Chapter 9

Book Chapter

Fintech: Self Organizing Maps for Fraud Detection

Amit Kumar Arora *1 and Shweta Gupta †2

[1]Associate Professor, KIET School of Management, KIET Group of Institutions, Ghaziabad, U.P.
[2]Assistant Professor, Sharda School of Business Studies, Sharda University, Greater Noida, U.P.

Abstract

Our relationships with the outside world have changed as a result of digitalization, which has also created new growth potential and drastically changed the banking sector. Large volumes of data were created as banks made the switch to digital operations; as of right now, the internet contains more than 44 zettabytes of data. This change brought about new vulnerabilities and enhanced efficiency, but it also exposed the financial sector to never-before-seen levels of fraud. In order to overcome this difficulty, machine learning becomes a vital instrument for spotting and stopping fraud. Large, precisely labeled datasets are necessary for standard machine learning techniques, but obtaining them can be challenging and time-consuming. This problem is avoided by deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which learn from raw data without explicit labeling. This allows for the development of reliable fraud detection systems. This chapter provides an account on Self-Organizing Maps (SOMs), a powerful deep-learning method that performs exceptionally well in grouping and dimensionality reduction.

Keywords: Financial Fraud. Detection. Deep Learning. Fintech. Self-Organizing Maps.

*Email: amitaroraicwa@gmail.com Corresponding Author
†Email: dr.shwetagupta04@gmail.com

1 Introduction

The financial industry has seen a variety of fraudulent actions in the financial markets recently, which has put professionals and auditors—who are responsible for assuring accuracy and transparency—in a difficult situation (Mittal, Kaur, & Gupta, 2021). Cyber attacks on banks have grown more frequent. Nearly 30,000 cyber attacks on banks have been documented globally in 2022. Financial damages from these assaults are also increasing, and are expected to exceed $10 billion by 2022. Phishing is the most popular sort of cyber attack targeting banks. Phishing is a sort of social engineering crime in which emails or text messages act like to come from an official source, such as a bank.These emails or text messages frequently include a link that, when clicked, redirects the victim to a phony website that seems to be the genuine bank's website. The attacker can take the victim's personal information if they input it on the bogus website (Alkhalil et al., 2021). Digitization has a strong impact on the financial services industry. We are witnessing a sea-change in Information and communication technology (ICT) reshaping the ways we interact and perform activities in everyday life. The term financial technology or short. FinTech reflects this development of an IT-induced transformation (Puschmann, 2017).Nowadays, credit card fraud detection is of great importance to finan-cial institutions (Zaslavsky & Strizhak, 2006). With the advancements in technology, the probability of online fraud and manipulation has also increased. There is a huge discussion about deep learning models in the community, as complex and unstructured data, manual computation gets more and more expensive with the use of the traditional (On paper method).

There are plenty of stories, of the wrong person caught due to human errors, hence where computers and artificial intelligence come in, with the power of computers and new learning techniques computers can identify irregularities in the data ("You are not similar to me, you are not a part of my family"), with time new techniques to identify these gaps in the data immerged, our main focus on this study brings one of these deep learning models to Fintech and help to identify these gaps without manually interpreting details of each account this is where computational power comes, surfing on this huge dataset and finding out outliers in the data or data points that don t make any sense. This is where machine intelligence comes into play, with this new field of innovative informative tech, before getting into Self Organizing we need to understand what K-Nearest Neighbor Algorithm is and how Self Organizing map solves its limitations.

2 Literature Review

Chicco, Napoli, and Piglione's (2003) published a paper on the application of clustering algorithms and self-organizing maps to classify Electricity Customers, the paper talks about the competition among the electricity markets distribution of service providers,

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

they proved that with the help of fuzzy-K-NN, hierarchical clustering, and Self Organizing Maps group together the identified customer patterns which exhibit load diagrams. The researchers had to examine 234 Non-Residential Customers. For this set valuable indicators to records quality data which were usually property-based, They came to conclusion that follow the leader and hierarchical clustering out performed all the other algorithms on the data, as the algorithms provided with highly detailed and separated the clusters, that isolated the load patterns with the unsettling or uncommon behavioral patterns where follow the leader algorithm toped the spot, measured on clusters adequacy and computational speed. Due to its isolation feature its performs best on such dataset.

