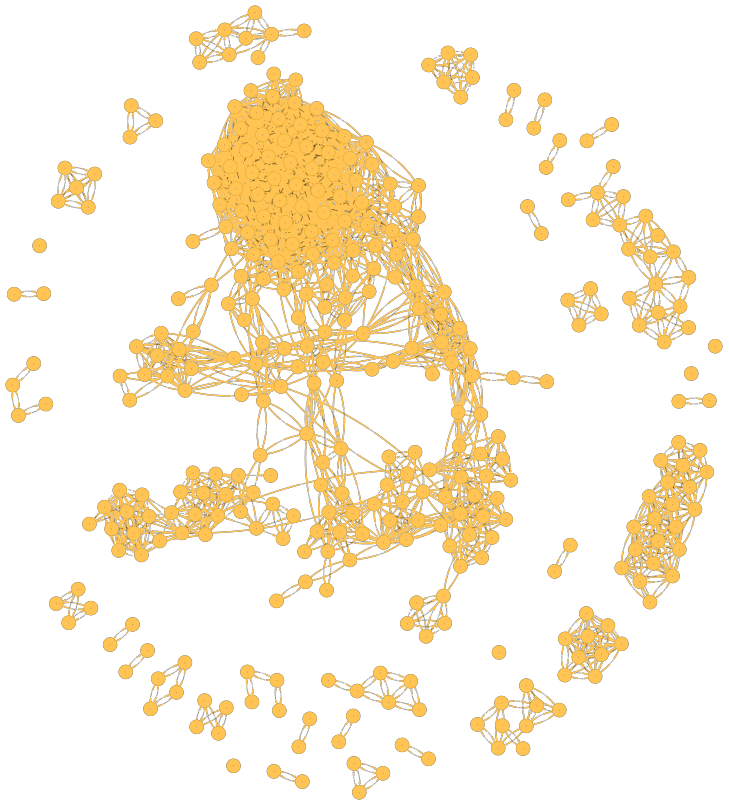
Credit Card Fraud Detection

Approximate Nearest Neighbour Search & Similarity Graph Analysis for Detecting Fraudulent Transactions

How do fraudulent and legitimate transactions contradict in network structure?

Sebastiaan A. M. Kragting



Utrecht University

Universiteit Utrecht

Credit Card Fraud Detection

Approximate Nearest Neighbour Search & Similarity Graph Analysis for Detecting Fraudulent Transactions

by

Sebastiaan A. M. Kragting

Contact	Student Number
S.A.M.Kragting@students.uu.nl	6935559

Supervisor I:	Dr. I. Karnstedt-Hulpus
Supervisor II:	V. Shahrivari Joghan
Examiner:	Prof. Y. Velegrakis
Project Duration:	April, 2022 - July, 2022 (10 Weeks)
Field of Science:	MSc. Applied Data Science
Faculty:	Science, Dept. of Computer Sciences

Cover:	Similarity Graph of Fraudulent Transactions
LaTeX:	https://dzwaneveld.github.io
License:	https://creativecommons.org/licenses/by-nc/4.0/
GitHub:	https://github.com/SebastiaanK97/NetworkSimilarity

Universiteit Utrecht

Preface

Approximate Nearest Neighbour Search
& Similarity Graph Analysis for
Detecting Fraudulent Transactions

The presented document is my thesis on credit card fraud detection which investigates open source data on how fraudulent transactions relate to further credit card transactions. The thesis has been written as closing research to graduate from the masters programme Applied Data Science. The engagement period of the research lasted from April up to July 2022.

The research is undertaken under supervision of Utrecht University whereas the university provided the topic in advance. The further aim of the research question has been formulated out of my own interest in underlying patterns in the data of credit card transactions. The research has been intensive, and the period has been of short duration, yet, the research was truly engrossing and I felt like I accomplished more than expected in the short period of time.

Therefore, I would like to address special thanks to my supervisors for their equal enthusiasm during the process. I both thank Dr. I. Karnstedt-Hulpus and V. Shahrivari Joghan for their discussions and feedback on my process. You have challenged me and I was happy to keep learning during the process.

I also thank my peer students from the masters programme for supporting me and sharing interest in my research. I appreciate the time spent together during the research process and of course I hope to keep seeing you as a friend and during our career. I would also like to thank my family for their stable support and caring when in need. The last year was a bless to me.

I hope the thesis is entertaining and engrossing to read.

Sebastiaan A. M. Kragting
Utrecht, July 2022

Abstract

Approximate Nearest Neighbour Search
& Similarity Graph Analysis for
Detecting Fraudulent Transactions

Fraudulent transactions of credit cards are a major problem for financial institutions and continues to grow along digital transformation. A conventional view states that fraudulent transactions are anomalies. A novel view suggests fraudulent transactions exists within fraud rings. An anonymous, sizeable, and unbalanced dataset of principal component analysis is investigated to juxtapose the perspectives on fraudulent transactions. Approximate nearest neighbour search identifies similar items in terms of Euclidean distance, which is applicable to create similarity graphs. The similarity graphs yield valuable metrics for the classification of fraudulent transactions. The findings in respect to the given approach are as following. First, the assortative mixing between fraudulent transactions is high in similarity graphs. Second, no topological difference exists between fraudulent and legitimate transactions. Third, fraudulent transactions are anomalies but also exist in fraud rings. Fourth, the effect of fraud rings is stronger than the effect of anomalies. Fifth, both perspectives make useful variables for a classification model which is competitive to the state-of-the-art.

Contents

Nomenclature

The following nomenclature describes the abbreviations and mathematical symbols used.

Abbreviations

Abbreviation	Definition
AIC	Akaike information criterion
ANN	Approximate Nearest Neighbour
ANN*	Artificial Neural Network
AUC	Area Under the Curve
FDS	Fraud-Detection System
HNSW	Hierarchical Navigable Small World
LOF	Local Outlier Factor
LOOCV	Leave-One-Out Cross-Validation
LR	Logistic Regression
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
RDBMS	Relational Database Management System
ROC	Receiver Operation Characteristics

Symbols

Symbol	Definition	Unit
r	Assortativity Coefficient	[1]
...		
L_2	Euclidean Distance	[1]
...		
ϕ	Fraction of Fraudulent Neighbours	[1]
...		
A	Transaction Amount	[€]
...		

1 Introduction

Online shopping, streaming, and communication services show us digital transformation is crucial to our daily life, and so digital transformation is inevitable for businesses to remain the competitive advantage they strive for. Yet, digital transformation yields various risks, for instance, businesses struggle with system functionality, control deficiency, and cybersecurity issues (Schwertner, 2017). Currently, with the universal adoption of credit cards, a prime concern of financial institutions is fraudulent behaviour from malicious hackers and scammers (Gürsoy & Varol, 2021b). Particularly, first-party credit card fraud costs financial institutions immense capital. Such frauds need to be discerned to avoid charging actual customers for what they did not procure. According to a global statement from the Nilson Report (2021), "over the next ten years, the industry will experience losses to fraud totaling $408.50 billion."

Credit card frauds broadly entail two behavioural traits. First, frauds as anomalies e.g., abnormal spending behaviour, frequency of transactions. Transactions also face restriction over time and space. For instance, a person is impossible to make two transactions within a short period of time in two different locations which are far apart. Such transactions thus are outliers (Kou et al., 2004). Second, frauds as part of a fraud ring i.e., at present criminals tend to collaborate, and therefore fraudulent transactions wire together into clusters of collective fraudulent behaviour, known as organised crime (Pourhabibi et al., 2020). Especially fraud rings cause a major loss of capital since they "behave very similarly to legitimate customers, until they bust-out, cleaning all their accounts and promptly disappearing". Concurrently, the amount of financial theft grows exponentially with the number of criminals involved in the fraud ring (Sadowski & Rathle, 2014). Notwithstanding, fraud rings are largely overlooked.

Financial institutions aim to invent robust frameworks to detect frauds. Former algorithms are rather rule-based and thence controversial. Though, with the rise of machine learning, current algorithms are less deterministic and learn underlying patterns. Therefore, current fraud-detection systems (FDS) are able to detect real-world fraudulent behaviour through learning from experience (Dal Pozzolo et al., 2017). The Université Libre de Bruxelles (2018) launched a Kaggle competition to contest machine learning algorithms and optimise automatic fraud detection solutions. Ongoing logistic regression (LR) and multilayer perceptron (MLP) models reach 93% and 97% $recall$ respectively, despite that, $precision$ remains futile.

Yet, frameworks tend to neglect connections and disconnections between transactions and therefore abandon causal information of the two fraud categories i.e., anomalies and fraud rings. Moreover, contemporary frameworks detect frauds posterior, but sophisticated algorithms necessitate exploring credit card transactions and identifying fraud rings a priori, before they bust-out. Therefore, exploratory research is essential to understand what credit card frauds constitute in topological means, and how criminals contradict legitimate customers.

1.1. Research Question

Accordingly, this research adopts the perspective of graphs to discern anomalies and fraud rings. In contradiction to research in the classification of frauds, the emphasis is put on unraveling patterns and associations within a network of credit card transactions. Hence, the research question, from the perspective of a graph, alludes:

How do fraudulent and legitimate transactions contradict in network structure?

To answer the research question a data set of 284,807 credit card transactions entailing 492 (0.172%) frauds is initialised from Kaggle (Université Libre de Bruxelles, 2018). Note, the data set is anonymous, sizeable, and imbalanced, and therefore various challenges need to be resolved before being able to do an analysis. First, the data is anonymous through vectorisation as a consequence of Principal Component Analysis (PCA). Thus, network relations do not exist yet, but instead, distances between vectors are computable. Second, the data is sizeable, and therefore computing all distances between each of the vectors is unfeasible due to the exponential growth in terms of computational complexity. Third, the classes of fraudulent and legitimate transactions are imbalanced, therefore conversion is requisite for proper analysis. Fourth, efficient querying is requisite to reduce the computational abundance.

Whilst analysis is initiated as inductive, once intriguing contrasts between fraudulent and legitimate transactions arise, deductive analysis inclines (Bryman, 2016). On the whole, the research consists of three parts.

First, approximate nearest neighbor (ANN) search to establish relationships between credit card transactions with near-linear computational complexity for scalability (J. Wang et al., 2021). Nonetheless, the data after PCA appears to be highly clustered in high dimensional space. Altogether, considering the similarity metric, curse of dimensionality, the size and clustering of the data, an appropriate algorithm must accommodate for the nature of the data. Once a number of neighbours for each node is discerned, a similarity graph structure arises. Therewith, graph analysis becomes applicable for detecting fraudulent behaviour.

Second, exploratory research to investigate patterns in the similarity graph. Specifically, a metric to express the existence of fraud rings would be the assortativity coefficient. Further exploratory research aims to discover clusters of fraudulent transactions. The exploratory research ultimately leads to the formation of hypotheses.

Third, three hypotheses arise from literature and exploratory research. The first two hypotheses express that fraudulent transactions are anomalies and exist in fraud rings. The third hypotheses assumes fraudulent and legitimate transactions differ in topological structure. In synchrony with the hypotheses, classifications models are built using significant metrics only. Last but not least, statistical inference is done to juxtapose anomalies with fraud rings.

1.2. Research Scope

The relevance of research is the prevention of theft and thereby cease loss of both public and private capital. Thus, solving cybersecurity issues in the domain of finance yields both societal and organisational relevance. Further organisational relevance lies in changing perspective from rule-based FDS to understanding connections between frauds, and so reducing dependency on large quantities of historical data. If fraudulent behaviour can be understood well, then real-time graph traversal is helpful to detect fraud rings, before they are ”burst-out”.

Scientific relevance is contributing to multiple problems in both the research fields of cybersecurity and computational science, but the essence of the research is in data science.

First, fraudulent behaviour is often poorly understood and therefore current algorithms neglect actual relationships. In other words, current machine learning models such as Artificial Neural Networks and Random Forests are too deterministic because they are overfitting to patterns within the black-box.

Second, the preferential tendency of connections within fraudulent behaviour is unknown and therefore is an open issue. Computing the assortativity coefficient for fraudulent transactions adds to current understanding of assortative mixing. The case of credit card transactions is remarkable because of the class imbalance and the inherent difference between fraudulent and legitimate transactions.

Third, ANN-search is a state-of-the-art approach to uncover nearby nodes, which is applicable for creating similarity graphs. Researching the viability of similarity graphs is imperative to initiate widespread adoption of the approach and ultimately leads to novel solutions in data science. Specifically, the extraction of metrics from graphs is novel (in open science) and could possibly be superior to conventional metrics.

1.3. Structure

The following chapter kicks-off with a literature review on ANN-search, graph theory, and logistic regression for classification. The chapter finishes with an overview of related work. Subsequently, the data is explored, and discussed in terms of descriptive statistics, visualisations, and associations with a similarity graph structure. The fourth chapter explains the methodology of ANN-search and similarity graph analysis, and how hypotheses occur. The fourth chapter also explains the approach to statistical modeling. The fifth and the sixth chapters are the empirical findings and the conclusions drawn. Last but not least is the discussion and further considerations for future research.

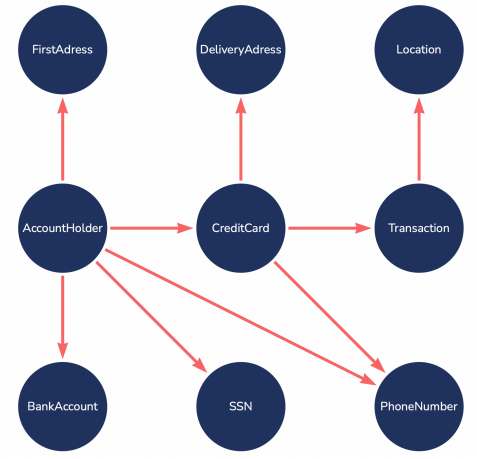
2
Theoretical Foundations

Under fraudulent behaviour within financial institutions exists i.a. credit card, mortgage, and money laundering fraud. This research focuses solely on credit card frauds. Credit card fraud signifies unauthorised procurement of a person's credit card without the person's consent nor conscience. Financial institutions are principally responsible for such theft. Fraudulent transactions may arise due to theft or loss of the credit card, online hacking and scamming, or fake ATM skimming, for instance (West & Bhattacharya, 2016). A regular graph model around the data of credit card transactions is depicted in Figure 2.1a.

2.1. Similarity Graphs

In this research, relationships between transactions are of interest, which requires the reconstruction of data. The reconstructed data alternates existence of relationships i.e., relationships do not exist yet, however, similar items are identifiable through searching for nearby data in vector space. Finding similar transactions in sizeable data is a problem for ANN-search (J. Wang et al., 2021). The similarity graph model is depicted in Figure 2.1b and is meant to artificially recreate the graph model in Figure 2.1a as a (distance) similarity graph.

Specifically, once a top-k of nearby neighbours is established through ANN-search, a graph structure emerges using transactions as nodes with undirected relationships between neighbouring vectors, therefore the graph is unipartite. Each transaction $n \in N$ has an equal amount of relationships to the top-k search, which is theoretically referring to the out-degree $d_n^+ = k$. Each relationship carries a weight of the inverse distance $1/L_2$ between vectors, the weight represents the similarity between nodes.



(a) Bank account information

(b) Similar transactions

Figure 2.1: Comparison of two graph models (Bastani, 2013).

2.2. Approximate Nearest Neighbour Search

If all similarities between each of all the daily transactions would be calculated (i.e., naive-kNN) then the computation time $O(n^2)$ would range up to 10^6 seconds i.e., weeks, which is not feasible. Therefore, ANN-search provides finding a top-k of similar items in near-linear computational complexity $O(n^{1+\epsilon})$ where $\epsilon > 0$ (Li et al., 2019). To do so, data is considered in vector space, whereas each transaction represents a vector.

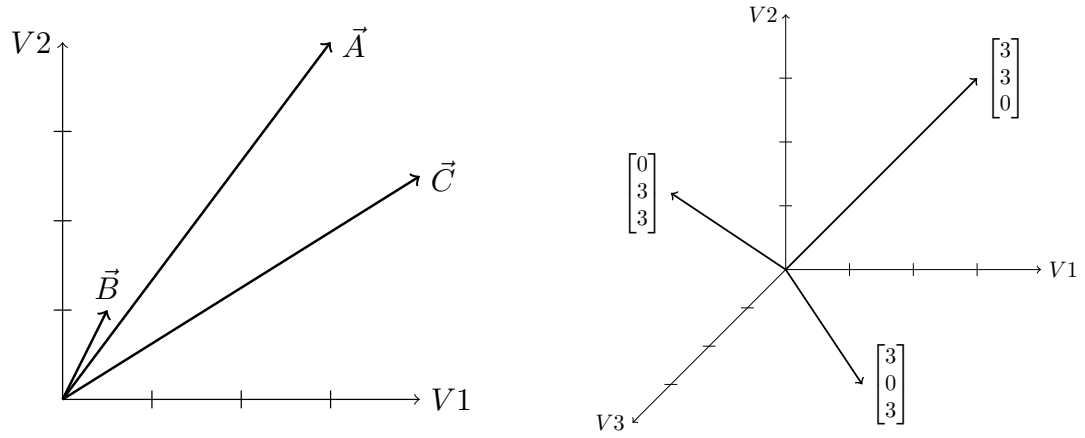
2.2.1. Vectors & Euclidean Geometry

In mathematical terms, a vector is a spatial relationship with a direction and a magnitude. In computer science, a vector is recognised as an array data structure. Vectors can be similar in various manners, two accessible methods are cosine similarity $cos(\theta)$ and Euclidean distance L_2, those metrics are expressed as, whereas V-dimensions exist of v:

$$cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{v=1}^{V} a_v \cdot b_v}{\sqrt{\sum_{v=1}^{V} a_v^2} \cdot \sqrt{\sum_{v=1}^{V} b_v^2}}, \tag{2.1}$$

$$L_2(A, B) = \sqrt{\sum_{v=1}^{V} (a_v - b_v)^2}. \tag{2.2}$$

Clearly, these metrics are inherently different, but since outliers are reliant on the magnitude of the vector, and not the angle, the cosine similarity is detrimental. Figure 2.2a depicts that vector $\vec{A}$ and $\vec{C}$ are approximate in distance L_2, whilst the angle $cos(\theta)$ is larger than for $\vec{A}$ and $\vec{B}$. Thus, we consider the Euclidean distance L_2 as the similarity metric of interest.



