EasyChair Preprint

№ 827

Abnormal Traffic Pattern Detection in Real-Time Financial Transactions

Sean Rastatter, Travis Moe, Amitava Gangopadhyay and Alfred Weaver

March 13, 2019

Abnormal Traffic Pattern Detection in Real-Time Financial Transactions

Sean T. Rastatter
Innovation Labs
SWIFT, Inc.
Manassas, VA 20110
sean.rastatter@swift.com

Travis B. Moe
Innovation Labs
SWIFT, Inc.
Manassas, VA 20110
travis.moe@swift.com

Amitava Gangopadhyay
Innovation Labs
SWIFT, Inc.
Manassas, VA 20110
amitava.gangopadhyay@swift.com

Alfred C. Weaver
Computer Science
University of Virginia
Charlottesville, VA 22901
weaver@virginia.edu

Abstract—We have developed a combined statistical analytical, machine learning (ML) and deep learning (DL) approach to detect abnormal traffic patterns in financial messages involving monetary payment instructions. We used optimally anonymized historical transaction data from multiple financial institutions from disparate geographic locations globally. Our objectives were to provide client institutions with customizable levels of alert notification based upon their risk tolerance, and the ability to detect and prevent fraudulent payment instructions in real time. Our statistical analytical approach demonstrates that a preliminary transaction-based calendar can be established based solely on historical transaction data containing message counts and their arrival times, and can be further improved based upon user input as necessary. Several ML and DL models were built and evaluated for each of their performance metrics (e.g., accuracy, confusion matrix). Our results suggest that a time series ML model (seasonal autoregressive integrated moving average (SARIMA)), and particularly two DL classification models (Autoencoder and Restricted Boltzmann Machine (RBM)) can consistently yield highly accurate predictions. Our study also suggests that ML and DL models in conjunction with a statistical analytical approach provide a powerful tool for real-time anomaly detection in financial transactions.

Keywords—statistical analytical, machine learning, deep learning, anomaly detection.

I. INTRODUCTION

The extent to which global financial fraud is reported in multiple surveys varies significantly among different sources and from one year to another [1, 2]. Yet, the reported rates of financial fraud around the globe are alarmingly high, and generally accelerating since 2009 [2]. For example, 84% of companies surveyed around the globe by Kroll's Economist Intelligence Unit (EIU) in 2017-18 experienced financial fraud, up slightly from 82% in 2016 [1]. Similarly, the PwC's 2018 Global Economic Crime and Fraud Survey reports that 49% of global organizations have been victims of economic crime in the past two years, an increase from 36% in 2016 ([2], Fig. 1). Notably, the US financial services firms alone have experienced an 8.5% increase in average cost of fraud from 2017 to 2018 [3].

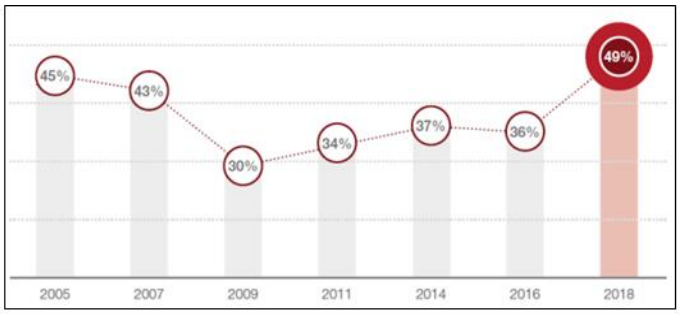

Fig. 1: **Percentage of companies reporting financial fraud during the period 2005-2018. [2]**

Financial fraud poses a challenging problem due to a variety of factors, including (a) the very small fraction of fraudulent transactions compared to the sheer volume of legitimate transactions processed by most financial institutions ("needle in the haystack" problem); (b) real- or near real-time transaction processing ("high velocity" of big data); (c) frequent lack of traceability of transactions from ultimate source to final destination; and (d) lack of information sharing among financial institutions. Further, smaller financial institutions usually lack the resources, capabilities, sophisticated tools and technology often necessary to detect and prevent financial fraud.