Globally, the application of artificial intelligence has significantly improved both traditional methods' and a machine's capacity to deal with the manipulation of financial information (Mehta et al., 2022).Abdulsattar and Hammad's (2020) in the paper talked about how advancement in e-commerce had an explosion in number of credit and debit card which is a comprehensive derive, they proposed that we can use machine learning algorithms like SGD Classifier, DT, RF, J48 and IBK machine learning algorithms to perform on UCSD Data Mining Contest 2009 Dataset for this particular problems they defined a pipeline Dataset Data Preprocessing Training Classifiers Testing Classifiers and Fraudulent Detection Legitimate or Fraudulent. They concluded that after the classifications they evaluated the models through evaluation matrixes like Confusion Matrix, Precision Score, Recall, F-measure, Kappa Statistics, MAE and RMSE, MCC and more which resulted in classifications were 97-98% accurate. Based on the Kappa evaluation Random Forest outperformed all the algorithms. In the paper by Mongwe and Malan's (2020) they have discussed the fraud in the financial bodies the total amount estimated to be about a Billion Dollars, this paper tries to asses the efficiency of the Financial Ratios in finding frauds in the financial statements at the local authorities. They try to make this into an unsupervised learning problem, as they might get handy in fraud detection in the public sector, after the training of their SOMs, the researchers conclude that financial ratios are useful in detecting frauds.

Organisational behavioral factors at the human-technology interface can help small and medium-sized firms (SMEs) embrace artificial intelligence (AI). The authors concentrate on the influence of AI deployment on sustainable practices and supply chain resilience (SCR) (Perifanis & Kitsios, 2023). One of the most important conclusions is that leadership is critical in pushing AI adoption in these SMEs. Effective leadership is seen in the establishment of a data-driven and digital organizational culture suitable to AI implementation. Furthermore, great leadership improves staff skills and competencies, which contributes to the successful implementation of AI (Lingam & Vanishree, 2024).

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

3 Method

3.1 K-Nearest Neighbours

K-Nearest Neighbors is the simplest algorithm of all the data algorithms available in the field of Data Science usually applied to find patterns in the data specifically used in classification problems (Bansal, Goyal, & Choudhary, 2022).Although it produces astonishing results usually domain-specific in nature for which the person forging the data for fitting in the algorithm, as the performance of the algorithm highly depends on the distance matrix usually talking about the L1 & L2 methods for calculating the distance as shown in figure 1. The approaches used the difference between 2 numeric values or a difference in points on a Cartesian plane using Euclidian Distance.

Euclidian Distance:$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} \left(w_r \left(a_r(x_i) - a_r(x_j)\right)^2 \right)}$

As shown in the Figure 1 below –

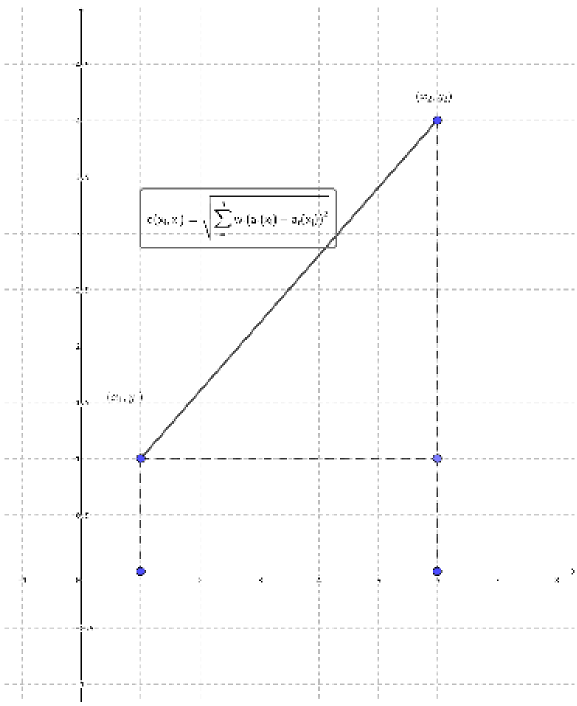

Figure 1. Euclidean Distance

Where $x = (a_1, \ldots, a_n)$ is the dimensionality of the feature vector, a_r is the rth attribute

