2019 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, OCT. 13–16, 2019, PITTSBURGH, PA, USA

INTERPRETABLE ONLINE BANKING FRAUD DETECTION BASED ON HIERARCHICAL ATTENTION MECHANISM

Idan Achituve *Sarit Kraus* *Jacob Goldberger*

Bar-Ilan Univesity, Israel

ABSTRACT

Online banking activities are constantly growing and are likely to become even more common as digital banking platforms evolve. One side effect of this trend is the rise in attempted fraud. However, there is very little work in the literature on online banking fraud detection. We propose an attention based architecture for classifying online banking transactions as either fraudulent or genuine. The proposed method allows transparency to its decision by identifying the most important transactions in the sequence and the most informative features in each transaction. Experiments conducted on a large dataset of real online banking data demonstrate the effectiveness of the method in terms of both classification accuracy and interpretability of the results.

***Index Terms*—** deep learning, attention, interpretability, Online banking, fraud detection

1. INTRODUCTION

In recent years there has been a continuous increase in online banking transactions. According to a recent study [1], global non-cash transaction volumes grew by 10.1% in 2016 to 482.6 billion and it is estimated to accelerate at a compound annual growth rate of 12.7% globally between 2016 and 2021. Online banking transactions are likely to become even more commonplace in the near future, as more financial services adopt digital banking platforms [2]. The increase in online banking activity seems to has gone hand in hand with online banking fraud. As reported in [3], annual online banking fraud losses in the UK almost doubled from 63.7 million British pounds in 2010 to 121.4 million in 2017.

Traditional approaches to fraud detection systems are rule based [4]. Rule based systems cannot be easily adapted to new fraud patterns since they need constant manual updates. Therefore, in recent years machine learning based techniques have been applied to online banking fraud detection. This research direction, however, is understudied in the literature. Most of the solutions we found describe a system based on smart domain expert features [5, 6, 7] or apply a naive assumption on the sequence of transactions [8]. To carry out an online payment transaction, a fraudster must, at the very least, log in to the bank system first. The login and the payment are two different transactions that must be done in that order. That is, even this simple process adheres to a natural order of events. Although most attempts at fraud are more complex, the majority of current approaches concentrate on data from the latest transaction or most recent transactions while relying on hand-engineered features that will hopefully catch global dependencies.

In this study we focus on real-time fraud detection of online banking transactions (e.g., logins, payments, view statements). The purpose of a fraud detection solution in this setting is to assess in real-time the risk of each individual transaction in the form of a fraud probability. Then the bank may choose to allow the transaction, deny the transaction or impose some form of authentication on the user upon successful completion the transaction will be allowed. We address this problem by formulating it as a sequence classification task in which the classifier has visibility to long sequences of transactions made by the user, and is able to extract global dependencies from it. A common solution for these kind of tasks are recurrent neural networks, and more specifically, a Long Short-Term Memory (LSTM) architecture. LSTM networks have gained a great deal of popularity in sequence classification tasks leading to state-of-the-art results in various fields. However, it is very difficult to interpret their decisions. Here, we take a different approach, we propose an attention based classifier. The attention mechanism enables a sequence-based neural network to automatically focus on the data item that is most relevant to the classification task by a data-driven weighted average of local information found in each term of the sequence.

A recent study [9] introduced an attention architecture for machine translation with state of the art results based solely on attention which entirely eliminated reliance on convolutional or recurrent networks. Inspired by this research trend we propose a fraud detector that relies solely on attention, without using any recurrent networks to process the sequences. We use a hierarchical form of attention [10], that is applied on both a single transaction for better feature representation and the entire sequence to identify the most relevant transactions. Our solution is simple yet powerful, can be easily trained and efficiently implemented in real-time systems. In addition, a major advantage of our attention mechanism is its ability to explain the fraud detection decision by the specific features and the most suspicious transactions. The main novel contribution of the proposed fraud detection method therefore is two-fold:

1. An attention based network that efficiently integrates cues from a sequence of transactions into a global fraud decision yielding improved detection results.
2. Interpretability - The decision made by our system can be explained in comprehensible to users terms.

The proposed method was applied to a large dataset of real bank transactions and we demonstrate its improved performance compared to different methods and its interpretability capabilities.

