Deep Learning Approach for Intelligent Financial Fraud Detection System

Aji Mubarek Mubalaike
Istanbul Technical University
Informatics Institute
Istanbul, Turkey
mubalaike15@itu.edu.tr

Eşref Adalı
Istanbul Technical University
Computer and Informatics Faculty
Istanbul, Turkey
adali@itu.edu.tr

***Abstract*—Law enforcement authorities continually confront new and vexatious frauds involving computer virus attacks or credit card frauds committed against financial institutions and banking card companies. Such phenomenon used to commit online banking fraudulent transactions are occurring continuously and lead to a loss of gargantuan amount of money. Information security specialists and associations usually refer to the process of selecting practitioners as safeguards of fraud assessment, fraud analysis, or fraud detection. Designating such safeguards based on fraud calculations can be a exasperating and high-price process over the past few decades. This paper aims to understand how deep learning (DL) models can be benevolent in detecting fraudulent transactions with high accuracy. Dataset is extracted from one month of genuine financial logs of a mobile money service company in Africa, containing about more than six million transactions. The most satisfactory machine learning techniques as ensemble of decision tree (EDT), and deep learning techniques as stacked auto-encoders (SAE) and Restricted Boltzmann Machines (RBM) classifiers are applied on the preprocessed data. The performance of generated classifier models is evaluated based on accuracy, sensitivity, specificity, precision, confusion matrix and ROC values. The results of optimal accuracy are 90.49%, 80.52% and 91.53% respectively. The comparative results manifest that restricted Boltzmann machine performs superior than the other techniques.**

Keywords—information security, fraud detection, deep learning, EDT, SAE, RBM, ROC value, accuracy

I. INTRODUCTION

Fraud is hackneyed but counterproductive subject that has been continuing to inhibit the development of economics and pose threats to the security systems. Criminals and crooks are finding various and ingenious ways to commit fraud against the security sectors. The loss due to unauthorized transactions alone is estimated to be billions of dollars each year. Computer fraud can be considered as any falsification or misrepresentation by customer, employee or any third party with the dishonest aim to gain pernicious benefit to compromise the confidentiality, integrity, and availability of vital information [1]. Although intrusion detection and intrusion prevention systems are necessary as system security functions, yet they are not sufficient to protect systems from all security threats. Intrusion detection systems (IDSs) or intrusion prevention systems (IPSs) aim to detect or prevent suspicious traffic and abnormal activities originating from inside or outside of the organization. They should be a part of a more comprehensive security strategy that includes vulnerability assessment, security policy and network firewalls, strong identification and authentication mechanisms, access control mechanisms, file and link encryption and file integrity checking [2].

Intrusion detection is typically split into two separate problems like host based and network based intrusion detection system. Host based intrusion detection system (HIDS) runs on individual hosts or devices on the network and monitors the inbound and outbound packets from the device while collecting data on a single host. Network based intrusion detection system (NIDS) is typically placed at a strategic point or points within the network to monitor traffic to and from all devices on the network [3]. It usually provides reliable, real-time information without consuming network or host resources, and plays a crucial role in defending computer networks. There are also some basic approaches to intrusion detection systems. Signature-based detection system (SIDS) which is also termed as misuse-based relies on some set of features that can be extracted from the data that indicate the existence of an attack. Limitations of these signature engines are that they are useless against new techniques for which they have no signature or patterns; they only detect attacks whose signatures are previously stored in database [4]. While anomaly-based intrusion detection system AIDS is dedicated to establishing normal activity profiles for the system and may find out the existing unknown risks.

Rest of the paper continues as follows. Section 2 discusses the previous researches that related to the intrusion and fraud detection systems using various kinds of machine learning or deep learning techniques. Our proposed approach is discussed in section 3 in detail theoretically. To implement a real-time network intrusion or frauds detection system, we use the synthetic dataset for real-time mobile money transactions generated by Paysim. In section 4, we demonstrate preprocessing work consists of feature engineering that reduces the dimensionality of the hypothesis search space and storage costs, enhance the performance of deep learning techniques, and simplify the experimental results. Finally, we present our experimental results of various classification and intrusion detection techniques and find the classification model generates the most prominent detection accuracy, then conclude the result in section 5.