Editors: Ankur Agrawal and Sadhana Tiwari
DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

and w_r is the weight of the rth attribute ranging from 1 to n. Classes are determined by the votes for each k nearest neighbor:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} \left(w_r \left(a_r(x_i) - a_r(x_j) \right)^2 \right)}$$

Where d_i is the testing example for which the class has to be estimated, x_j is its nearest neighbor in the training set, $y(x_j, c_k)$ tells which class x_j belongs to. The above equation predicts the classes based on the number of members in the nearest class:

$$d_i = \arg\max_k \sum_{x_j \in \mathrm{NN}} y(x_j, c_k)$$

The limitation of KNN algorithm:

- Strict Decision Boundaries: In the predicting process as algorithm strictly classified the data into k neighbors but in case if there is a data point that lies exactly at the middle of 2 or more neighbor decision boundaries it has to select one of the classes it cannot go around by telling it s a 50-50 probability, which reduces that accuracy of the model.
- Labeled Data Required: KNN being a supervised learning models which requires pre-labeled data for training of the data after which predictions need to be done hence for preparing such a labeled classification, the data needs to be prepared by manually analyzing these frauds and classing them to be a fraud or a regular transaction.

3.2 Artificial Neural Networks

ANN is an ML algorithm that tries to tries to mimic the synchronic bhaviors of human brains adapting to the envirments as iterations for training in a particular environment, its consist of neurons interconnected with each other, which consists of 3 layers namely Input Layer, Hidden Layer and an Output Layer, if ANN works when the inputs are initialized and multiplied with a weights and a small about of bias which prevents the model from memorizing the data, which through a output which is then compared to the original value and cost functions is calculated which then feed back in the inner layers through backprogation and weight are modified accordingly (Dey et al., 2023).

3.3 Self Organising Maps

Self-Organizing Maps were first introduced by Kohonen's (2001), a Finnish academic, who made a huge contribution to the artificial neural network, some of his other algorithms also include the Learning Vector Quantization algorithm, which is a supervised classification technique, theories of distributed associative memory, it s a super memory system that is

Editors: Ankur Agrawal and Sadhana Tiwari
DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

capable of searching through every high dimensional application, as in this memory the data is stored in the cache and memory forms. But the most famous and know contribution is the self-organizing maps.

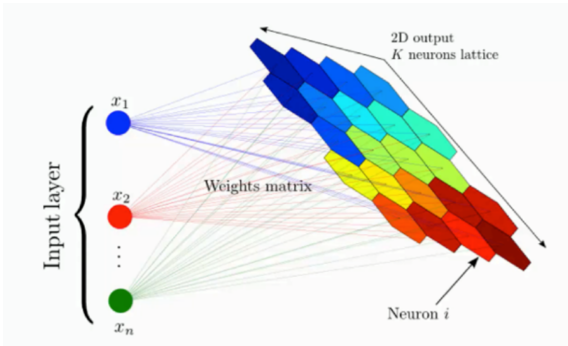

Figure 2. SOM Working

The SOM or the Kohonen Map starts with a hypothesis that the model is based on learning rules and interactions, the processing units maintain the proximity as they grow.(see figure 2). The algorithm uses neurons (array of post synaptic neurons).Van Hulle's (2012) demonstrated with input vectors that are connected to each of these units of the Lattice (Map), with some weights assigned to these attachments, as iterations go by the weights are then updated according and the whole process goes on and on till the models converge and no new points converge into the neighborhood, On this lattice, there is a concept of neighborhoods, as the neurons are interconnected to each other, each neuron has its neighbors as the whole concepts of SOM is that the neurons grow together. The goal here is to find the weights that adjacent values have similar values, and centroids with most neighbors become clusters and we can find irregularities in the data that don't lie in the vectors space or points that are too far from the neighborhood or in the wrong neighborhood. The objective of the SoM is to find weight values such that the adjacent values have similar values, then inputs are assigned to each unit that similar to the unit, the objective here is to be as close as to the input but we are not copying it all, then each identified unit becomes a center of the clusters.

Given input x, find the ith term with the closest weight vectors by competition, which means we fix the centroid and the data movies around it, for each unit j in the neighborhood Ni of the winning neuron i we update the weight j (wj) keeping in mind weights outside the Ni are not updated. SOM has 3 Stages –> Compatition – Collaboration – Weight updates (Van Hulle, 2012).