While accurate fraud detection is an essential part of financial crime prevention, misclassifications of legitimate transactions as fraud ("false positive"), on the other hand, can lead to significant loss of revenues for merchants and financial institutions. For example, the retail, e-commerce and financial services industries all had an estimated more than 20% false fraud alerts in 2017-18 contributing to signicant "opportunity cost" due to lost revenue [3]. Similarly, the risk of false fraud alerts in the financial services industry involving wholesale payments may amount to unwarranted cost of manual review and/or investigation, and also loss of trust. A combined approach that employs multiple and diverse fraud detection techniques is thus necessary in order to achieve an optimum balance between risk tolerance and cost.

In this study, we have developed an approach using combined dynamic statistical analyses, and artificial intelligence (AI) and ML models in order to detect anomalous traffic patterns in financial messages for wholesale payment instructions in real time. Our objective was to provide client institutions with tools for financial fraud detections and

generate customizable alerts based upon the risk tolerance of individual institution. We have used available anonymized historical transaction data from 13 consecutive months in 2017-18 from 10 financial institutions. Selected "features" of the historical datasets (e.g., identity of sender or receiver financial institution, exact transaction amounts) were anonymized prior to their use in our study. Thus, the identities of specific institutions were withheld in compliance with applicable data protection and privacy laws. The financial institutions were, however, selected from disparate geographical locations around the globe in order to avoid geographical control on transaction patterns (e.g., country- or region-specific holidays, seasonal patterns). The historical data span a large range of traffic volumes per day (a few hundred to a few thousand). Thus, our data collectively represent both small and large financial institutions.

Our multi-prong solution for financial fraud detection involves statistical analyses of historical data in conjunction with the use of a time series ML model (SARIMA [4-6]) and a pair of DL models (Autoencoder [7] and Restricted Boltzmann Machine (RBM) [8,9]). We analyzed and modeled the subsets of historical data for each financial institution separately. The separate analyses and modeling for each institution were important because the transaction patterns among financial institutions in our datasets vary greatly, and, as a result, one single model is not suitable to generalize patterns across all institutions. Here we report our implementation of this multi-faceted approach as different phases in the fraud detection workflow, and demonstrate that the AI-powered prototype web application we have built can successfully identify fraudulent transactions with high accuracy and generate alerts in real time.

II. Overall Approach

Here we briefly describe the three separate phases of our modeling.

A. Dynamic Statistical Analysis

Unlike fraud detection based upon a fixed set of pre-defined rules, statistical analyses of several features from 13 months of historical data for each of the 10 institutions were used for alert generation. Different types of days (work days, non-work days, part-time work days, holidays) were identified for each institution based upon transaction counts for each day in the historical data. These days were used to define the transaction-based calendar. Further, the mean and standard deviation of features, namely, "Transaction count" and "Transaction Arrival Time," were used to derive a transaction-based calendar that describes the window of time when a given transaction is permissible. This transaction-based calendar may or may not be identical to the business calendar of the financial institutions. For example, a financial institution that opens for business at 8 a.m. and closes at 5 p.m. may have historically received its first transaction for a given day (e.g., Monday) at 9:00 a.m. and its last transaction at 4:30 p.m. Thus, the business calendar for that day (8 a.m. to 5 p.m.) is different from its transaction-based calendar (9 a.m. to 4:30 p.m.).

We derived the transaction-based calendar for each day based upon the mean of the first and last transaction arrival times, within default 3-sigma uncertainties. Transactions are considered legitimate when they arrive within the window of the transaction-based calendar, whereas transactions arriving outside of this window are either "blocked" or sent for "review request." When a transaction arrives at a time outside of both the business and transaction based calendars, it is "blocked" from further processing. Alerts for "review requests" are sent when a transaction arrives outside of transaction-based calendar hours but still within the business hours of the specific institution.

Each feature was approximated to a Normal Gaussian Distribution and alert rules were formulated based upon probabilities and z-scores, which in turn can be customized depending upon the risk tolerance of a particular institution. Also, such an approach is dynamic in that the statistical parameters that define the alert rules can change from one financial institution to another and also as new training data are added over time.