(a) Comparison of cosine similarity and Euclidean distance

(b) Curse of dimensionality in vector search

Figure 2.2: Visualisation of vector space.

Figure 2.2b depicts an issue of dimensionality in ANN-search, the distance between vectors is equal whilst orientations differ (Indyk & Motwani, 1998). If dimensionality and cardinality is high, then numerous quasi-identical vectors subsist over various dimensions. Consequentially, query performance degrades as dimensionality increases (Weber et al., 1998).

2.2.2. Hopkins Statistic

Aside from the curse of dimensionality, data can be clustered. Both the curse of dimensionality and clustering lead to searching through dense vector space, which is emanating further complexities. The Hopkins Statistic is a measure of clustering tendency which is applicable to vectors (Hopkins & Skellam, 1954):

$$H = \frac{\sum_{i=1}^{n} u_i^d}{\sum_{i=1}^{n} w_i^d + \sum_{i=1}^{n} u_i^d} \tag{2.3}$$

Here, a random sample n from data X with features (i.e., dimensions) x_i. For an equal number of dimensions and entities, a set Y is generated with a uniform but random distribution of vectors. In Equation 2.3, u_i represents the distance of $y_i \in Y$ to the nearest neighbour in X, and w_i is the distance of the sample $x_i \in X$ to the nearest neighbour in X. In other words, the Hopkins Statistic signifies nearby vectors in comparison to the uniform random distribution, which is 1.00 for complete clustering and 0.00 for an absolute uniform distribution (Banerjee & Dave, 2004).

2.2.3. Hierarchical Navigable Small-World Graphs

ANN-search relies on vector indexing for efficiently organising data and optimsing query speed. Four types of vector indexing exist; quantisation, tree, hash, and graph-based indexing. Especially, graph-based indexing focuses on high-speed queries with high $recall$ for memory intensive data. The Hierarchical Navigable Small-World (HNSW) graph is an index of ANN-search which is performing efficiently on clustered data. Herein, a vector index is built on basis of connected neighbourhoods, then greedy heuristics navigate proximity graphs of these connected neighbourhoods for a certain query (Li et al., 2019).

The structure of the HNSW algorithm is a multi-layer graph whereas the query initiates vector search in the top-layer. The search in the initial top-layer is highly approximate but quickly improves over multiple iterations towards lower layers. The structure of the HNSW graph evolves by inserting unseen vectors successively. After acquiring sufficient connections, a relative neighbourhood arises (cluster). Extra connections between neighbourhoods are selected to retain connection to a global component. Therefore, the algorithm remains viable for highly clustered data (Malkov & Yashunin, 2018). The $recall$ of the HNSW algorithm is dependent on the data structure and the size of the database.

Yet, the $recall$ and query speed performance of HNSW are optimisable over three parameters: "maximum degree of nodes on each layer of the graph" (M), search scope of building an index (efC), and search scope of target retrieval (ef) to define a search range.

2.3. Assortative Mixing by Enumerative Characteristics

Assortative mixing by enumerative characteristics is a measure of the difference between the actual number A_E and the expected number E_E of relationships between nodes of identical classes (Newman, 2002). The corresponding assortativity coefficient r is a measure of homophily i.e., the tendency to associate with similar entities. The number of actual relationships between nodes of an identical class is:

$$A_E = \sum_{(i,j)\in E} \delta(c_i, c_j) = \frac{1}{2} \sum_{i,j} a_{i,j} \delta(c_i, c_j) \tag{2.4}$$

where E is the set of relationships in the graph and $a_{i,j}$ is the number of actual relationships between node i and j. The factor one-half accounts for the relationships being undirected. The Kronecker delta mathematically accounts for the nodes to be of identical class:

$$\delta(c_i, c_j) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases} \tag{2.5}$$

The expected number of relationships between nodes of an identical class is a mathematical estimation as if the classes are spread randomly over the graph:

$$E_E = \frac{1}{2} \sum_{i,j} \frac{d_i d_j}{2m} \delta(c_i, c_j) \tag{2.6}$$

m is the number of edges in the graph. Nodes i and j yield a degree d_i and d_j respectively. Thus, $\frac{d_i d_j}{m}$ refers to the expected number of relationships between node i and j. The modularity is a measure of difference between the actual and the expected number of relationships[1]:

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j) \tag{2.7}$$

whereas the maximum possible modularity is the difference between the total and the expected number of relationships:

$$Q_{max} = \frac{1}{2m} \left(2m - \sum_{i,j} \frac{d_i d_j}{2m} \delta(c_i, c_j) \right). \tag{2.8}$$

All in all, normalising modularity results in the assortativity coefficient r:

$$-1 \leq r(E) = \frac{Q}{Q_{max}} = \frac{\sum_{i,j}(a_{i,j} - d_i d_j / 2m)\delta(c_i, c_j)}{2m - \sum_{i,j}(d_i d_j / 2m)\delta(c_i, c_j)} = \frac{A_E - E_E}{m - E_E} \leq 1 \tag{2.9}$$

In this research the above equations require alteration for class imbalance. Appendix E is an explanation of how assortative mixing is relevant to this research.

[1] http://users.dimi.uniud.it/~massimo.franceschet/teaching/datascience/network/assortative.html

2.4. Multinomial Logistic Regression

Multinomial Logistic regression (LR) is applicable to model the probability that an observations belongs to a particular class (James et al., 2013). Hastie et al. (2009) define the probability for binary classification as following:

$$\log\left(\frac{P(X,k)}{1-P(X,k)}\right) = \beta_0 + \sum_{j=1}^{S} X_j \cdot \beta_j \tag{2.10}$$

$$P(X,k) = \frac{e^{\beta_0+\sum_{j=1}^{S} X_j\cdot\beta_j}}{1+e^{\beta_0+\sum_{j=1}^{S} X_j\cdot\beta_j}} \tag{2.11}$$

here the probability is a function of a set of variables S. The model coefficients β_j and variables X_j plus the intercept β_0 determine the probability of the class for an observation.

2.4.1. Binary Classification

Table 2.1 is an overview of classification, which is not a hard cut between classes. Namely, each classification is probabilistic $P(c = f) = 100\% - P(c = l)$. Conventionally, the boundary T between a binary class lies at $P(c = f) > T = 50\%$. Though, this boundary is a parameter to optimise the trade-off between recall and precision. Increasing the boundary would lead to less TP and FP, but would lead to more TN and FN. Vice versa for decreasing the boundary. The terminology and derivations from a confusion matrix for evaluation are as following:

$$PPV(T) = precision = \frac{TP}{TP+FP} \tag{2.12}$$

$$TPR(T) = recall = \frac{TP}{TP+FN} \tag{2.13}$$

$$TNR(T) = specificity = \frac{TN}{TN+FP} = 1 - FPR(T) \tag{2.14}$$

The area under the curve (AUC) from Receiver Operating Characteristic (ROC) curves is for viable model evaluation independent of threshold T. Hand (2009) defines the AUC as:

$$AUC = \int_{v=0}^{1} TPR(FPR^{-1}(v)) \cdot dv = \int_{+\infty}^{-\infty} TPR(T)FPR'(T) \cdot dT \tag{2.15}$$

Thus formally, $recall$ and $specificity$ are functions of boundary T. The integral defines the AUC and is straightforwardly interpretable over the whole model.

Table 2.1: Confusion matrix for binary classification.

		Predicted	
		0	1
Actual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

2.5. Related Work

The data for credit card fraud detection is open for science and therefore is surrounded by publications (Université Libre de Bruxelles, 2018). The major focus lies on solving the problem of imbalanced data through undersampling and oversampling techniques (Dal Pozzolo, 2015; Dal Pozzolo et al., 2015; Dal Pozzolo et al., 2014), further focus lies on the nonstationarity of timely data (Carcillo et al., 2018; Dal Pozzolo et al., 2017). Yet, the core perspective is on fraudulent transactions as anomalies, not as fraud rings.

For instance, anomalies exhibit in the variants of credit card fraud, mobile phone fraud, and cyber attacks. Three types of anomalies exist: point anomaly, contextual anomaly, and collective anomaly (Ahmed et al., 2016). The topical priority of point anomaly detection lies in detecting outlying patterns using Artificial Neural Network, Support Vector Machine, and Random Forest (Gürsoy & Varol, 2021b; Kou et al., 2004). Such models are deterministic because they operate inside the black-box, thus are neglecting underlying causalities. Moreover, models ordinarily score high on $recall$ but $precision$ is futile, therefore considering the AUC is crucial (Dal Pozzolo, 2015). Further work argues the Local Outlier Factor (LOF) captures collective anomalies to a certain extent (Gogoi et al., 2011). kNN algorithms are applicable for FDS (Gürsoy & Varol, 2021b), particularly, kNN is well established for classification tasks (Guo et al., 2003; Zhang et al., 2017). Currently, the problem is finding neighbouring vectors in sizeable data (Deng et al., 2016), and whether the theorem holds on fraudulent credit card transactions is an open issue. Fortunately, ANN-search is offering new opportunities for sizeable data and may overcome the current barriers of kNN classification (J. Wang et al., 2021).

Graphs support detecting fraud rings in real-time through analysing direct relationships (Sadowski & Rathle, 2014). For example, social network analysis allows for finding roles and patterns of money laundering by analysing graph topology and clusters (Dreżewski et al., 2015). In the case of similarity graphs, C. Wang et al. (2019) propose a measure of anomalies for small-scale data using kNN as a parameter. Further, community detection methods deem appropriate to detect fraud rings in similarity graphs (Needham & Hodler, 2018). Howbeit, if the PCA-data in this research is valid for building similarity graphs is an open issue.

Prior research shows cross-validation constructs useful evaluation of real-time FDS (Chen et al., 2005; Thennakoon et al., 2019). Namely, a model is trained on prior knowledge and an assumption made is that credit card transactions enter the database successively. Note, training models on imbalanced data typically requires undersampling of legitimate transactions to produce classifications neutrally. Therefore, leave-one-out cross-validation (LOOCV) appeals for model evaluation utilising an optimum of available resources (Wong, 2015).

All in all, prior research tends to neglect the graph structure of credit card transactions. This research aims to fabricate a similarity graph in order to extract significant metrics. Subsequently, the metrics are fed to logistic regression models for classification and inference. Therefore, the models do not operate within the black-box. Besides, The models become interpretable and thus yield theoretically relevant information. As a result the perspectives of fraudulent transactions as anomalies and fraud rings can be juxtaposed to assess prior research.

3
Data Description

The dataset is retrieved from Kaggle (Université Libre de Bruxelles, 2018) and consists of $284,807$ credit card transactions over two full days entailing 492 (0.172%) frauds. The data set is sizeable, anonymous, and imbalanced. The data set contains the following features: a timestamp, the amount of the transaction, and 28 unidentifiable features as a result of PCA. The 28 features correlate nil. Solely the distance between the vector representations of these 28 features is meant to seek for similar transactions. Table A.1 in Appendix A indicates descriptive statistics of the dataset, several observations are made:

- The class for legitimate transactions is 0, and the class for fraudulent transactions is 1. The data is imbalanced with roughly 1 out of 600 transactions being fraudulent.
- Data is non-standardised, but requires a standard deviation $\sigma = 1$ and mean $\mu = 0$ before computing distances L_2 for equal treatment of vector dimensions.
- The Hopkins Statistic of a random sample after standardisation is consistently 1.00.
    * Note, standardisation of data worsens the clustering issue arising from the curse of dimensionality, increasing the dependence on an appropriate ANN-algorithm.
- The 'Amount' variable ranges from €0.00 to €$25,691.16$, and therefore has a right-skewed distribution with an immense standard deviation.
- Further investigation in raw data of the 28 features indicates the classes show remarkable different distributions. Figure 3.1 depicts three visible cases of anomalous values.

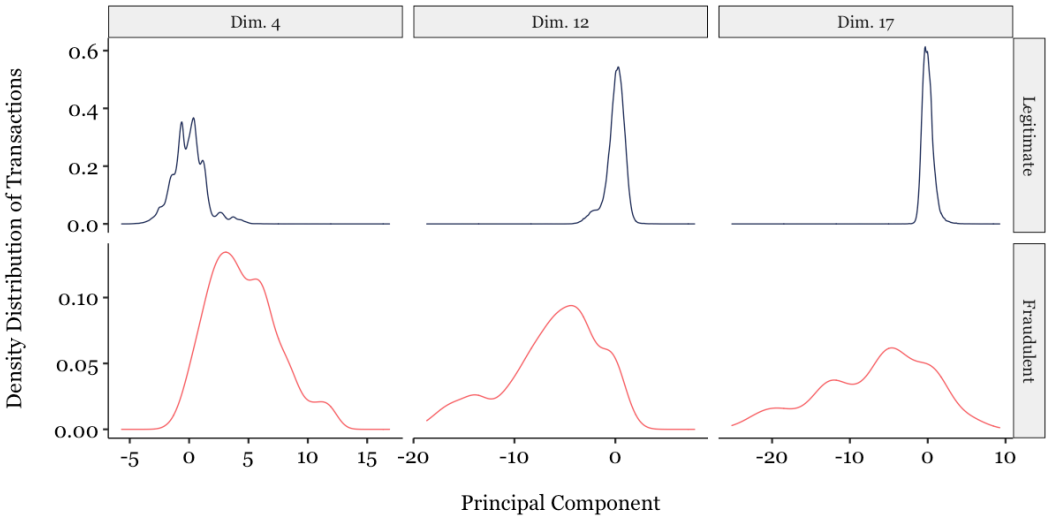


Figure 3.1: Density distributions per class for principal component dimensions four, twelve, and seventeen. Further visualisation in **Figure B.1** in Appendix B shows how the distributions relate in conjunction.

Figure 3.2 visualises the transaction frequency over time. Two fraudulent peak times exist at 12pm on the first day and 5am on the second day, therefore fraudulent behaviour is non-seasonal. Yet, the peak indicates discontinuous fraudulent behaviour as of criminals initiate theft and bust-out before fraudulent transactions are intercepted.

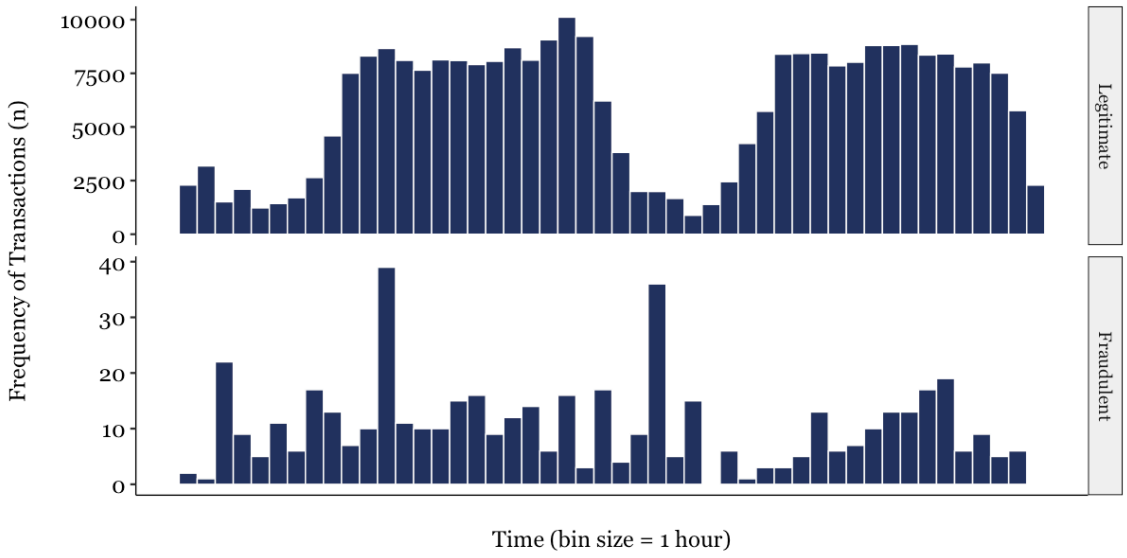


Figure 3.2: Transaction frequency over time per class.

Figure 3.3 visualises the cumulative distribution of transaction amounts. Noticeable is that fraudulent transactions chiefly consists of several cents, €1.00, and €100.00.