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

Stage 1 - Competition
Find the most similar unit:

$$i(x) = \arg\max_j \|x - w_j\|_2 \quad \text{where } j = 1, \ldots, m \text{ and } m \neq \text{units}$$

Stage 2 – Collaboration
Use the lateral distance d_{ij} between the winner unit i and unit j:

$$h_{ij}(d_{ij}) = \exp\left(\frac{-d_{ij}}{2\delta^2}\right)$$

which is also known as the Gaussian Neighborhood.

$$\delta(n) = \delta_0 \exp\left(\frac{-n}{T}\right)$$

where n is the number of iterations and T is a constant.
Stage 3 – Weight Updates

$$w_j(n+1) = w_j(n) + \Delta w_j \quad \text{—- (1)}$$

Where Δw_j is made up of 2 theories, namely Hebb's Rule and the Forgetting Rule:

$$\Delta w_j = \eta y_j X - g(y_j) w_j \quad \text{where } \eta \text{ is a constant}$$

In the above equation $g(y_j) = \eta y_j$ which is also the neighborhood $\eta h_{ij}(X)$.
Now, from (1):

$$w_j(n+1) = w_j(n) + \eta(n) h_{ij}(n)[X - w_j(n)]$$

And:

$$\eta(\eta) = \eta_0 \exp\left(\frac{-n}{T_2}\right)$$

where T_2 is a constant.
Things to Keep in Mind

- Many Iterations are required (Almost 1000 times the data shape)
- Stop when no notisible change is visible
- Takes a long time to converge
- Variable Results (Not all runnings will give the same number of clusters as k is not defined like in K-Neighbours Clusters)

Editors: Ankur Agrawal and Sadhana Tiwari
DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

4 Data Analysis

4.1 Data description

In the methodology section the SOM have been used to detect anamolies in the dataset which has been dicusses thoroughly in the sections below Credit Card Transactions Fraud Detection Dataset was utilised in this research, which contains credit card approval records with 18 attributes. A brief description about the attributes are given below :

- cc_num - Account number of the person making the transaction
- merchant - Merchant ID receiving the transaction
- category - Type of the shop
- amt - Amount of the transaction
- gender - Gender of the person making the payment
- street - Street address of the merchant
- city - City where the merchant's shop is located
- state - State where the merchant's shop is located
- zip - Zip code
- lat - Latitude of the person making the transaction
- long - Longitude of the person making the transaction
- city_pop - Population of the city where the transaction takes place
- job - Job of the person making the transaction
- trans_num - Transaction number
- unix_time - Unix timestamp of the transaction
- merch_lat - Latitude of the merchant's shop
- merch_long - Longitude of the merchant's shop
- is_fraud - Indicator of whether the transaction is fraudulent

4.2 Data Preprocessing

To prepare the data, exploratory analysis is performed initially, which includes looking at summary statistics from the dataset, such as the number of occurrences, characteristics, accepted credits, and rejected credits. The Kolmogorov Smirnov test and the Scipy library are then used to verify the data for normalcy. Because the data is not normally distributed, and after checking the splits between genuine and fraudulent entries 50-50 sampling needed to done which is a classic case of over sampling data. After the sampling of the the data we tend to reduce the data between 0 and 1 using the standard scaller after which the data is ready to fit the model, after extracting the accounts that might be misread as frauds and some genuine.

The analyzer can print out these fraud cases and hand them over to the inspection

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

teams to look into these accounts and people and get the frauds out (Figure1), for this minisom library is used to create the SOM, and extracts the relevant frauds from the map being produced.

4.3 Training the SOM

For training a Self-Organizing Map (SOM), we can either build one from scratch or use a pre-built Python library. In this paper, we will use a pre-built library named MiniSom. The user can install the library with pip. There are several steps to training a SOM:

Step 1 – Initializing the MiniSom Library

After importing the MiniSom library, we need to initialize the SOM by setting the required hyperparameters: x and y (the dimensions of the grid), and input_length (the number of variables on which the model will be trained, in this case, 15).

Step 2 – Randomly Initializing the Weights of the SOM

We will use the random_weights_init function to randomize the weights of the SOM, which will be adjusted accordingly as the model iterates multiple times over the dataset and finally converges.

