UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Mining Heterogeneous Enterprise Data

by

Xinxin Jiang

A thesis submitted in partial fulfilment
of the requirements for the degree

Doctor of Philosophy

Dissertation Directed by

Principal Supervisor Professor Chengqi Zhang and Co-Supervisor Doctor Guodong Long

Centre for Artificial Intelligence, FEIT

Sydney, Australia

2018

Certificate of Authorship/Originality

I, Xinxin Jiang declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

© Copyright 2018 Xinxin Jiang

Production Note:
Signature removed
prior to publication.

Dedication

To my dear parents, parents-in-law, husband and daughter

Acknowledgements

First and foremost, I would like to express my earnest thanks to my supervisors, Professor Chengqi Zhang and Doctor Guodong Long, who have provided tremendous support and guidance for my PhD study and research. Prof Zhang provided me with an opportunity to study in the stimulating and interactive Centre for Artificial Intelligence, where I met and learned a lot from many smart and talented people. I have benefitted significantly from his generous help and invaluable suggestions on my research. I would like to thank Doctor Guodong Long for his continual guidance and supervision during my PhD study. He has always encouraged me to think about and explore my research interests. His creativeness and enthusiasm in solving challenging problems have greatly inspired my work. Without their professional guidance and consistent help, this dissertation would not have been possible.

I would also like to thank all the people that had a positive influence on my day-to-day enjoyment of research. My teachers, my leaders, my officemates, my friends, past and present: Prof Longbing Cao, Prof Ivor Tsang, Prof Guandong Xu, Mr Yongjun Lin, Mr Xiaorong Huang, Dr Shirui Pan, Dr Ling Chen, Dr Peng Zhang, Dr Jing Jiang, Dr Wei Liu, Dr Xueping Peng, Dr Can Wang, Dr Qinzhe Zhang and Dr Qin Zhang. They are the ones who have given me great support during both joyful and stressful times, and the ones to whom I will always be thankful.

Finally, I would like to thank my family, my husband Charles, my daughter Carolyn, my parents and parents-in-law for their unconditional encouragement and support. No words could possibly express my deepest gratitude for their endless love, self-sacrifice and unwavering help. To them, I dedicate this dissertation.

Xinxin Jiang

Sydney, Australia, 2018.

ABSTRACT

Mining Heterogeneous Enterprise Data

by

Xinxin Jiang

Enterprise data is complicated real-world data that is shared by the users of an organization, generally across departments and geographic regions. Heterogeneity is becoming one of the key characteristics inside enterprise data, because the current nature of globalization and competition stress the importance of leveraging huge amounts of enterprise accumulated data, according to various organizational processes, resources and standards. Effectively mining heterogeneous enterprise data is challenging. First, enormous amounts of enterprise data are being amassed from a greater variety of sources, which means traditional analysis methods are becoming less efficient and less effective. Second, enterprise data is becoming more and more heterogeneous. Primarily relate to social and technological trends and pertain to the shift in customer needs and expectations, today's enterprise data have more structures, greater complexity, and more intricate correlations than in years past. Third, learning tasks in enterprise applications have become more complex. Enterprise requirements are placing tighter constraints on data analysis, such as imbalanced data, class distributions, and multiple learning tasks, etc. Therefore, effectively deriving meaningful insights from complex large-scaled heterogeneous enterprise data poses an interesting, but critical, challenge.

The aim of this thesis is to investigate the theoretical foundations of mining heterogeneous enterprise data in light of the above challenges and to develop new algorithms and frameworks that are able to effectively and efficiently consider heterogeneity in four elements of the data: objects, events, context, and domains.

Objects describe a variety of business roles and instruments, for example, the or-

ganisations, departments, people, products, or services involved in business systems. Object heterogeneity means that object information at both the data and structural level is heterogeneous. The inherent complexity of heterogeneous objects, and the dynamic nature of business processes, means that learning heterogeneous business objects in serial architectures is either computationally infeasible or ineffective because of heavy pre-processing. However, the cost-sensitive hybrid neural network (Cs-HNN) proposed in this thesis leverages parallel network architectures and an algorithm specifically designed for minority classification to generate a robust model for learning heterogeneous objects. Results from experiments with real-world data indicate that Cs-HNN demonstrates superior performance over baseline procedures.

