A Natural Language Processing Approach for Financial Fraud Detection

Javier Fernández Rodríguez[1,2], Michele Papale[1], Michele Carminati[1,*] and Stefano Zanero[1]

[1]*Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, Italy*
[2]*Universidad Politécnica de Madrid Madrid, Spain*

Abstract

Due to the proliferation of online banking, people are more exposed than ever to attacks. Moreover, frauds are becoming more sophisticated, bypassing the security measures put in place by the financial institutions. In this paper, we propose a novel approach to fraud detection based on Natural Language Processing models. We model the user's spending profile and detect frauds as deviations from it. To do so, we employ the attention mechanism that allows us to model and fully exploit past transactions. Our evaluation on real-world data shows that our model achieves a good balance between precision and recall, outperforming traditional approaches in different scenarios.

Keywords

Fraud Detection, Natural Language Processing, Transformer Model

1. Introduction

Frauds are becoming more sophisticated as time goes along, bypassing the protection mechanisms put in place. The Fraud Detection and Prevention market is valued at 19.5 billion dollars and raising [1]. Consequently, financial institutions are demanding up-to-date solutions. Financial fraud detection is challenging due to various reasons that make usual techniques ineffective. In fact, frauds are difficult to detect due to the **lack of data**, the **concept drift** of spending profiles, and the **temporal dimension** of data. Financial datasets are hard to obtain due to privacy concerns. However, thanks to the collaboration with a large Italian bank, we were able to work on a real-world dataset. Additionally, frauds are rare by definition and mixed with legitimate transactions. The imbalance between classes seriously hurts the performance of detection models. Users, as well as fraudsters, behave differently as time goes by. Consequently, the temporal dimension is crucial for achieving good performance.

Natural Language Processing (NLP) studies the interactions between machines and human language. This field has seen a dramatic change in recent years thanks to models based on the Transformer [2]. Moreover, Transformer-based models have been proven to be universal approximators of any sequence-to-sequence functions [3]. The objective of this work is to

ITASEC'22: Italian Conference on Cybersecurity, June 20–23, 2022, Rome, Italy
*Corresponding author.
javier.fernandez@mail.polimi.it (J. F. Rodríguez); michele.papale@mail.polimi.it (M. Papale); michele.carminati@polimi.it (M. Carminati); stefano.zanero@polimi.it (S. Zanero)

exploit the advances made in the Natural Language Processing (NLP) field to fully exploit user's past transactions to build the user spending profile. This model directly tackles the domain's challenges and has shown outstanding performance in many fields [4, 5, 6] that share similarities with the fraud detection one.

In this paper, we present a novel approach for fraud detection based on the Transformer model [2], a state-of-the-art technique in the Natural Language Processing field to model the user's spending profile and detect frauds as deviations from it. We compute a risk score by comparing the predicted transactions with the actual transactions. To do so, we employ attention-based mechanisms [2] and unsupervised multitasks learners [7], which allow us to model and fully exploit past transactions. In particular, we exploit the transformer model and the attention mechanism to tackle the domain's challenges directly. The transformed model is designed to deal with sequences of data and time-series, while the attention mechanism over the user's past transactions has been proven to be robust against concept drift [8]. Our approach hinders explainability but enables the model to optimize the feature representation of the input generated by neural networks, which usually achieve better performance [8, 9, 10]. The evaluation on a real-world dataset shows that our model outperforms state-of-the-art approaches in different scenarios, going from realistic to adversarial attacks. Finally, we also demonstrate the better performance of the proposed generative solution with respect to NLP-based discriminative approaches [11].

In summary, we make the following contributions:

- We present, to the best of our knowledge, the first study on the application of the Transformer model to the fraud detection task. By doing this, we take into account the time dimension and fully exploit the users' past spending patterns, tackling the concept drift of the user's profile and the data scarcity issues.
- We evaluate our approach on a real-world dataset, showing that it outperforms state-of-the-art methods in different fraudulent scenarios.
- We compare the performance of NLP-based generative solutions against NLP-based discriminative approaches deployed in the fraud detection domain.

2. Background and Motivation

Most fraud detection approaches define fraud as a deviation from the normal spending pattern. However, frauds are difficult to detect due to the **lack of data**, the **concept drift** of spending profiles, and the **time dimension**. In fact, financial datasets are hard to obtain due to privacy concerns. However, thanks to the collaboration with a large Italian bank, we were able to work on a real-world dataset. Additionally, frauds are rare by definition and mixed with legitimate transactions. The imbalance between classes seriously hurts the performance of detection models. Users, as well as fraudsters, behave differently as time goes by. Consequently, the temporal dimension is crucial for achieving good performance.

Related Work. Existing approaches can be categorized based on how they model normal user's behavior [12]: local models are user-centric and aggregate features by users; global models are system-centric and try to model the behavior of the system.