B. Machine Learning Approach

The ML methods involved modeling aggregated message counts at individual minute intervals using the following four separate time series models: Auto-Regressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), Multi-Layer Perceptron (MLP) and Long-Short Term Memory (LSTM). These four models were trained and tested with the same set of sample data over multiple runs. Among these four models, the Seasonal ARIMA model seemed to yield the most consistently accurate results across multiple runs based on our test data. Accordingly, here we limit our discussion to the Seasonal ARIMA model.

The seasonal ARIMA model (SARIMA) combines both seasonal and non-seasonal factors in a multiplicative model using a paired set of 3 parameters. The common shorthand notation of the mode is as follows [4-6]:

$$\text{ARIMA}(p, d, q) \times (P, D, Q)_m$$

where:

p is the non-seasonal autoregressive (AR) order,

d is the non-seasonal differencing needed for stationarity, and

q is the non-seasonal moving average (MA) order.

The upper case P, D, Q terms represent the equivalent parameters (as in lower case p, d and q) for seasonal factors. The term *m* represents the time span of repeating seasonal pattern (e.g. m = 4 for quarterly data). The coefficients in the model are all positive integers, and determine the structure of the full prediction equation [4-6].

C. Deep Learning Approach

Our sample data from financial transactions include multiple fields ("features") such as currency type, geographic route, message type, message counts and others that were used to identify anomalous transactions. These features, however, have complex interdependencies that, in some cases, make it difficult to detect abnormal traffic patterns using simpler ML

models, as discussed in the last section. Accordingly, we also used two deep learning (DL) models, Autoencoder [7] and Restricted Boltzmann Machine (RBM: [8, 9]), in order to leverage the higher resolving power of Artificial Neural Networks (ANNs) used in these models. Further, we used two separate DL models (rather than one single model) in order to build a more robust solution for classification using two independent models for deep learning. Although DL models usually yield higher accuracy in predictions, the training and testing of these models typically require more computing power than in simpler ML models. As such, we used a pair of Nvidia GeForce GTX 1080 graphic cards to accelerate the training and testing of our deep learning models. We used a binary classification scheme for each transaction as either fraudulent (class 1) or legitimate (class 0). Below is a brief description of the Autoencoder and RBM models used in this study.

1) Autoencoder

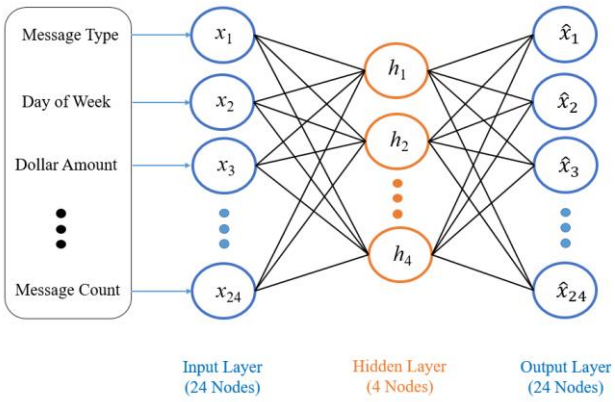

Fig. 2. Architecture of the Autoencoder Model

The Autoencoder model architecture (Fig. 2) consists of three layers: an input layer, a hidden layer, and an output layer. The nodes in the input layer have a 1-1 mapping with the corresponding nodes in the output layer. The transformations that take place in the hidden layer result in an indirect mapping from input to output. Each node in the input layer represents a feature in a transaction (e.g. message type, message count). The categorical data from the input layer were converted to numeric data using an enumeration process, and then standardized between 0 and 1 to allow the optimization algorithm to minimize the "reconstruction error" to work more efficiently [11].

The Autoencoder model attempts to reconstruct each transaction based on computed weights of features corresponding to each node in the hidden layer. The number of nodes in the hidden layer can be tuned during training to allow an optimal trade-off between resolving insights the model can offer versus additional computing power the ANN requires in order to yield desired levels of model accuracy. The "reconstruction error" for each transaction is analogous to the "cost function" in other ML models. This error is minimized using a Gradient Descent algorithm during the model training [11].

