuristics used by human analysts.

Neural networks have also been applied to automating business-related judgement, particularly loan application processing [8]. Results confirmed that a neural network trained with data from defaulted and non-defaulted companies was able to successfully replicate human decision criteria to grant or deny a loan.

When applied to medical diagnosis [10], deep models showed a higher performance than rule-based systems and shallow neural networks. Researchers concluded that deep learning is more promising to simulate the thinking procedure of human experts than other traditional approaches.

Lastly, results obtained in clinical imaging classification [11] suggest that deep learning techniques might even outperform humans. A deep Convolutional Neural Network (dCNN) was trained with an image set from the Breast Imaging Reporting and Data System (BI-RADS). In order to test and compare the resulting accuracy, an internal dataset with 101 images and an external dataset with 43 images were used. For the internal dataset, classification accuracy ranged from 87.1% to 93.1% for the dCNN and from 79.2% to 91.6% for human readers. For the external dataset, the AUC was 96.7% for the dCNN and 90.9% for the humans.

C. FRAUD DETECTION SYSTEMS

Although the focus of this paper is not on FDS, a brief background on this topic is provided to understand what techniques have proven to be more successful in delivering the first layer of defense against fraud, before human analysts' decisions comes into play.

Recent research [19] has benchmarked multiple approaches leveraging new or enhanced methods which improve state of the art in several aspects. Researchers concluded that unsupervised learning is the most promising approach in terms of classification accuracy.

Another recent study [20] concludes that neural networks are the most effective method for credit card fraud detection, by comparing their performance to other traditional machine learning techniques, including: association rules, Support Vector Machines (SVM), Artificial Immune System, Dempster Shafer Theory, cost-sensitive decision tree, aggregation, logistic regression, K-nearest Neighbor (KNN), active learning, invariant diversity and Cardwatch [21].

1) GENERAL ASPECTS OF FDS

In credit card fraud detection, frequent input features include [22]: credit card details, transaction amount, geographical data, time, and customer personal details. Common metrics for performance evaluation are accuracy, precision, true positive rate or recall, and false positive rate [23].

Comparing metrics across systems remains impractical nowadays, because there is no reference dataset to be used for benchmarking. However, there are ongoing proposals [24] based on federated learning, geared towards securely exchanging transactions data among banks, that could potentially help on this issue.

Key performance drivers in FDS include feature selection and engineering [25], [26], choice of supervised or unsupervised models [27], ability to leverage ensembles of

models [28], [29] or hybrid models [30], [31], and incorporated domain expertise [20]. Moreover, novel feature engineering frameworks [32] have proven to yield consistent performance gains across a wide range of deep learning and machine learning algorithms.

2) SUPERVISED TECHNIQUES

The underlying assumption is that fraudulent transactions follow stable patterns that can be learnt from historical records of labelled transactions [30]. Among several common classifiers (Random Forest, K-NN, Decision Tree, Logistic Regression), logistic regression has proven to produce the more accurate results according to recent research [22].

When factoring in skewness, logistic regression, C5.0 classifier, Support Vector Machines and Artificial Neural Networks have proven [33] to perform better on imbalanced data than other techniques (Naïve Bayes, Bayesian Belief Network, Artificial Immune Systems, K Nearest Neighbors). Similar research [34] concluded that neural networks can better deal with imbalanced data than Support Vector Machines, Random Forest and decision trees.

Supervised learning (Extreme Gradient Boosting, Random Forest) delivered better performance [27] than best performing unsupervised techniques (Restricted Boltzmann Machine, Generative Adversarial Networks) when measuring Area Under the Receiver Operating Curves (AUROC), with values ranging from 0.988 and 0.989. Feature selection can also influence performance: filter and wrapper methods [25] led to higher accuracy with J48 decision tree and PART, and higher precision and sensitivity for J48, AdaBoost and random forest.

Recent research [35] has proposed a non-parametric novel approach with subset of relevant transactions created through data reduction. Another novel technique [36] based on Gradient Boosting Decision Tree (GBDT) and DeepWalk, achieved promising results, with F1 score ranging from 61.43% and 71.84%.

3) UNSUPERVISED TECHNIQUES

The underlying hypothesis is that authorized transactions follow patterns, while fraudulent transactions deviate from those patterns.

A recent unsupervised method built on top of the hypersphere model captures legitimate user behavior [37] and claims to yield better results than other traditional approaches. Also, a method [38] based on Local Outlier Factor (LoF) and Isolation Forest has shown superior performance compared to other machine-learning techniques.

Another approach [39] based on sequence-based neural network shows high performance, model interpretability and resilience against concept drift. Moreover, factoring in concept drift [40] in credit card customer behavior has proven to yield better detection results. Lastly, an approach [26] leveraging a Hidden Markov Model (HMM) proved to increase precision-recall AUC by 15% compared to other FDS.

TABLE 1. Dataset label distribution.

Label	# Records	Percentage (%)
Confirmed	195,265	43.77%
Discarded	250,811	56.22%

Unsupervised learning outperforms supervised methods when considering skewness [19]. However, oversampling methods like SMOTE have proven [41] to improve accuracy in supervised models. Lastly, recent research concludes that combining multiple outlier scores can negatively impact accuracy [30].

4) HYBRID AND ENSEMBLE-BASED TECHNIQUES

The assumption is that no single technique can detect all types of fraud, and so combining techniques yields better results.

Accuracy is improved by combining Support Vector Machine (SVM), K-Nearest Neighbor (K-NN) and Multilayer Perceptron (MLP) classifiers, compared to standalone classifiers: Naïve Bayes, Extreme learning machine (ELM), K-NN, MLP and SVM [29], [31]. Another research [42] that leverages a variation of the stacking ensemble method and AdaBoost delivered higher accuracy (94.5%) than other techniques: stacking (92.6%), AdaBoost (91.4%), Random Forest (90%), Decision Tree (89.1%), and Logistic Regression (87.2%).