Local models usually rely on neural networks designed to deal with sequences of data. Among them, LSTM and GRU based solutions are the more popular options [13, 14, 15]. The main shortcoming of these solutions is the need for a fixed-length input and the LSTM cells' bottleneck. This problem hinders the processing of both users with very few transactions and the ones with a large amount of them. Instead, our solution can process users who perform from 1 transaction up to 1024 thanks to the attention-based mechanism. Fraudmemory [8] adds attention on top of the LSTM output. The model outperforms state-of-the-art solutions in terms of Precision, Recall, and AUC. Our solution uses attention-based mechanisms, but it applies them directly to the input to mitigate the the bottleneck for the flow of information inside the neural networks [2]. Zamini et al. [16] use an autoencoder to model legitimates credit card transactions, and it leverages the reconstruction error to detect frauds. In Veeramachaneni et al. [17], the authors propose an ensemble of unsupervised methods, including a Density-based model, a Matrix Decomposition-based model, and a Replicator Neural Network. By combining the anomaly scores computed by the three models, their system ranks the instances based on the most anomalous ones and then presents them to the subject matter expert for review; the feedback collected is used to train a Random Forest model. One shortcoming of this kind of solution is the impossibility of training the model as a whole. Hence, they fail to extract rich differential features to detect outliers [12].

Global models usually consist of clustering and are based on the probability distribution of the data, which is later used to spot anomalies. Among the most known techniques, there are k-NN [18, 19]. Although simple, k-NN performs well compared to more complex approaches, according to Campos et al. [20]. OC-SVM [21] is a SVM modified to find a separation plane that englobes all the legitimate samples. Frauds will fall outside this plane, and therefore, they will be detected. Another promising unsupervised technique is OC-NN [22, 23], which partially solves the Autoencoders' problem and can extract a rich representation of the input optimized for fraud detection. Although these models show promising results [12], the training times grow exponentially with the input dimension. Thanks to the use of the attention mechanism, our model does not suffer from this issue.

Banksealer [24, 25, 26] is a semi-supervised model that builds a local, global, and temporal profile using methods with a well-known statistical meaning, which adds explainability to the final result. In a subsequent works [27, 28], the temporal profile is improved with the application of signal processing techniques to exploit the end user's recurrent vs. non-recurrent spending pattern, and the authors exploit analyst feedback to self-tune and improve Banksealer's detection performance using a multi-objective genetic algorithm.

Research Goal. The objective of this work is to exploit the advances made in the NLP field to fully exploit user's past transactions to build the user spending profile. To do so, we exploit the transformer model and the attention mechanism to tackle the domain's challenges directly. The transformed model is designed to deal with sequences of data and time-series, while the attention mechanism over the user's past transactions has been proven to be robust against concept drift [8].

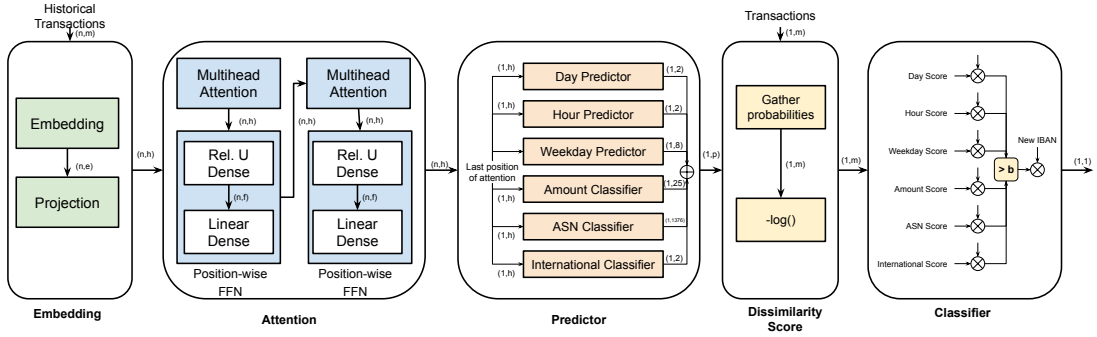


Figure 1: Overall architecture

3. Approach

Our approach is based on the Transformer model [2], which is a sequence-to-sequence model where both input and output are sequences. It exploits an encoder and a decoder: The encoder builds a representation of the input and passes it to the decoder; the decoder takes the information from the encoder and the output generated so far, and predicts the next item in the output sequence. The Transformer model was originally engineered to perform language translation, and, therefore, it excels at modeling long sequences of interrelated data.