2. RELATED WORK

Online banking fraud detection is understudied in the literature. We found few papers on this topic, perhaps because of data privacy, even though most of today's banks employ some kind of fraud detection module. Wei et al. [5] presented a complete system in which data are pre-processed and then classified by combining 3 models: contrast pattern mining, cost sensitive neural network and decision

978-1-7281-0824-7/19/$31.00 ©2019 IEEE

forest. Kovach and Ruggiero [6] combined local behavior changes on the user level with global evidence across all users to generate a risk score for a transaction. Their system requires that the end user download and install a component to have good device identification, which could be a nuisance. Carminati et al. [7] suggested taking a semi-supervised approach to rank users' transactions so they could be inspected more efficiently by an analyst. All these methods classify based on a single transaction while relying on domain-expert features. However, none look at a sequence of transactions to make a decision. In [8] the use of Hidden Markov Models (HMM) was suggested for fraud detection. HMM assumes a first order Markovian property on the hidden states which is not true in our case since fraudulent transactions may be interleaved in genuine transactions.

Online credit card fraud and intrusion detection are somewhat similar domains that have received more attention in the literature. They are similar to the online banking use case in the sense that they also deal with a highly imbalanced class distribution in which classifying the minority class correctly has greater importance. However, they may differ from the online banking use case in the amount of data per user, the data richness or the variability in procedures.

Similar to the online banking case, several studies in the credit card domain [11, 12, 13] have put forward techniques to create powerful features using aggregations over time windows that could be used by classifiers such as random forest, SVM or logistic regression. In recent years several studies have addressed the fraud detection problem as a sequence classification. Srivastava et al. [14] used HMM over tokens representing spending profiles of cardholders and Heryadi and Hendric [15] compared the performance of CNN, LSTM and stacked LSTM, after applying PCA on the input sequences. Both methods rely heavily on the fact that all transactions have an amount attached to them; however, in our case not all transactions involve money. Li et al. [16] generated 3 sets of features: artificial features based on domain knowledge, latent combinations between features using gradient tree boosting and sequential features using GRU. They used all sets of features as input to a random forest classifier. Jurgovsky et al. [17] integrated several feature engineering strategies to generate representations of transactions. Using these representations, they suggested classifying credit-card sequences of transactions with LSTM. Wang et al. [18] developed a system for classifying sessions with LSTM as either fraudulent or genuine based on data generated from user clicks.

In the intrusion detection domain, several studies have utilized sequences of user logs for classification. Yuan et al. [19] suggested using LSTM to extract temporal features from sequences of user actions and then using these features as input to a CNN. Because CNN expects a fixed size input they ignored short sequences and trimmed long sequences. In our case this kind of sequence processing could seriously damage performance. In both [20, 21] an unsupervised approach was taken using RNNs to rank events or users for analyst inspection. Tuor et al. [20] combined continuous features generated from aggregations over time with categorical features to train an auto-encoder for detecting anomalies online, while in [21] the problem was formulated as a language model. The authors described a 2-tier approach that considers individual log-lines and users' actions over time to generate scores for all events in a single day.

3. ATTENTION BASED FRAUD DETECTION SYSTEM

The fraud detection network proposed here is made up of two main components. The first component involves embedding of categorical features in a continuous space and combining the features into a single vector using an attention mechanism. The second component is responsible for the actual fraud detection that is carried out by sequence level attention. We next describe the input structure and then explain the network components.

3.1. Sequence Definition

Let $r = \{f_1, f_2, ..., f_k\}$ be a transaction that is represented by k categorical features consisting of either raw transaction data or generated by a domain specialist. Each feature value is a token from a pre-defined set of possible feature values. We define a sequence of m transactions made on the same account as $S = \{r_1, r_2, ..., r_m\}$. The transactions in the sequence are ordered according to their time of execution and are generated over a fixed period of time from the last transaction in the sequence. That is, a sequence is defined as all the transactions made on the same account in a fixed time window from the last transaction (e.g., 1 day, 7 days, 30 days). Note that by defining sequences in this manner allows us to address an online detection scenario in which the system evaluate how risky is the last transaction based on data available from all transactions in the sequence. The goal is finding the sequence binary class (Fraud or Genuine). We use the abbreviation F or G to denote the two classes.