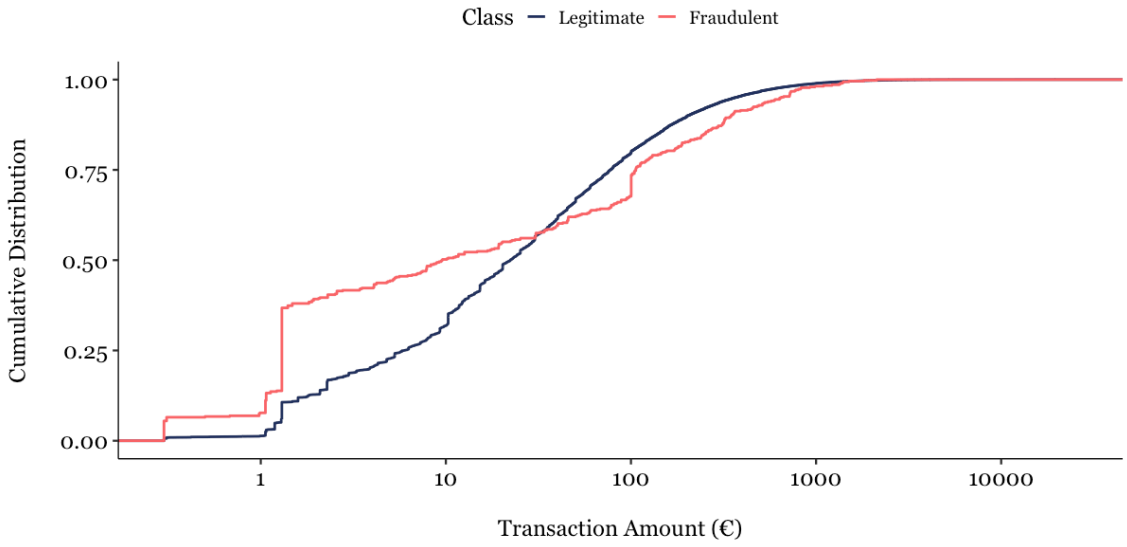


Figure 3.3: Cumulative distribution of transaction amounts per class.

The data in general consists of no missing values, but no statements are disclosed whether all fraudulent transactions are labelled to their corresponding class or whether these cases have not been identified. Since the data set is anonymous no ethical conflicts arise in exploiting the data (Calvino et al., 2017). The data is widely utilised on https://www.kaggle.com.

3.1. Preliminary Research

The given data after PCA yields 28 features and therefore is considerably high in dimensions. Moreover, the data is sizeable due to the high amount of transactions each day. The fact that the Hopkins Statistic is equal to 1.00 is remarkable. The properties of the data are unfavourable for ANN-search. Altogether, considering the similarity metric, curse of dimensionality, the size and clustering of the data, an appropriate ANN-search algorithm must accommodate for the nature of the data. Preliminary research through trial and error reveals that the HNSW graph performs well in comparison to alternative algorithms[1] e.g., IVF_FLAT and ANNOY. See https://github.com/SebastiaanK97/NetworkSimilarity for an explanation on the process.

3.2. Classification for Fraud Detection

Table 3.1 denotes the confusion matrix for binary classification for fraudulent transactions. The labels are as following. TN: A legitimate transaction classified correctly, FP: A fraudulent transaction classified as legitimate; FN: A legitimate transaction classified as fraudulent; TP: A fraudulent transaction classified correctly. The optimisation of the classification is a consequence of boundary T. Increasing the boundary leads to criminals escaping, and decreasing the boundary leads to a higher workload of correcting legitimate transactions.

Table 3.1: Confusion matrix for binary classification.

		Predicted	
		Legitimate	Fraudulent
Actual	Legitimate	True Negative (TN)	False Positive (FP)
	Fraudulent	False Negative (FN)	True Positive (TP)

Generally, the financial institutions' aim is to detect all fraudulent behaviour because the theft outweighs the remuneration of the workload. Thus, the goal is to strive for high $recall$ (Awoyemi et al., 2017; Université Libre de Bruxelles, 2018). Yet, the extent of $recall$ optimisation is a business decision. Besides, a model may perform decently in $recall$ but poorly in $precision$. Therefore, the area under the curve (AUC) from Receiver Operating Characteristic (ROC) curves is a viable model evaluation which is independent of threshold T. Though, for optimisation, $recall$ needs to be taken into account as a specific evaluation measure to grasp the actual aim of the model i.e., ceasing fraudulent behaviour (Hand & Anagnostopoulos, 2013).

[1] https://milvus.io/docs/index.md

4
Methodology

The research proposes analysing credit card transactions from a similarity graph. Figure 4.1 depicts the methodology as a pipeline, which is described throughout this chapter. The methodology yields four main purposes; ANN-search, setting up a similarity graph, extracting significant metrics from the graph, and fitting statistical learning models using these metrics. The purposes are executed exclusively, but, progressing requires the data from prior steps.

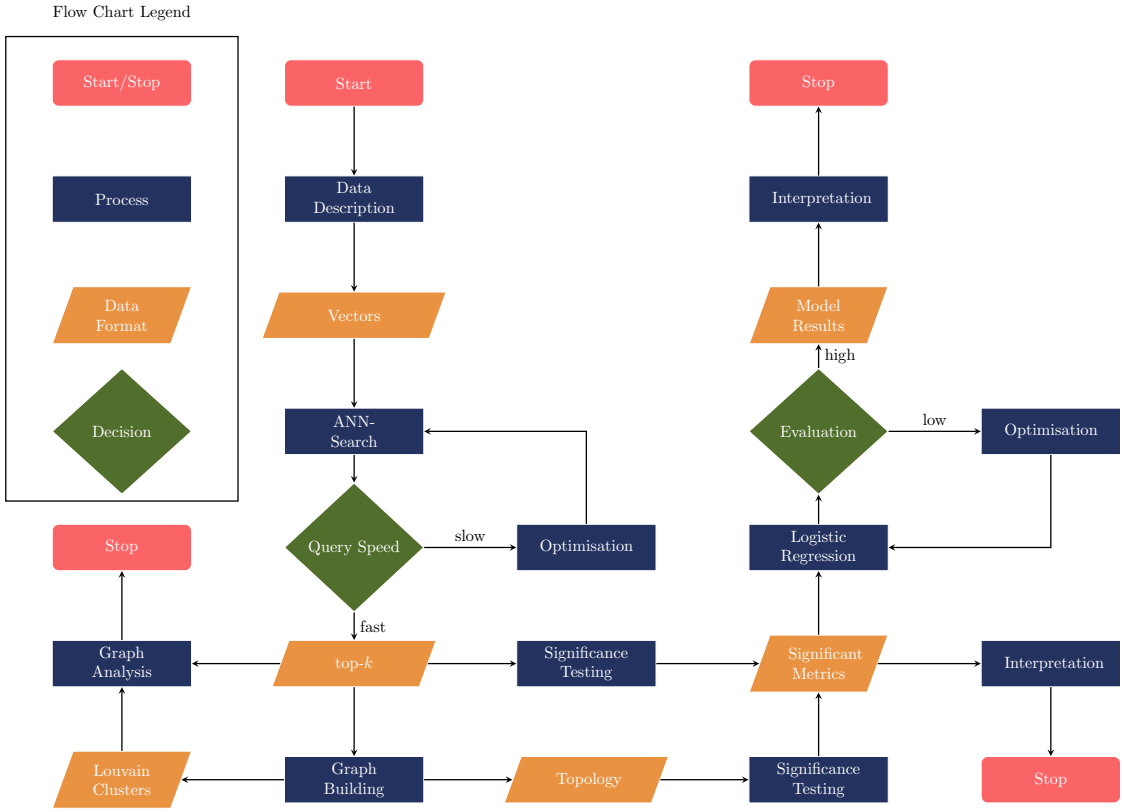


Figure 4.1: Flow Chart of the research process. The arrows indicate the flow of the methodology.

4.1. Vector Database

The vectors are inserted to a vector database for efficient ANN-search through vector indexing. The vector database utilised is Milvus[1], which is open-source and cloud-native. Milvus is able to query multiple vector search for billion-scale data and therefore is state-of-the-art for ANN-search (J. Wang et al., 2021). Milvus is mounted in the programming language Python as a software development toolkit (SDK). Python communicates with Milvus' cloud via Docker.

[1]https://milvus.io

Milvus requires the definition of the database field schema, collection schema, and collection name. Subsequently, data is inserted into the database and parameters are set in Python to define the search query. The vector search revolves around finding a set of top-k nearby neighbours, where k is a parameter on which needs to be decided on. Increasing k increases the query time linearly (approximately). Milvus enables the specification of a number shard to divide vectors into buckets, sufficient shards are requisite to ensure load balance over the vectors. Inadequate shards cause the search-space to be too extensive, but an overabundance of shards causes an increase in latency in communication to the cloud-database. The effect of the parameters: number of neighbours (k), number of shards, and number of vectors will be investigated in reciprocal conjunction to scrutinise query speed.

The type of vector index is detrimental to the search, namely, indexing organises data favourable for high-speed information retrieval. The types of vector indices are IVF_FLAT, HSNW, ANNOY, which are quantisation, graph, and tree-based indices respectively. HNSW handles clustered data efficiently and appears to be non-erroneous for the given data. Further, HNSW functions best for scenarios of high-speed querying, optimal $recall$, and large memory sources[2]. Three parameters exist within HNSW to finetune query speed and the $recall$ of ANN-search. The effect of the parameters: ”maximum degree of nodes on each layer of the graph” (M), search scope of building an index (efC), and search scope of target retrieval (ef) will be investigated in reciprocal conjunction to optimise query speed.

The output of the ANN-search is saved into a data frame for export as a comma-separated values (CSV). As in Figure 4.1, the data on top-k relations for each vector (a.k.a. credit card transactions) is in functional structure for graph building, graph analysis, and extracting metrics for significance testing. For practical implementation and explanation of ANN-search towards creating similarity graphs see https://github.com/SebastiaanK97/NetworkSimilarity.

4.2. Graph Database

When moving from relational data to a graph database the structure alternates on which particular algorithms are applicable. Specifically, a graph database is in functional structure for graph clustering and extracting topological metrics. Moreover, a graph database is efficient in retrieval because nodes are directly linked for perpetuation. The graph database management system made use of is Neo4j[3] which utilises the querying language Cypher. In the graph database, each transaction is a node with a k number of relationships. The distance is a property of the relationships and e.g., the amount of the transactions is a property of the nodes.

Numerous analyses are applicable on graphs. Up to the present time, nil is published about the topology of fraudulent behaviour in social networks. Besides, whether the data of credit card transactions after PCA and ANN-search is valid to build eloquent graphs is unknown. Accordingly, graph analysis in this research could either be groundbreaking or controversial, thus, the nature of graph analysis in this research is utmost exploratory. The graph database requires admin import, consult Appendix C on how to.

[2]https://milvus.io/docs/v2.0.x/index.md
[3]https://neo4j.com

4.3. Exploratory Research

The research pursues with exploratory research to induce novel propositions. Subsequently hypotheses can be build to explain fraudulent behaviour in credit card transactions.

4.3.1. Components, Clustering & Structure

As a first analysis of the graph, the structure of components and clusters is scrutinised. Multiple graph algorithms are ran. Analysis of weakly connected components is done to investigate individual components whilst thresholding the maximum distance of relationships in the graph. The aim of investigating components is to separate densely connected fraudulent transactions from legitimate transactions. Analysis of louvain clustering is done to investigate densely connected fraudulent transactions in a flexible manner. Also the graph is vectorised through Node2Vec to exploit features of the graph. Appendix D is partially an overview of the algorithms for exploratory research.

4.3.2. Assortative Mixing

If fraud rings truly exist, then suggestively fraudulent transactions possess relationships to further fraudulent transactions. Therefore, the graph asserts homophily i.e., the tendency for entities to relate to those which are similar to themselves. (McPherson et al., 2001). Thus, the activity of fraudulent behaviour clusters into fraud rings which can be measured by the assortativity coefficient r. The distinction between the classes of fraudulent and legitimate classes is an enumerative characteristic i.e., a finite set of possible values. However, due to the class imbalance Equation 2.9 necessitates alteration to a sub-graph (see Appendix E).

Prior research by Newman (2003) shows the assortativity coefficient r in social networks ranges from -0.03 ± 0.04 for student relationships up to $+0.36 \pm 0.00$ for coauthorship in the field of physics. The preferential tendency of connections within fraudulent behaviour is unknown and therefore is an open issue.

Therefore, as the second analysis of the graph, the assortativity coefficient r is calculated as a function of the k number of neighbours. The assortativity coefficient r helps understand the preferential tendency of connections within fraudulent behaviour. If fraud rings truly exist then $r > 0.00$. Namely, fraudulent transactions will connect to further fraudulent transactions. Note, for low numbers of k, assortative mixing is dependent on a smaller neighbourhood of the nodes. As a consequence, the assortativity coefficient r becomes more local. Equation E.7 is detrimental to the calculations and is easily extractable through Cypher queries, Appendix F depicts an example query and calculation for $k = 256$.

4.4. Explanatory Research

As the research aim is to scrutinise contradictions in network structure between fraudulent and legitimate transactions, metrics such as the local clustering coefficient (transitivity), eigenvector, betweeness, closeness, and (personalised) PageRank yield information on the network structure. Unguided hypotheses testing helps discover contradictions in such metrics.

Hypothesis 1 (H1) *Fraudulent and legitimate transactions differ in topological structure.*

Let τ_i define any topological metric of a node in a graph, then the null-hypothesis would state there is no contradiction between legitimate and fraudulent transactions:

$$H_{\tau 0}(c,k) : \tau(c = l, k) = \tau(c = f, k) \tag{4.1}$$

here the hypothesis is an outcome of topology metrics by class difference c which vary over the top-k. If fraudulent transaction contradict legitimate transactions in a graph then:

$$H_{\tau 1}(c,k) : \tau(c = l, k) \neq \tau(c = f, k). \tag{4.2}$$

4.4.1. Anomalies

A common view on fraudulent behaviour is that their presence is anomalous a.k.a. being an outlier. According to Gogoi et al. (2011), outliers can either be distance or density-based. Individual outliers are detectable through distance-based methods, such as setting exceeding a minimum distance, or exceeding a certain percentile of neighbours. Collective anomalies are more or less detectable through density-based methods, yet these approaches are computationally intensive and incompatible with ANN-search. In general, the hypothesis is:

Hypothesis 2 (H2) *A positive relationship exists between the distance to a set of nearby neighbours and the classification of a transaction as fraudulent.*

The distance L_2 to neighbours would be higher for fraudulent transactions in comparison to legitimate transactions. Whether it is the the maximum, mean, or the standard deviation of the distance L_2 does not matter, although, one metric may be more distinguishable than the other. The robustness of difference is discernible by significance-levels between the classes.

The null-hypothesis states no difference in distances to neighbours exists between classes:

$$H_{L0}(c,k) : L_{2*}(c = l, k) = L_{2*}(c = f, k). \tag{4.3}$$

The mathematical formulation of the alternative hypothesis is as following:

$$H_{L1}(c,k) : L_{2*}(c = l, k) < L_{2*}(c = f, k) \tag{4.4}$$

let the $L_{2*}(k)$ denote any aggregation of the distance measure such as the maximum, mean, or the standard deviation, the one with highest significance is of interest. The distance metrics and thus also the hypotheses are dependent on the k number of nearby neighbours.

4.4.2. Fraud Rings

A novel view on fraudulent behaviour is that their presence is surrounded by other fraudulent behaviour a.k.a. fraud rings. Therefore, a set of nearby neighbours of a fraudulent transaction would entail further fraudulent transactions (Gürsoy & Varol, 2021b). Since credit card transactions enter databases successively, there is high prior knowledge of classification. Therefore, considering the class of nearby neighbours is feasible. For example, Gürsoy and Varol (2021a) apply KNN-search to make classifications in clinical data. Expectantly, such an approach would also persist for ANN-search on a large dataset of credit card transactions:

Hypothesis 3 (H3) *A positive relationship between the tendency to connect to fraudulent transactions exists within fraudulent transactions.*

In contrast to assortative mixing, a new metric needs to be defined at the local level as a characteristic of each node. Theretofore, a nodes' direct neighbours are considered. The formal expression of the fraction of fraudulent transactions is as following:

$$\phi_i(j,k) = \frac{\sum_j \delta(c_j = f)}{k} \tag{4.5}$$

here the fraction of fraudulent transactions ϕ_i of node i is dependent on the neighbouring nodes j being fraudulent, relative to the total number of neighbours k.

The null-hypothesis states no difference in fractions exists between classes:

$$H_{\phi 0}(c,k) : \phi(c = l, k) = \phi(c = f, k) \tag{4.6}$$

The mathematical formulation of the alternative hypothesis is as following:

$$H_{\phi 1}(c,k) : \phi(c = l, k) < \phi(c = f, k) \tag{4.7}$$

4.4.3. Overview of Hypotheses

Figure 4.2 is a visual exemplar of the hypotheses for two individual nodes of each class. Expectantly, legitimate transactions are surrounded by nearby legitimate nodes, and fraudulent transactions are surrounded by more fraudulent nodes, whilst being an outlier, relatively.