Step 3 – Running the Iterations

After setting the parameters and initializing the weights of the model, we train the model. As it completes, it produces the winning nodes.

Step 4 – Visualizing the Results to Identify Outliers

In Figure 3, colors in the boxes depict the mean interval distances (MIDs) between nodes. We use the pylab library to make the plot that represents the MIDs of the points. The further the MID, the higher the chances of the account being fraudulent. The brighter the box or portion, the further the node is from its neighborhood, and it is considered a potential fraud. The green boxes represent people whose credits were approved, and the red circles represent those that were rejected.

Step 5 – Extracting the Customer IDs

To get the Customer IDs, we need to extract the IDs from the map. The SOM has an inbuilt function win_map that extracts the data behind the color grid. By concatenating the mapped coordinates, we can get the fraud IDs. Analysts can also match timestamps and check all these accounts to gain a better understanding of the data.

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

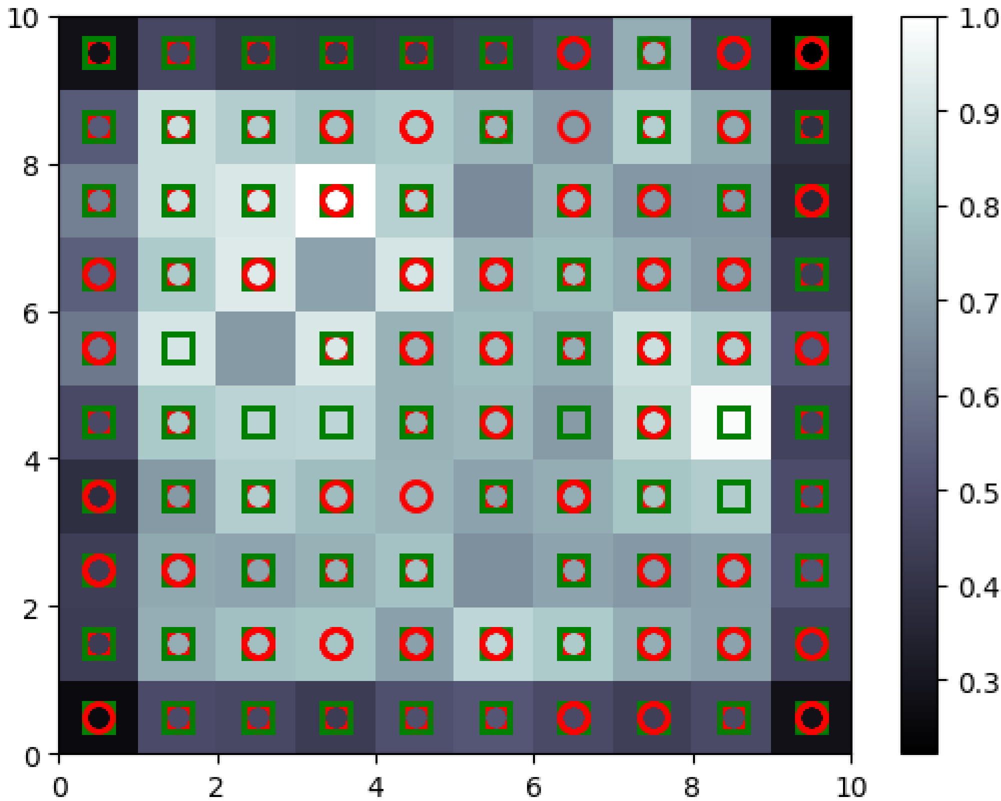

Figure 3

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2 | Copyright ©2024 QTanalytics®

Table 1. Fraud Customers Table

cc_n	merc	cate	amt	G	stre	city	state	zip	lat	long	city_job	tran	unix	mer	merc1	merc2	
###	352	4	369	0	693	281	14	###	38	-89	###	396	###	###	39	-89	
###	64	4	348	0	742	461	37	###	44	###	###	443	###	###	44	###	
###	472	4	309	0	742	461	37	###	44	###	443	###	###	###	44	###	
###	20	4	358	0	742	461	37	###	44	###	443	###	###	###	44	###	
###	583	9	51	0	742	461	37	###	44	###	443	###	###	###	44	###	
###	212	3	68	1	924	597	4	###	38	###	43	###	###	###	37	###	
###	63	9	8.8	1	924	597	4	###	38	###	43	###	###	###	38	###	
###	291	3	60	1	924	597	4	###	38	###	###	43	###	###	38	###	
###	452	9	1.7	1	969	812	26	###	49	###	192	205	###	###	50	###	
###	24	2	12	1	622	806	16	###	39	-96	###	88	###	###	39	-96	
###	106	0	8.2	1	909	493	14	###	41	-89	532	42	###	###	41	-89	
###	250	9	25	1	969	812	26	###	49	###	192	205	###	###	49	###	