Combining k-means and artificial bee colony algorithm (ABC) through semantic fusion increases accuracy [43]. Moreover, clustering customers ahead of the classification task [44] improves accuracy for certain types of customers.

Lastly, combining supervised (decision tree) and semi-supervised (transaction sequences and user behavior) led to an increase of 7% in detection rate [45].

III. METHODOLOGY

A. STATEMENT OF THE PROBLEM

In our research, several deep neural network architectures were trained to automatically discard alerts associated with false positives, thus effectively reducing the number of alerts requiring manual investigation.

B. DATASET DESCRIPTION

The dataset used for our research contained 446,076 real alerts related to suspicious credit card transactions. They were obtained from a Spanish payments processing organization. The alerts in this dataset span a period of six months. The proportion of confirmed and discarded transactions is shown in Table 1.

Available features are shown in Table 2.

C. FEATURE ENGINEERING

The dataset contains both numerical and categorical features. Features were encoded as fixed-length vectors of binary

TABLE 2. Input columns.

Feature	Description	Type	Cardinality	Range
Amount	Transaction amount in local currency.	Numerical	N/A	[0, 231.014,48]
Day of month	Day of month.	Numerical	N/A	[1, 31]
Hour	Hour.	Numerical	N/A	[0, 23]
Data Input	Method used by Point of Sales terminal for credit card identification.	Categorical	21	N/A
Authentication method	Method used by Point of Sales terminal for customer authentication.	Categorical	12	N/A
Response code	Response code supplied by the payment processing system after the transaction was processed.	Categorical	59	N/A
Merchant type (international)	It identifies merchant's type of business, according to international classification codes.	Categorical	507	N/A
Merchant type (domestic)	It identifies merchant's type of business, according to Spanish classification codes.	Categorical	325	N/A
City	City where the transaction was recorded.	Categorical	54	N/A
Country	Country where the transaction was recorded.	Categorical	187	N/A
Issuing bank	It identifies which bank issued the credit card used in the transaction.	Categorical	107	N/A
Score	It represents the risk score assigned by the FDS.	Numerical	100	N/A
Label	It represents the analyst's decision: confirmed or discarded alert.	Binary	2	N/A

values, that were supplied as input to each of the neural network architectures.

TABLE 3. Binary encoding example.

Label	# Records	Percentage (%)
B_1	0	000
B_2	1	001
B_3	2	010
B_4	3	011
B_5	4	100

TABLE 4. One hot encoding example.

Feature value	Index	Encoded value
O_1	0	00001
O_2	1	00010
O_3	2	00100
O_4	3	01000
O_5	4	10000

TABLE 5. Bin thresholds.

Feature name	Bin thresholds
Score	[-∞, 10, 20, 30, 40, 50, 60, 70, 80, 90, ∞]
Day of month	[-∞, 10, 20, ∞]
Hour	[-∞, 4, 8, 12, 16, 20, ∞]
Amount	[-∞, 10, 100, 1000, 10000, 100000, ∞]

The encoding types used were:

1) BINARY

Used for categorical features. Each feature value was indexed with a positive (greater than or equal to zero) integer. The resulting index was converted to a binary value. Vector length was chosen based on the number of bits required to represent the highest binary value.

Table 3 shows an example of the encoding process for a feature with five unique values.

$$B = \{B_1, B_2, \ldots, B_5\} \quad (1)$$

2) ONE HOT ENCODING

Used for categorical features. Each feature value was indexed with a positive (greater than zero) integer. Vector length equaled the amount of unique feature values. All bits in the vector were set to zero, except from the bit at the position represented by the index, which was set to one.

Table 4 shows an example of the encoding process for a feature with five unique values.

$$O = \{O_1, O_2, \ldots, O_5\} \quad (2)$$

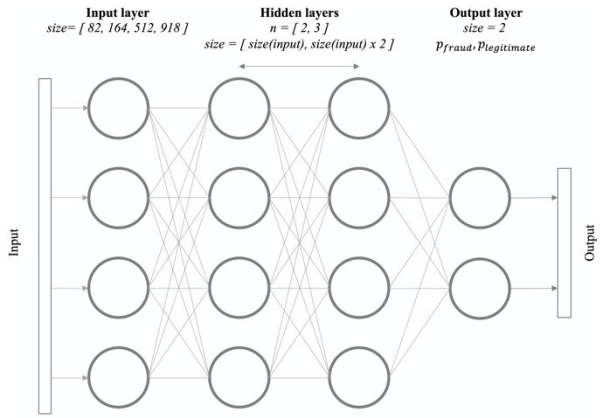

FIGURE 1. MLP architecture.

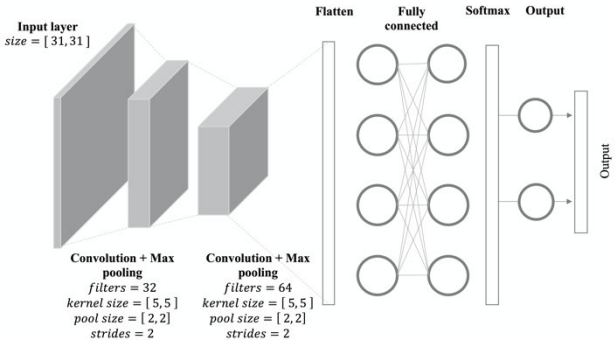

FIGURE 2. CNN architecture.