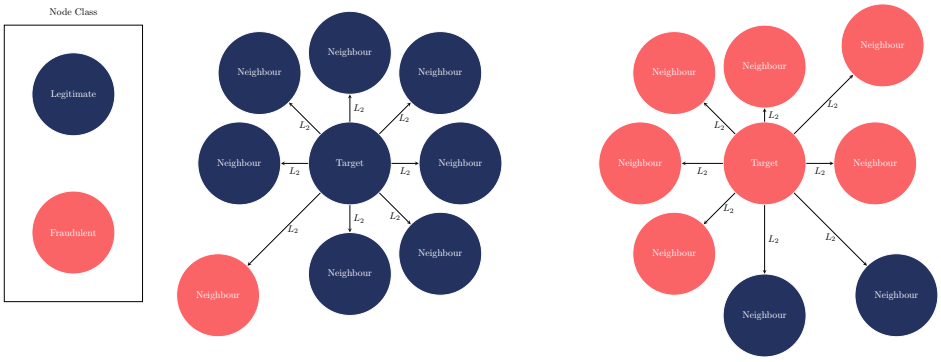


Figure 4.2: Exemplar of neighbouring transactions per class for $k = 8$.

The two theoretical hypotheses on fraudulent transactions as anomalies and fraud rings are tested through a one-sided Welch's t-test for unequal variance. Namely, variances are unequal because of the inherent difference between legitimate and fraudulent transactions given the metrics. For instance, the fraction of fraudulent neighbours ϕ scatters around 0.00 for legitimate transactions, and the distance to neighbours $L_{2\sigma}$ ranges widely depending on how far an outlying fraudulent transaction is. The hypothesis on legitimate and fraudulent transactions differing in topological structure is tested through a two-sided Welch's independent t-test for unequal variance. Namely, whether a metric would be larger or smaller for the given class is unknown. All hypotheses are tested over parameter k to investigate the effect of ANN-search.

4.4.4. Statistical Modeling

Classification of legitimate and fraudulent transactions is done through logistic regression. Model evaluation is key to improving the validity of classification, which is in this case measurable through the AUC and $recall$. Model evaluation is done by undersampling and LOOCV. However, undersampling causes the measures of AUC and $recall$ to be case-specific on undersampling the training set. Therefore, it is necessary to pool outcomes over multiple iterations i.e., the mean and standard deviation of the AUC and $recall$ are explored over several parameters considering 50 reiterations. The parameters are the variable input of the model and the discretisation of the k number of neighbours. Through stepwise logistic regression, three sets of models come about with the variables: fraction of fraudulent neighbours ϕ, the standard deviation of the distance to neighbours $L_{2\sigma}$, and the transaction amount A. Both a linear and quadratic polynomial fit is evaluated. A linear fit returns a single coefficient for each variable, plus the model intercept. A quadratic fit returns also the quadratic transformation of variables, plus interaction effects and the model intercept. On top of these models, also the logarithmic transformations of $L_{2\sigma}$ and A are considered. Table 4.1 is an overview of the models.

Table 4.1: Overview of logistic regression models. Note, the No. Coefficients includes the model intercept.

Label	Model Variables	No. Coefficients (Linear)	No. Coefficients (Quadratic)
Model 1	$P(c = f\|\phi)$	2	3
Model 2	$P(c = f\|\phi, L_{2\sigma})$	3	6
Model 3	$P(c = f\|\phi, L_{2\sigma}, A)$	4	10

Search parameters for logistic regression modeling are as following. The maximum number of iterations to settle coefficients is set to $10,000$ (instead of 100) for reaching a global optima. The penalty of the model set to "$none$" for consistent treatment of coefficients over varying models, moreover, the AUC and $recall$ deem to be higher in this case. To ensure justified variables, an analysis of the variation inflation factor (VIF) is done to track multicollinearity.

As final result, after model evaluation, a theoretically relevant logistic regression model is fit over all available data to investigate the overall influence of model coefficients. Namely, the process of statistical inference yields valuable information on the coefficients of the model.

5
Empirical Findings

The following chapter kicks-off with query speed optimisation and descriptive findings of ANN-search on the data. Subsequently, findings from exploratory research reveal descriptive analysis on graphs and proposals for further investigation. Last but not least is explanatory findings on graph analysis. Explanatory research naturally arises out of exploratory findings.

5.1. Approximate Nearest Neighbour Search

If no ANN-search i.e., naive kNN is performed the computational complexity scales exponentially. In Table 5.1 the computation time for multiple sets of nodes is tracked, clearly, doubling the number of nodes results in quadrupling the computation time. The computation time per node will be compared to the performance of ANN-search. Extrapolating the time in Table 5.1 to the total $284,807$ nodes leads to an approximation of:

$$t_i = \alpha \cdot N_i^2 \approx 1 \times 10^{-5} \cdot 284,807^2 \approx 1 \times 10^6 \,\text{s}. \tag{5.1}$$

Therefore, the rough approximation estimates computing the Euclidean distance between all nodes take 1×10^6 s which equals to weeks i.a. months of time.

Table 5.1: Overview of computation time in seconds [s] for naive kNN.

Data Size N	64	128	256	512	1024	2048
Computation time [s]	0.22	0.91	3.62	14.41	57.49	231.73
Computation time per node [s]	0.0035	0.0071	0.0142	0.0281	0.0561	0.1131
Growth (t_i/t_{i-1})	-	4.00	3.96	3.98	3.99	4.03

Note, the data only exists of two days, years of data would straightforwardly be problematic. Evidently, ANN-search is requisite to perform kNN and compute the Euclidean distance to nearby neighbours efficiently. Table 5.2 yields the query time in seconds for ANN-search.

Table 5.2: Query time in seconds [s] of ANN-search over k and N considering 4 shards.

No. k	Data Size N									
	256	512	1024	2048	4096	8192	16384	32768	65536	131072
256	4.47	4.83	5.75	7.81	11.46	23.22	54.51	180.52	527.36	1928.69
1024	5.03	6.11	8.83	12.85	24.86	45.25	105.23	251.27	753.43	2496.40
4096	6.76	9.15	15.97	29.61	68.11	135.14	275.08	706.52	1850.24	5309.83

Distinctly, query time is near-linear and is viable over large sets of nodes. Benchmarking of queries is necessary to investigate optimal query speed and avoid inaccurate parameter selection. Benchmarking is specific to the data set. Figure 5.1 visualises the trend of query time relatively per node. Note, the x-axis refers to the data size N as in Table 5.2 and the y-axis is the computation time per node $\frac{t_i}{N_i}$. The y-axis ranges roughly from 0.003 seconds per node for $k = 256$ and $N = 4,096$ up to 0.041 seconds per node for $k = 4,096$ and $N = 131,072$. The query speed yields a u-curve relationship with the number of nodes. Specifically, the u-curve depicts the trade-off between latency and data abundance i.e., ANN-search is slow for a few nodes because of latency to the server, and a high number of nodes causes the query to be complex. The number of shards does not affect the query speed considering the given data.

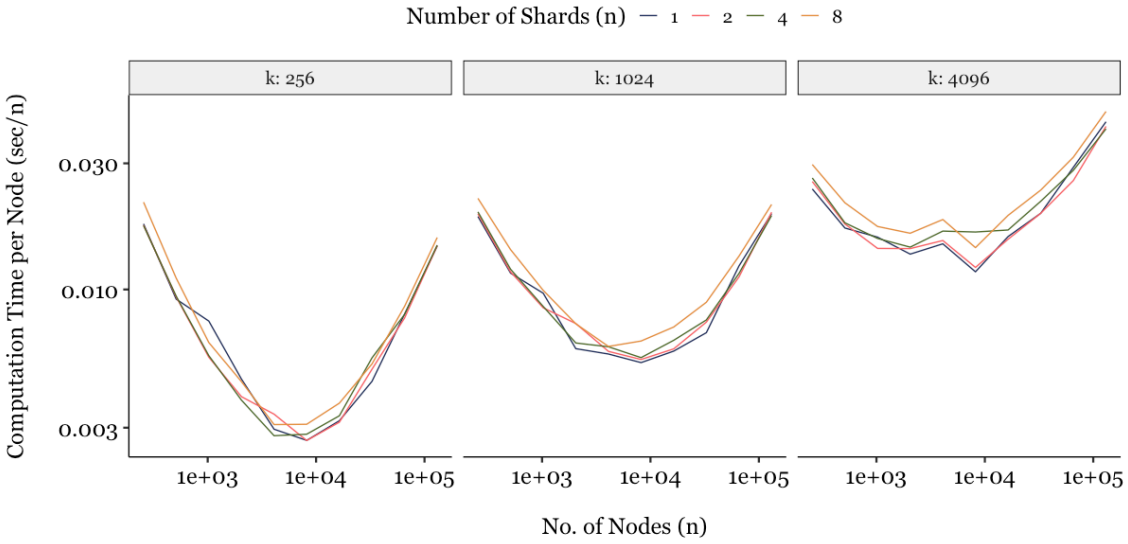


Figure 5.1: Query speed as computation time per nodes in relation Table 5.2.

Further parameters affect the query speed, but may also affect the $recall$ of the query i.e., whether the approximation of ANN-search performs rigorously. These parameters are as following. The ”maximum degree of nodes on each layer of the graph” (M), search scope of building an index (efC), and search scope of target retrieval (ef). The reciprocal conjunction of these parameters is investigated in Figure 5.2. It transpired that the parameters do not affect the query speed for the given data. Furthermore, a brick investigation reveals that the ANN-search remains stable over the parameters. Namely, the results appear to remain stable over the parameters as well. Perchance the data is straightforward enough the results are not approximations but clear-cut kNN. In other words, conventionally ANN-search is implemented for image and video search, thus array structures. Therefore, the vector search is not complex and the algorithm decidedly seems to find the exact nearest neighbours.

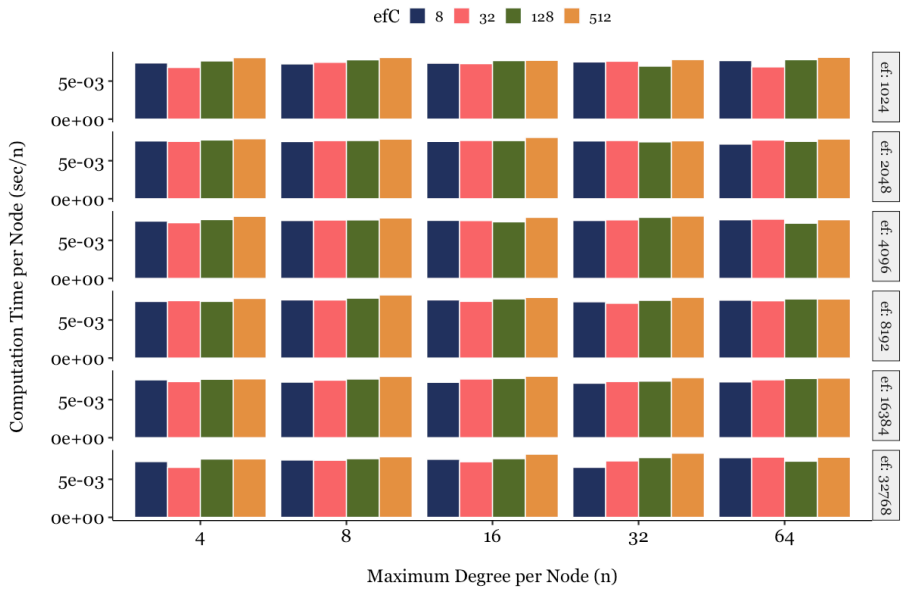


Figure 5.2: Query speed over HNSW parameters.

It appears that $k = 256$ for the full data set $N = 284,807$ balances the query speed well with a computation time of 1×10^4 s, about 2,5 hours, for HNSW. Benchmarking for IVF_FLAT reveals that HNSW is up to twice as fast and less erroneous, which is confirming that data is clustered. Besides, setting $k = 256$ prevents conflicts in running Neo4j on an excessively large graph. Further, processing of data and file sharing becomes inconvenient for larger k. The total number of relationships is $(256 + 1) \cdot 284,807 = 73,195,399$, the extra relationship stems from the ANN-search finding the closest distance to vectors themselves first. Figure 5.3 depicts the distribution of Euclidean distance L_2 to neighbours over varying k number of neighbours, a higher k results in discovering further neighbourhoods. Interestingly, the distribution depicts peaks which hint toward the existence of clusters in vector space.

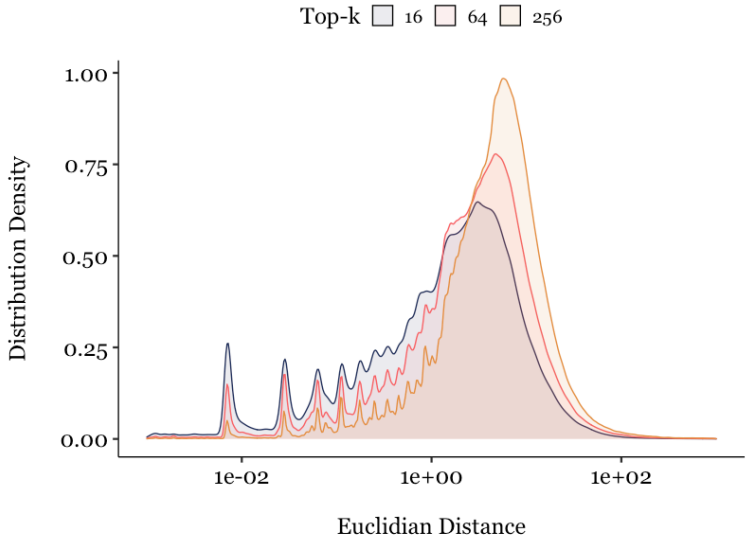


Figure 5.3: Distribution of Euclidean Distance over varying k number of neighbours.

5.2. Exploratory Research

ANN-search establishes nodes' relationships, subsequently a graph is set-up in a local Neo4j database. Multiple graph algorithms are ran to investigate the graph structure and propose objectives for research deduction. Note, the Euclidean distance is often referred to as distance which is a numerical (float) property of a relationship. See Appendix D for Cypher code.

5.2.1. Weakly Connected Components

As a first analysis, weakly connected components (disconnections) are investigated over thresholding the distance, which is resembling "cutting off" distant relationships. Through adjusting the distance threshold, Table 5.3 records the quantity of isolated frauds and sizes of disconnected components. Fraudulent transactions become isolated rather quickly compared to legitimate transactions, therefore they are outliers. Reducing the distance threshold causes components to fragmentise and additional components arise, up to a point the number of nodes per component start to reduce, and components become isolates themselves. The components hint towards existing clusters in data in relation to Figure 5.3.

Table 5.3: Overview of unconnected components over distance thresholds.

Threshold	Isolate Frauds	Clusters $N > 10$	Clusters $N > 100$	Clusters $N > 1000$
1/64	485	568	63	0
1/32	481	760	79	0
1/16	479	933	85	1
1/8	469	1,336	114	1
1/4	456	1,638	155	2
1/2	436	1,776	223	6
1	354	1,344	214	34
2	222	638	150	37
4	100	227	36	12
8	39	73	20	11
16	11	21	2	1
32	3	14	2	1
64	0	4	1	1

The focus lies on finding components which contain a high number of fraudulent transactions i.e., a collective anomaly. Figure 5.4 depicts the distribution of the number of fraudulent transactions relative to the aggregation of transactions in each component. The threshold indicates fraudulent transactions are either anomalies or either form a collective anomaly from which roughly 0.10% of the component is fraudulent. Hence, a considerable amount of noise (legitimate transactions) are present in the component. Further, the fraudulent transactions become isolates first, before forming a large component full of fraudulent transactions. Therefore, fraudulent transactions are distant among each other, and thus metrics of density-based outliers (such as LOF) would a be appropriate measure to detect fraudulent behaviour.

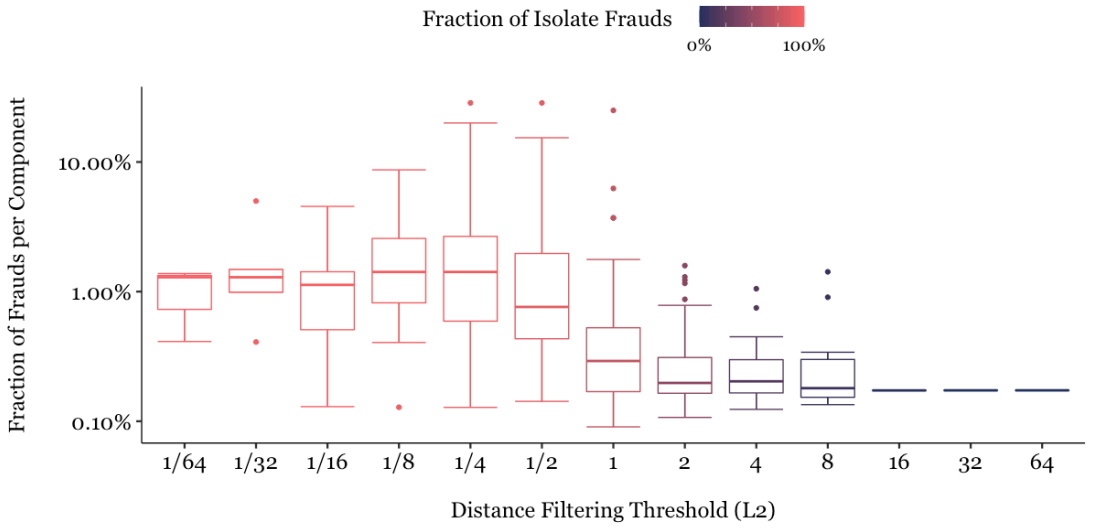


Figure 5.4: Fraction of fraudulent transactions in weakly connected components.