Events trace an object's behaviours or the sequence of an object's activities. Event heterogeneity reflects the level of variety in business events and is normally expressed in the type and format of features. Mining event heterogeneity aims to build effective and efficient mining models by considering the heterogeneous event-related factors. The natural complexity of event heterogeneity in real-world business means that traditional pattern mining approaches tend to be insufficient. Most models are based on the assumption of homogeneity, but real-world events, activities, and behaviours are more often connected by heterogeneous features types in complex ways. These connections carry critical information for mining fruitful results. Practical and efficient sequential pattern mining approaches are needed to overcome these problems. The approach proposed in this thesis focuses on fleet tracking as a practical example of an application with a high degree of event heterogeneity. Complex fleet rental event patterns are discovered by combining heterogeneous features with measurement algorithms to provide valuable insights for business. The results from experiments on real-world datasets verify the effectiveness of the approach.

Context describes the environment and circumstances surrounding objects and events. Context heterogeneity reflects the degree of diversity in contextual features. Mining context heterogeneity aims to design algorithms to efficiently analyse complicated, context-aware correlations in business prospects. Again, the complexity inherent in heterogeneous contexts and the lack of a straightforward way to capture

complicated context-aware factors both present significant challenges to heterogeneous data mining. However, the coupled collaborative filtering (CCF) approach proposed in this thesis is able to provide context-aware recommendations by measuring the non-independent and identically distributed (non-IID) relationships across diverse contexts using the inter-item, intra-context, and inter-context correlations between items, users and context-aware factors as the basis for the coupled similarity calculations. The results form a set of experiments that compare and contrast a range of baseline models to demonstrate the effectiveness of this approach.

Domains are the sources of information and reflect the nature of the business or function that has generated the data - for example, different industries, or a sales function vs a supply function. As with the previous types of heterogeneity, domain heterogeneity is reflected in the number of sources the data has been derived from and manifests in the both the data type and format. Mining domain heterogeneity aims to effectively and efficiently model the correlations among real-world heterogeneous domains. However, cross-domain deep learning (Cd-DLA) provides a potential avenue to overcome the complexity and nonlinearity of heterogeneous domains. The approach presented in this thesis learns the correlations within a domain, across several domains, and between time-series data, within a parallel multi-task learning architecture. In a setting that uses multiple financial markets to represent heterogeneous domains, Cd-DLA is able to analyse complex domain-related correlations by capitalising on attention mechanisms and a series of recurrent neural networks (RNNs), and then predict market trends in the next trading window. Experimental results on 10 years of financial data prove that this approach produces more accurate predictions than other baselines.

Each of the approaches, algorithms, and frameworks for heterogeneous enterprise data mining presented in this thesis outperform the state-of-the-art methods in a range of backgrounds and scenarios, as evidenced by a theoretical analysis, an empirical study, or both. All outcomes derived from this research have been published or accepted for publication, and the follow-up work has also been recognised, which demonstrates scholarly interest in mining heterogeneous enterprise data as a

research topic. However, despite this interest, heterogeneous data mining still holds increasing attractive opportunities for further exploration and development in both academia and industry.

Dissertation directed by Professor Chengqi Zhang and Doctor Guodong Long
Faculty of Engineering and Information Technology

List of Publications

The papers of my PhD research that have been submitted, accepted, and published follow.

Conference papers

C-1. **X. Jiang**, S. Pan, G. Long, J. Chang, J. Jiang, and C. Zhang, Cost-sensitive hybrid neural networks for heterogeneous and imbalanced data, in *International Joint Conference on Neural Networks (IJCNN)*, 2018. IEEE, 2018. (full paper accepted on 15th March 2018)

C-2. **X. Jiang**, S. Pan, J. Jiang, and G. Long, Cross-domain deep learning approach for multiple financial market prediction, in *International Joint Conference on Neural Networks (IJCNN)*, 2018 . IEEE, 2018. (full paper accepted on 15th March 2018)

C-3. **X. Jiang**, W. Liu, L. Cao, and G. Long, Coupled collaborative filtering for context-aware recommendation, in *AAAI*, 2015, pp. 41724173.

C-4. **X. Jiang**, X. Peng, and G. Long, Discovering sequential rental patterns by fleet tracking, in *International Conference on Data Science*. Springer, 2015, pp. 4249.