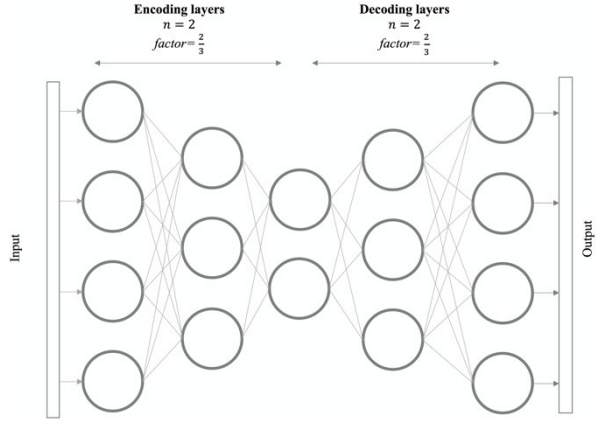

FIGURE 3. DAE architecture.

3) BINNING

Used for numerical features. Each feature value was assigned to a bin, based on a predefined set of bin thresholds, defined from the feature value range.

Table 5 shows the bins used for each numerical feature.

Lastly, Label was formatted as 1 for confirmed cases (fraud class), and as 0 for discarded cases (false positive class).

D. NEURAL NETWORK ARCHITECTURES

The set of neural network architectures tested in our research belong to one of the following types:

TABLE 6. Neural network key parameters.

Label	# Records
MLP	Number of hidden layers
	Hidden layer sizes
CNN	Number of convolutional layers
	Input layer shape
DAE	Number of encoding/decoding layers
	Layer sizes
Common (MLP, CNN, DAE)	Batch size
	Learning rate
	Number of epochs

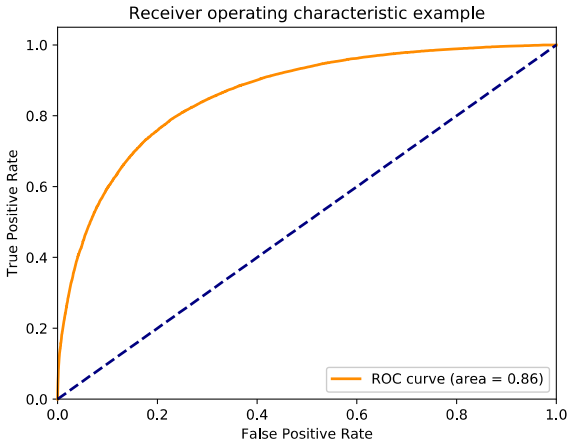

FIGURE 4. MLP2BE256H82 ROC.

Multi-Layer Perceptron (MLP). MLP is type of feedforward, fully connected, neural network with three or more layers of neurons. It uses backpropagation for learning. It contains one input layer, one or more hidden layers, and an output layer. Each neuron applies a non-linear activation function to its input.

A visual representation of the MLP architecture used in our research is shown in Figure 1.

Convolutional Neural Network (CNN). CNN also has an input layer, several hidden layers and an output layer. Hidden layers implement sequences of convolution and max pooling operations and are followed by a fully connected layer. A convolution operation is a sliding dot product of the input and a ReLu (rectifier linear unit) as non-linear activation function. Max pooling does non-linear down-sampling.

A visual representation of the CNN architecture used in our research is shown in Figure 2.

Deep Autoencoder (DAE): DAE is a type of neural network used for unsupervised learning. It performs dimensionality reduction across a number of layers. It then reconstructs the signal back from its compressed representation. It's used to learn patterns and spot outliers in a dataset, that is, inputs with a high reconstruction error.

A visual representation of the DAE architecture used in our research is shown in Figure 3.

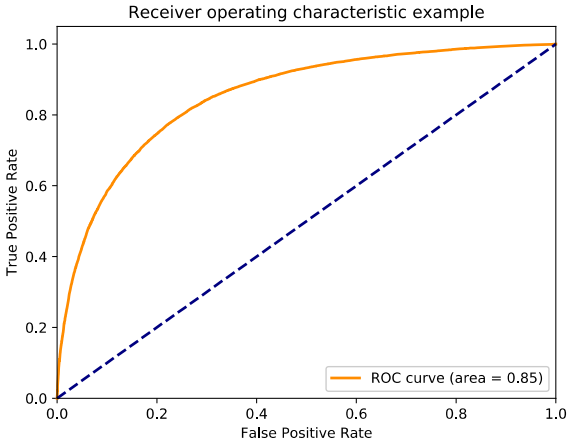

FIGURE 5. MLP2BE128H164 ROC.

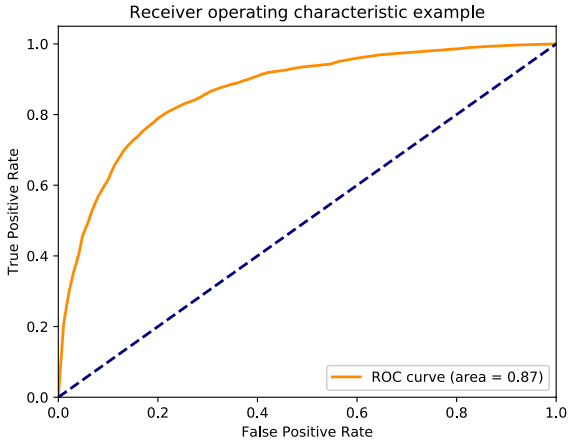

FIGURE 6. MLP2OH128H918 ROC.

TABLE 7. MLP neural network architectures.