3.2. Network Architecture

Dense Feature Representation Given a sequence of recorded transactions, we first move from categorical features to a dense representation of each feature value. Let $r = \{f_1, f_2, ..., f_k\}$ be a transaction where f_i is a one-hot representation of a categorical feature value. We apply an embedding matrix M_i on f_i to represent it by a d-dimensional dense feature vector $e_i = M_i f_i$. The matrices $M_1, ..., M_k$ are learned in the training phase. The choice of d is a hyper parameter of the model and both d and k are fixed across all features and transactions in the database respectively.

Attention over Features In order to represent a transaction as a single vector, we can either concatenate the features or average them. The former is linearly increasing with the number of features and thus can be computationally expensive. Hence, we prefer the latter approach. In addition, we want to determine which features were the most influential to the final decision. Therefore, we employed attention over the features, which provides this property. The data driven weight of each feature e_i is computed as follows:

$$\alpha_i = \frac{\exp(w^\top \cdot g(e_i))}{\sum_{j=1}^{k} \exp(w^\top \cdot g(e_j))} \tag{1}$$

where g is a non-linear function implemented by a neural network and the vector w is a parameter that implements the attention mechanism. A given transaction is finally represented by a weighted average of the input features:

$$x = \sum_{i=1}^{k} \alpha_i \cdot e_i \tag{2}$$

Attention over Transactions Given a transaction sequence $S = (x_1, ..., x_m)$, our goal is to find its class y that can be either fraudulent or genuine (F or G). We first apply a local decision network on each transaction separately:

$$p(y = F|x_t) = 1 - p(y = G|x_t) = \sigma(h(x_t)) \tag{3}$$

such that h is a non-linear function of x that is implemented by a neural network and σ is the sigmoid function.

A naive way to integrate the information from all of the transactions in the sequence S is to average their local decisions:

$$p(y=F|S)=\frac{1}{m}\sum_{t=1}^{m}p(y=F|x_t) \tag{4}$$

Alternatively we can weight the decisions using a fixed distribution under the reasonable assumption that later transactions in the sequence are more important. Let z be a (hidden) random variable that represents the transactions that convey the most relevant information for decision making. We sample the transaction location z using exponentially decaying weights:

$$p(z=t)=\frac{\exp(b^{t-m})}{\sum_{j=1}^{m}\exp(b^{j-m})}, \quad t=1,...,m \tag{5}$$

such that $b>1$ is the parameter of the exponential time decay. The final fraud detection decision is:

$$p(y=F|S)=\sum_{t=1}^{m}p(z=t)\cdot p(y=F|x_t) \tag{6}$$

In this study, we propose a data driven approach for weighting the local decisions. That is, the importance of the transactions in a sequence is extracted from the data itself. We can view the model as a two-step process that produces a F/G decision given an input sequence S. We first use an attention network component to select a transaction location z and then use the selected transaction to obtain the F/G decision. The attention selection is done as follows:

$$p(z=t|S)=\frac{\exp(u^\top\cdot l(x_t))}{\sum_{j=1}^{m}\exp(u^\top\cdot l(x_j))}, \quad t=1,...,m \tag{7}$$

such that l is a non-linear function that is implemented by a neural network and u is the attention parameter. The fraud decision is finally obtained as a data driven average of the transaction level decisions:

$$p(y=F|S)=\sum_{t=1}^{m}p(z=t|S)p(y=F|x_t) \tag{8}$$

Learning the Model Parameters We next describe the training procedure. Assume we are given n transaction sequences $S_1,...,S_n$ with corresponding binary labels $y_1,...,y_n\in\{F,G\}$. Each sequence S_i is composed of m_i consecutive transactions $(x_{i,1},...,x_{i,m_i})$. Denote by θ the parameter set of the network (the dense feature representation, the feature level attention and the sequence level attention). The log-likelihood function of the model parameters θ is:

$$L(\theta)=\sum_{i=1}^{n}\log p(y_i|S_i;\theta)=\sum_{i=1}^{n}\log(\sum_{t=1}^{m_i}p(z_i=t|S_i)p(y_i|x_{i,t})) \tag{9}$$

where,

$$p(y_i|x_{i,t})=\begin{cases}\sigma(h(x_{i,t})), & y_i=F\\ 1-\sigma(h(x_{i,t})), & y_i=G\end{cases}$$

and $p(z_i=t|S_i)$ is defined in Eq. (7). To find the network parameters we can maximize the likelihood function using the standard back-propagation network training algorithm. The fraud detection algorithm is summarized in Algorithm 1.