Journal papers

J-1. **X. Jiang**, S. Pan, G. Long, F. Xiong, J. Jiang, and C. Zhang, Cost-sensitive parallel learning framework for insurance intelligence operation, *IEEE Transactions on Industrial Electronics*, 2018. (full paper accepted on 9th September 2018)

Contents

Abbreviation

AUC - area under curve

BiLSTM - bi-directional long short-term memory

CCF - coupled collaborative filtering

CdNN - cross-domain neural network

CNN - convolutional neural network

Cs-HNN - cost-sensitive hybrid neural network

DNN - deep neural network

DsNN - description neural network

ELM - extreme learning machines

HNN - hybrid neural network

HUSPM - high-utility sequential pattern mining

IdNN - inner-domain neural network

IID - independent and identically distributed

LSTM - long short-term memory

MAE - mean absolute error

MF - matrix factorization

MLP - multi-layer perceptron

MTL - multi-task learning

non-IID - non-independent and identically distributed

RMSE - root mean squared error

RNN - recurrent neural network

SqNN - sequence neural network

SVM - support vector machine

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Mining heterogeneous data

Organisations today have more data than ever at their disposal [26] [48]. However, deriving meaningful insights from that data and converting it into actionable knowledge is still quite challenging for most enterprises [48] [62] [95] [118] [180] [190]. Enterprise heterogeneous data mining is the process of discovering patterns in large enterprise data sets, involving specialized machine learning algorithms and techniques with various data heterogeneities, such as object, event, context and domain heterogeneities, for better business decision making. It is an interdisciplinary subfield of artificial intelligence with an overall goal to extract information (with intelligent method) and transforming them into the valuable patterns and insights. Once identified, these patterns can help businesses spot sales trends, develop marketing campaigns, prepare smarter management strategies, and so on.

Heterogeneous data mining algorithms and techniques help industries generate new business opportunities in several ways.

1)*Predicting trends and behaviours*: Integrating heterogeneous mining technology into business processes can help companies make valuable predictions using large-scale datasets. Targeted market analysis is a good example of the power of predictive knowledge, where heterogeneous data mining techniques are used on past promotional mailings to identify the targets most likely to respond. Such predictions would help to maximise returns on future mailings. Other predictive problems in-

clude forecasting bankruptcy or potential loan defaults, or identifying the segments of a population that are likely to participate in similar events.

2)*Discovering hidden patterns*: The difficulties with manually analysing large volumes of data mean that valuable patterns in customer behaviour often go unnoticed. Heterogeneous data mining can reveal these patterns. For example, identifying seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions or identifying anomalous data.

With the growing accumulation of large-scale data, more and more business applications are seeing the benefits of mining heterogeneous data [56] [89] [118] [180]. A small selection of these applications follows.

- customer segmentation - identifying the characteristics common to customers who purchase specific products;
- fraud detection - identifying which transactions are likely to be fraudulent;
- direct marketing - identifying which prospects should be included in a mailing list to obtain the highest response rate; and
- trend analysis - revealing increases, decreases, or seasonal variations in business activities over a given time frame.

Given the great benefits data mining brings, scholars have developed a range of algorithms and analysis techniques to suit a variety of problems common to business [56] [89] [96] [180]. These include:

- decision trees [56] - tree-shaped structures that represent sets of decisions. These decisions generate rules for classifying a dataset;

- genetic algorithms [89] - optimisation techniques based on the concepts of genetic combination, mutation, and natural selection;
- nearest neighbour [89] - a classification technique that categorises each record based on the most similar records in a historical database; and
- artificial neural networks [180] - predictive models that resemble biological neural networks and learn through training.

However, these numerous legacy applications are continually adding complexity and diversity to enterprise data, particularly when combined with the immense processing power, cheap storage, a multitude of data sources, and the vast and dynamic landscapes that characterise modern business scenarios [26] [48] [62] [95] [118] [190]. Hence, enterprise data is increasingly becoming large-scale with extremely complex heterogeneity. As one of the main characteristics of real-world enterprise data, heterogeneity presents great challenges to traditional data mining technologies and has, therefore, attracted a great deal of recent attention in both scholarly studies and industry endeavours.

A typical end-to-end heterogeneous mining process comprises six phases, as shown in Table.1.1. The sequence of each phase is not fixed, and moving backward or forward between phases is allowed. However, the output of one phase is always the input for the next phase.

1.1.2 Types of heterogeneity in enterprise data

Compared to traditional datasets, heterogeneous enterprise data presents dramatic challenges to current data mining algorithms and techniques. However, modelling the complexity inherent in heterogeneous data becomes much simpler when the heterogeneity is broken down by data element and addressed accordingly. Fig 1.1 illustrates the four main type of heterogeneity in enterprise data.