5.2.2. Louvain Clustering

As a second analysis, Louvain clusters are investigated. Namely, weakly connected components are too determinisitc, contrarily, Louvain clustering is flexible for aggregation. Since Louvain clustering is a greedy algorithm modularity optimisation is requisite. Modularity is computed in an equal manner as the assortativity coefficient r, but instead, an algorithm aims to maximise the modularity given the relationships within clusters. Thus, the clusters are approached as an enumerative characteristic. Note, the Louvain clustering also considers the weight of relationships.

Ten reiterations of louvain clustering are performed over $k = 16, 64, 256$. A high k leads to unstable results i.e., no consensus of the number of clusters, though, modularity hits high. A low k is stable with lower modularity, however, the number of clusters are unreasonably high considering $284,807$ nodes. Therefore, Louvain clustering is inconvenient because of the lack of consensus in modularity i.e., it is difficult to pinpoint an optimum. The analysis is controversial, and due to lack of time, is halted, arguably the relationships in the graph are too artificial to cluster nodes.

In depth-configuration of Louvain clustering would be beneficial since clusters yield important information. Namely, fraudulent transactions may lie in-between clusters. Therefore, fraudulent transactions would have a significant higher count of relationships outside their assigned cluster in comparison to legitimate transactions. Yet, the hypothesis is unfavourable considering the given data.

5.2.3. Assortative Mixing of Fraudulent Transactions

As a third analysis, the assortativity coefficient r is calculated over a range of k number of neighbours. Figure 5.5 depicts the trend of assortative mixing. A positive value indicates a tendency for connections between fraudulent transactions exists. The assortativity coefficient ranges from $r = 0.66$ for $k = 2$ up to $r = 0.41$ for $k = 256$.

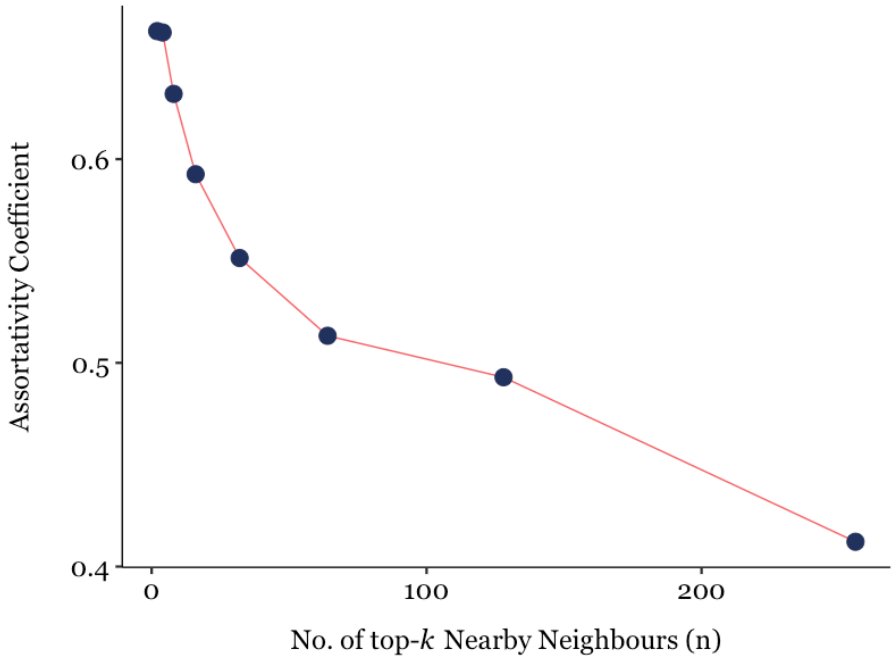


Figure 5.5: Assortative mixing within fraudulent nodes and their neighbours.

The assortativity coefficient r decreases over k because the neighbourhood of search increases, and more legitimate transactions become involved. For the given range of k, the tendency is rather large, especially in comparison to what is found for other social networks (Newman, 2003). The findings are intriguing since 1 out of 600 transactions are fraudulent, yet, fraudulent transactions tend to mingle. Suggestively, fraud rings do exist in the given data. Moreover, due to the high assortative mixing, the majority of fraudulent behaviour is probably stemming from criminal organisations, and not from individual criminal activity.

5.3. Explanatory Research

The following section tests three hypotheses. Subsequently is the evaluation of the classification model and statistical inference of a theoretically relevant model.

Table 5.4: Overview of hypotheses and overall significance.

	Metric	Formal Definition	Result	Significance Level
H1	Topology	$H_{\tau 1}(c,k) : \tau(c=l,k) \neq \tau(c=f,k)$	False	$0.1 < p$
H2	Anomaly	$H_{L1}(c,k) : L_{2*}(c=l,k) < L_{2*}(c=f,k)$	True	$p << 1 \times 10^{-5}$
H3	Fraud Ring	$H_{\phi 1}(c,k) : \phi(c=l,k) < \phi(c=f,k)$	True	$p << 1 \times 10^{-5}$

Table 5.4 summarises the three hypotheses in concordance to the given data. In simple words, fraudulent transactions are unalike their neighbours in comparison to legitimate transactions, and thus are anomalies. Also, fraudulent transactions tend to connect to further fraudulent transactions. Therefore, fraudulent behaviour exists in fraud rings, and credibly, criminal organisations do exist. The following section explains the hypothesis testing and depicts the trends of significance in-depth over the k number of neighbours.

5.3.1. Topology

The significance test in Figure 5.6 shows no difference between classes, namely the corresponding p-values range from 10% up to 100%. Therefore, no trends exist over k neighbours. Note, the graph could be too artificial to extract meaningful metrics. Further topological metrics are incomputable due to the size of the graph and require approximations by considering a sample of nodes. Such approximations occur random and therefore is refrained from for research.

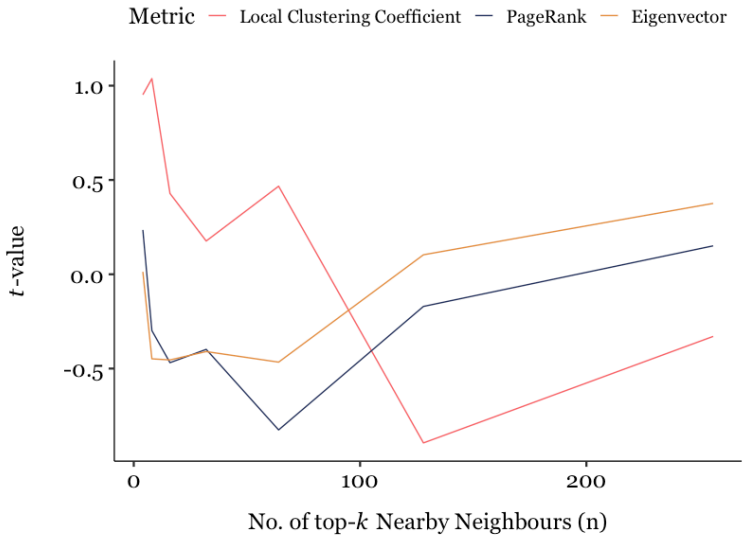


Figure 5.6: Two-sided Welchs' t-test on topological metrics τ.

5.3.2. Anomalies

A general fact is that fraudulent transactions are anomalies i.e, outliers in terms of data. Unsurprisingly, the distance to neighbours in vector space is higher for fraudulent transactions than for legitimate transactions. For all p-values yields $p < 1 \times 10^{-9}$, therefore the difference is highly significant. Figure 5.7 depicts that the significance increases over the k number of neighbours. The standard deviation over the distance L_2 is least significant over a low k, but exceeds other metrics when k increases. Therefore, $L_{2\sigma}$ is considered as most appropriate for classification with a p-value of $p \approx 1 \times 10^{-50}$ at $k = 256$.

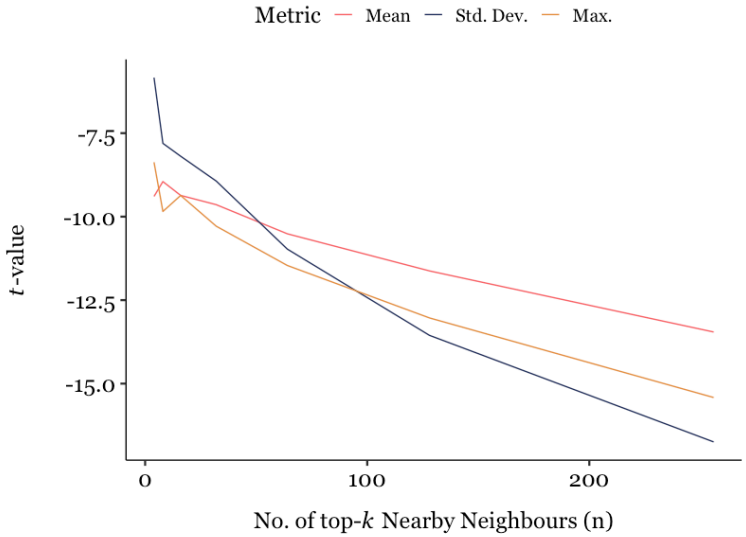


Figure 5.7: One-sided Welchs' t-test on aggregated distance metrics L_{2*}.

A further look in Figure 5.8 shows how the values are distinct for the classes over k. Only a small overlap of distributions remains for a high k, but since the classes are imbalanced, the overlap is deceiving. In actual, numerous legitimate transactions are outliers as well.

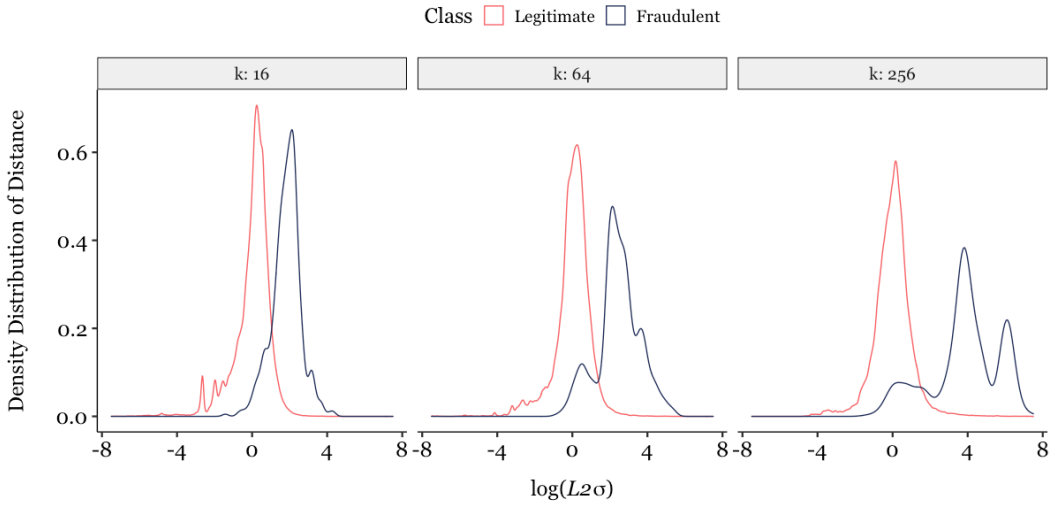


Figure 5.8: Density Distribution of the (**standardised**) distances $L_{2\sigma}$ over classes and $k = 16, 64, 256$.

5.3.3. Fraud Rings

A novel (rather qualitative) concept is that fraudulent transactions exist in fraud rings i.e, clusters in terms of data. In line with the assortativity coefficient r in Figure 5.5, the difference between classes their neighbourhood sets the classes apart. Specifically, fraudulent transaction are similar to fraudulent transactions whilst legitimate transactions are not. For all p-values yields $p << 1 \times 10^{-100}$, therefore the difference is highly significant. Figure 5.9 depicts that the significance decreases over the k number of neighbours. Namely, increasing the (diameter of the) neighbourhood in vector space, by increasing k leads to finding more transactions of the opposite class. Notwithstanding, ANN-search is highly effective for detecting fraud rings and therefore is likely to be successful for classifying transactions.

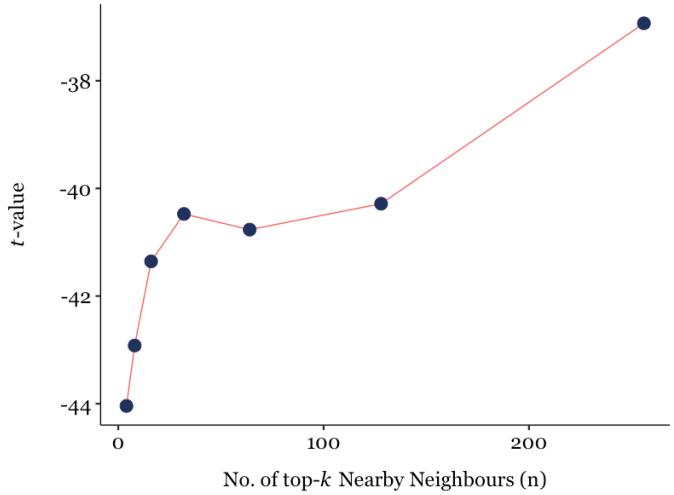


Figure 5.9: One-sided Welchs' t-test on fraction of fraudulent neighbours ϕ.

Figure 5.10 shows the distribution is extremely skewed for legitimate transactions, which is beneficial, because the distinctness sets the legitimate and fraudulent transactions apart.

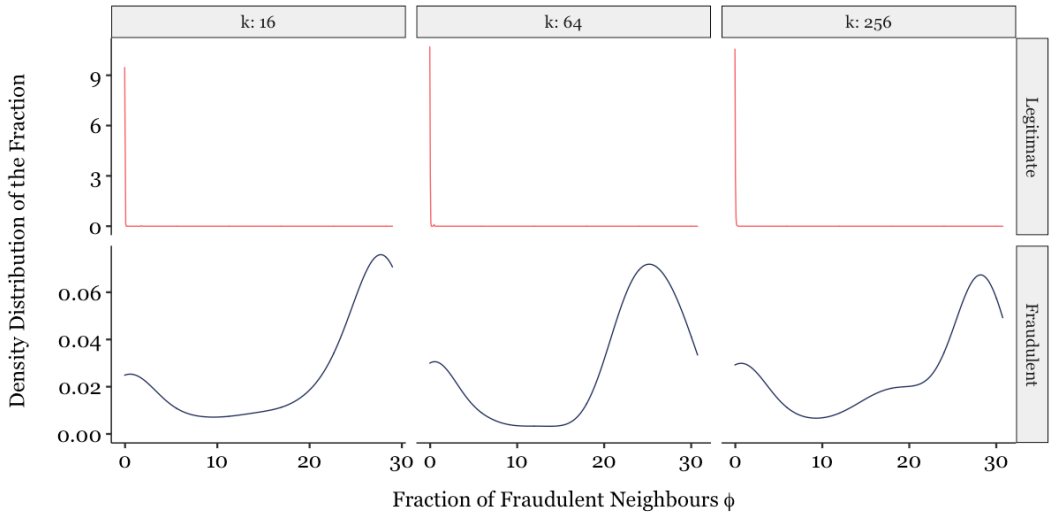


Figure 5.10: Density Distribution of the (**standardised**) fraction ϕ over classes and $k = 16, 64, 256$.

5.3.4. Classification

Optimising classification requires careful model evaluation. Three models are evaluated in terms of $recall$, $precision$, and AUC over the parameters k, and polynomial and logarithmic transformations. Table 5.5 is a recapitulation of the three models. The variables are stepwise added in order of improvement. The fraction of fraudulent neighbours ϕ is key to the models. Concerning Figure 5.11 up to Figure 5.13, the trend depicts the mean of evaluation scores with whiskers depicting the standard deviations (uncertainty) over 50 reiterations. Note, undersampling leads to deceiving results and is not representative for the whole population.

Table 5.5: Overview recapitulation of logistic regression models.

Label	Model Variables	Stepwise Added Variable
Model 1	$P(c = f\|\phi)$	Fraction of Fraudulent Neighbours
Model 2	$P(c = f\|\phi, L_{2\sigma})$	Std. Dev. of Distance to Neighbours
Model 3	$P(c = f\|\phi, L_{2\sigma}, A)$	Transaction Amount

Figure 5.11 depicts the $recall$ as function of the three parameters. Concerning $recall$, the models improve over k. Both the quadratic polynomial and logarithmic transformations lead to a higher $recall$ with a decrease in uncertainty. The test reveals model 3 with a quadratic polynomial and logarithmic transformation on $k = 128$ is optimal with $recall = 88.47\%$.