Identifier	Test dataset size	Test dataset size (legitimate)	Test dataset size (fraud)
MLP2BE256H82	89,216	50.168	39,048
MLP2BE128H164	89,216	50.168	39,048
MLP2OH128H918	89,216	49,624	39,592
MLP3OH256H512	89,216	49,907	39,309
MLP3BE256H512	89,216	50,117	39,099
MLP3OH256H918	89,216	49,911	39,305
CNN2OH100LR10-3	89,216	50,231	38,985
CNN2OH100LR10-1	89,216	50,137	39,079
DAE4BE256	244,081	48,816	195,265
DAE4OH256	244,081	48,816	195,265

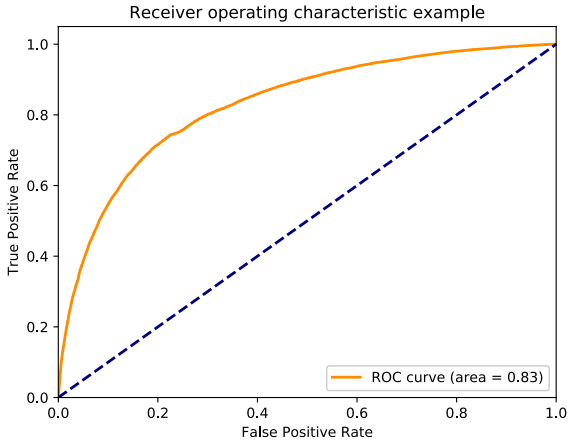

FIGURE 7. MLP3OH256H512 ROC.

E. ARCHITECTURE DESIGN CRITERIA

The architectures selected in our research represent both supervised and unsupervised learning settings.

MLP and CNN are popular neural network types for classification tasks on fixed length input data. They apply supervised learning, and, therefore, they require labeled data. DAE are used for unsupervised learning and require no labeled data. They perform the classification task with no prior knowledge of fraud patterns. Each neural network architecture requires different parametrization.

Table 6 shows key parameters of each architecture.

F. PERFORMANCE EVALUATION CRITERIA

Precision and recall are popular performance evaluation metrics in classification tasks.

They are also used for fraud detection [23].

$$Precision = \frac{tp}{tp + fp} \tag{3}$$

$$Recall = \frac{tp}{tp + fn} \tag{4}$$

where true positives (*tp*) refer to alerts related to actual fraud cases, false positives (*fp*) are legitimate transactions that erroneously triggered an alert, and false negatives (*fn*) are fraud cases that triggered no alert.

In our research, precision and recall were measured for each neural network architecture. The Receiver Operator Characteristic (ROC) curve was also obtained, and the resulting Area Under the Curve (AUC) metric was displayed. For MLP and CNN, thresholds were based on the probability associated with the fraud (1) class.

For DAE, thresholds were set based on reconstruction error values.

$$e = -log_{10}\left(\frac{1}{N}\sum_{i=0}^{N}(x_i - \tilde{x}_i)^2\right) \tag{5}$$

where e represents reconstruction error, N represents input length, x_i is each dimension of the input and $\tilde{x}_i$ is each dimension of the output (that is, the reconstructed input).

Based on those performance metrics, alert reduction rate of each configuration was obtained. This rate represents proportion of alerts requiring no review by a human analyst, that is, those automatically classified as false

TABLE 8. Confusion matrix (MLP2OH128H918, threshold = 0.1).

Predicted class	Legitimate	Fraud	Total
Real class			
Legitimate	28,120	21,504	49,624
Fraud	3,249	36,343	39,592
Total	31,369	57,487	89,216

TABLE 9. Confusion matrix (MLP2OH128H918, threshold = 0.2).

Predicted class	Legitimate	Fraud	Total
Real class			
Legitimate	32,367	17,257	49,624
Fraud	4,849	34,743	39,592
Total	37,216	52,000	89,216

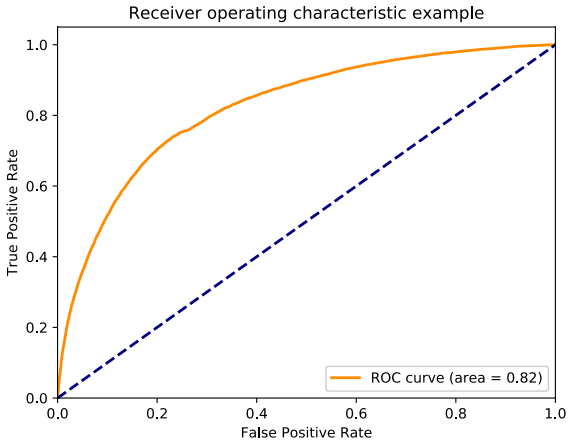

FIGURE 8. MLP3BE256H512 ROC.

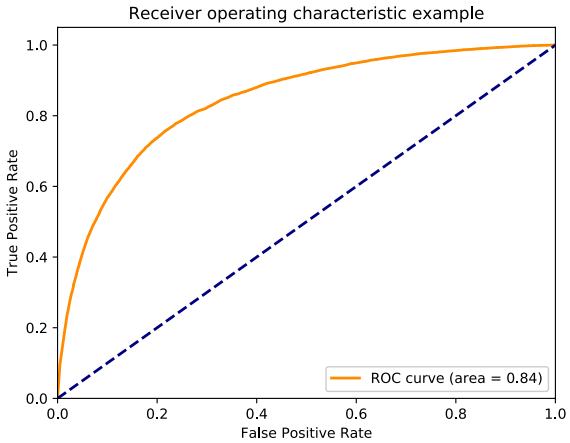

FIGURE 9. MLP3OH256H918 ROC.

positives. Each reduction rate has an associated misclassification rate (true positives erroneously discarded as false positives).

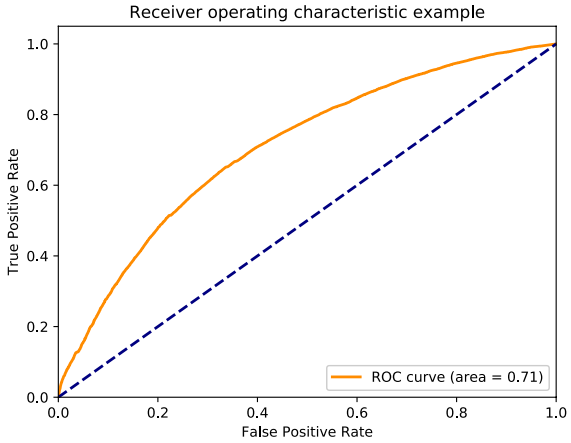

FIGURE 10. CNN2OH100LR10-3 ROC.