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

Table 1 contains the extracted accounts that are considered to have potential frauds, on which the banks can conduct investigations to identify the actual fraudulent accounts.

4.4 Evaluating the Fit of the Model

To evaluate fit of the SOM, toplogical error and quantization error the best for the task, they initially tell whether the model depects the inputs effectively and accuracy trained on the dataset, lower error tell how well the model fits into the dataset.

4.4.1 Topological Error

Topological error, sometimes referred to as topographic error or mapping error, quantifies how effectively the SOM maintains the topological relationships of the input data in the map (Birgitta Dresp-Langley John Mwangi Wandeto et al., 2018).The spatial arrangement of data points in the original feature space is referred to as topology. The purpose of a SOM is to maintain as many of the neighborhood relationships of input data points as feasible. The topological defect identifies cases in which this neighborhood association is not correctly retained in the SOM. This error is measured mathematically as the fraction of data instances for which the Best Matching Unit (BMU) (the winning neuron) lacks nearby neurons on the map that belong to the same class or category. In other words, it measures the frequency with which the BMUs of neighboring data instances end up on distinct regions of the map. A reduced topological error suggests that the SOM accurately represents the topological structure of the data. Thus, Topological and quantization errors are both utilized to fine-tune the SOM's training process, measure convergence, and compare the quality of different SOMs for different datasets.

4.4.2 Quantization Error

Quantization error, also known as mean quantization error or representation error, is a metric that quantifies how well the SOM reflects the original input data in the reduced map space.(see figure 4). It evaluates how well the codebook vectors (the weights of neurons) mimic the input data. A lower quantization error indicates that the neurons of the SOM closely match the input data points in the feature space. The average Euclidean distance between each input data point and its corresponding BMU (the neuron that best reflects that data point) is used to compute the quantization error. It essentially estimates how far each input data point must "travel" in the SOM's map space to reach its BMU. A reduced quantization error suggests a better data representation in the SOM. Acording to the error the SOM fits pretty nicely with the data as it is able to represent amost 70% of the data right off the bat which is considered to be pretty good fit for a model.

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

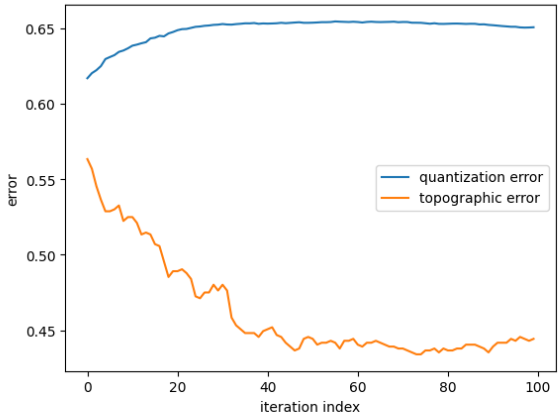

Figure 4. Quantization &Topological Error

5 Findings and Arguments

The key finding from our work is that putting a simple or a super complex dataset, SOMs can be applied to any industry, if put in the correct form, just like the fintech case, these companies can easily charge the banks for detecting potential frauds and charge huge chunks of Data, the same application can be used in the education sectors to finding out potential students who need additional help as they might be passing through unfair means, which is not at all good for them in the long run, and cause problems for them in the future. For detecting fair charges for real estate, and locations too good to be true can be found in case of parameters relating to the price of the property, which might come in handy to rule out potential customers. More trustworthy reports can be formed with greater insights that might be a great asset for any business organization. Businesses can use these methods for suppliers and frequent buyers to understand frauds on supplies and buyers who might be shoplifting, whole data is currently available to the business.