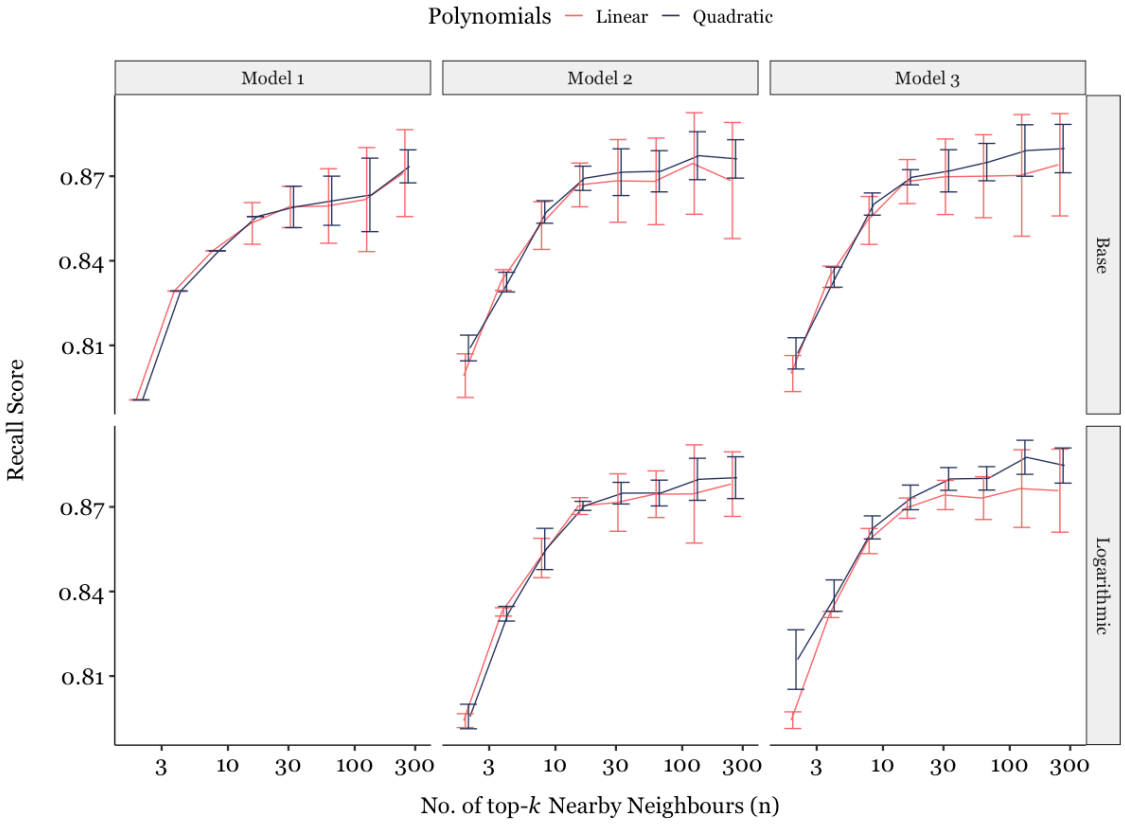


Figure 5.11: $recall$ as function of k, polynomial transformation, and logarithmic transformation.

Figure 5.12 depicts the $precision$ as function of the three parameters. Concerning $precision$, the models do not necessarily improve over k. Neither does the quadratic polynomial transformation lead to a higher k, in contrary, $precision$ decreases overall whilst the uncertainty decreases. The test reveals model 1 with a linear polynomial transformation on $k = 4$ is optimal with $precision = 99.93\%$. Yet, all trends approach a maximum of 100.00% at mean.

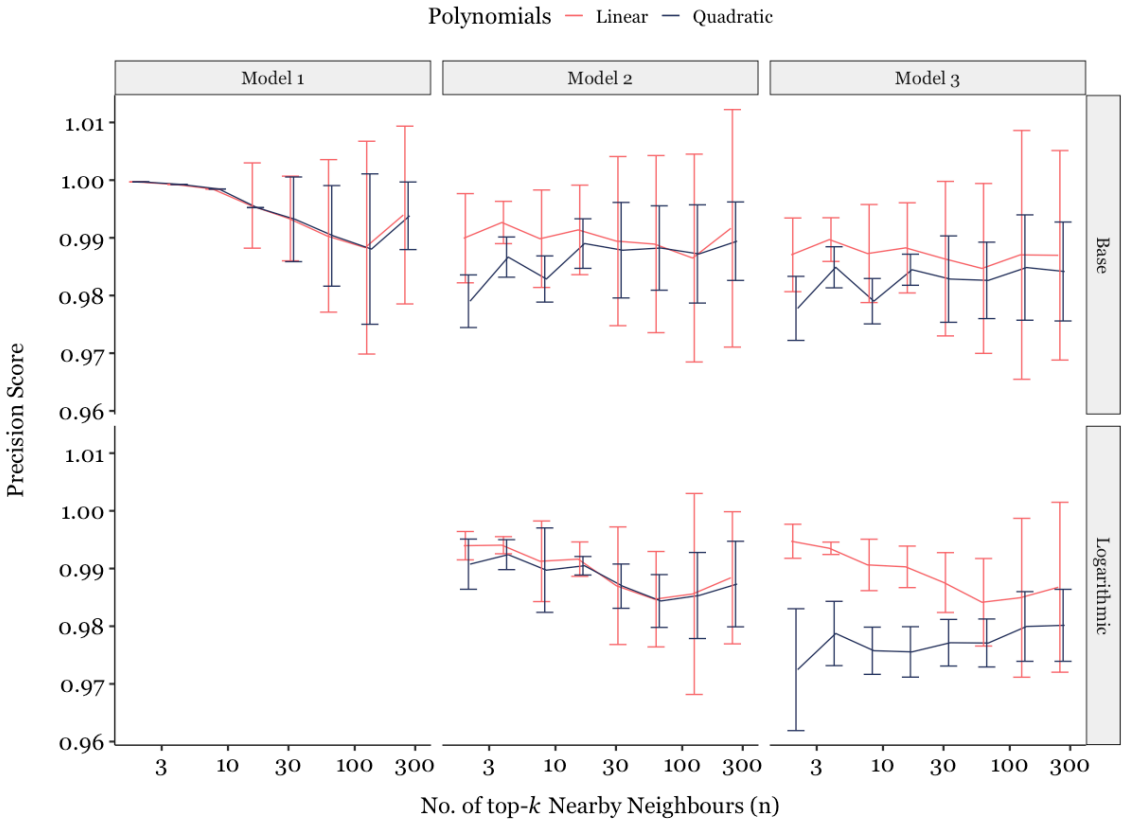


Figure 5.12: $precision$ as function of k, polynomial transformation, and logarithmic transformation.

Figure 5.13 depicts the AUC as function of the three parameters. Concerning AUC, the models improve slightly over k on the whole. Both the quadratic polynomial and logarithmic transformations lead to a higher AUC with a decrease in uncertainty. The test reveals model 3 with a quadratic polynomial and logarithmic transformation on $k = 256$ is optimal with AUC $= 96.80\%$. Therefore, in overall, the aforesaid model is highest performing because it is independent of the threshold parameter. Besides, also the $recall$ appears to be highest for the corresponding parameters. The $precision$ is consistently adequate, yet, the aforesaid model is relatively low in $precision$, but still plausible due to the high overall performance.

The model evaluation shows which parameters are viable and which are not. A general takeaway is that a low k number of neighbours if not accurate, also, models improve over logarithmic transformation. The variable selection only leads to slight improvements, though, $precision$ decreases when including the transaction amount A as third variable, yet, the AUC is unaffected. Model 2 among logarithmic transformation and a high k is optimal. Fine-tuning of the threshold to balance $precision$ and $recall$ is a business decision.

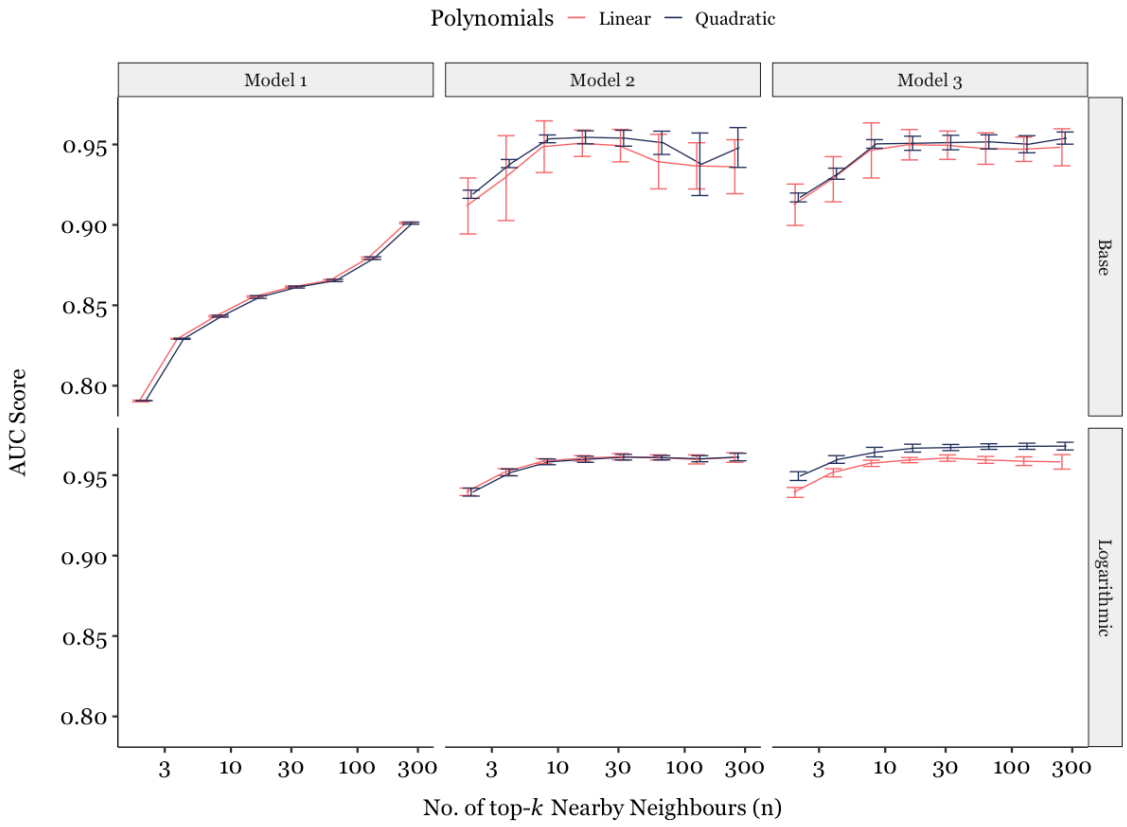


Figure 5.13: AUC as function of k, polynomial transformation, and logarithmic transformation.

The models are ran over 50 reiterations of undersampling, therefore confusion matrices and ROC-curves are iteration specific, thus controversial. However, a business decision exists between theft of capital and the workload of restoring FN for which Figure 5.14 is visualising the trade-off. A low thresholds leads to a high loss of capital whilst a high threshold leads to a high FN, therefore $precision$ matters moderately to restrain the remuneration of the workload.

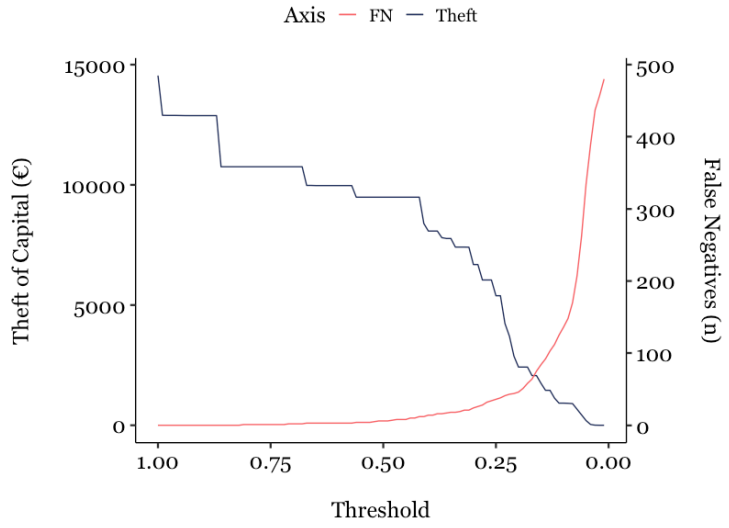


Figure 5.14: Trade-off between theft of capital and workload (FN) as function of the classification threshold T.

Figure 5.15 depicts a comparison over four models. Here, two models are popular instances from Kaggle (Université Libre de Bruxelles, 2018), and two models are part of this research. The model evaluation depicts the mean and standard deviations over 50 reiterations of under-sampling and is a zoom (left) of the whole bar plot (right). The models from Kaggle operate on the raw data from PCA and utilise a 70%/30% split of training and test data instead of LOOCV. The first model is a logistic regression (LR) classifier with a regularisation (L2 penalty). The second model is a multilayer perceptron (MLP) classifier with 200 hidden layers. Therefore, the models operating on 28 features of PCA and the transaction amount as 29^{th} feature are relativly slow and thus less compatible with LOOCV due to their computational complexity.

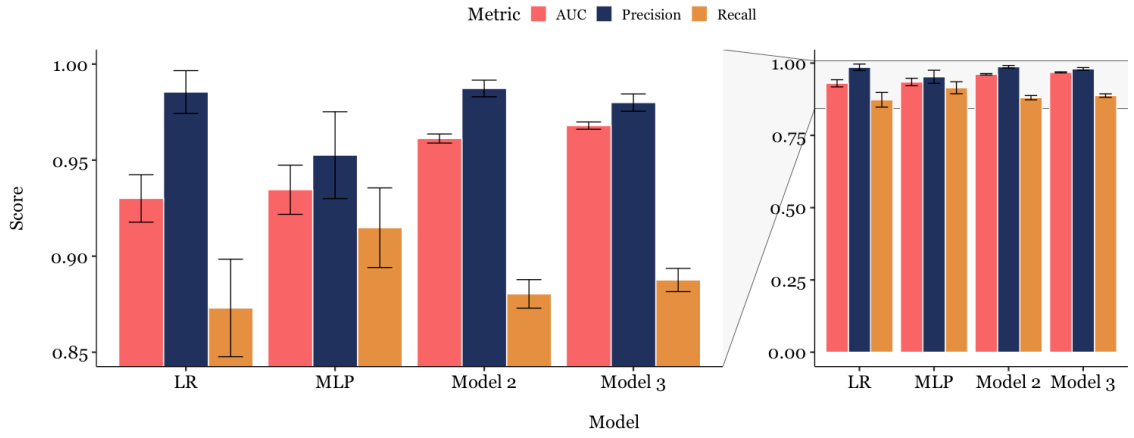


Figure 5.15: Evaluation of AUC, *precision*, *recall* over multiple models (Université Libre de Bruxelles, 2018).

Table 5.6 is an overview and summary of the model outcomes. The LR model performs well in terms of *precision*, but less in AUC and *recall*. The MLP model performs superior in *recall* and therefore captures the aim of classification best, yet, AUC and *precision* are relatively low. Model 2 and 3 perform superion in AUC and yield high *precison* whilst the *recall* is lower than the MLP model but higher than the LR model. Foremost, model 2 and 3 are theoretically relevant because coefficients are interpretable. Concurrently, the LR and MLP model operate inside the black-box and therefore are fragile to change. Therefore, the LR and MLP yield rather large standard deviations, yet, model 2 and 3 are relatively steady. In accordance, model 2 and 3 rely less on large quantities of historic data for training the models in comparison to the LR and MLP model. Thus overall, model 2 and 3 outperform the LR model in multiple practices. Whether model 2 and 3 are superior to the MLP model is a matter of business.

Table 5.6: Overview of models, variables, their sources, and outcomes of evaluation.

Label	Model Variables	Source (Hyperlink)	High Metric	Low Metric
LR	Raw PCA	Joparga (Kaggle LR)	*precision*	AUC, *recall*
MLP	Raw PCA	Javier (Kaggle MLP)	*recall*	AUC, *precision*
Model 2	ϕ and $L_{2\sigma}$	$k = 256$, quadratic	AUC, *precision*	*recall*
Model 3	ϕ, $L_{2\sigma}$, and A	$k = 128$, quadratic	AUC, *precision*	*recall*

5.3.5. Statistical Inference

A statistical model is fit to the whole data using two variables and their interaction effect: the standard deviation of the distance to neighbours $\log(L_{2\sigma}(k))$, the fraction of fraudulent neighbours $\phi(k)$, and their interaction effect $\log(L_{2\sigma}(k)) \cdot \phi(k)$. The aim of the model is to juxtapose the weight of anomalies and the weight of fraud rings on the classification as fraudulent.

Table 5.8 is an overview of the model fit over the k number of neighbours. All variables over all k yield a significant effect on the dependent variable. The coefficient weight of $\log(L_{2\sigma}(k))$ is positive and first increases slightly, and then decreases harshly over k. The coefficient weight of $\log(L_{2\sigma}(k))$ is positive, and the interaction effect $L_{2\sigma}(k) \cdot \phi(k)$ is negative. The latter two coefficients do not show a particular pattern. To consider the respective weight of the variable coefficients, the distribution[1] of the variables needs to be taken into account. Therefore, Table 5.7 multiplies the maximum value of the variable range with their corresponding coefficient to calculate the respective weight on the dependent variable being fraudulent.

Table 5.7: Range of (**standardised**) variables and respective weight over k in conjunction with Table 5.8.