978-1-5386-7893-0/18/$31.00 ©2018 IEEE

II. Related Work

Noticeably, there has been numerous existing researches related to the detection of network intrusion or banking credit card frauds. The most popular machine learning techniques include gradient boosting machines (GBM), decision trees (DT), dimensionality reduction principle component analysis (PCA), k-nearest neighbor clustering (KNN) have been used to detect fraudulent incidents. Last few decades, deep learning techniques, separated from neural networks in machine learning, have been taking off in an excessive way. Mentioned deep learning techniques compromise restricted Boltzmann machines (RBM), deep belief network (DBN), convolutional neural network (CNN), Recurrent Neural network (RNN) and stacked auto-encoders (SAE). It is impossible to know beforehand which technique will be most effective. Every algorithm has its specialty for specific type of dataset, for instance some classification models can only detect well some specific type of attacks after comparing different type of accuracy. The datasets that researchers analyzed include KDD99, NSL-KDD, UNSW-NB15, and Tor client datasets. Implementing a real-time network intrusion or fraud detection system for real networks are the main demand.

A. Network Intrusion Detection

In [5] implemented the IDS classifier based on long-short term memorization (LSTM) with the combination of RNN and evaluated the IDS model by achieving highest detection rate (DR) and accuracy. In [6] proposed intrusion detection methods based on deep belief network (DBN) and probabilistic neural network (PNN) to shorten the training and testing time by converting the raw data into low-dimensional data. After combining the generated model with particle swarm optimization (PSO) algorithm, authors obtained the best learning performance by optimizing the DBN number of hidden-layer nodes. In [7] demonstrated the application of stacked auto-encoder (SAE) and stacked restricted Boltzmann machines (RBM) to anomaly-based NIDS. Experimental results testified the problem of the training time consuming in SAE is much more than that in RBM due to much computations in SAE. In [8] researched the application of deep recurrent neural network (DRNN) algorithm for prediction of user behavior in Tor networks by constructing a Tor server and getting data by using Wireshark network analyzer, then achieved better performance for malware classification in comparison to previously published results. In [9] achieved promising results offering high levels of accuracy with reduced training time by using graphics processing unit (GPU) techniques. They discussed the problems existing on NIDS techniques, and proposed the novel nonsymmetrical deep auto-encoder (NDAE) method for unsupervised feature learning with the combination of random forest (RF) classification algorithm for two types of datasets, NSL-KDD and KDD99. In [10] tested some shallow and deep neural networks on the well-known network traffic NSL-KDD dataset, and achieved a result that shallow neural networks have lower error rates for network intrusion detection compared to deep neural networks, while be able to classify network data more accurately. In [11] presented a deep auto-encoder (DAE) approach with the combination of greedy layer wise fashion in order to avoid overfitting and local optima. The experimental results on the KDD99 dataset resulted substantial improvement over the other deep learning-based approaches in terms of accuracy, detection rate and false alarm rate.

B. Credit Card Fraud Detection

In [12] demonstrated the performance comparison of some machine learning techniques, like naive Bayes (NB), k-nearest neighbor (KNN) and logistic regression (LR) on highly skewed credit card fraud dataset obtained from European cardholders, containing 284,807 transactions. Comparative results showed that the classifier model of KNN outperformed than the other techniques with the highest value of accuracy, sensitivity and specificity. In [13] proposed some detecting models generated by different types of machine learning techniques, like RUSMRN, RUS-Boost, AdaBoost, and Naïve Bayes algorithms. Combining of boosting and sampling to RUSMRN algorithm based on RUS and MRN data sampling techniques, can improve classification accuracy of unbalanced characteristic data while acquiring the best performance in terms of accuracy and sensitivity, and predicting the data transactions issued by a bank from Taiwan in 2005. In [14], neural network model was used to the UCSD-FICO dataset, generated on data mining contest in 2009. The proposed detecting model was generated using H2O libraries and achieved the prominent accuracy, finding out the importance of all features. In [15] generated detection of fraudulent transactions classifier models by referring to logistic regression, gradient boosted trees and deep learning algorithms, also used unsupervised auto-encoder and domain expertise algorithms for feature engineering. The created features using domain expertise offered a notable improvement in predictive unseen real transactions and the generated model by using deep learning successfully achieved the largest area under curve (AUC) value, while gradient boosted trees had the second highest values across the feature sets.