In this work, we exploit the similarities between human language and transactions, focusing on their temporal dependency. In particular, we exploit this architecture's modeling capacities to model users' behavior in terms of transactions: Instead of words and sentences, we have transactions and user records. Consequently, instead of predicting the next word, we exploit the transformer model to predict the next transaction in the sequence belonging to the user's spending pattern. Significant changes have been made to adapt the Transformer model to the fraud detection domain's particularities. More formally, we train the Transformer model to predict the next transaction given the user's past transactions. The model takes the last 1024 $(t_0, \ldots, t_{1023})$ transactions and will output a representation of the next transaction t'. We select 1024 as input size since it is the value that in our experimental evaluation obtained the best trade-off between performance and computation requirements. This parameter can be adjusted as needed, but it must be a power of two to allow efficient computation [2]. Then, we compare t' with the actual transaction, and if their difference is higher than a threshold p^{th}, the actual transaction is marked as fraud.

In Figure 1 we present an overview of the proposed solution. The model consists of 47 layers arranged in different blocks depending on their function. All the neural network layers sum a total of 1961358 weights that are optimized with the Adam optimizer. The main building blocks are the Embedding, the Attention mechanism (implemented with Multihead attetention and Position-wise feed-forward network), the Predictor, the Dissimilarity Score, and the Classifier.

3.1. Embedding

Natural language models rely on an embedding layer to translate words to numbers, usually called Word2Vec. The objective is to map the one-hot encoded-word, a sparse high dimensional space, to a reduced dense space. The Embedding allows knowing if two words are similar or not. To do so, Transformer models are trained on large vocabularies, which are preprocessed. There are several algorithms like BPE [29], WordPiece [30] or SentencePiece [30]. The idea is to break down words into smaller pieces. In the case of transactions, we build the vocabulary by considering each transaction's feature as a word of a different vocabulary. We avoid building a vocabulary for each transaction since it would have made the Embedding not scalable. We use the concept of Embedding introduced by Tomás Mikolov [31]. These layers transform the sparse one-hot encoded transactions into a dense, fewer dimensional space. The resulting vectors share an interesting property: similar elements in terms of meaning are mapped closer. This property helps the model to focus on the relevant elements. For instance, if the model is interested in transactions with amounts in the 5th bin, the search will also return transactions belonging to bins 4th and 6th as they are "close in meaning". There are different types of inputs. Therefore, a different embedding is used for each one. The last layer consists of a dense layer that projects the input in a higher space. The concept is similar to the one used in the Electra model [11]. Another issue to keep into consideration is Positional Encoding. The Transformer models need the original position of the encoded input, which is lost during the attention mechanism's application. Existing techniques involve coding a cosine and sine signal from the input, marking the position of each word in the sentence. This technique assumes that all the input elements are equidistant, as in words in a sentence, but transactions are not. Instead of using positions, timestamps are used. Similar to the usual Positional Encoding, cosines and sines are used to encode the information.

3.2. Attention mechanism

The Attention mechanisms are the core of our approach. They were introduced by Bahdanau [32] and Luong [33]. They exploit the similarity extracted from the embeddings to compute the dot product. If the vectors of two similar words are similar, the dot product between them will be higher. The attention mechanism allows the model to search the input for useful information. Thanks to attention, the model can deal with long sequences of inputs. The implementation is similar to the one presented in the original Transformer paper [2]. However, while the original approach is a sequence-to-sequence model conceived for language translation, predicting the next transaction is a sequence-to-one problem. Therefore, we adapt our approach taking inspiration from GPT-2 [7], which is engineered for prediction. Therefore, the proposed architecture is similar to the one used by GPT-2.

Multihead Attention Mechanism. The matrix calculated in Equation 1 indicates which positions of the input V are more relevant given the query Q.

$$Attention(Q, K) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{1}$$

Attention is later multiplied by the value V, as shown in Equation 2. The result V' contains the more relevant input given V, K, and Q. It worth noticing that V' will have the same dimensions

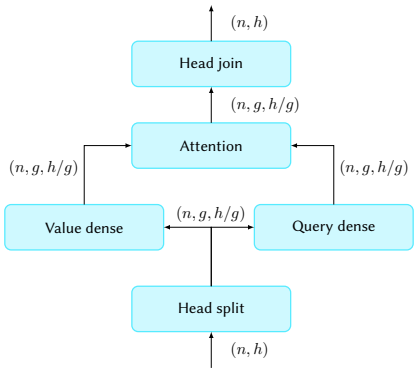


Figure 2: Multihead attention architecture

of Q. In fact, in Transformer-alike models, $K = V$.

$$V' = Attention(Q, K) * V \tag{2}$$

There are different types of attention depending on how Q and V are calculated. In this case, all the blocks use self-attention except for the last one. Self-attention is the case in which Q and V are obtained, applying a linear transformation to the input. It is called self-attention because each input position pays attention to the other positions of the input. In this way, each input is enriched with the context around them. The output has the same dimensions as the input. The last layer of the model consists of cross-attention. Instead of getting Q from all the input, only the last position is used. This forces V' to have the same dimension of Q, which is one transaction's dimension.