Table 1.1 : The six phases of an end-to-end heterogeneous mining process

Phase name	**Description**
Understanding business	Determining the nature of the business that requires data mining
Data collection	Collecting heterogeneous data from various sources
Data selection	Selecting the relevant data features
Data learning	Applying various data mining techniques to the embedded data
Data evaluation	Verifying whether the selected approaches and their results meet business requirements based on measures
Knowledge presentation	Presenting the mined knowledge in a way business users find easy to interpret

Enterprise data heterogeneity is defined with four elements: object, event, context, and domain in Fig 1.1. In brief, object represents the business role or instrument in enterprise system, which has various data types and structures, reflecting the object's static and dynamic properties in its lifecycle. Event is used to trace the important behaviours or activities of one or more objects. Then context describes the contextual environment and circumstance surrounding the objects and events. Domain reflects the nature of the business or function area that has generated the objects, events and contexts. These four elements compose the whole picture of general enterprise data heterogeneity.

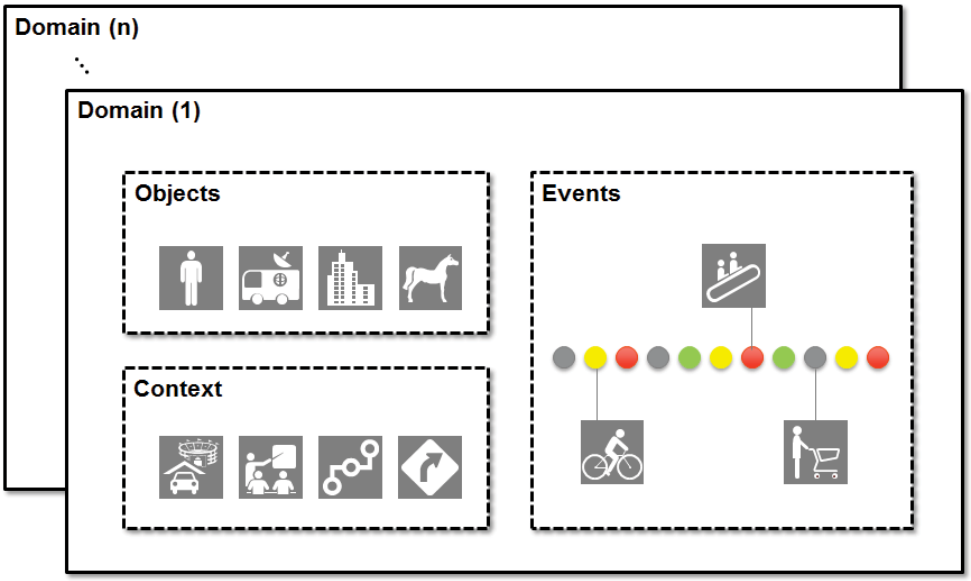

Figure 1.1 : Heterogeneity by element: object, event, context, and domain

Object heterogeneity

Objects describe business roles and instruments, for example, the organisations, departments, people, products, or services involved in business systems. Object heterogeneity reflects the variety of these roles in a dataset, and is normally found at the data level or the structural level. For example, an object's data features

might vary from department to department, facility to facility, or corporation to corporation in both the type of data and the structure of that data.

Heterogeneity at the data level means that the data values are stored as a mixture of types, e.g., integers, floats, characters, texts, images, media, etc., while heterogeneity at the structural level means that that data exists in different formats, e.g., static properties that tend not to change, and dynamic data that describes how the object changes over time. Fig. 1.2 illustrates object heterogeneity at the data and structural levels.

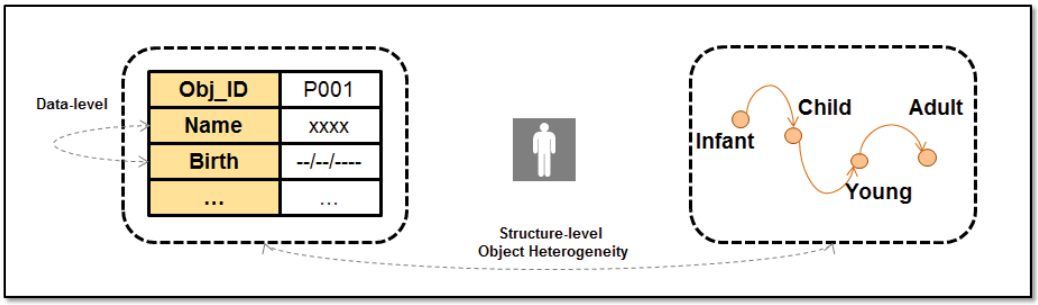

Figure 1.2 : An example of object heterogeneity

Event heterogeneity

Events trace an object's behaviours or activities. Event heterogeneity reflects the level of variety in business events and is normally expressed in the type and format of features. For example, common event features include time, effect, status, and so on - attributes that describe behaviour or action or reflect the priorities in a business strategy. Fig. 1.3 demonstrates that heterogeneous events (shown in different colours, such as red, green, gray, and yellow), can be evaluated using a similarity function that compares different features or different aspects of a feature, e.g., usage or utility. Even if a feature is described using different data types or formats, such functions are able to measure the effectiveness or efficiency of a business event.