III. Experimental Methods

Deep learning is a type of artificial intelligence and machine learning that has become prodigiously important in the past few years and allows us to teach machines how to complete complex tasks without explicitly programming them to do so [16]. Allowing private information such as credit card numbers belonging to thousands of people causes to be stolen and distributed on the Internet. In this paper, to prevent mentioned frequent problems we implement and evaluate the application of deep learning to the combination of anomaly-based NIDS and a real-time financial dataset, and compare the detection results with the previous researches.

A. Stacked Auto-Encoder

An auto-encoder is an artificial neural network used for unsupervised learning, which does not need labelled data. The aim of an auto-encoder is to learn an encoding for a set of data, generally for propose of dimensionality reduction by preserving lots of vital information [17]. Recently, SAE in deep learning has become more widely used for learning classification models of data. It learns a good weight initialization that can be used to further train the network. Since dimension of the dataset, which we will use, does not need to be reduced after feature preprocessing, so we chose to use stacked auto-encoders to generate classification model

for fraud detection. First two layers are encoders last two layers are decoders. Then L1 regularization is used.

The simplest form of an auto-encoder is a feed forward, It is very similar to the multilayer perceptron (MLP) – having an input layer, an output layer and one or more hidden layers connecting them – An auto-encoder framework always consists of two parts, the encoder and the decoder [18], which can be defined as:

- *Encoder*: The deterministic mapping function $f(x)$ use (1), which transforms an input vector x through parameter set θ= {W, b} into hidden representation *y*, is called the encoder. Its nonlinear typical form is like:

$$f(x) = s(\mathrm{W}x + \mathrm{b}) \quad (1)$$

Where W is a weight matrix and b is the offset vector of dimensionality.

- *Decoder*: The resulting hidden representation *y* is then mapped back to a reconstructed d-dimensional vector z by a parameters set $\theta' = \{\mathrm{W}', \mathrm{b}'\}$ in input space as *g(y)*, this mapping *g(y)* is called decoder represented as (2). Its nonlinear typical form is like:

$$g(y) = s(\mathrm{W}'y + \mathrm{b}') \quad (2)$$

- *Learning objective*: The idea is that the weights learned in an unsupervised manner to minimize reconstruction error for the representation learning task offer a good starting point to initialize a network for a supervised discriminative task such as classification or similarity [19]. In this research, the network data learns something about the underlying distribution by looking at the unlabeled data, allowing it to discriminate between labeled data. However, the weights still need to be "fine-tuned" for this new task. So add a logistic regression layer on the top of the network and then do supervised learning with a labeled dataset [20]. The fine tuning step will do gradient descent and adjust the weights for all layers in the network simultaneously.

B. Restricted Boltzmann Machines

Restricted Boltzmann machines are an alteration of Boltzmann machines; it puts a limit on the connectivity between hidden units to make learning easier, so connections between hidden units or between visible units could be ignored. This restriction allows for more efficient training algorithms than are available for the general class of Boltzmann machines [21]. Since visible units and hidden units are conditionally independent, we can quickly generate an unbiased sample from the posterior distribution when given a data-vector.

Energy for the RBM is defined as:

$$E(v, h; \theta) = -\Sigma v^{T}Wh - \Sigma b^{T}v - \Sigma a^{T}h \quad (3)$$

Where E is the energy for given RBM and parameters in (3), include a, b, W represent weights for hidden layer bias, weights for visible layer bias and combined weights respectively.