Figure 2 shows the attention layer. This layer is called Multihead attention. Instead of using only one *head*, several are used. Each *head* performs the Equations 1 and 2 on different parts of the input. Therefore, the model can pay attention to several positions at once with the same computational cost.

Position-wise Feed-Forward Network. All the dense layers used in the model are linear except the ones used in this layer. The Point-wise feed-forward network adds non-linearity to the model. This operation is Position-wise since it is a non-linear transformation applied to each input position with the same parameters.

3.3. From Prediction to Fraud Detection

Some adjustments are needed to convert a predictive model to a generative one. The model outputs the probability vector for each feature. With that, it is easy to get the probability of a given transaction. Then, the anomaly score is obtained by applying the $-log(\cdot)$ to the probability [4]. Lastly, a meta-learner is trained to convert the anomaly scores of the features to the probability of fraud, by following the stacking ensembling technique [34].

Predictor. The first part of the model has the task of predicting the next transaction. The predictor consists of several dense layers. Each layer performs a linear transformation of the hidden state and is trained to predict the next transaction feature. A transaction is composed

of different features with different loss functions. Each feature contributes to the final loss function in an equal manner: $L_{total} = L_{day} + L_{hour} + L_{amount} + L_{week} + L_{asn} + L_{int}$. In the case of the continuous variables (e.g., day or month), the MSE is used as the loss function. For the discrete variables instead, the layer is trained using Sparse Cross-entropy. The magnitudes of the losses are different as they account for different problems. For example, the ASN loss is larger than the day loss because it is a prediction between 1375 different options whilst the day MSE usually ranges between 0 and 1. This issue is mitigated by choosing Adam optimizer, which scales the loss according to the learning rate [35].

Dissimilarity Score. This layer receives in input the prediction from the model and the current transaction and generates as output the probability of the current transaction given the prediction. The prediction given by the model consists of a vector of probabilities for each feature. Lastly, $-log(\cdot)$ is computed for each feature, yielding a dissimilarity score. This approach is similar to the one taken by Brown et al. [4]. The output of the model is the probability of fraud of the current transaction.

Classifier. The classifier can be seen as a meta-learner. Its function is to weigh the dissimilarity scores of each feature to get a proper classification of frauds. It is a classification problem with two classes. Thus, Binary Crossentronpy is used.

4. Experimental Evaluation

We demonstrate the effectiveness of the model in different scenarios and against state-of-the-art approaches. The experiments are divided depending on the attacker's knowledge [36, 37].

Dataset. The dataset used to validate our model contains the transactions of a large Italian banking group. The transactions belong to two periods. One goes from December 2012 to September 2013, and the other goes from October 2014 to February 2015. This accounts for a total of 1043478, comprising 6195 different users. A detailed analysis of the data can be found in [27, 25]. Each transaction has 31 features, of which only 9 can be used due to the anonymization process. The proposed model takes as input 6 of them: the amount, the hour, the day of the month, the telecommunications operator from which the transaction is issued, and information about the destination IBAN.

Experimental Setup. First, we preprocess the dataset. Then, we split the dataset into the training set to train the model, the validation test to assess the model's performance at each set of the training, and the test set to get the final results of the model. The test set is the only one used to create the different scenarios and injected with synthetic frauds. We refer the reader to Appendix A for the description of the metrics used.

4.1. Black Box Attacks

In the black box attacks scenarios the attacker does not have prior information of the system he or she tries to attack. We consider four different attacking strategies that model real-life situations: Stealing, hijacking, persistent, and Mix. The **Stealing** scenario simulates a phishing attack in which the user's credentials are stolen. The amount transferred is very high. The connection can be originated from a national or foreign IP. The **Hijacking** scenario simulates