2) Restricted Boltzmann Machine (RBM)

This model is similar to Autoencoder in that it automatically detects patterns in data by reconstructing the inputs. RBM is considered a shallow ANN because it only has two layers, the visible layer and the hidden layer. Each node in the visible layer connects to each node in the hidden layer. The RBM model is called 'restricted' because the nodes within each layer are not connected to other nodes in the same layer [11-13]. The training process starts with receiving the inputs to the nodes in the visible layer and moving them to the hidden layer. During each pass, the weights of inputs are modified in the input layer. Next, the hidden layer modifies the inputs again and produces new outputs. These new outputs are sent back to the input layer where they are compared with their corresponding original inputs and their differences are computed. These weight factors are adjusted during each pass until the differences between the input feature values and the reconstructed values are minimized.

III. DATA ANALYSES AND MODELING

A. Dynamic Statistical Analysis

Four main features of the transaction data were explored for the statistical analysis: message arrival time at the minute level (timestamps), the type of message (categorical values), the message count (integers), and the sender institution (categorical). The historical sample data for 30 months were split into a 24-month training set and a 6-month test set. Based upon the user-entered risk tolerance level, a z-score was determined.

Possible holidays were determined based on the transaction counts per day for the entire training period. The holidays were removed from the training set in order to yield the transaction counts for each working day.

Different types of alerts were considered in this study. For example, if a message is received on a non-working day of the sender institution, an alert is generated. A particular day of the week was classified as a "non-working" day if the number of messages received on that day never exceeded 5% of the total messages for the week across all weeks in the training dataset. Notably, this definition of "non-working day" can be customized and configured based upon thresholds on transaction counts or proportions provided by individual financial institutions. Similarly, if a message is received before or after the "transaction window" of the sender financial institution, an alert is generated. This window was determined by taking the mean and standard deviation of the earliest and latest transaction arrivals for each day and generating a 3-sigma bound before and after the respective means. Finally, some financial institutions send transactions as a batch (and usually with a cap on the maximum number of transactions). In those cases, a "hard ceiling" is set for transaction batches, and alerts are generated if a given batch carries more transactions than the cap determined for the entire training period.

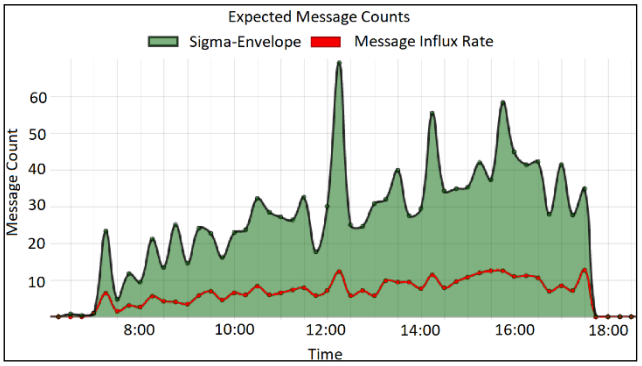

Fig. 3. EWMA graph along with its sigma envelope on a Monday for a particular bank.

We used a rolling 15-minute time window, and computed the message influx rate (count of incoming messages per minute). The exponential weighted moving average (EWMA) and exponential weighted standard deviation (EWSD) of the message influx rates were computed for any given time within the window (Fig. 3). The EWMA assigns higher weights for more recent data compared to older data. Consequently, the EWMA yields accurate predictions on the expected transaction count at a given minute in time series datasets. Alerts are generated when transaction counts at a given minute fall outside of the 3-sigma error envelope at a given minute (EWSD), thus enabling our detection of the first order unusual transaction patterns.

B. *Machine Learning*

For the solution presented here, we have revised the original python source code from [14] and modified based on the specifics of our use case. For our seasonal ARIMA model, we used 13 months of historical sample data that were split into a 12-month training set and a one-month test set. Each financial institution had unique message traffic patterns over the time period in the training set, which, in turn, required separate sets of tuning parameters for each financial institution. We employed a grid-search method in order to find the optimal set of tuning parameters via continuously fitting SARIMA models to the training set and calculating the mean-squared error (MSE). With m = 12, all possible combinations of 0s and 1s were fitted to a particular financial institution's data (ARIMA (0,0,0) $(0,0,0)_{12}$, ARIMA (0,0,0) $(0,0,1)_{12}$, …, through ARIMA (1,1,1) $(1,1,1)_{12}$). The result with the lowest MSE was then selected. The results of our SARIMA model for one specific institution are shown in Fig. 4.