No. k	$\log(L_{2\sigma}(k))$			$\phi(k)$		
	Min.	Max.	$\beta_1(k)\cdot$Max.	Min.	Max.	$\beta_2(k)\cdot$Max.
2	−9.36	3.67	4.63	−0.04	26.37	11.84
4	−10.71	3.99	5.94	−0.04	27.05	13.25
8	−11.44	4.39	6.78	−0.04	28.00	15.18
16	−14.1	4.97	7.33	−0.04	28.98	13.01
32	−15.32	5.49	7.26	−0.05	29.86	11.85
64	−10.86	6.42	7.36	−0.05	30.73	12.54
128	−8.96	7.64	7.35	−0.05	30.13	12.35
256	−4.75	8.71	7.42	−0.05	30.74	20.17

Note, the relations of k are artificially created, thus scepticism is necessary, yet, the respective weight of $\phi(k)$ is higher than the respective weight of $\log(L_{2\sigma}(k))$ and grows further over k. Therefore, the effect of fraudulent neighbours is stronger than the effect of outliers. In other words, the effect of being situated in a fraud ring is more determinant on the classification as fraudulent than the effect of being an anomaly. Conventionally, the view on fraudulent transactions is that they are outliers. However, in contrast to this a priori assumption, the novel results indicate that fraudulent transactions exist in clusters and thus are *collective* anomalies.

Note, the interaction effect $\log(L_{2\sigma}(k)) \cdot \phi(k)$ is negative and strong in respective weight considering both variables. Therefore, the effect diminishes if both variables increase, though, $\log(L_{2\sigma}(k))$ also yields negative values. Thus, if a transaction is a global outlier, but not a local outlier, then the negative effect of the interaction effect and the negative values of the $\log(L_{2\sigma}(k))$ variable become positive together. In other words, fraudulent transactions are not necessarily outliers within their corresponding fraud ring.

[1] Also consider the conjunction of Figure 5.8 and Figure 5.10 between the distribution and the respective weight.

Table 5.8: Logistic regression model on whole data (a sample of credit card transactions over two days) with two variables and their interaction effect.

	Dependent variable:							
	Classification as Fraudulent							
k	2	4	8	16	32	64	128	256
$\log(L_{2\sigma}(k))$	1.262^{***}	1.489^{***}	1.545^{***}	1.475^{***}	1.322^{***}	1.147^{***}	0.962^{***}	0.852^{***}
	(0.165)	(0.168)	(0.135)	(0.101)	(0.087)	(0.068)	(0.053)	(0.042)
$\phi(k)$	0.449^{***}	0.490^{***}	0.542^{***}	0.449^{***}	0.397^{***}	0.408^{***}	0.443^{***}	0.656^{***}
	(0.013)	(0.046)	(0.048)	(0.040)	(0.028)	(0.023)	(0.024)	(0.032)
$\log(L_{2\sigma}(k)) \cdot \phi(k)$	-0.042^{***}	-0.047	-0.089^{***}	-0.047^{**}	-0.032^{**}	-0.043^{***}	-0.048^{***}	-0.083^{***}
	(0.010)	(0.033)	(0.027)	(0.020)	(0.013)	(0.008)	(0.006)	(0.006)
Constant	-8.379^{***}	-8.587^{***}	-8.636^{***}	-8.605^{***}	-8.509^{***}	-8.403^{***}	-8.283^{***}	-8.168^{***}
	(0.148)	(0.157)	(0.151)	(0.139)	(0.131)	(0.123)	(0.117)	(0.110)
Observations	284,807	284,807	284,807	284,807	284,807	284,807	284,807	284,807
Log Likelihood	−994.511	−894.892	−921.593	−965.256	−1,009.150	−1,065.808	−1,102.526	−1,152.981
Akaike Inf. Crit.	1,997.022	1,797.785	1,851.187	1,938.513	2,026.300	2,139.616	2,213.053	2,313.962
Note:								*p<0.1; **p<0.05; ***p<0.01

6
Conclusion

Approximate nearest neighbour (ANN) search is unparalleled faster than a naive approach and is convenient to set up kNN for sizeable data sets. An optimum exists between the search of the k number of neighbours and the size of data N which speeds up the process significantly. Further, tuning of parameters does not affect the outcome and query speed considering the given data. ANN-search is applicable for the set up of similarity graphs and kNN classification.

The assortativity coefficient r is calculated for similarity graphs. The equations are altered to ommit the bias of the class imbalance. The assortativity coefficient ranges from $r = 0.66$ for $k = 2$ up to $r = 0.41$ for $k = 256$. The tendency of within class connection is relatively high in comparison to supplemental empirical findings. Therefore, the presumption is that criminals tend to collaborate in fraud rings, and thus fraudulent transactions exist in clusters. Yet, analyses of weakly connected components and Louvain clusters are unfavourable to distinguish the fraudulent clusters in similarity graphs. The belief is that the relationships from a similarity graph are too artificial to be valid for clustering techniques.

The similarity graph is taken to test three hypotheses to contradict fraudulent and legitimate transactions. First, topological measures such as the local clustering coefficient, PageRank, and the eigenvector are indifferent over classes. The belief is that the structure of the graph is futile because the graph is too artificial. Second, fraudulent transactions yield a significantly higher distance to their neighbouring similar transactions and thus are anomalies. Third, fraudulent transactions connect to a significantly higher amount of further fraudulent transactions and therefore exist in fraud rings. Furthermore, statistical inference indicates the effect of fraud rings outweighs the effect of anomalies, therefore fraudulent transactions should be considered as *collective* anomalies. The variables interaction effect suggests that fraudulent transactions are no definite local outliers, but are near to further fraudulent transactions.

Thorough model evaluation leads to two creditable logistic regression classification models. The first model entails two variables; the standard deviation of the distance to neighbours $\log(L_{2\sigma}(k))$ and the fraction of fraudulent neighbours $\phi(k)$. The variables stem from $k = 256$ number of neighbours. The model includes quadratic polynomials and an interaction effect. The second model also entails the transaction amount A as third variable. The variables stem from $k = 128$ number of neighbours. The model includes quadratic polynomials and interaction effects. The models are compared to ongoing state-of-the-art logistic regression (LR) and multilayer perceptron models (MLP) from Kaggle. The models achieve a relatively high AUC and $precision$, yet, the $recall$ is lower than the MLP model. Remarkable is that the models from this research are theoretically relevant and do not operate inside the black-box. The fact that a theoretical relevant model competes with an Artificial Neural Network (MLP) is engaging and puts emphasis on causal learning over machine learning.

7 Discussion

The outline of the discussion is as following. First is a retrospect on validity and usability of the given data. Second, the methodology is reviewed on what obstacles exist and where opportunities lie. Third are further suggestion and remarks. All sections also entail a discussion of future work.

7.1. Data

The data is anonymous and therefore open to use for science. Openness leads to high collaboration e.g., sharing of code. Unfortunately, therefore the data is altered from its original structure and actual relationships are lost. Establishing a similarity graph through ANN-search is a workaround, and therefore the data quality diminishes twice. First the transformation to PCA and second the limited k in ANN-search both cause loss of variance in the data. Therefore the validity of creating similarity graphs is low. Performing the current research on the original data may lead to immense improvements of AUC, $precision$, and $recall$ causing the models from this research likely to be superior over the MLP model. Besides, the original data yields actual relationships of fraud rings e.g., fraud rings exist over geographical space and therefore are straightforwardly identifiable by locations of transactions. Yet, the use of the original data is unethical to publish. A solution may lie in simulating credit card transactions[1].

Notwithstanding, the data is sizeable and therefore various metrics are ought to be non-extractable i.e., computationally impossible. Furthermore, the data is by nature imbalanced which hinders the classification process. Both these issues are a universal challenge for data science and requires advancements within computational sciences in general.

7.2. Methodology

7.2.1. Approximate Nearest Neighbour Search

The ANN-search performs well on the given data i.e., there is no sacrifice of accuracy and the query does not require optimisation of parameters. Likely the data is simple enough for the algorithm to find the exact neighbours instead of the ANN. Namely, conventionally vector indexing is applied on million-, billion-, or even trillion-vector data sets for complex data e.g., image and video search. Therefore, the 1-dimensional data of credit card transactions is straightforwardly scalable. Yet, the implementation of time dependency and expanding similarity graph for real-time FDS remains an open issue.

[1] https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_3_GettingStarted/SimulatedDataset.html

7.2.2. Similarity Graphs

In concordance to the data, the original data exists of explicit relationships. The anonymous PCA data and computation of the Euclidean distance alters the view on relationships, therefore it is unclear what a top-k actually entails. Hence, what does a similarity graph actually entail and when is it realistic? Computations of the assortativity coefficient r are artificial, yet, the coefficient is high over numerous k and therefore could be seen as truth. Therefore, the given data remains as an exceptional case of assortative mixing.

Various metrics could not be extracted due to the size of the graph. Moreover, several algorithms are computational intensive and incompatible with similarity graphs in general. Such computational intensive algorithms and queries are analysis of transitivity, betweeness, closeness, personalised PageRank, and assortative mixing in ego-networks. Making Cypher and Python communicate would be a solution to query graph algorithms based on feedback from Python. Due to the lack of time such queries are refrained from.

The extraction of Node2Vec is attempted over various k number of neighbours considering a varying n number of vector embedding. The data is fed to an MLP model (due to flexibility of fit), however, the AUC, $precision$, and $recall$ where futile, and therefore Node2Vec is refrained from. Possibly, if the graph is not artificial there is potential in Node2Vec.

Louvain clustering seems incompatible on the similarity graphs. Scaling parameters such as ”maxLevels”, ”maxIterations”, and ”tolerance” for thorough search (up to Neo4j crashes) does not lead to a stable number of communities. Potentially, the Leiden algorithm could overcome the randomness of Louvain clustering. Last but not least, running Neo4j in the could on a GPU could increase functionality of algorithms .

7.2.3. Exploratory Research

The assortativity coefficient r required alterations to equate (Appendix E). The alterations are unconventional and aim to reduce the bias of class imbalance. The alterations focus on a subgraph of fraudulent transactions instead of the complete graph of fraudulent and legitimate transactions. The computation of assortative mixing is straightforward considering a fixed k number of neighbours (Appendix F). First, whether the alterations are applicable on artificial similarity graphs deserves scrutiny. Second, whether the alterations of equations hold for varying data requires further empirical investigation.

7.2.4. Explanatory Research

Statistical inference explains scientific truths of populations. The data in this research suggests a sample of two days, therefore notable is that the results are not representative for the population i.e., a year. In other words, shopping (and fraudulent) behaviour differ over the year, and thus entail different data. Besides, due to the imbalanced data the constant (i.e., the intercept) is biased. For these two reasons modeling on the entire given data set is not applicable for classification of the population.

The fact that no significant difference exists in topological structure between fraudulent and legitimate transactions does not mean that the hypothesis is untrue. Namely, the graph is artificially created. Likely, differences do exist, but not considering the approach of this research. Future research could be done through simulating credit card transactions.

A practical assumption of LOOCV is that transactions enter the database one-by-one and can be classified successively. Yet, the time order is neglected for ease of model evaluation. LOOCV also assumes all fraudulent and legitimate transactions are classified, except one. However, for ease of computation, the variable $\phi(k)$ is computed over the full data a priori. Therefore, bias exist due to the fact that several relationships can not be known a priori. The bias is large for a low k number of neighbours, fortunately, the bias diminishes for $k >> 1$ since the error for such computations is $1/k$. All in all, a low k number of neighbours is undesirable.

The classification model assumes each fraudulent transactions is equal. But in actual, one fraudulent transactions weighs more than another fraudulent transaction because of the transaction amount. Cost-sensitive learning could be considered to overcome this issue.

7.3. Further Remarks

In this research the metrics of anomalies L_{2*} and fraud rings ϕ are straightforwardly and intuitively defined. Further research in L_2 could focus on the improvement of these metrics. For example, the local outlier factor (LOF) is a useful metric for detecting density-based outliers, so to say, numerous supplementary metrics exist that could enhance the models. In terms of fraud rings ϕ, further research in (approximate) kNN classifiers is necessitate.

The emphasis of this research is put on theoretically relevant variables. Namely, fraudulent transactions are anomalies and exist within fraud rings. A follow-up research could focus on how fraudulent behaviour occurs. A research gap remains in questions such as; ”what are fraud rings?”, and ”how do fraudulent transactions relate in fraud rings?” Therefore, qualitatively research in fraudulent behaviour is necessary. Specifically, qualitative understanding (inductivism) of fraudulent transactions would lead to novel propositions.

Overall, this research is a step closer to causal learning and therefore diverges from machine learning. The research shows that theoretical understanding is useful because it is less reliant on algorithms that operate within the black-box. Further understanding in causalities would reduce the dependency on historic data if coefficients are directly applicable without training a model. Alas, the class imbalance throws a spanner in the works and coefficients are only representative to undersampling of the data set.

References

Ahmed, M., Mahmood, A. N. & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31.

Awoyemi, J. O., Adetunmbi, A. O. & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. *2017 international conference on computing networking and informatics (ICCNI)*, 1–9.

Banerjee, A. & Dave, R. N. (2004). Validating clusters using the hopkins statistic. *2004 IEEE International conference on fuzzy systems (IEEE Cat. No. 04CH37542)*, *1*, 149–153.

Bastani. (2013). Bank fraud detection. https://github.com/neo4j-contrib/gists/blob/master/other/BankFraudDetection.adoc

Bryman, A. (2016). *Social research methods*. Oxford university press.

Calvino, A., Aldeguer, P. & Domingo-Ferrer, J. (2017). Factor analysis for anonymization. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 984–991.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y. & Bontempi, G. (2018). Scarff: A scalable framework for streaming credit card fraud detection with spark. *Information fusion*, *41*, 182–194.

Chen, R.-C., Luo, S.-T., Liang, X. & Lee, V. C. (2005). Personalized approach based on svm and ann for detecting credit card fraud. *2005 international conference on neural networks and brain*, *2*, 810–815.

Dal Pozzolo, A. (2015). Adaptive machine learning for credit card fraud detection.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C. & Bontempi, G. (2017). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, *29*(8), 3784–3797.

Dal Pozzolo, A., Caelen, O., Johnson, R. A. & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. *2015 IEEE symposium series on computational intelligence*, 159–166.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S. & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, *41*(10), 4915–4928.

Deng, Z., Zhu, X., Cheng, D., Zong, M. & Zhang, S. (2016). Efficient knn classification algorithm for big data. *Neurocomputing*, *195*, 143–148.

Dreżewski, R., Sepielak, J. & Filipkowski, W. (2015). The application of social network analysis algorithms in a system supporting money laundering detection. *Information Sciences*, *295*, 18–32.

Gogoi, P., Bhattacharyya, D. K., Borah, B. & Kalita, J. K. (2011). A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, *54*(4), 570–588.

Guo, G., Wang, H., Bell, D., Bi, Y. & Greer, K. (2003). Knn model-based approach in classification. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 986–996.

Gürsoy, G. & Varol, A. (2021a). Prediction of arrhythmia with machine learning algorithms. *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, 1–5.

Gürsoy, G. & Varol, A. (2021b). Risks of digital transformation: Review of machine learning algorithms in credit card fraud detection. *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, 1–6.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the roc curve. *Machine Learning*, *77*, 103–123.

Hand, D. J. & Anagnostopoulos, C. (2013). When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? *Pattern Recognition Letters*, *34*(5), 492–495.

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Hopkins, B. & Skellam, J. G. (1954). A new method for determining the type of distribution of plant individuals. *Annals of Botany*, *18*(2), 213–227.

Indyk, P. & Motwani, R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 604–613.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Kou, Y., Lu, C.-T., Sirwongwattana, S. & Huang, Y.-P. (2004). Survey of fraud detection techniques. *IEEE international conference on networking, sensing and control, 2004*, *2*, 749–754.

Li, W., Zhang, Y., Sun, Y., Wang, W., Li, M., Zhang, W. & Lin, X. (2019). Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1475–1488.

Malkov, Y. A. & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, *42*(4), 824–836.

McPherson, M., Smith-Lovin, L. & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, *27*(1), 415–444.

Needham, M. & Hodler, A. E. (2018). A comprehensive guide to graph algorithms in neo4j. *Neo4j. com*.

Newman, M. (2002). Assortative mixing in networks. *Physical review letters*, *89*(20), 208701.

Newman, M. (2003). Mixing patterns in networks. *Physical review E*, *67*(2), 026126.

Newman, M. (2018). *Networks*. Oxford university press.

Nilson Report. (2021). Card fraud worldwide - nilsonreport.com. https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf

Pourhabibi, T., Ong, K.-L., Kam, B. H. & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, *133*, 113303.

Robinson, I., Webber, J. & Eifrem, E. (2015). *Graph databases: New opportunities for connected data*. " O'Reilly Media, Inc."

Sadowski & Rathle. (2014). Fraud detection: Discovering connections with graph databases. *White Paper-Neo Technology-Graphs are Everywhere*, *13*.

Schwertner, K. (2017). Digital transformation of business. *Trakia Journal of Sciences*, *15*(1), 388–393.

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. & Kuruwitaarachchi, N. (2019). Real-time credit card fraud detection using machine learning. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 488–493.