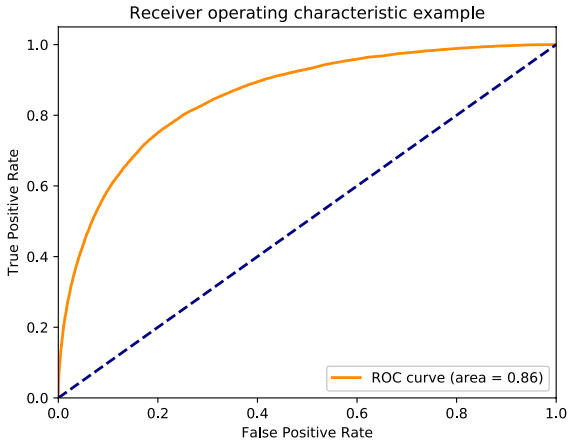

FIGURE 11. CNN2OH100LR10-1 ROC.

IV. EXPERIMENTAL SETTING

A. TRAINING AND TEST DATASETS

Train and test datasets were produced for each architecture (supervised or unsupervised). For MLP and CNN, 20% of the alerts (89,216) were randomly assigned to the test dataset, with the remaining 80% (356,860) being assigned to the train dataset. For DAE, 201,995 alerts raised for legitimate transactions were used for training. 48,816 alerts associated with legitimate transactions and 195,265 alerts associated with fraud were used for testing, totaling 244,081.

Table 7 shows the amount of train and test alerts in each setting:

B. PARAMETRIZATION

For each architecture type, reference (initial) parameter values were assigned. They are shown in Tables 10, 11 and 12. These values were adjusted using grid search, in order to evaluate their degree of influence on performance.

C. RESULTS AND DISCUSSION

Obtained results are summarized in Tables 13 to 22. ROC curves and their respective AUC are included in

TABLE 10. MLP parametrization.

Identifier	Hidden layers	Encoding (categorical)	Encoding (numerical)	Batch size	Learning rate	Hidden layer size	Epochs
MLP2BE256H82	2	Binary	Binning	256	10^{-3}	82	250
MLP2BE128H164	2	Binary	Binning	128	10^{-2}	164	1,000
MLP2OH128H918	2	OHE	Binning	128	10^{-2}	918	250
MLP3OH256H512	3	OHE	Binning	256	10^{-3}	512	250
MLP3BE256H512	3	Binary	Binning	256	10^{-3}	512	250
MLP3OH256H918	3	OHE	Binning	256	10^{-3}	918	250

TABLE 11. CNN parametrization.

Identifier	Convolutional layers	Encoding (categorical)	Encoding (numerical)	Batch size	Learning rate	Input layer shape	Epochs
CNN2OH100LR10-3	2	OHE	Binning	100	10^{-3}	31x31	25,000
CNN2OH100LR10-1	2	OHE	Binning	100	10^{-1}	31x31	25,000

TABLE 12. DAE parametrization.

Identifier	Layers	Encoding (categorical)	Encoding (numerical)	Batch size	Learning rate	Layer sizes	Epochs
DAE4BE256	4	Binary	Binning	256	10^{-2}	55 (1) 37 (2)	250
DAE4OH256	4	OHE	Binning	256	10^{-2}	612 (1) 408 (2)	500

TABLE 13. MLP2BE256H82 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.92	0.43	0.89	0.59	0.85	0.71	0.81	0.78	0.78	0.84
Fraud	0.44	1.00	0.57	0.95	0.63	0.90	0.69	0.84	0.73	0.77	0.77	0.70
Average / Total	0.19	0.44	0.77	0.66	0.78	0.73	0.78	0.76	0.78	0.78	0.78	0.78

TABLE 14. MLP2BE128H164 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.92	0.43	0.89	0.58	0.86	0.68	0.83	0.74	0.80	0.81
Fraud	0.44	1.00	0.56	0.95	0.63	0.91	0.67	0.85	0.71	0.81	0.75	0.74
Average / Total	0.19	0.44	0.76	0.66	0.77	0.72	0.78	0.76	0.78	0.77	0.78	0.78

figures 1 to10. MLP and CNN (supervised setting) achieved significantly better results than DAE.

DAE recorded an AUC of 0.52 for binary encoding and 0.48 for one-hot encoding (OHE). A possible reason behind this poor performance is that alerts data don't represent all patterns of legitimate credit card transactions, but just a subset of them (those that were tagged as suspicious by the FDS). In this case, training inputs would be strongly biased, and

TABLE 15. MLP2OH128H918 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.90	0.57	0.87	0.65	0.82	0.81	0.74	0.90	0.74	0.81
Fraud	0.44	1.00	**0.63**	**0.92**	**0.67**	**0.88**	0.76	0.78	0.83	0.61	0.84	0.69
Average / Total	0.20	0.44	**0.78**	**0.72**	**0.78**	**0.75**	0.79	0.79	0.78	0.77	0.78	0.76

TABLE 16. MLP3OH256H512 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.77	0.81	0.76	0.84	0.75	0.86	0.74	0.87	0.73	0.88
Fraud	0.44	1.00	0.74	0.70	0.76	0.66	0.78	0.63	0.79	0.61	0.80	0.59
Average / Total	0.19	0.44	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.75

TABLE 17. MLP3BE256H512 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.83	0.66	0.82	0.70	0.80	0.74	0.79	0.76	0.78	0.78
Fraud	0.44	1.00	0.66	0.83	0.68	0.80	0.70	0.77	0.71	0.75	0.72	0.72
Average / Total	0.19	0.44	0.76	0.73	0.76	0.74	0.76	0.75	0.76	0.76	0.76	0.76

TABLE 18. MLP3OH256H918 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.84	0.69	0.82	0.73	0.81	0.77	0.80	0.80	0.79	0.81
Fraud	0.44	1.00	0.68	0.83	0.70	0.80	0.73	0.76	0.74	0.74	0.75	0.73
Average / Total	0.19	0.44	0.77	0.75	0.77	0.76	0.77	0.77	0.77	0.77	0.77	0.77

therefore not suitable for the unsupervised approach used in DAE.