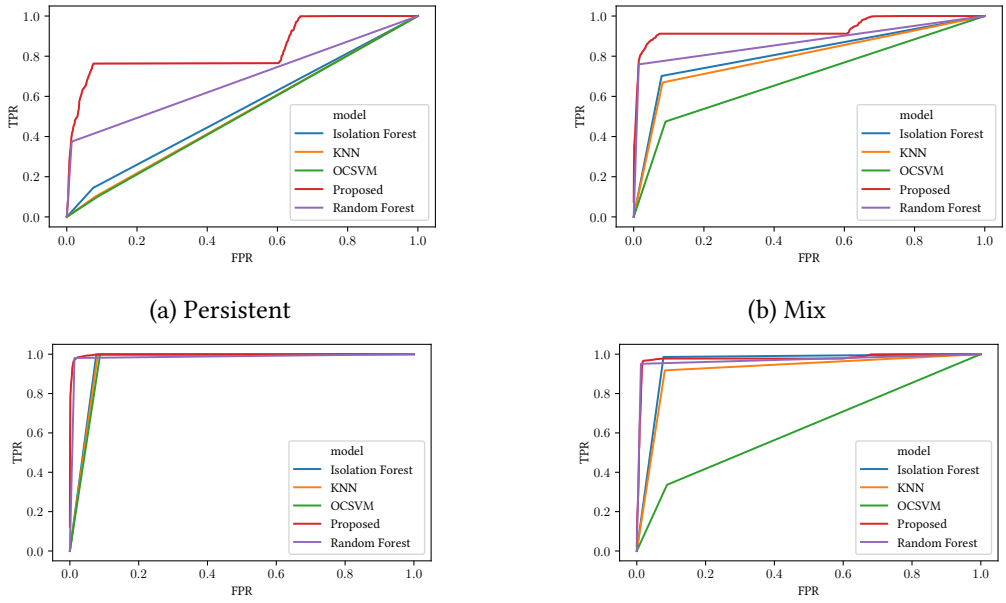
Table 1
Black Box attacks scenario results

Scenario	Model	Precision	Recall	AUROC	AUPR	MCC	F1	FPR
Persistent	Proposed	**0.751**	**0.499**	**0.834**	**0.652**	**0.559**	**0.600**	0.030
	Random F.	0.737	0.387	0.657	0.507	0.525	0.507	**0.014**
	Isolation F.	0.234	0.135	0.528	0.099	0.123	0.171	0.032
	OC-SVM	0.169	0.103	0.506	0.052	0.015	0.128	0.041
	k-NN	0.181	0.103	0.510	0.099	0.072	0.033	0.033
Stealing	Proposed	0.869	**0.986**	**0.997**	**0.987**	0.910	0.924	**0.001**
	Random F.	**0.926**	0.978	0.982	0.804	**0.943**	**0.951**	0.014
	Isolation F.	0.695	0.788	0.961	0.152	0.801	0.739	0.031
	OC-SVM	0.667	0.809	0.956	0.166	0.779	0.731	0.041
	k-NN	0.684	0.801	0.959	0.158	0.793	0.738	0.034
Hijacking	Proposed	0.864	**0.968**	**0.978**	**0.922**	0.897	0.913	**0.007**
	Random F.	**0.927**	0.954	0.970	0.794	**0.929**	**0.940**	0.013
	Isolation F.	0.696	0.921	0.952	0.687	0.791	0.793	0.032
	OC-SVM	0.407	0.336	0.624	0.270	0.269	0.368	0.041
	k-NN	0.668	0.918	0.918	0.647	0.739	0.773	0.034
Mixed	Proposed	0.828	**0.833**	**0.936**	**0.871**	**0.801**	**0.830**	**0.006**
	Random F.	**0.902**	0.756	0.871	0.703	0.800	0.823	0.015
	Isolation F.	0.602	0.707	0.814	0.532	0.589	0.650	0.033
	OC-SVM	0.467	0.474	0.691	0.365	0.382	0.470	0.043
	k-NN	0.576	0.669	0.793	0.503	0.552	0.619	0.035

a Man-in-the-browser attack. The connection details are legitimate. The amount transfer is high. The transfer happens no later than ten minutes from a legitimate one. The **Persistent** scenario simulates the infection of a banking Trojan [38]. The frauds have a low amount, and the connection details are legitimate. The **Mix** scenario combines all previous scenarios.

Table 1 compares the proposed model against the baselines algorithms. The proposed model is better in almost all scenarios. The lower performance of baseline algorithms is due to concept drift. The overall low FPs, demonstrate how the proposed approach can correctly model (i.e., it does not negatively impact) the user's "spending pattern". Traditional algorithms have a hard time detecting frauds that have not been seen before. The baseline algorithms' performance is higher in the scenarios that are more similar to the training set and lower in the most complex scenarios as the Persistent or the Mix scenario.

Figure 3 shows the ROC curves of each model. All the models have been trained in identical conditions, i.e., using the same dataset. Regarding the persistent (a) and mix (b) scenarios, the proposed model is better and presents an overall lower FPR. The curve is similar in both cases due to the persistent frauds. Also, it is possible to distinguish two groups of frauds in these scenarios. The ones easy to detect, which correspond to the first ramp, and the hard ones belong to the second ramp. In the stealing (c) and hijacking (d) scenarios, almost all the models perform similarly. The proposed model works better than baseline algorithms for very low values of FPR. Because of that, the proposed model has a higher AUROC.