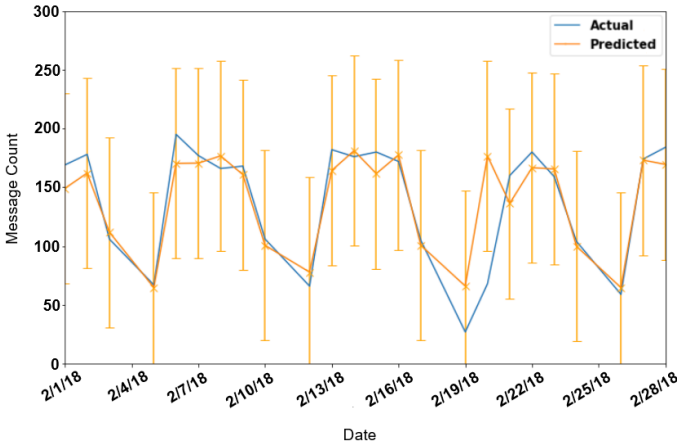

Fig. 4. SARIMA model predictions for daily total message counts for a specific bank during February 2018. The uncertainty bars represent 2-sigma deviation from the corresponding predicted values.

C. *Deep Learning*

Deep learning models typically require large volume of data for model training. Accordingly, we simulated several hundred thousand transactions following the same statistical distribution patterns in the real data, and also anonymized the data appropriately. Further, unlike the statistical analyses and ML modeling discussed earlier, where only limited number of features were used, the data used in our Deep Learning models included a total of 24 features, including message type, arrival time, sender financial institution, receiver financial institution, currency type, transaction amount bucket (in USD), message count, sender and receiver countries, etc. Also, these simulated data included both "negative" ("legitimate" or class 0) and a suitable number of "positive" ("suspicious" or class 1) samples. The "suspicious" samples were created via artificially injecting outliers in the feature values (e.g. unusually high transaction amount, transaction arrival time outside of typical transaction window). The "positive" samples were used to test the accuracy of the models in predicting suspicious transactions.

1) Autoencoder

The simulated dataset was split into 85% training data and 15% test data. Relevant features were extracted and the resulting feature matrix was fed into our Autoencoder model (Fig. 2). The model was initialized with four nodes within the hidden layer. The model was trained using 50 epochs, and the test set was subsequently fed into the model [10]. For each transaction in the test set, the model computes a Reconstruction Error (RE). Based upon the RE values, a threshold was established to classify the transactions, whereby all RE values above the threshold were flagged as suspicious transactions. The Autoencoder source code from [10] was modified to suit our model requirements.

2) Restricted Boltzmann Machine (RBM)

The same simulated data used for Autoencoder was also used for the RBM model. Unlike the Autoencoder model where an 85/15 split was used for training and test data, we used a 50/50 split for the RBM model because our research into RBM implementations showed more favorable results with a 50/50 split. Still, the training and testing processes for RBM are very similar to those used with Autoencoder. The python source codes for RBM published in [11-13] were used and modified for our solution. The training data is passed to the RBM over 50 epochs. Similar to the ranking of RE in the Autoencoder model, RBM uses Free Energy to mark a message as either suspicious (class 1) or legitimate (class 0). Records with free energy values close to zero, as computed by the RBM model, are assigned to class 0 ("legitimate") [8]. Any record with computed free energy values above a defined threshold, on the other hand, is marked as class 1 ("suspicious"). This threshold can be customized based on the risk tolerance level of the financial institution.

IV. RESULTS AND DISCUSSION

The results of statistical analyses, SARIMA ML model and the DL models (Autoencoder and RBM) were analyzed and evaluated for their predictions on known test data. Here we briefly discuss these results along with the alert rules we have implemented for our fraud detection application.