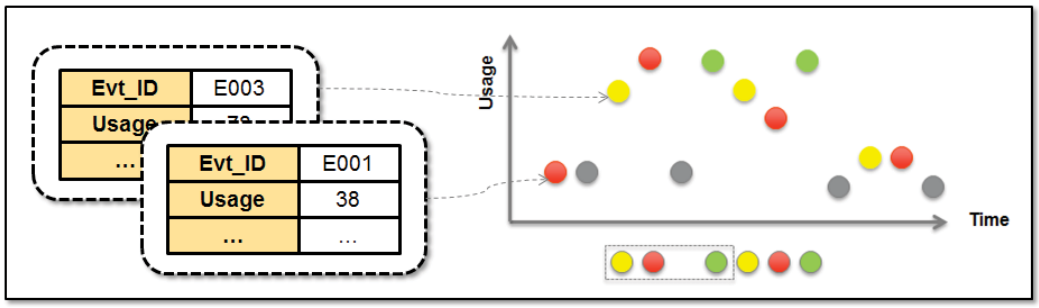

Figure 1.3 : An example of event heterogeneity

Context heterogeneity

Context describes the environment and circumstance surrounding objects and events, i.e., the business's behaviours and activities. Context heterogeneity reflects the degree of diversity in contextual features. Context heterogeneity is typically found in the type and format of the data. Fig. 1.4 illustrates a real-world business example of context heterogeneity with spatial data. Consider a series of devices that each collect data from a specific geographic location and a range of retail products that are each linked to a social community. Every piece of data collected would be sourced from a different context (the geographic location) and be applied to a different context (the retail product) and both contexts would be characterised by different data types and formats. Thus, the geographic location, the social relationships, and the various data types and formats associated with the contextual features all reflect context heterogeneity. Interestingly, this example also reflects some of the complexity associated with decision making in modern business.

Domain heterogeneity

Domains are the sources of information and reflect the nature of the business or function that has generated the data – for example, different industries, or a sales function vs a supply function. As with the previous types of heterogeneity, domain

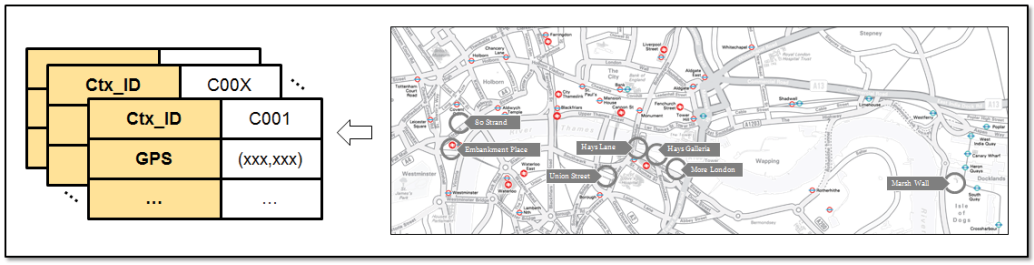

Figure 1.4 : An example of context heterogeneity

heterogeneity is reflected in the number of sources the data has been derived from and manifests in the both the data type and format. Fig. 1.5 illustrates domain heterogeneity using tourism, climate, and economics domains as an example. Each domain has its own business objects, events, and contexts. The internal features and sequences within a single domain trace the trends associated with that business or industry, such as historical tourist, accommodation, dining, and transport data within tourism domain demonstrating the trends of tourism. However, in practice, the data information associated with one domain often correlate to other heterogenous domains in some regard. In other words, for instance, trends in tourism may affect, or be affected by, trends in climate, economics or other heterogenous domains. These relationships are known as cross-domain correlations.