(a) Persistent

(b) Mix

(c) Stealing

(d) Hijacking

Figure 3: ROC curves in different scenarios

4.2. Grey Box Attacks

Grey box attacks consist of attacks performed by an attacker with information about the target system, as described in [36]. The attacker has acquired (e.g., thanks to a malware infection or leak) all the transactions from December 2012 to September 2013. The attacker uses the available data to train a Random Forest classifier that is treated as an oracle. The attacker will try the fraud against the oracle before injecting it into our model. If the oracle flags the transaction as fraud, the attacker will change the transaction and will try again. Table 2 summarizes the result obtained for each model in the grey box scenarios. The proposed model outperforms by far the baseline algorithms. As expected, the Random Forest is the worse since it is used as an oracle by the attacker. The results also show the benefits of exploiting different approaches to fraud detection. The proposed model suffers less from the grey box attack because it is based on user modeling, while the oracle and baseline algorithms rely on finding a separation plane between frauds and legitimate transactions.

4.3. Generative Versus Discriminative Approach

The proposed model is generative: It generates the expected transaction probabilities, which are then used to detect fraud. However, there are also discriminative models: they are trained to discriminate between frauds and legitimate transactions. Random Forest or OC-SVM are examples of discriminative models. Following the architecture of the generator, a discriminative model is proposed here. The differences between the two models are in the last layers of

Table 2
Grey-box attacks scenario results

Model	Precision	Recall	AUROC	AUPR	MCC	F1	FPR
Proposed	**0.769**	**0.556**	**0.867**	**0.691**	**0.606**	**0.645**	**0.020**
Random F.	0.548	0.096	0.541	0.237	0.182	0.163	0.026
Isolation F.	0.094	0.044	0.483	0	-0.047	0.060	0.033
OC-SVM	0.146	0.084	0.497	0	-0.007	0.107	0.042
k-NN	0.141	0.075	0.496	0	-0.011	0.098	0.035

Table 3
Comparison between Discriminator and Generator approaches

Scenario	Model	Precision	Recall	AUROC	AUPR	MCC	F1
Persistent	Generator	**0.751**	0.499	0.834	**0.652**	**0.559**	**0.600**
	Discriminator	0.549	**0.509**	**0.886**	0.578	0.446	0.528
Stealing	Generator	**0.869**	**0.986**	**0.997**	**0.987**	**0.910**	**0.924**
	Discriminator	0.674	0.869	0.972	0.896	0.799	0.759
Hijacking	Generator	**0.864**	**0.968**	**0.978**	**0.922**	**0.897**	**0.913**
	Discriminator	0.672	0.862	0.966	0.882	0.779	0.755
Mixed	Generator	**0.828**	**0.833**	0.936	**0.871**	**0.801**	**0.830**
	Discriminator	0.622	0.743	**0.938**	0.800	0.690	0.677

the model and the training. The discriminative model does not have the Predictor nor the Dissimilarity score shown in Figure 1. Regarding training, the generator is trained for prediction, while the discriminator is trained for classification. The discriminative model uses cross-attention instead of self-attention and uses the incoming transaction as a query to search in the past transactions. The results are reported in Table 3. Overall, the generative approach performs better than the discriminative one.

4.4. LSTM-based Model Comparison

Banking information is not public due to obvious reasons. Due to the lack of standard databases and benchmarks, it is not easy to compare different models. Therefore, we have developed an ensemble model composed by LSTM, Random Forest, and XGBoost based on the model proposed by Jurgovsky et al. [13]. To combine the three models' output, we compute the Cumulative Distribution Function of the exponential distribution applied to each model output y, which gives the probability k of the input sample. Then, the ensemble gives the final decision to the model that has the maximum distance between k and the mean CDF value seen in training. The ensemble improves the performance of each of the individual models. The improvement over the LSTM model is 22% in terms of AUROC. We used the same dataset to train both models. To provide an accurate comparison, the Random Forest's performance is set as a baseline, and the models are compared in terms of improvement. Table 4 summarizes the results for the mix scenarios, where all types of frauds are considered. The improvements of both models over the baseline are close. Overall, the proposed model performs slightly better than the ensemble

Table 4
Comparison between LSTM-based approach and the proposed model.

Model	Precision	AUROC	AUPR	MCC	F1	FPR
LSTM Ensemble	0.422	0.975	0.0.872	0.588	0.583	0.145
Baseline	0.505	0.961	0.803	0.634	0.649	0.098
Improvement	-0.083	0.014	0.069	-0.046	- 0.066	0.046
Proposed model	0.828	0.936	0.871	0.801	0.830	0.006
Baseline	0.902	0.871	0.703	0.800	0.823	0.015
Improvement	**-0.074**	**0.065**	**0.168**	**0.001**	**0.007**	**-0.009**

one. It is also important to notice that the proposed model can deal with users with very few transactions, while the LSTM-based ensemble needs at least 50 transactions per user.