Algorithm 1 Attention based fraud detection algorithm.

Input: A transaction sequence $S=(r_1,...,r_m)$ where each transaction is composed of k features: $r_t=(f_{t1},...,f_{tk})$.

Output: Fraud/Genuine classification

Transaction Level Processing:

- Feature embedding:

$$e_{ti}=M_i f_{ti}, \quad i=1,..,k, \quad t=1,..,m$$

- Feature level attention:

$$\alpha_{ti}=\frac{\exp(w^\top\cdot g(e_{ti}))}{\sum_{j=1}^{k}\exp(w^\top\cdot g(e_{tj}))}$$

$$x_t=\sum_{i=1}^{k}\alpha_{ti}\cdot e_{ti}$$

Sequence Level Decision:

- Transaction level decisions:

$$p(y=F|x_t)=\sigma(h(x_t)), \quad t=1,...,m$$

- Sequence level attention:

$$p(z=t|S)=\frac{\exp(u^\top\cdot l(x_t))}{\sum_{j=1}^{m}\exp(u^\top\cdot l(x_j))}$$

- Weighted averaging of local decisions:

$$p(y=F|S)=\sum_{t=1}^{m}p(z=t|S)p(y=F|x_t)$$

4. EXPERIMENTS & RESULTS

4.1. Dataset

Our method was tested on 6 months of data from 2017 belonging to a South American bank. In that period there were 26.1 million transactions. 22,500 of those transactions were marked by a bank's analyst as fraud and the rest were genuine transactions. We generated a total of 57 categorical features by taking fields from the raw data, which are, for the most part, categorical by nature (e.g., browser type, operating system). Non-categorical features such as payments amounts were discretized manually. In order to incorporate the transaction time, we extracted from the time-stamp the minute, the hour of the day, the day of the week and the week of the month. We replaced features with high domain space such as IP with alternative data such as geo-location information. In addition, automatic predefined banks policies sometimes deny the execution of transactions or oblige the user to pass an authentication measure in order to complete the transaction successfully. Therefore, to incorporate this information when we generated the sequences we erased this information from the final transaction of the sequence to which we need to give a risk assessment. Finally, Missing values were replaced with a special token.

We generated sequences by taking all of the transactions executed 7 days prior to each transaction for each user to simulate online detection scenario. As a result, the generated sequences were of

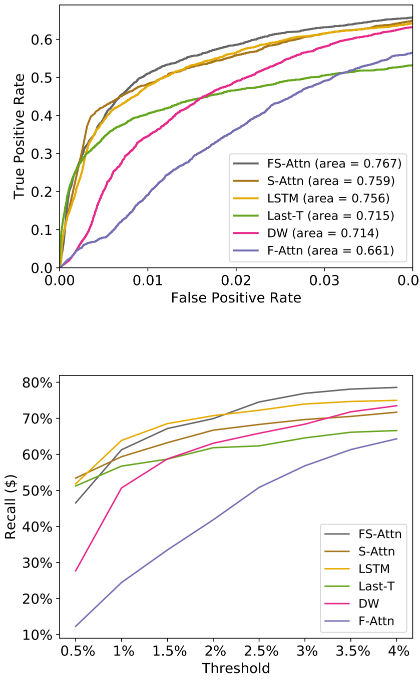


Fig. 1: Area under the top 4% FPR of the ROC curve (top) and recall in US dollars presented in percentages on 8 thresholds (bottom).

varying lengths with average of 28 frames. We split the data to train, validation and test according to the time of the last transaction in each sequence. All sequences that ended in the first 4 months were taken to be the training set, all sequences that ended in the following 1 month were taken to be the validation set and all the sequence that ended in the final month were taken to be the test set.