MLP configurations with less depth (that is, less hidden layers) showed slightly higher performance in terms of AUC. One-hot encoding showed slightly higher performance than binary encoding, also in terms of AUC. On the other hand, batch size, number of epochs and learning rate didn't seem to be relevant parameters.

MLP2OH128H918 obtained the best AUC (0.87). Setting the threshold to 0.2 led to best tradeoff between average precision (0.78) and average recall (0.75), with fraud class showing a precision of 0.67 and a recall of 0.88.

Table 8 and Table 9 shows the confusion matrixes for threshold values 0.1 and 0.2:

In order to measure alert reduction performance, alert reduction rate (arr) and misclassification rate (mr) were calculated for optimal thresholds.

Alert reduction rate is calculated as follows:

$$arr = \frac{pred_l}{N} \tag{6}$$

TABLE 19. CNN2OH100LR10-3 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.00	0.00	0.85	0.09	0.78	0.35	0.73	0.60	0.67	0.78
Fraud	0.44	1.00	0.44	1.00	0.46	0.98	0.51	0.88	0.58	0.71	0.65	0.51
Average / Total	0.19	0.44	0.19	0.44	0.68	0.48	0.66	0.58	0.66	0.65	0.66	0.66

TABLE 20. CNN2OH100LR10-1 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.00	0.00	0.95	0.27	0.91	0.47	0.86	0.65	0.82	0.76	0.78	0.84
Fraud	0.44	1.00	0.51	0.98	0.58	0.94	0.66	0.87	0.72	0.78	0.77	0.69
Average / Total	0.19	0.44	0.76	0.58	0.77	0.68	0.77	0.74	0.77	0.77	0.78	0.78

TABLE 21. DAE4BE256 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.21	0.65	0.20	0.52	0.21	0.42	0.23	0.34	0.23	0.24	0.25	0.13
Fraud	0.82	0.39	0.80	0.49	0.81	0.61	0.81	0.71	0.81	0.80	0.81	0.91
Average / Total	0.69	0.44	0.68	0.50	0.69	0.57	0.69	0.64	0.69	0.69	0.70	0.75

TABLE 22. DAE4OH256 performance.

Threshold	0.0		0.1		0.2		0.3		0.4		0.5	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Legitimate	0.20	0.62	0.19	0.49	0.17	0.32	0.17	0.25	0.17	0.17	0.18	0.09
Fraud	0.81	0.39	0.79	0.49	0.78	0.59	0.79	0.69	0.79	0.80	0.80	0.90
Average / Total	0.69	0.44	0.67	0.49	0.66	0.54	0.66	0.60	0.67	0.67	0.68	0.74

where $pred_l$ is the number of alerts classified as legitimate, and N is the total number of alerts.

Misclassification rate is calculated as follows:

$$mr = \frac{fn}{real_f} \tag{7}$$

where fn is the number of false negatives, and $real_f$ is the total number of real fraud cases.

The resulting efficiency (alert reduction rate) was 35.16% ($threshold = 0.1$):

$$arr_{0.1} = \frac{31,369}{89,216} = 0.3516\,(35.16\%) \tag{8}$$

to detect 91.79% of the fraud cases (8.21% misclassification rate):

$$mr_{0.1} = \frac{3,249}{39,592} = 0.08206\,(8.21\%) \tag{9}$$

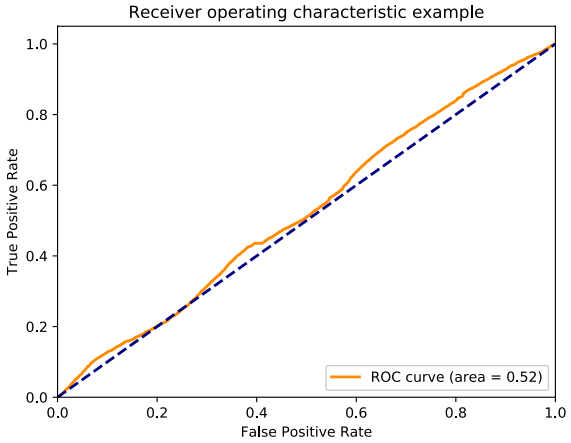

FIGURE 12. DAE4BE256 ROC.

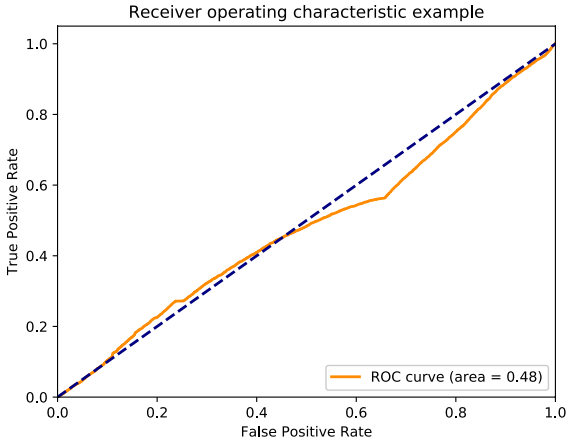

FIGURE 13. DAE4OH256 ROC.