5. Conclusions

This paper presented a novel approach for fraud detection based on the Transformer model, a state-of-the-art technique in the Natural Language Processing field. Besides demonstrating the feasibility of applying NLP techniques to model the user's spending behavior, we showed how the proposed model overcame the domain's limitations and achieved better performance than state-of-the-art algorithms in complex scenarios and against adversarial attacks. Future works will work in the direction of providing explainability to the proposed framework, which is of paramount importance in the fraud detection domain. In light of the results obtained, we deem that the future of banking fraud detection is standardization and transfer learning. The field needs standard benchmarks like ImageNet [39] for image classification or GLUE [40] for NLP models. It is a daunting task and will require open datasets, which are difficult to obtain because of the data's sensitivity. Withal, it would bring enormous advantages. Updated models could be fine-tuned for specific tasks in a matter of days, allowing financial institutions and researchers to reach new frontiers.

References

[1] Markets, Markets, Market research report, https://www.researchandmarkets.com/, 2018.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, arXiv e-prints (2017) arXiv:1706.03762. arXiv:1706.03762.

[3] C. Yun, S. Bhojanapalli, A. Rawat, S. Reddi, S. Kumar, Are transformers universal approximators ofsequence-to-sequence functions? (2019).

[4] A. Brown, A. Tuor, B. Hutchinson, N. Nichols, Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection, arXiv e-prints (2018) arXiv:1803.04967. arXiv:1803.04967.

[5] G. Branwen, Gpt-2 folk music, 2020. URL: https://www.gwern.net/GPT-2-music.

[6] S. Alexander, https://slatestarcodex.com/2020/01/06/a-very-unlikely-chess-game/, 2020. URL: https://slatestarcodex.com/2020/01/06/a-very-unlikely-chess-game/.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners (2019).

[8] Y. Kunlin, A memory-enhanced framework for financial fraud detection, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 871–874. doi:10.1109/ICMLA.2018.00140.

[9] L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. non-handcrafted features for computer vision classification, Pattern Recognition 71 (2017) 158 – 172. URL: http://www.sciencedirect.com/science/article/pii/S0031320317302224. doi:https://doi.org/10.1016/j.patcog.2017.05.025.

[10] L. Cai, J. Zhu, H. Zeng, J. Chen, C. Cai, Deep-learned and hand-crafted features fusion network for pedestrian gender recognition, in: J. Cao, E. Cambria, A. Lendasse, Y. Miche, C. M. Vong (Eds.), Proceedings of ELM-2016, Springer International Publishing, Cham, 2018, pp. 207–215.

[11] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, arXiv e-prints (2020) arXiv:2003.10555. arXiv:2003.10555.

[12] R. Chalapathy, S. Chawla, Deep Learning for Anomaly Detection: A Survey, arXiv e-prints (2019) arXiv:1901.03407. arXiv:1901.03407.

[13] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, O. Caelen, Sequence classification for credit-card fraud detection, Expert Systems with Applications 100 (2018) 234 – 245. URL: http://www.sciencedirect.com/science/article/pii/S0957417418300435. doi:https://doi.org/10.1016/j.eswa.2018.01.037.

[14] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, P. Beling, Deep learning detecting fraud in credit card transactions, in: 2018 Systems and Information Engineering Design Symposium (SIEDS), 2018, pp. 129–134. doi:10.1109/SIEDS.2018.8374722.

[15] B. Wiese, C. Omlin, Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 231–268. URL: https://doi.org/10.1007/978-3-642-04003-0_10. doi:10.1007/978-3-642-04003-0_10.

[16] M. Zamini, G. Montazer, Credit card fraud detection using autoencoder based clustering, in: 2018 9th International Symposium on Telecommunications (IST), 2018, pp. 486–491. doi:10.1109/ISTEL.2018.8661129.

[17] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, K. Li, Aiˆ 2: training a big data machine to defend, in: 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), IEEE, 2016, pp. 49–54.

[18] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, SIGMOD Rec. 29 (2000). URL: https://doi.org/10.1145/335191.335437. doi:10.1145/335191.335437.

[19] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, Proceedings of the Sixth European Conference on the Principles of Data Mining and Knowledge Discovery