Joint likelihood of RBM is given by:

$$P(v, h| \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \quad (4)$$

Where Z(θ) in (4) is the partition function defined as the summation distribution of exp(-E(v,h; θ)) over all possible configurations as (5).

$$Z(\theta) = \Sigma_{v, h} \exp(-E(v,h; \theta)) \quad (5)$$

RBMs can also be used in deep learning networks. For instance, deep belief networks can be formed by stacking RBMs and optionally fine-tuning the resulting deep network with gradient descent and backpropagation [22].

IV. Performance Evaluation and Result

There is a paucity of public available datasets on financial services, predominantly in the emerging mobile money transactions domain of fraud detection. Paysim [23], a financial mobile money simulator, aggregated data from the private dataset to generate a synthetic data, which can evaluate the performance

of fraud detection methods. Our challenge is to build a decent model that is able to identify the fraud transactions with the highest accuracy in a real-time network by using the most up-to-date techniques like deep learning.

A. Dataset Description

Mobile banking fraudulent transaction incidents are growing rapidly worldwide, which costs upwards of billions of dollars per year. The real-time dataset, we use in this research, had been collected by Paysim during a month time period [23]. Mobile banking fraudulent transactions from the proposed dataset contain 8,213 thousands out of approximately 6 million transactions over one month time in September 2015. It totally contains two types of transaction results, fraud and genuine. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise, depicted as TABLE I.

TABLE I. Original Feature Columns and Description

Features	Description
Step	1 step is 1 hour of time. Total steps 744 (30 days simulation)
Type	Cash_in, Cash_out, Debit, Payment, Transfer
Amount	amount of the transactions in local currency
NameOrig	customer who started the transaction
OldBalanceOrg	initial balance before the transaction
NewBalanceOrg	new balance after the transaction
NameDest	customer who is the recipient of the transaction
OldBalanceDest	initial balance recipient before the transaction.
NewBalanceDest	new balance recipient after the transaction.
Class	Fraud, Genune (1, 0)
IsFlaggedFraud	Transaction whose amount more than 200,000 (1, 0)

Some challenges we encountered at the beginning were the huge imbalanced in the dataset, that the positive class values, frauds, only account for 0.129 percentage out of all transactions. As is illustrated in Fig.1 that the number of transactions which are the actual fraud in per transaction type. It is clear to see that frauds only appeared in two of the

five types of transactions, cash out and transfer. Fortunately, there are no any missing values in the original dataset.

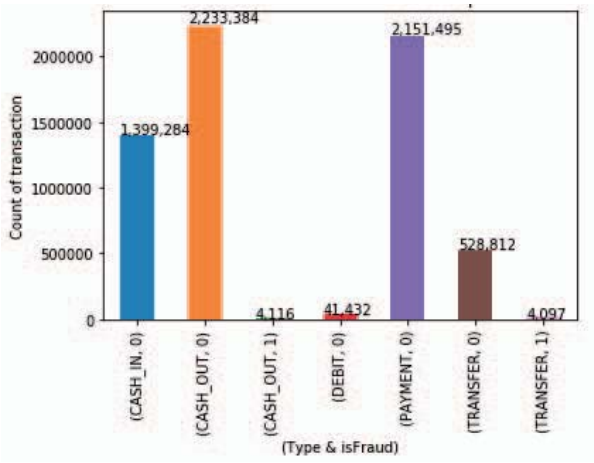

Fig. 1. Number of transactions which are fraud on per transaction type

B. Data Preprocessing

In order to make it easier to run deep learning techniques, another data set was created by preprocessing the original raw data and extracting the relevant features. After doing some experimental preparatory test for the original data, it is easy to conclude that the feature named NameOrig, NameDest and IsFlaggedFraud are irrelevant and can be discarded without losing any vital information. Since for isFlaggedFraud feature value, isFraud is always set when it is set, and the number of appearance is just 16 times in a meaningless way. Furthermore, frauds only appeared in two of the five types of transactions, cash out and transfer. Because of our main purpose, that mainly focusing on detecting the fraudulent transactions, we decided to discard the unnecessary transactions whose type values are cash in, debit or payment. After doing mentioned feature engineering process, we achieved new dataset that are more balanced. Extracted new features are given in TABLE II. specifically. Motivated by [25], in order to differentiate between fraudulent and genuine transactions, two new feature columns named ErrorBalanceOrig and ErrorBalanceDest, recording errors in the originating and destination accounts for each transaction were generated. Followings are the description of generated new dataset.