A. Dynamical Statistical Analysis

The results of the dynamic statistical analyses were used as the basis of some of our alert rules. First, as expected, no alerts are generated for class 0 ("legitimate") transactions. The application generates alerts only for the class 1 ("suspicious") transactions. Secondly, two types of alerts were created for suspicious transactions: "forbidden" and "review required." For example, a transaction is considered "forbidden" when it is received from a sender institution at a time outside of its business hours (e.g., non-working days or holidays), as defined via statistical analyses of its historical transaction data. In this case, our fraud detection system can trigger an alert to notify the appropriate transaction support team on the likely fraudulent nature of the transaction. The authorized user can block this transaction, if deemed necessary, from further processing. If a transaction, on the other hand, is received within business hours, yet outside of the transaction window of the originating institution (discussed in section III A), the transaction is considered "suspicious". In this case, an alert is sent, and it requires a review of the legitimacy of the transaction by an authorized user prior to further processing.

Our application is capable of ingesting real-time streaming transaction data, and can display the alerts on a web application approve a single transaction at a time or a batch of transactions together as necessary for faster processing.

The use of EWMA, as opposed to regular moving averages, appears to more accurately define the trends in our historical time series transaction data. As noted in the previous section, the message influx rate (message counts per minute) was dynamically computed using EWMA for a 15-minute rolling window (Fig. 3). This message influx rate, in conjunction with an estimate of uncertainty (3-sigma), yields a useful measure of expected message count at a given minute within the business hour of the institution. When more messages are received than expected at a given minute, the system triggers alerts. Similar to alerts at minute levels, we also formulated alerts based on daily aggregates of transaction counts and their total monetary values. Further, alert rules were formulated based on aggregate daily, weekly, and monthly message counts in order to forecast trends in transaction at those levels.

Notably, the majority of the alerts, formulated for our selected financial institutions, are due to deviations from our minute level predictions. There is also a trade-off between false negatives and false positives in model predictions, and their relative frequencies can be optimized based on the risk tolerance of an individual institution. For example, fewer alerts are triggered when the standard deviation threshold is raised (3-sigma instead of 2-sigma), which is likely to reduce the number of false positives at the expense of increased risk of missing fraudulent transactions.

B. Machine Learning Models

As mentioned, we used historical data for a total of 13 months in 2017-18 for our SARIMA model. The model was trained using data for the first 12 months in the dataset, and tested using data for the remaining month. Aggregate daily message counts were predicted for the last month, and compared against the actual daily message count in our historical records. The results of these predictions for a single institution are shown in Fig. 4.

Note in Fig. 4 that the prediction equation used in the SARIMA model (section II B), creates a sinusoidal projection that, within bounded uncertainties ("error envelope"), accurately reproduces the expected daily message count for a given financial institution bank over the period of a month. The uncertainty is defined by a 3-sigma envelope around the sinusoidal curve. If the message influx rate for a particular day plots outside of the error envelope, an alert is generated to suggest abnormal activity for that particular day. For example, if the actual message count is significantly above the predicted value, it may suggest suspicious or fraudulent transactions. On the other hand, if the message count is significantly below the predicted value, it may suggest possible system or network failure, an unscheduled financial institution closure, etc.

C. Deep Learning Models

1) Autoencoder

Our Autoencoder model correctly predicted 6,539 out of 6,544 legitimate (99.92%) and 26 out of 27 (96.3%) fraudulent transactions. One false negative (actual fraud misclassified as legitimate) and five false positive results (legitimate transaction misclassified as fraud) were predicted by the model. Combined, these results demonstrate the high degree of accuracy in predicting both legitimate and suspicious transactions.

2) Restricted Boltzmann Machine (RBM)

The results of the RBM model were also highly accurate. The free energies of the vast majority of simulated fraudulent transactions were distinctively above the threshold set for legitimate transactions. Accordingly, the RBM model correctly predicted 13 fraud transactions (true positives). There was only one false negative and no false positives in the RBM model predictions. These results suggest that the RBM model could be applied with a high degree of confidence in detecting fraudulent transactions.

V. CONCLUSIONS

First, while fraud detection can prevent financial loss, misclassification of legitimate transactions as fraudulent can lead to potential revenue loss as an unintended consequence. Thus, the optimal level of alert generation ("aggressive" versus "conservative") is best determined based upon the level of risk tolerance ("risk appetite") of a particular institution. Accordingly, the statistical analyses in our approach enable financial institutions to define alert rules and customize them based upon their specific level of risk tolerance. For example, different statistical measures of the transaction data, such as probability densities, z-scores, and exponentially weighted moving averages (EWMA), all can be used in our approach to

define a set of alert rules that best suit the risk tolerance of an institution.