2431 (2002) 15–26. doi:10.1007/3-540-45681-3_2.
[20] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, M. E. Houle, On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study, 2016. URL: https://doi.org/10.1007/s10618-015-0444-8. doi:10.1007/s10618-015-0444-8.
[21] B. Lamrini, A. Gjini, S. Daudin, F. Armando, P. Pratmarty, L. Travé-Massuyès, Anomaly detection using similarity-based one-class svm for network traffic characterization, 2018.
[22] R. Chalapathy, A. K. Menon, S. Chawla, Anomaly detection using one-class neural networks, 2019. arXiv:1802.06360.
[23] L. Ruff, R. Vandermeulen, N. Görnitz, L. Deecke, S. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, 2018.
[24] M. Carminati, R. Caron, F. Maggi, I. Epifani, S. Zanero, Banksealer: A decision support system for online banking fraud analysis and investigation, Computers & Security 53 (2015) 175 – 186. URL: http://www.sciencedirect.com/science/article/pii/S0167404815000437. doi:https://doi.org/10.1016/j.cose.2015.04.002.
[25] M. Carminati, R. Caron, F. Maggi, I. Epifani, S. Zanero, Banksealer: An online banking fraud analysis and decision support system, in: N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. Abou El Kalam, T. Sans (Eds.), ICT Systems Security and Privacy Protection, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 380–394.
[26] M. Carminati, M. Polino, A. Continella, A. Lanzi, F. Maggi, S. Zanero, Security evaluation of a banking fraud analysis system, ACM Transactions on Privacy and Security (TOPS) 21 (2018) 1–31.
[27] M. Carminati, A. Baggio, F. Maggi, U. Spagnolini, S. Zanero, Fraudbuster: temporal analysis and detection of advanced financial frauds, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2018, pp. 211–233.
[28] M. Carminati, L. Valentini, S. Zanero, A supervised auto-tuning approach for a banking fraud detection system, in: Cyber Security Cryptography and Machine Learning, CSCML 2017, Springer International Publishing, 2017, pp. 215–233.
[29] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, 2015. arXiv:1508.07909.
[30] M. Schuster, K. Nakajima, Japanese and korean voice search, 2012.
[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013. arXiv:1301.3781.
[32] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014. arXiv:1409.0473.
[33] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, 2015. arXiv:1508.04025.
[34] J. Rocca, Ensemble methods: bagging, boosting and stacking, https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205, 2019.
[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014. arXiv:1412.6980.
[36] M. Carminati, L. Santini, M. Polino, S. Zanero, Evasion attacks against banking fraud detection systems, in: 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020), 2020, pp. 285–300.
[37] A. Erba, R. Taormina, S. Galelli, M. Pogliani, M. Carminati, S. Zanero, N. O. Tippenhauer,

Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems, in: ACSAC '20: Annual Computer Security Applications Conference, Virtual Event / Austin, TX, USA, 7-11 December, 2020, ACM, 2020, pp. 480–495. URL: https://doi.org/10.1145/3427228.3427660. doi:10.1145/3427228.3427660.

[38] A. Continella, M. Carminati, M. Polino, A. Lanzi, S. Zanero, F. Maggi, Prometheus: Analyzing webinject-based information stealers, Journal of Computer Security 25 (2017) 117–137.

[39] S. V. Lab, Imagenet, http://www.image-net.org/, 2020.

[40] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Glue: A multi-task benchmark and analysis platform for natural language understanding, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355. URL: https://www.aclweb.org/anthology/W18-5446. doi:10.18653/v1/W18-5446.

[41] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, BMC Genomics 21 (2020). doi:10.1186/s12864-019-6413-7.

A. Metrics

Fraud Detection is a classification problem in which the classes are very unbalanced. Therefore, the usual classification metrics, such as accuracy, could lead to wrong conclusions. For example, a model predicting that all the transactions are legitimate in a dataset containing 1% of frauds have a 99% of accuracy but will not detect frauds. Hence, we propose the use of the following metrics, more appropriate to assess the quality of a model in an unbalance scenario as fraud detection

A.0.1. Precision and Recall

Precision measures the number of predicted frauds that are actually frauds. It is similar to the accuracy of the positive class.

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

Recall measures how many frauds are detected from the total number of frauds. Provides an indication of missed frauds.

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

Both metrics are related. Often, increases in one metric imply decreasing the other.

A.0.2. Curves

There are two curves:

- Receiver operating characteristic, or ROC, shows the behaviour of the system for different thresholds.
- Precision-Recall, or PR, shows the tradeoff between precision and recall.

ROC curve relates True Positive Rate with False Positive Rate. Given a desired FPR, the model is better as higher the TPR is. A common metric to measure the quality of the curves is the area under the curve or AUC. Higher values indicate better models.

A.0.3. Matthews correlation coefficient

Also called phi coefficient, Matthews correlation coefficient measures the correlation between the observed and predicted classification.

As shown in Equation A.0.3, MCC takes into account all the metrics given by the confusion matrix. It is regarded as one of the most reliable statistics for imbalance problems [41].

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

A.0.4. F-score

F-score is a statistical accuracy measure. It is calculated from the Precision and Recall, as shown in Equation 6.

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall} \tag{6}$$

β is a weighting parameter. In our case, we use the F_1 score, which is the same as the harmonic median between Precision and Recall.

A.0.5. False Positive Rate

The False Positive Rate, also known as fall-out, is the ratio between the number of misclassified negative samples and all negatives samples. Equation 7 shows its calculation, FP is the number of False Positive, and TN the number of True Negatives.

$$FPR = \frac{FP}{FP + TN} \tag{7}$$