Alert reduction rate could grow to 41.47% (*threshold* = 0.2) with a slightly higher misclassification rate:

$$arr_{0.2} = \frac{37,216}{89,216} = 0.4147\,(41.47\%) \tag{10}$$

to detect 87.75% of the fraud cases (12.25% misclassification rate):

$$mr_{0.2} = \frac{4,849}{39,592} = 0.12247\,(12.25\%) \tag{11}$$

V. CONCLUSION

In our research, deep neural networks were assessed from a perspective of credit card fraud alert reduction. The goal was to reproduce the ability to capture and automate decision criteria used by humans reported by previous literature.

A set of alerts triggered by an FDS (associated with suspicious transactions) were classified as either valid alerts, representing real fraud cases, or wrong alerts, representing false positives, by ten neural network architectures.

Optimal configuration (MLP2OH128H918) achieved an alert reduction rate (*threshold* = 0.1) of 35.16% when capturing 91.79% of fraud cases (8.21% misclassification rate), and a reduction rate (*threshold* = 0.2) of 41.47% when capturing 87.75% of fraud cases (12.25% misclassification rate). It should be noted that such level of reduction would entail a significant reduction in cost of human labor, because alerts classified as false positives by the neural network wouldn't require human inspection.

Therefore, deep neural networks can be considered as a promising choice for assisting fraud detection teams in payment organizations looking to achieve efficiencies in their fraud investigations.

ACKNOWLEDGMENT

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] T. Pietraszek, "Using adaptive alert classification to reduce false positives in intrusion detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2004, pp. 102–124.

[2] H. W. Njogu and L. Jiawei, "Using alert cluster to reduce IDS alerts," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, vol. 5, Jul. 2010, pp. 467–471.

[3] R. Harang and P. Guarino, "Clustering of snort alerts to identify patterns and reduce analyst workload," in *Proc. MILCOM-IEEE Mil. Commun. Conf.*, Oct. 2012, pp. 1–6.

[4] R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan, "Using neuro-fuzzy approach to reduce false positive alerts," in *Proc. 5th Annu. Conf. Commun. Netw. Services Res. (CNSR)*, May 2007, pp. 345–349.

[5] N. A. Bakar, B. Belaton, and A. Samsudin, "False positives reduction via intrusion alert quality framework," in *Proc. 13th IEEE Int. Conf. Netw. Jointly IEEE 7th Malaysia Int. Conf Commun.*, vol. 1, Nov. 2005, p. 6.

[6] F. Xiao, S. Jin, and X. Li, "A novel data mining-based method for alert reduction and analysis," *J. Netw.*, vol. 5, no. 1, p. 88, Jan. 2010.

[7] S. McElwee, J. Heaton, J. Fraley, and J. Cannady, "Deep learning for prioritizing and responding to intrusion detection alerts," in *Proc. MILCOM-IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 1–5.

[8] R. P. Srivastava, "Automating judgmental decisions using neural networks: A model for processing business loan applications," in *Proc. ACM Annu. Conf. Commun. (CSC)*, 1992, pp. 351–357.

[9] T. Hill and W. Remus, "Neural network models for intelligent support of managerial decision making," *Decis. Support Syst.*, vol. 11, no. 5, pp. 449–459, Jun. 1994.

[10] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with EMRs," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2014, pp. 556–559.

[11] A. Ciritsis, C. Rossi, M. Eberhard, M. Marcon, A. S. Becker, and A. Boss, "Automatic classification of ultrasound breast lesions using a deep convolutional neural network mimicking human decision-making," *Eur. Radiol.*, vol. 29, no. 10, pp. 5458–5468, Oct. 2019.

[12] C. L. Tan, T. S. Quah, and H. H. Teh, "An artificial neural network that models human decision making," *Computer*, vol. 29, no. 3, pp. 64–70, Mar. 1996.

[13] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[14] W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings*, San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 115–123.

[15] W. Lee, "A date mining framework for constructing features and models for intrusion detection systems," Ph.D. dissertation, Dept. Comput. Sci., Columbia Univ., New York, NY, USA, 1999.

[16] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, Oct. 2000.

[17] D. Nauck, U. Nauck, and R. Kruse, "NEFCLASS for Java-new learning algorithms," in *Proc. 18th Int. Conf. North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, Jun. 1999, pp. 472–476.

[18] Z. He, X. Xu, Z. Huang, and S. Deng, "FP-outlier: Frequent pattern based outlier detection," *Comput. Sci. Inf. Syst.*, vol. 2, no. 1, pp. 103–118, 2005.