Secondly, we have demonstrated that our ML model (Seasonal ARIMA) yields highly accurate predictions in detecting legitimate versus fraudulent transactions based upon the message influx rate (counts per minute). Further, the classification algorithms used in particularly two DL models (Autoencoder and RBM) similarly yield highly accurate predictions. Combined, this study suggests that a statistical analytical approach can be used in conjunction with the use of ML and DL models in order to detect anomalous traffic patterns in financial transactions.

Finally, our results suggest that the dynamic statistical analytical methods, combined with the ML and DL methods, provides a powerful tool to detect suspicious transactions with high accuracy in real time traffic. While our multi-prong approach followed in this study was applied only on financial messages carrying wholesale payment instructions, the same approach can be potentially applied to other business areas (e.g. retail banking, lending) via carefully selected set of relevant features of data on analogous financial transactions. Thus, the results of this study suggest that our approach can potentially reduce fraud-related monetary loss in financial transactions.

Disclaimer

The views articulated in this paper are personal to the authors and do not represent the views of their employers or any other organization.

Acknowledgement

This research was funded and supported by the Innovation Labs and Summer Internship Program at SWIFT. We acknowledge the help and support from the Innovation Lab members in various ways during this work. We also thank Soumitra Dutta (Cornell Univ.) and Uwe Aickelin (Univ. of Melbourne) and Tony Wicks for their very insightful and constructive informal reviews of earlier versions of this manuscript.

REFERENCES

[1] Kroll Global Fraud and Risk Report, Forging New Paths in Times of Uncertainty, 10th Annual Edition, 2017/18, https://www.kroll.com/en-us/global-fraud-and-risk-report-2018

[2] Pulling Fraud out of the shadows, PwC's Global Economic Crime and Fraud Survey, 2018, https://www.pwc.com/gx/en/services/advisory/forensics/economic-crime-survey.html

[3] LexisNexis Risk Solutions, True cost of Fraud Study, 2018, https://risk.lexisnexis.com/-/media/files/financial-services/research/2018-true-cost-of-fraud-overall-rep-pdf.pdf?la=en-us&hash=61447FA7436D743955711ECA8F102DCA8685A320

[4] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Liung, G. M., Time Series Analysis: Forcasting and control (5th ed), John wiley & Sons, 2015.

[5] Brockwell, P. J, & Davis, R. A., Introduction to time series and forecasting (3rd ed), Springer, 2016.

[6] Wu, J. P., & Wei, S. *Time series analysis*. Hunan Science and Technology Press, ChangSha, 1989.

[7] Hinton, G. E., and Salakhutdinov, R. R., Reducing the Dimensionality of Data with Neural Networks, Science, 313, (5786): 504-507, 2006.

[8] Hinton, G., Sejnowski, T., and Ackley, D., Boltzmann machines: Constraint satisfaction networks that learn, Tech. rep. TR-CMU-CS-84-119, Carnegie-Mellon University, Dept. of Computer Science, 1984.

[9] Ackley, D., Hiton, G., & Sejnowski, T. (1985). A learning algorithm for Boltzmann machines, Congnitive Science, 9, 1985.

[10] V. Valkov, 'Credit Card Fraud Detection using Autoencoders in Keras', 2017. https://medium.com/@curiousily/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd.

[11] Skymind, 'Restricted Boltzmann Machine'. [Online]. Available: https://skymind.ai/wiki/restricted-boltzmann-machine

[12] 'Restricted Boltzmann Machines (RBM)', http://deeplearning.net/tutorial/rbm.html

[13] W. Wang, 'Credit Card Fraud Detection using Restricted Boltzmann Machine in TensorFlow', 2017, https://weiminwang.blog/2017/08/05/credit-card-fraud-detection-2-using-restricted-boltzmann-machine-in-tensorflow/

[14] S. Abu, 'Seasonal ARIMA with Python', 2016. http://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-python/