1.1.3 Challenges with mining heterogeneous enterprise data

As a result of the diverse objects, events, contexts, and domains within today's business systems, most enterprise data is heterogeneous, which means traditional data mining techniques often fall short [42] [56] [92] [180]. However, to improve the quality of products and customer service, and gain an in-depth understanding of hidden business opportunities, the ability to mine heterogeneous data has become a basic and fundamental part of doing business. For the most enterprises, the vast datasets they collect are effectively black boxes of untapped knowledge. The

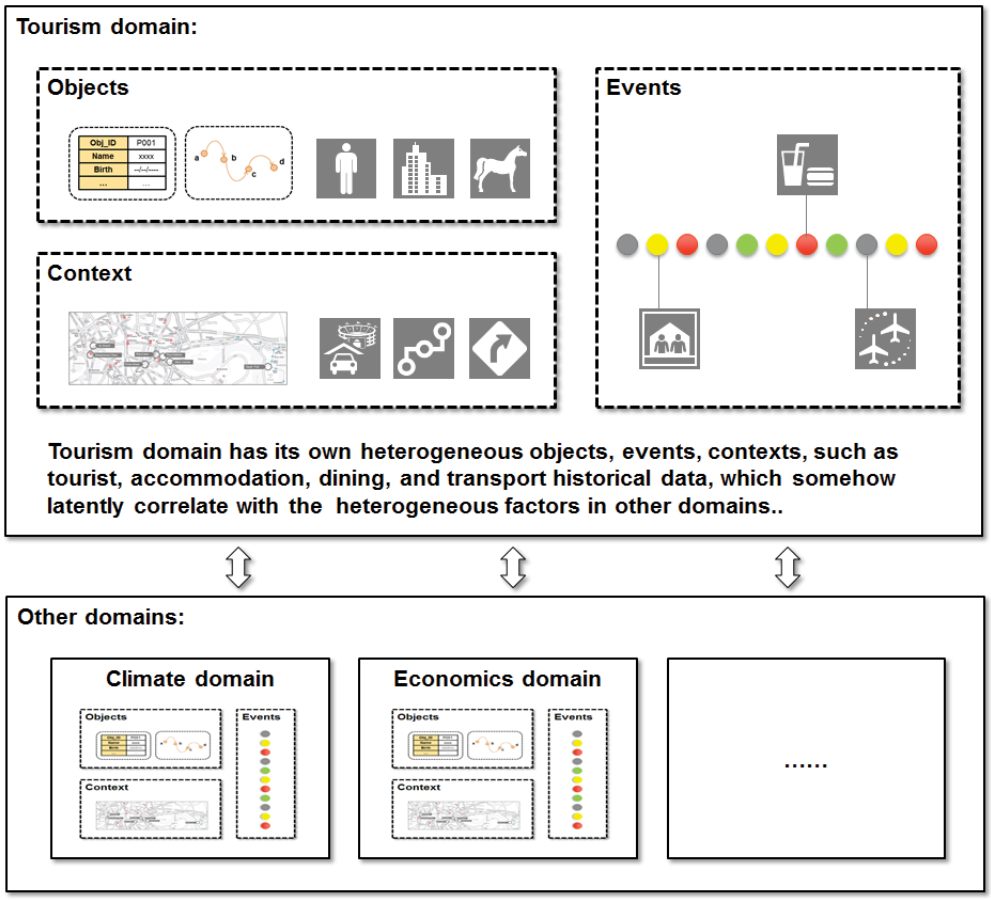

Figure 1.5 : An example of domain heterogeneity

methods and techniques needed to extract the valuable information they require from historical data are either elusive or beyond the technical ability of most managers and analysts, particularly in small-to-medium-sized enterprises. Further, the learning curve associated with heterogeneous data mining is growing every day. And, beyond the complexities associated with heterogeneous data discussed above, the data being collected may be inaccurate or ambiguous, have missing values or high levels of redundancy.

In general, most enterprises are confronted with a common set of problems to overcome when attempting to apply data mining techniques to heterogeneous data. However, the specific knowledge requirements, and therefore the most significant

challenges, do vary depending on the organisation [24] [42] [180].

Each of these issues is also rising new challenges for data mining technology. The major three technical challenges are huge volumes of data, complex heterogeneity, and diverse business requirements. Mining heterogeneous enterprise data demands highly suitable strategies and algorithms, more effective preprocessing steps such as data filtering and integration, advanced parallel learning architecture, and intelligent and effective user interaction. These common challenges are illustrated in Fig. 1.6.

Huge volumes of data from multiple sources

The enormous volume of data is, arguably, one of the biggest and most common challenges in heterogeneous data mining [26] [48] [71] [72]. Appropriate and efficient analysis methods to leverage massive volumes of heterogeneous data often simply do not exist. This is because different information collectors use their own schemata to record data, and the nature of different business processes also results in diverse representations of the data. In reality, trends like e-commerce, mobility, social media and the Internet of Things are generating so much information that almost every organisation likely faces this challenge.