4.2. Compared Classifiers

In our experiments we compared the following approaches. The first three methods either used only the last transaction or a simple sequence averaging. The last three used a more sophisticated data driven sequence processing:

- **Last Transaction (Last-T)** Attention on the features and using only the last transaction in the sequence.
- **Decaying Weight (DW)** Attention on the features and a fixed weighted averaging of the sequence items with decaying parameter $b = 1.5$ (see Eq. (6)).
- **Features Attention (F-Attn)** Attention over the features and unweighted averaging of the items in the sequence.
- **LSTM** Attention on the features and using LSTM to process sequences of transactions for a binary F/G decision.
- **Sequence Attention (S-Attn)** Attention over the transactions in the sequence and unweighted feature averaging.
- **Features and Sequence Attention (FS-Attn)** Our proposed method of applying attention on both the features and the transactions.

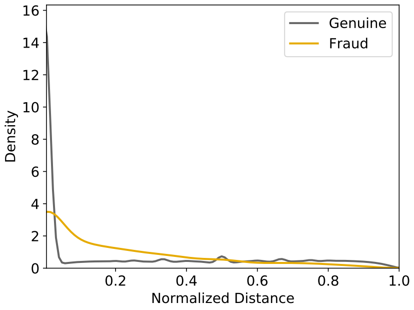
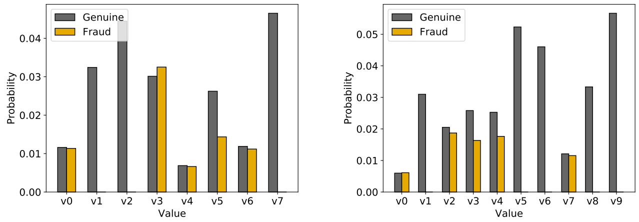


Fig. 2: Distribution of the location of the A-transaction for each class.



(a) Operating System

(b) Browser Type

Fig. 3: Mean attention weights assigned to each value of the features (a) operating system and (b) browser type at the A-transaction.

4.3. Evaluation Metrics

In imbalanced settings, a popular evaluation metric is the area under the Receiver Operating Characteristic (ROC) curve. Banks often set a probability threshold above which a user's transaction may be stopped. By doing so banks can maintain a friendly user experience and a low workload that is expressed in a small number of cases for analyst inspection. Therefore, we chose the main evaluation metric to be the area under a portion of the ROC curve, which in our use case was the standardized area on the top 4% false positive rate (FPR), a threshold used currently by the bank.

In addition, since the ultimate goal is to prevent monetary loss to the banks as a result of transactions with money involved, we suggest another metric that compares the recall in money values. Similar to the previous metric we evaluated the model performance based on the recall of the top 4% scored transactions.

4.4. Implementation Details

The feature level attention was obtained by setting w as a parameter vector and modelling $g(e_{ti})$ by a 2 layer NN. The transaction level decision function, $h(x_t)$, was modeled by a 6 layer NN with residual connection on each of the hidden layers. The sequence level attention was obtained by modeling u as a 2 layer NN and by taking the 4th layer of $h(x_t)$ to be $l(x_t)$. All layers and vectors were set to size 32 apart from the first 2 hidden layers and the final hidden layer of the transaction level NN that were set to 64 and 16 accordingly.

All the parameters were initialized using Glorot initialization [22] and were updated using the ADAM [23] optimizer with $\beta_1 =$

Table 1: Use cases

Seq. Class	Highest Weighted Transactions	Highest Weighted Features	Details
F	6,8,9	Time features, transaction type, amount, beneficiary	**True positive**, The last transactions in the sequence were all payments attempts committed by the fraudster. All the transactions were executed in a short period of time, the amount at each transaction was slightly different, and the beneficiary was the same in most of them.
F	8	Change information, device details	**True positive**, The fraudster tried to fool the system by changing some of the user's personal information in the 8^{th} transaction. In addition, in both the 8^{th} transaction and the last transaction the fraudster connected from a device that differed from previous devices of the user.
F	10	Location, device details	**False Negative**, The sequence extended across several days and only the last transaction was a fraud attempt. The information in the last transaction was similar to information in previous transactions (e.g., device elements, location) and there wasn't anything unusual in the sequence or in the last transaction. Therefore we speculate that the system considered the sequence as genuine.
G	7-9	Transaction status	**False Positive**, All transactions were made by the actual user, however, because of predefined bank policies transactions 7-9 were not allowed to complete. We speculate that because it is a suspicious pattern the network assigned a high score to the sequence indicating a possible fraud attempt.