6 Conclusion

We came to understand that the field of AI, Machine Learning, and Deep Learning will be setting the future norms through which we would be living our lives, and it brings great opportunities for businesses to get the bites of these technologies, as the field of advanced machine learning and deep learning models will bring new opportunities and many ways to getting things done, from fraud detection finding innovations into the world. The same was the aim of the project to let the readers know about the potential of simple deep

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

learning models such as the SOMs. If we attach the results we received from this fraud detection and we attach it to Artificial Neural Network for detecting the probability of the frauds of these identified customers the banks can priorities their investigations more efficiently.

References

Abdulsattar, K., & Hammad, M. (2020). Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms. International Conference on Innovation and Intelligence for Informatics, Computing and Technologies. https://doi.org/10.1109/3ICT51146.2020.9312025

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. Frontiers in Computer Science, 3. https://doi.org/10.3389/fcomp.2021.563060

Bansal, M., Goyal, A., & Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. Decision Analytics Journal, 3. https://doi.org/10.1016/j.dajour.2022.100071

Birgitta Dresp-Langley John Mwangi Wandeto, H. O. N., Dresp-langley, B., Wandeto, J. M., & Nyongesa, H. O. (2018). Using the quantization error from Self Organizing Map (SOM) output for fast detection of critical variations in image time series. Des Données à la Décision -From Data to Decision, 2(1), 1–32. https://www.openscience.fr/Using-the-quantization-error-from-Self-Organizing-Map-SOM-output-for-fast

Chicco, G., Napoli, R., & Piglione, F. (2003). Application of clustering algorithms and Self Organising Maps to classify electricity customers. IEEE Bologna PowerTech - Conference Proceedings, 1, 373–379. https://doi.org/10.1109/PTC.2003.1304160

Dey, P. K., Chowdhury, S., Abadie, A., Vann Yaroson, E., & Sarkar, S. (2023). Artificial intelligence-driven supply chain resilience in Vietnamese manufacturing small- and medium-sized enterprises. International Journal of Production Research, 1–40. https://doi.org/10.1080/00207543.2023.2179859

Kohonen, T. (2001). Learning Vector Quantization. In Handbook of brain theory and neural networks (pp. 245–261). MIT Press. https://doi.org/10.1007/978-3-642-56927-2_6

Lingam, M. S., & Vanishree, J. (2024). Leadership in implementing artificial intelligence (AI) for Strategic purposes. International Development Planning Review, 23(1), 1–11. https://idpr.org.uk/index.php/idpr/article/view/104

Mehta, K., Mittal, P., Gupta, P. K., & Tandon, J. K. (2022). Analyzing the Impact of Forensic Accounting in the Detection of Financial Fraud: The Mediating Role

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2

of Artificial Intelligence. International Conference on Innovative Computing and Communications, 585–592. https://doi.org/10.1007/978-981-16-2597-8_50

Mittal, P., Kaur, A., & Gupta, P. K. (2021). THE MEDIATING ROLE of BIG DATA to INFLUENCE PRACTITIONERS to USE FORENSIC ACCOUNTING for FRAUD DETECTION. European Journal of Business Science and Technology, 7(1), 47–58. https://doi.org/10.11118/ejobsat.2021.009

Mongwe, W. T., & Malan, K. M. (2020). The Efficacy of Financial Ratios for Fraud Detection Using Self Organising Maps. IEEE Symposium Series on Computational Intelligence, SSCI, 1100–1106. https://doi.org/10.1109/SSCI47803.2020.9308602

Perifanis, N. A., & Kitsios, F. (2023). Investigating the Influence of Artificial Intelligence on Business Value in the Digital Era of Strategy: A Literature Review. Information (Switzerland), 14(2). https://doi.org/10.3390/info14020085

Puschmann, T. (2017). Fintech. Business and Information Systems Engineering, 59(1), 69–76. https://doi.org/10.1007/s12599-017-0464-6

Van Hulle, M. M. (2012). Self-organizing maps. In Handbook of natural computing (pp. 585–622). Springer Verlag. https://doi.org/10.1007/978-3-540-92910-9_19

Zaslavsky, V., & Strizhak, A. (2006). Credit Card Fraud Detection Using Self-Organizing Maps. Information Security: An International Journal, 18, 48–63. https://doi.org/10.11610/isij.1803

DOI:10.48001/978-81-966500-3-2-9 | ISBN: 978-81-966500-3-2