- *Efficient learning architectures:* Typically, the architecture, software, and storage solutions for large-scale data are hosted on cloud computing platforms. However, to provide efficient solutions for heterogeneous data mining tasks, most current approaches are based on serial architectures deep learning, for example. As serial architectures is more adapted to specific learning areas, feeding the processed data in turn to utilize the advantages of each neural network without considering what kind of data it is more suitable for processing, processing huge volumes of heterogeneous business data usually results in performance problems.

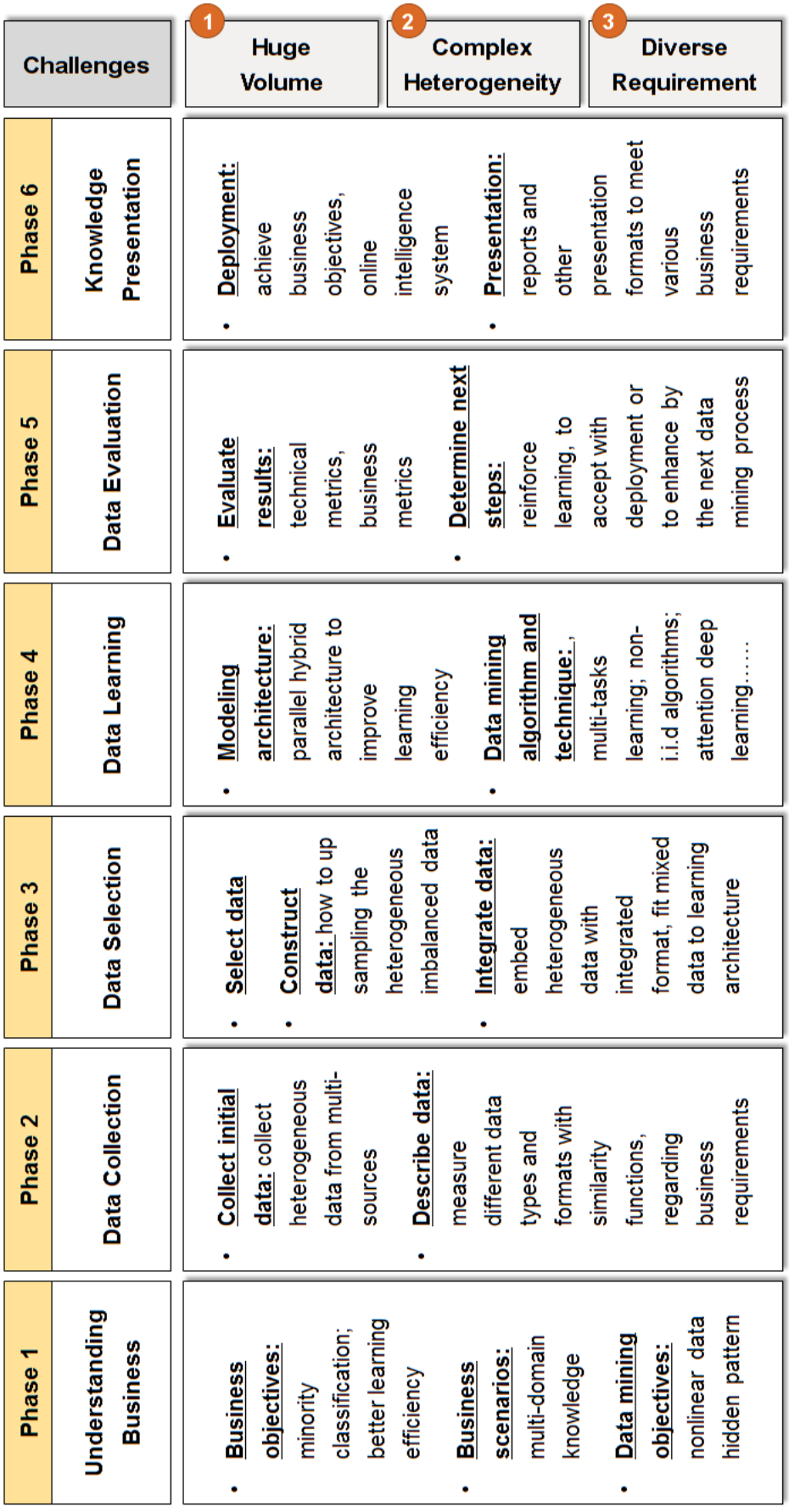

Figure 1.6 : Challenges associated with mining heterogeneous enterprise data

- *Multi-task learning:* The high competition in modern business requires real-world data analysis to be efficient and effective. Multi-task learning is a machine learning paradigm with the aim of leveraging useful information gleaned from multiple related tasks to help improve the overall performance of one specific task. Usually, each task in multi-task learning relies on a limited amount of training data, which is not large-scale. However, the size of the overall training set grows as the number of tasks increases, which makes the number of tasks required to produce the desired result one of the most significant factors in multi-task learning. Hence, finding an efficient architecture that can support effective multi-task learning with more tasks is an important problem.