a) Accounted percentage of fraudulent transactions increased from 0.13% to 0.3%.

b) Total amount of transactions decreased from approximately 6 millions to 2 millions while frauds remained at the same balance.

c) Type of transactions only consists of Cash_out and Transfer, while Cash_out accounts for 80.8% and Transfer is only about 19.2% of all dataset.

d) New feature columns named ErrorBalanceOrig and ErrorBalanceDest were generated.

e) Values of features named Amount, OldBalanceOrg, NewBalanceOrg, OldBalanceDest, NewBalanceDest, ErrorBalanceOrig, and ErrorBalanceDest will be normalized.

f) One hot encoding techniques will be used for the catogorical feature values, as Type and Class.

TABLE II. GENERATED FEATURE COLUMNS AFTER FEATURE ENGINEERING

Features	Description
Step	1 step is 1 hour of time. Total steps 744 (30 days simulation)
Type	Cash_out, Transfer (1, 0)
Amount	amount of the transactions in local currency
OldBalanceOrg	initial balance before the transaction
NewBalanceOrg	new balance after the transaction
OldBalanceDest	initial balance recipient before the transaction
NewBalanceDest	new balance recipient after the transaction
Class	Fraud, Genuine (1, 0)
ErrorBalanceOrig	NewBalanceOrg + Amount - NewBalanceOrg
ErrorBalanceDest	OldBalanceDest + Amount - NewBalanceDest

For the range of feature values of raw data varies widely in a random way, in some deep learning algorithms, objective functions will not work properly without normalization. For instance, the majority of classifier models calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized in some specific area so that each feature contributes approximately proportionately to the final distance. Proposed numerical feature values are computed using equation (6) normalized on a scale of 0 to 1:

$$Z = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (6)$$

One hot encoding is a procedure, in which categorical features are splittedform into a form that could supply for deep learning algorithms a better accuracy in prognostication of testing model. Using this technique to the categorical two feature columns, as is illustrated in TABLE III., they can be separated into four different columns in case of generating more decent classification model and achieving higher accuracy results in prognostication of unseen testing data.

TABLE III. SPLITTED FOUR DIFFERENT FEATURE COLUMNS AFTER USING ONE HOT ENCODING

Type		Class	
Cash Out	*Transfer*	*Fraud*	*Genuine*
0	1	0	1
1	0	1	0

C. Evaluation Measures

Accuracy shows the percentage of the correct classifications with respect to all samples. But it does not say anything about the performances for negative and positive classes. Precision measures how many of the positively classified samples were really positive. There are other performance measures such as recall, sensitivity, specificity, true positive rate (TP-Rate) and false positive rate (FP-Rate), as shown in TABLE IV.

TABLE IV. PERFORMANCE MEASURES AND CORRESPONDING FORMULA

Name	Formula
Error	(FP + FN) / N
Accuracy	(TP + TN) / N = 1 - Error
TP-Rate	TP / P
FP-Rate	FP / N
Precision	TP / P'
Recall	TP / P = TP-Rate
Sensitivity	TP / P = TP-Rate
Specificity	TN / N = 1 – FP-Rate

Both metrics above work with a fixed threshold value on the class probability. But dissimilar thresholds may result in divergent accuracy values, or different precision and sensitivity values. If we are interested in a high sensitivity, we would calculate the performance at a low threshold value. Thus, area under the ROC curve (AUC) has a value for the overall performance of the classifier. AUC reports the curve between TP-Rate and FP-Rate and the area under the curve. It is a better measure than accuracy based on formal definitions of consistency [27]. Then we make a plot of sensitivity versus specificity, which is the ROC curve. An ROC curve can be generated by modifying the classification threshold from 0 to 1 in small steps, and measuring sensitivity and specificity for each value of the threshold. Finally, we can compute the ROC to get a value for the overall performance of the classifier.