0.9, $\beta_2 = 0.999$ and a learning rate of 0.0003. We enforced regularization by a weight decay of 0.002. Since we are dealing with a highly imbalanced class distribution, in order to prevent the classifiers from being biased towards the genuine class, during training at each epoch we balanced the dataset so there would be an equal number of fraud sequences and genuine sequences. In addition, since most transaction are genuine and there is a small number of fraudulent transactions, at each epoch we used all of the fraud sequences in the training set while sampling random genuine sequences from the training set each time. We trained our model on the GPU GeForce GTX 1080 Ti for 80 epochs until convergence and chose the model that had the highest area under the top 4% of the ROC curve on the validation set. The total training time was 20 hours.

4.5. Fraud Detection Results

Fig. 1 presents the fraud detection results on the test set according to the evaluation metrics defined above. We can first observe that later transactions in the sequence are more relevant to the classification decision. The area under the ROC curve (AUC) of the DW and Last-T classifiers are higher than the F-Attn. The Last-T and the DW performance is similar with a slight advantage to the Last-T, mainly because of the better results in the higher regions.

Our second observation is that when modeling the entire sequence with adaptive classifiers, either by using transaction attention or by using LSTM, there is an improvement in the performance. It can be seen that the LSTM, S-Attn and FS-Attn achieve a better AUC by a substantial margin compared to the other classifiers and prevent larger monetary loss on almost all thresholds.

Finally, it can be seen that the attention based classifiers surpassed the LSTM according to the AUC metric, where the highest AUC was achieved by our proposed method, the FS-Attn model. The FS-Attn and the LSTM prevent the largest amount of monetary loss on almost all thresholds. On the first 3 thresholds the LSTM was better by a small margin, whereas on the other thresholds the FS-Attn was the same or better. When looking on the 4% threshold, the FS-Attn model prevented almost 80% of the monetary loss from frauds, which corresponds to hundreds of thousands of dollars.

4.6. Attention Mechanism Analysis

The most important advantage of our hierarchical attention method (apart from performance) is its ability to explain how the decision was made. This interpretability ability is due to the attention mechanism that localizes the decision to the relevant transaction and to the relevant features within that transaction. We first analyze the transaction attention mechanism. Denote the transaction with highest attention weight by A-transaction. The index of the A-transaction in a sequence $S = (x_1, ..., x_m)$ is:

$$\text{index} = \arg\max_t p(z = t|S) = \arg\max_t (u^\top l(x_t)) \qquad (10)$$

Fig. 2 presents the A-transaction location distribution for the two classes in the test set. We normalized the index locations to [0,1] (where 0 corresponds to the most recent transaction) by dividing them in the sequences length. The graph shows that in genuine sequences the attention is prone to give the highest weight to the last transaction in the sequence whereas in fraud sequences the distribution is smoother, indicating that earlier transactions in the sequence were the trigger for the fraud detection.

We next analyze the feature attention. We collected the feature attention distributions (Eq. (1)) at the A-transaction and for every feature i we calculated the average feature attention probability per each value v according to:

$$\overline{\alpha_{iv}} = \frac{1}{n_v} \sum_{t=1}^{n} \alpha_{ti} \cdot \mathbb{1}\{f_{ti} = v\} \qquad (11)$$

where n is the total number of sequences in the test set, n_v is the number of sequences where the value of feature i in the A-transaction is v, α_{ti} is the attention allocated to the i^{th} feature in the A-transaction of the t^{th} sequence and f_{ti} is the corresponding feature value. Fig. 3 depicts the average feature attention distribution for the features operating system and browser type. It is interesting to notice that feature values which are unique to genuine sequences tend to get higher probability, indicating that the feature was more significant in these kind of sequences while values that are not unique to either class tend to get lower probabilities without much difference between the classes.