[19] S. Mittal and S. Tyagi, "Performance evaluation of machine learning algorithms for credit card fraud detection," in *Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2019, pp. 320–324.
[20] G. Rushin, C. Stancil, M. Sun, S. Adams, and P. Beling, "Horse race analysis in credit card fraud—Deep learning, logistic regression, and gradient boosted tree," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2017, pp. 117–121.
[21] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A neural network based database mining system for credit card fraud detection," in *Proc. IEEE/IAFE Comput. Intell. Financial Eng. (CIFEr)*, Mar. 1997, pp. 220–226.
[22] W. W. Soh and R. M. Yusuf, "Predicting credit card fraud on a imbalanced data," *Int. J. Data Sci. Adv. Anal.*, vol. 1, no. 1, pp. 12–17, 2019.
[23] P. Kumari and S. P. Mishra, "Analysis of credit card fraud detection using fusion classifiers," in *Computational Intelligence in Data Mining*. Singaore: Springer, 2019, pp. 111–122.
[24] W. Yang, Y. Zhang, K. Ye, L. Li, and C. Z. Xu, "FFD: A federated learning based method for credit card fraud detection," in *Proc. Int. Conf. Big Data*. Cham, Switzerland: Springer, 2019, pp. 18–32.
[25] A. Singh and A. Jain, "Adaptive credit card fraud detection techniques based on feature selection method," in *Advances in Computer Communication and Computational Sciences*. Singapore: Springer, 2019, pp. 167–178.
[26] Y. Lucas, P.-E. Portier, L. Laporte, S. Calabretto, O. Caelen, L. He-Guelton, and M. Granitzer, "Multiple perspectives HMM-based feature engineering for credit card fraud detection," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1359–1361.
[27] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," 2019, *arXiv:1904.10604*. [Online]. Available: http://arxiv.org/abs/1904.10604
[28] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S.-K. Nam, Y. Song, J.-A. Yoon, and J.-I. Kim, "Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning," *Expert Syst. Appl.*, vol. 128, pp. 214–224, Aug. 2019.
[29] D. Prusti and S. K. Rath, "Fraudulent transaction detection in credit card by applying ensemble machine learning techniques," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–6.
[30] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, early access, May 16, 2019, doi: 10.1016/j.ins.2019.05.042.
[31] D. Prusti, S. S. Padmanabhuni, and S. K. Rath, "Credit card fraud detection by implementing machine learning techniques," in *Proc. 1st Int. Conf. Mach. Learn., Image Process., Netw. Secur. Data Sci. (MIND)*, Kurukshetra, India. Rourkela, India: National Institute of Technology, 2019.
[32] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Inf. Sci.*, early access, May 16, 2019, doi: 10.1016/j.ins.2019.05.023.
[33] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019.
[34] G. Parthasarathy, L. Ramanathan, Y. JustinDhas, J. Saravanakumar, and J. Darwin, "Comparative case study of machine learning classification techniques using imbalanced credit card fraud datasets," in *Proc. Int. Conf. Sustain. Comput. Sci., Technol. Manage. (SUSCOM)*. Jaipur, India: Amity Univ. Rajasthan, 2019, pp. 1–8.
[35] P. R. Vardhani, Y. I. Priyadarshini, and Y. Narasimhulu, "CNN data mining algorithm for detecting credit card fraud," in *Soft Computing and Medical Bioinformatics*. Singapore: Springer, 2019, pp. 85–93.
[36] S. Cao, X. Yang, C. Chen, J. Zhou, X. Li, and Y. Qi, "TitAnt: Online real-time transaction fraud detection in ant financial," 2019, *arXiv:1906.07407*. [Online]. Available: http://arxiv.org/abs/1906.07407
[37] L. Chen, Z. Zhang, Q. Liu, L. Yang, Y. Meng, and P. Wang, "A method for online transaction fraud detection based on individual behavior," in *Proc. ACM Turing Celebration Conf.-China-ACM TURC*, 2019, pp. 1–8.
[38] V. C. Sharmila, K. K. R., R. Sundaram, D. Samyuktha, and R. Harish, "Credit card fraud detection using anomaly techniques," in *Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. (ICIICT)*, Apr. 2019, pp. 1–6.
[39] K. Yang and W. Xu, "FraudMemory: Explainable memory-enhanced sequential neural networks for financial fraud detection," in *Proc. 52nd Hawaii Int. Conf. Syst. Sci.*, 2019, pp. 1023–1032.
[40] Y. Lucas, P.-E. Portier, L. Laporte, S. Calabretto, L. He-Guelton, F. Oble, and M. Granitzer, "Dataset shift quantification for credit card fraud detection," in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Jun. 2019, pp. 97–100.
[41] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection-machine learning methods," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2019, pp. 1–5.
[42] E. Prabhakara, M. N. Kumarb, K. Ponnarb, A. Sureshb, and R. Jayandhiranb, "Credit card fraud detection using boosted stacking," *South Asian J. Eng. Technol.*, vol 8, no. S1, pp. 149–153, Apr. 2019.
[43] S. M. Darwish, "An intelligent credit card fraud detection approach based on semantic fusion of two classifiers," *Soft Comput.*, vol. 24, no. 2, pp. 1243–1253, Jan. 2020.
[44] N. Kasa, A. Dahbura, C. Ravoori, and S. Adams, "Improving credit card fraud detection by profiling and clustering accounts," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2019, pp. 1–6.
[45] A. Eshghi and M. Kargari, "Introducing a method for combining supervised and semi-supervised methods in fraud detection," in *Proc. 15th Iran Int. Ind. Eng. Conf. (IIIEC)*, Jan. 2019, pp. 23–30.

RAFAEL SAN MIGUEL CARRASCO received the degree in telecommunications engineering and the degree in computer science and engineering from the University Alfonso X El Sabio. He is currently pursuing the Ph.D. degree in artificial intelligence with the University of Alcalá.

He has worked in the fields of security operations and ethical hacking, and has also strong background and experience in developing security analytics and fraud detection capabilities by leveraging statistical modeling, machine learning, and artificial intelligence techniques. He has developed his professional career in multinational firms such as British Telecom, Deloitte, Santander, and Silicon Valley start-up FireEye. He is currently a Senior Expert of cybersecurity practice with Minsait (Indra).

MIGUEL-ÁNGEL SICILIA-URBÁN received the degree in computer science from the Pontifical University of Salamanca, the degree in information science from the University of Alcalá, Madrid, Spain, and the Ph.D. degree in computer science from Carlos III University.

Before joining academia, he was a part of the Research and Development and E-commerce Architecture Staff of the iSOCO, a spinoff from the Artificial Intelligence Institute of the Spanish Council of Science. He is currently a Full Professor with the Department of Computer Science, University of Alcalá. He has developed his research activity in the fields of artificial intelligence, machine learning, and analytics applied to different fields, including learning, health, computational science, command and control systems, and information security.