D. Comparing Accuracies and Result

In this experiment, we have evaluated the deep learning approaches with stacked auto-encoder and restricted Boltzmann machines. In order to compare the results obtained from our proposed deep learning methods, different related work applying machine learning approaches as ensemble of decision tree (EDT) to the same dataset have been used [24]. The experiment is conducted for 100 iterations for binary classification with 10 input features and two output features. From the results of confusion matrix and ROC, it can be clearly observed that the RBM is better than SAE. It is illustrated in Fig. 2 that the ROC value results of SAE model for training process, at about 0.8183.

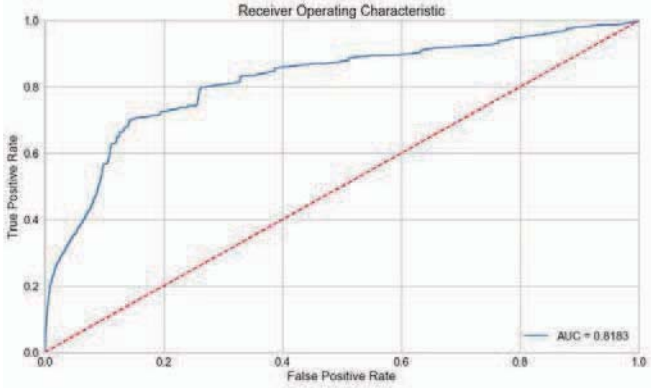

Fig. 2. ROC result for training SAE model

Confusion matrix contains the number of actual and predicted classifications achieved by the classifier. TABLE V. displays the confusion matrix results of the run performance during the testing phase. 690,442 and 1,551 records were correctly classified as genuine and fraud respectively, with 610 records being misclassified.

TABLE V. CONFUSION MATRIX OF SAE TESTING PHASE

Confusion Matrix	Genuine	Fraud
Genuine	690442	64
Fraud	546	1551

TABLE VI. shows the confusion matrix of the run achieved during the testing phase for restricted Boltzmann machine, where 682,586 and 994 records were correctly classified as normal and attacks respectively, with only 131 records being misclassified.

TABLE VI. CONFUSION MATRIX OF RBM TESTING PHASE

Confusion Matrix	Genuine	Fraud
Genuine	682586	45
Fraud	86	994

V. CONCLUSION

It is necessary that banking or credit card companies should be able to recognize fraudulent transactions, so that customers are not charged for items that they did not purchase. In this research, there are many parameters need to be trained for deep neural networks. What's more, our proposed network layer is large and the training process includes 100 iterations, it will take a lot of time, hence we chose to use GPU acceleration technology to speed up the experiment process, the ultimate goal is to build a system similar to real-time intrusion detection with the real network dataset.

In this paper, we demonstrated a real-time synthetic financial fraud detection system using stacked auto encoder (SAE) and restricted Boltzmann machines (RBM) that could be a good contribution to the field of anomaly based network intrusion classification. As is illustrated in TABLE VII., the proposed RBM based fraud detection system achieved around 0.92 testing accuracy for only 30 % of total dataset and a hundred iterations. Moreover, this approach was compared with proposed SAE approach and it is shown that RBM model performs noticeably better compare to SAE, accuracy with about 0.81, by comparing their ROC and confusion matrix results. In the near future, we would like to apply the convolutional neural network (CNN) method to the real network for detecting real-time fraud transactions with high accuracy.

TABLE VII. COMPARISON OF DIFFERENT METHOD'S DETECTION RATE ACCURACY RESULTS

DR Accuracy (%)	Comparison of Accuracies		
	DT (%)	*SAE (%)*	*RBM (%)*
Training set	91.76	81.83	92.86
Test Set	90.49	80.52	91.53

ACKNOWLEDGMENT

Thanks Kaggle platform for providing the various and high-challenging competitions that requires deeper research.