Finally, In Table 1 we demonstrate the ability to explain our model decisions by the most important transactions and features. We present 4 sequences, each one made of 10 transactions. The first 2 examples were fraudulent attempts that were rightfully allocated a high score and in the last 2 examples our classifier was wrong. We explain the decision by the weights assigned to the features in the highest weighted transactions. We numbered the transactions in the sequence according to their order such that 1 is the oldest transaction and 10 is the most recent transaction.

5. CONCLUSION

In this study we proposed modeling the online banking fraud detection problem as a sequence classification. We developed a classifier that applies attention over the features and attention over the transactions without any use of recurrent or convolutional connections. Our model is simple, yet powerful and makes it possible to interpret its decisions. We demonstrated our method on real data from a South American bank. We showed that there is substantial benefit in attention based models that dynamically assign weights to transactions compared to models with constant weights and we showed how our model outperformed LSTM; a common approach for sequence classification. Finally, we analyzed the attention mechanism of our network and demonstrated the effectiveness of our classifier's ability to spot the most relevant features and the most relevant transactions for its final decision.

6. REFERENCES

[1] "World payments report 2018," World-Payments-Report-WPR18-2018.pdf, 2018.

[2] Urs Gasser, Oliver Gassmann, Thorsten Hens, Larry Leifer, Thomas Puschmann, and Leon Zhao, "Digital banking 2025," 2017.

[3] "Value of annual online banking fraud losses in united kingdom (uk) from 2010 to 2017 (in million gbp)," March 2018.

[4] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[5] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2013.

[6] Stephan Kovach and Wilson Vicente Ruggiero, "Online banking fraud detection based on local and global behavior," in *Proc. of the International Conference on Digital Society*, 2011, pp. 166–171.

[7] Michele Carminati, Roberto Caron, Federico Maggi, Ilenia Epifani, and Stefano Zanero, "Banksealer: an online banking fraud analysis and decision support system," in *Proc. of the IFIP International Information Security Conference*. Springer, 2014, pp. 380–394.

[8] Sunil Mhamane and L.M.R.J Lobo, "Fraud detection in online banking using HMM," *International Proceedings of Computer Science & Information Technology*, pp. 200–204, 2012.

[9] Ashish Vaswani et al., "Attention is all you need," in *Proc. of the Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[10] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, "Hierarchical attention networks for document classification," in *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[11] Christopher Whitrow, David J Hand, Piotr Juszczak, D Weston, and Niall M Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Mining and Knowledge Discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[12] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Björn Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.

[13] Sanjeev Jha, Montserrat Guillen, and J Christopher Westland, "Employing transaction aggregation strategy to detect credit card fraud," *Expert systems with applications*, vol. 39, no. 16, pp. 12650–12657, 2012.

[14] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[15] Yaya Heryadi and Harco Leslie Hendric Spits Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, stacked LSTM, and CNN-LSTM," in *Proc. of the IEEE International Conference on Cybernetics and Computational Intelligence*, 2017, pp. 84–89.

[16] Xurui Li et al., "Transaction fraud detection using GRU-centered sandwich-structured model," in *Proc. of the IEEE International Conference on Computer Supported Cooperative Work in Design*, 2018, pp. 467–472.

[17] Johannes Jurgovsky et al., "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[18] Shuhao Wang, Cancheng Liu, Xiang Gao, Hongtao Qu, and Wei Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in *Proc. of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 241–252.

[19] Fangfang Yuan, Yanan Cao, Yanmin Shang, Yanbing Liu, Jianlong Tan, and Binxing Fang, "Insider threat detection with deep neural network," in *Proc. of the International Conference on Computational Science*. Springer, 2018, pp. 43–54.

[20] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," *arXiv preprint arXiv:1710.00811*, 2017.

[21] Aaron Tuor, Ryan Baerwolf, Nicolas Knowles, Brian Hutchinson, Nicole Nichols, and Rob Jasper, "Recurrent neural network language models for open vocabulary event-level cyber anomaly detection," *arXiv preprint arXiv:1712.00557*, 2017.

[22] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[23] Diederik P Kingma and Jimmy Ba, "ADAM: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.