Université Libre de Bruxelles, M. L. G. .-. (2018). Credit card fraud detection. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?datasetId=310&sortBy=voteCount

Wang, C., Liu, Z., Gao, H. & Fu, Y. (2019). Vos: A new outlier detection model using virtual graph. *Knowledge-Based Systems*, *185*, 104907.

Wang, J., Yi, X., Guo, R., Jin, H., Xu, P., Li, S., Wang, X., Guo, X., Li, C., Xu, X. et al. (2021). Milvus: A purpose-built vector data management system. *Proceedings of the 2021 International Conference on Management of Data*, 2614–2627.

Weber, R., Schek, H.-J. & Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *VLDB*, *98*, 194–205.

West, J. & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & security*, *57*, 47–66.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, *48*(9), 2839–2846.

Zhang, S., Li, X., Zong, M., Zhu, X. & Wang, R. (2017). Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, *29*(5), 1774–1785.

A
Descriptive Statistics

Table A.1: Raw Data Descriptive Statistics Summary.

Statistic	N	Mean	St. Dev.	Min	Max
Time	284,807	94,813.86	47,488.15	0	172,792
V1	284,807	0.00	1.96	−56.41	2.45
V2	284,807	0.00	1.65	−72.72	22.06
V3	284,807	0.00	1.52	−48.33	9.38
V4	284,807	0.00	1.42	−5.68	16.88
V5	284,807	0.00	1.38	−113.74	34.80
V6	284,807	0.00	1.33	−26.16	73.30
V7	284,807	0.00	1.24	−43.56	120.59
V8	284,807	0.00	1.19	−73.22	20.01
V9	284,807	0.00	1.10	−13.43	15.59
V10	284,807	0.00	1.09	−24.59	23.75
V11	284,807	0.00	1.02	−4.80	12.02
V12	284,807	0.00	1.00	−18.68	7.85
V13	284,807	0.00	1.00	−5.79	7.13
V14	284,807	0.00	0.96	−19.21	10.53
V15	284,807	0.00	0.92	−4.50	8.88
V16	284,807	0.00	0.88	−14.13	17.32
V17	284,807	0.00	0.85	−25.16	9.25
V18	284,807	0.00	0.84	−9.50	5.04
V19	284,807	0.00	0.81	−7.21	5.59
V20	284,807	0.00	0.77	−54.50	39.42
V21	284,807	0.00	0.73	−34.83	27.20
V22	284,807	0.00	0.73	−10.93	10.50
V23	284,807	0.00	0.62	−44.81	22.53
V24	284,807	0.00	0.61	−2.84	4.58
V25	284,807	0.00	0.52	−10.30	7.52
V26	284,807	0.00	0.48	−2.60	3.52
V27	284,807	0.00	0.40	−22.57	31.61
V28	284,807	0.00	0.33	−15.43	33.85
Amount	284,807	88.35	250.12	0.00	25,691.16
Class	284,807	0.002	0.04	0	1

B
Scatter Plot

Figure B.1 shows the distributions in Figure 3.1 distinct into a fraudulent neighbourhood as a collective outlier. Yet, some fraudulent transactions mingle among legitimate transactions. Note, the plot depicts 3 out of 28 features and is undersampling the legitimate transactions (which is vastly deceiving), and therefore is not a holistic view on the data.

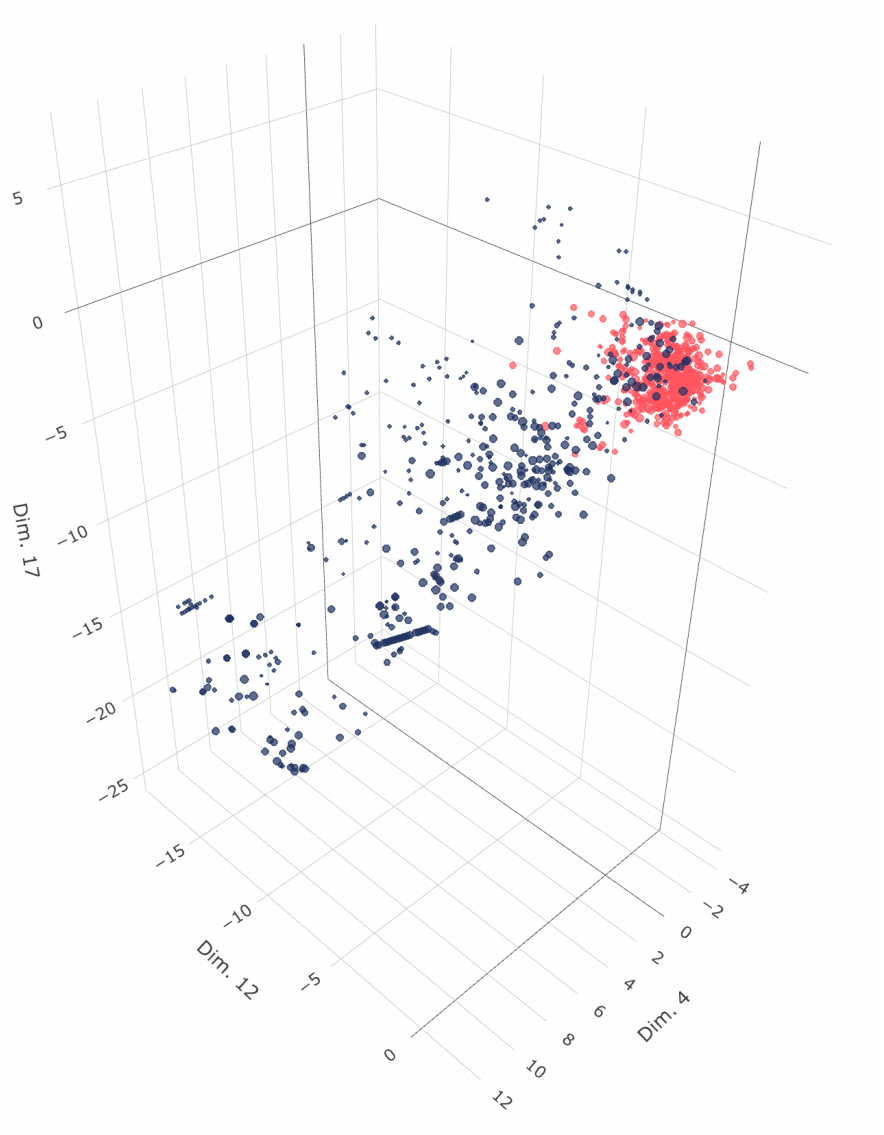


Figure B.1: Distribution of data (i.e., vectors) over three-dimensional space for principal component dimensions four, twelve, and seventeen. Fraudulent transactions is in red, and a sample of legitimate transactions is in blue.

C
Neo4j Admin Import

The following is an explanation on how to perform Neo4j admin import a.k.a. bulk loading data. The admin import exists of terminal instructions to force a graph database which either entirely fails to execute or completes flawlessly. The admin import requires a specific (relational) data structure. The admin import is able to process billions of relationships per minute and is the only viable import for sizeable data (Robinson et al., 2015).

The importation of a single graph requires four CSVs. Two CSVs for creating nodes and their properties and two CSVs for creating relationships between nodes and their properties. One of the two CSVs defines the headers of columns, the other CSV entails the data. The structure is as following:

node_header.csv

```
id:ID(node-ref), time:int, amount:float, class:int, :LABEL
```

relationship_header.csv

```
:START_ID(node-ref), distance:float, inverse:float, priority:int, :END_ID(node-ref
    ), :TYPE
```

The type of the column and the type of numerical data needs to be clearly specified for a competent import. The execution is as following. First, create a new Neo4j local database, namely, it is not allowed to have any entities stored yet because the files will be imported through an admin command which creates the database from scratch. Second, locate `/Users/{user_name}/Library/ApplicationSupport/Neo4jDesktop/Application/relate-data/dbmss/dbms-{security_id}`, this is the local database (for OS X). Third, locate the files and move them to the folder `/import` in the local database directory. Fourth, in terminal, change directory to the local database and run as a single line:

```
bin/neo4j-admin import --nodes import/creditcard_header.csv,import/creditcard.csv
    --relationships import/relations_header.csv,import/relations.csv
```

Fifth, confirm there's no errors in the terminal and whether the database yields about all nodes and relationships. For more information about the admin import consult https://neo4j.com/docs/operations-manual/current/tools/neo4j-admin/neo4j-admin-import/ and the instructional book by Robinson et al. (2015).

D
Cypher Queries & Algorithms

The following is an overview of cypher queries and algorithms in the research process.

First the projection of the graph puts the data in memory on which algorithms are applicable. Nodes' relationships to themselves are removed, and filtering on k is made possible. The weight is the inverse of the Euclidean distance. The name of the projection is 'transactions'.

```
CALL gds.graph.project.cypher('transactions',
  'MATCH (n:Node) RETURN id(n) AS id',
  'MATCH (f:Node)-[s:SIMILAR_TO]->(t:Node) WHERE s.priority <= 256 AND NOT s.
      distance = 0 RETURN id(f) AS source, id(t) AS target, s.inverse AS weight')
YIELD graphName AS graph, nodeQuery, nodeCount AS nodes, relationshipQuery,
    relationshipCount AS rels
```

The weakly connected components (WCC) is extracted as following:

```
CALL gds.wcc.stream('transactions') YIELD nodeId, componentId
RETURN nodeId, componentId
```

The statistics of Louvain clustering are as following:

```
CALL gds.louvain.stats('transactions', {relationshipWeightProperty: 'weight',
maxIterations: 1000, tolerance: 0.000001, maxLevels: 100})
YIELD communityCount, modularity, modularities
```

The topological metrics are called as following:

```
CALL gds.degree.stream('transactions') YIELD nodeId AS id, score AS degree
CALL gds.localClusteringCoefficient.stream('transactions') YIELD nodeId AS id,
    localClusteringCoefficient AS lcc
CALL gds.pageRank.stream('transactions') YIELD nodeId AS id, score AS PageRank
CALL gds.eigenvector.stream('transactions') YIELD nodeId AS id, score AS
    eigenvector
CALL gds.betweenness.stream('transactions') YIELD nodeId AS id, score AS
    betweenness
CALL gds.beta.closeness.stream('transactions') YIELD nodeId AS id, score AS
    closeness
```

Personalised PageRank is extracted as following:

```
MATCH (f:Node) WHERE f.class = 1 WITH collect(f) AS frauds
CALL gds.pageRank.stream('transactions', {relationshipWeightProperty: 'weight',
    maxIterations: 10, dampingFactor: 0.85, sourceNodes: frauds, tolerance:
    0.0001}) YIELD nodeId AS id, score AS PPR
```

The vector embedding (with e.g., 8 output-dimensions) is extracted as following:

```
CALL gds.beta.node2vec.stream('transactions', {embeddingDimension: 8,
    relationshipWeightProperty: 'weight'}) YIELD nodeId, embedding
RETURN nodeId, embedding
```

E
Assortative Mixing for a Sub-Graph

The class imbalance between fraudulent and legitimate transactions causes the assortativity coefficient to be biased. The alteration of Equation E.7 is as following.

The activity of fraudulent behaviour clusters into fraud rings which can be measured by the assortativity coefficient r. The distinction between the classes of fraudulent and legitimate classes is an enumerative characteristic i.e., a finite set of possible values. Due to the class imbalance the equations are altered to be class-specific for a sub-graph of fraudulent transactions. The number of actual relationships between nodes of an identical class is:

$$A_E = \sum_{(i,j)\in E} \delta(c_i, c_j) = \frac{1}{2} \sum_{i,j} a_{i,j} \delta(c_i, c_j) \tag{E.1}$$

where E is the set of relationships in the graph and $a_{i,j}$ is the number of actual relationships between node i and j. The factor one-half accounts for the relationships being undirected. The Kronecker delta mathematically accounts for the nodes to be of identical class:

$$\delta(c_i, c_j) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases} \tag{E.2}$$

The total number of undirected relationships in the sub-graph m_f depends directly on the k number of neighbours and the number of fraudulent transactions N_f:

$$m_f = N_f \cdot k. \tag{E.3}$$

The expected number of relationships between nodes of an identical class is a mathematical estimation as if the classes are spread randomly over the graph. The degrees d are reliant on the top-k and the total possible relationships m_f as given in Equation E.3:

$$E_E = \frac{1}{2} \sum_{i,j} \frac{d_i d_j}{2m_f} \delta(c_i, c_j) = \frac{1}{2} \cdot \frac{k^2}{2m_f} \cdot N_f \cdot (N_f - 1) = \frac{1}{2} \cdot \frac{k^2}{2 \cdot N_f \cdot k} \cdot N_f \cdot (N_f - 1) \tag{E.4}$$

conventionally, nodes i and j yield a degree d_i and d_j respectively. But, ANN-search returns a set number of nearby vectors $d_n = k$, and therefore $d_i d_j = k^2$. The total number of class-specific relationships $2 \cdot m_f$ is considered as all relationships from and to fraudulent nodes. Since the attributes d_i, d_j, and m_f remain constant over all nodes, the sum over the Kronecker delta is replaceable by the total number of relationships possible between all fraudulent nodes in a directed graph is equal to $N_f \cdot (N_f - 1)$ (Newman, 2018).

The modularity of the sub-graph is a measure of difference between the actual and the expected number of relationships:

$$Q = \frac{1}{2m_f} \sum_{i,j} \left(a_{i,j} - \frac{d_i d_j}{2m_f} \right) \delta(c_i, c_j) \tag{E.5}$$

whereas the maximum possible modularity is the difference between the total and the expected number of relationships:

$$Q_{max} = \frac{1}{2m_f} \left(2m_f - \sum_{i,j} \frac{d_i d_j}{2m_f} \delta(c_i, c_j) \right). \tag{E.6}$$

All in all, normalising modularity results in the assortativity coefficient r:

$$-1 \leq r(E,k) = \frac{Q}{Q_{max}} = \frac{\sum_{i,j}(a_{i,j} - d_i d_j / 2m_f)\delta(c_i, c_j)}{2m_f - \sum_{i,j}(d_i d_j / 2m_f)\delta(c_i, c_j)} = \frac{A_E - E_E}{m_f - E_E} \leq 1 \tag{E.7}$$

all in all, the total relationships m_f is modified for a class-specific sub-graph, because otherwise, the assortativity coefficient approaches the maximum $r \simeq 1.00$ due to the class imbalance. Therefore, only the assortativity between fraudulent nodes is of interest. All the relationships between fraudulent and legitimate nodes are neglected in the above equations because of the Kronecker delta. The amount of relationships between legitimate nodes is biased due to class imbalance, and therefore is not of interest. The amount of relationships between the fraudulent nodes is variable and fabricates an intriguing case for investigating whether fraud rings exist. Appendix F exemplars a calculation from results for $k = 256$.

Further assumptions are made. Namely, Neo4j requires relationship to be directed, whilst a similarity metric is actually undirected. Though, from the perspective of top-k, a certain vector can be a nearby neighbour of another vector but not vice versa. All nodes have an outdegree of $d_n^+ = k$, however, the indegree d_n^- is variable, thus $d_n^+ \neq d_n^-$. The equation for the expected number of relationship in a directed graph is:

$$E_E = \frac{1}{2} \sum_{i,j} \frac{d_i^+ d_j^-}{2m_f} \delta(c_i, c_j). \tag{E.8}$$

Fortunately, rewiring d_n^- leads to the an equal result as of $d_n^+ = d_n^-$, because in total an equal number of relationship remains. Therefore, the equation in Appendix E is stated as in Equation E.4, which is straightforwardly calculable through the number of fraudulent transactions m_f and number of nearby neighbours k, thus $d_i d_j = k^2$.

F
Example Assortativity Coefficient

As following is an example of a calculation on how the assortativity coefficient r is calculated for a sub-graph considering fraudulent-to-fraudulent transactions for $k = 256$.

Overall the total number of relationships for the fraudulent sub-graph is:

$$m_f = N_f \cdot k = 492 \cdot 256 = 125,952 \tag{F.1}$$

The first step is counting the actual number of relationships between all nodes of identical class. The Cypher query to count all unidirectional relationships is given below:

$$A_E = \sum_{(i,j) \in E} \delta(c_i, c_j) = \frac{1}{2} \sum_{i,j} a_{i,j} \delta(c_i, c_j) = \frac{1}{2} \cdot 140,774 = 70,387 \tag{F.2}$$

```
MATCH (from_node)-[s:SIMILAR_TO]-(to_node)
WHERE from_node.class = 1 AND to_node.class = 1
    AND s.priority <= 256 AND NOT s.priority = 0
RETURN count(s)
```

The second step is estimating the expected number of relationships between fraudulent transactions, note, the latter part of the equation is simplified by deducting k and N_f:

$$E_E = \frac{1}{2} \sum_{i,j} \frac{d_i d_j}{2m_f} \delta(c_i, c_j) = \frac{1}{2} \cdot \frac{k^2}{2 \cdot N_f \cdot k} \cdot N_f \cdot (N_f - 1) = \frac{1}{2} \cdot \frac{256}{2} \cdot (492 - 1) = 31,424 \tag{F.3}$$

All in all, the assortativity coefficient r is calculable by:

$$r(E, k) = \frac{A_E - E_E}{m_f - E_E} = \frac{70,387 - 31,424}{125,952 - 31,424} = 0.41 \tag{F.4}$$