REFERENCES

[1] E. Adali, Bilgisayar ve Bilgi Guvenligi Yonetimi, 1st ed, Turkey, 2016.

[2] S. Bosworth, M. E. Kabay, Computer Security Handbook, 6th ed, Willey, USA, 2014.

[3] C. Eric, Network Security Bible, 2nd ed, John Wiley & Sons, 2009.

[4] A. Ghorbani, L. Wei, Network Intrusion Detection and Prevention, Advances in information security, Springer, Canada, 2010.

[5] K. Jihyun, K. Jaehyun, L. T. Huong, and K. Howon, "Long short term memory recurrent neural network classifier for intrusion detection," IEEE International Conference on PLATCON, South Korea: Jeju, February 2016.

[6] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," IEEE International Conference on CSE and EUC, China: Guangzhou, July 2017, pp. 639–642.

[7] N. T. Van, T. N. Thinh, and L. T. Sach, "An anomaly-based network intrusion detection system using deep learning," IEEE International Conference on ICSSE, Vietnam: Hochiminh, July 2017, pp. 210–214.

[8] T. Ishitaki, R. Obukata, T. Oda, and L. Barolli, "Application of deep recurrent neural networks for prediction of user behavior in tor networks," IEEE 31st International Conference on WAINA, Taiwan: Taipei, March 2017, pp. 238–243.

[9] N. Shone , T. N. Ngoc, V. D. Phai , and Q. Shi, "A deep learning approach to network intrusion detection," IEEE International Conference on emerging topics in computational intelligence, January 2018, vol. 2, pp. 41–50.

[10] E. K. Daniel, M. Gofman, "Comparison of shallow and deep neural networks for network intrusion detection," IEEE 8th International Conference on CCWC, USA: Lasvegas, January 2018, pp. 204–208.

[11] F. Fahimeh, H. Jukk, "A deep auto-encoder based approach for intrusion detection system," IEEE 20th International Conference on ICACT, South Korea, February 2018, pp. 178–183.

[12] J. O. Awoyemi, A. O. Adetunmbi, S. A. Oluwadare, "Credit card fraud detection using machine learning techniques," IEEE International Conference on ICCNI, Nigeria: Lagos, December 2017.

[13] C. Anusorn, "Credit card fraud detection using RUS and MRN algorithms," International Conference on MITiCON, Thailand, 2016, pp. 73–76.

[14] Y. Pandey, "Credit card fraud detection using deep learning," International Journal of ARCS, India: Delhi, June 2017, vol. 8, pp. 981–984.

[15] G. Rushin, C. Stancil, M. Sun, S. Adams, and P. Beling, "Horse race analysis in credit card fraud–deep learning, logistic regression, and gradient boosted tree," IEEE International Conference, USA: Virginia, 2017, pp.118–121.

[16] D. Sumeet and D. Xian, Data Mining and Machine Learning in Cybersecurity, Taylor and Francis Group, London, 2011.

[17] E. Alpaydın, Introduction to Machine Learning, 2nd ed, Massachusetts Institute of Technology, London, 2010.

[18] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, USA, 2006.

[19] P. Vincent, H. Larochelle, I. Lajoie, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, Canada, 2010, pp. 3371-3408.

[20] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," JMLR Workshop and Conference Procedings, Canada, 2012, pp. 37–50.

[21] M. Z. Alom , V. Bontupalli, and T. M. Taha "Intrusion detection using deep belief networks," IEEE International Conference on NAECON, USA, June 2015, pp. 339–344.

[22] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," IEEE International Conference on CCSI, USA: North Dakota State University, December 2017.

[23] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "Paysim: a financial mobile money simulator for fraud detection," International Conference on Research Gate, Norway: Norwegian University of Science and Technology, September 2016.

[24] https://www.kaggle.com/ntnu-testimon/paysim1/data

[25] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," IEEE International Conference on NAECON, USA, June 2017, pp. 63–69.

[26] https://www.kaggle.com/arjunjoshua/predicting-fraud-in-financial-payment-services

[27] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the ROC curve," Springer Science and Business Media, June 2009, pp. 103–123.