Complex heterogeneity

As previously mentioned, the complexity of heterogeneous data can be defined more acutely defined through the elements of data: objects, events, context, and domain (see Fig. 1.2), reflecting a tremendous complexity of interrelated data elements with diverse types, formats, structures, and sources. Mining enterprise data regarding such complex heterogeneities, the great challenge is how to investigate the suitable methods and algorithms unveiling hidden patterns or knowledge dwelt at the non-linear intersections within heterogeneous enterprise data.

- *Data embedding:* Data embedding aims to pre-process heterogeneous data into suitably integrated formats for further stages of training and data analysis. Clearly, improper embeddings will reduce the value of the original data and may even obstruct data analysis entirely. Hence, studying efficient data embedding methods that are able to represent a range of data types, formats, and structures is a vital step in developing an effective solution to overcome complexity. Suitable technologies that engender efficiency in subsequent stages of analysis also need to be explored.

- *Learning algorithms:* Traditional machine learning algorithms work well with homogeneous data, assuming the dataset has been carefully prepared in the first steps of the knowledge discovery process. However, enterprise data typically contains a diverse range of data types, formats, and sources. Further, heterogeneous data usually occupy different positions in the data space. Hence, using a single learning method or one projection to extract patterns from heterogeneous data tends not to produce the accurate learning results. Therefore, any solid solution to heterogeneous data mining will require a series of learning algorithms.

- *Non-IID data:* Another important challenge to overcome is mapping the relationships between complex non-IID data. Most current machine learning methods typically treat all data features as independent and equal, i.e., IID. However, in practice, the information about one feature is usually linked to other features with a coupled relationship, and these non-IID relationships can have a significant influence over the accuracy or appropriateness of a prediction. For example, a customer's preferred drink at an airport in the morning may be very different from their preferred drink at a vacation spot in the evening, so a good customer recommendation system would need to consider both the time and location features when making predictions. Even in the same location, the underlying reasons of items and users may affect user preferences in different moments. Identifying these non-IID relationships presents another challenge for real-world heterogeneous data mining.

Diverse business requirements

The nature of globalisation and competition is driving the companies to rapidly turn heterogeneous data into significant insights to guide their marketing, investment, and management strategies. Diverse business requirements bring the chal-

lenges of heterogeneous data mining in high competition. For instance, efficiently learning the relevant knowledge from multiple domains, and accurately gauging customer responses to changes in business rules, would produce a powerful competitive advantage.

- *Transfer learning across multiple domains:* The widespread use of technology is bringing greater opportunities for companies to acquire knowledge from multiple domains. Analysing data from a variety of sources has substantial power to highlight aspects of business from different meaningful perspectives. As a result, there is great demand in industry to learn from knowledge that has been transferred from other domains. However, combining these data sets usually results in heterogeneity. Traditional approaches to knowledge transfer mostly focus on historical data from similar domains but suffer from several shortcomings due to nonlinear data and limitations on the number of features that can be correlated. Take financial market analysis as an example, most existing financial analysis methods rely on data from homogeneous markets and find modelling the complex nonlinear relationships between homogeneous and heterogeneous markets quite difficult.

- *Minority classifications for optimising business:* In real business, more and more enterprises are retooling their operational functions by focusing on leveraging the risk response with advancing business strategies. In data analysis, risks are often characterised by small probability events, which shifts the focus of the analysis to target minor instance classifications. For example, the incidence of claims on term life insurance policies is usually only about 0.3-0.5 per thousand, and the incidence of major illness claims is usually only 1-3 per thousand. Small probability rates lead to imbalanced data with extremely skewed outcomes that are difficult to optimise to meet business objectives. For

data-driven organisations, developing an effective approach to solving these heterogeneous data mining challenges is urgent.

1.2 Research objectives

Finding an effective way to exploit the knowledge that heterogeneous data offers has been a long-term challenge for both academia and industry. The business environment is constantly changing and, without a series of efficient data-driven methodologies as an intelligence centre to support end-to-end business operations, organisations gradually become inefficient, inflexible, and unable to meet changing customer demands.

Therefore, the research objectives of this thesis are to improve the theoretical foundations of algorithms and models designed to mine heterogeneous enterprise data, and to develop practical frameworks, algorithms, and analysis techniques for heterogeneous data that can meet current business demands. The research objectives in this thesis have been divided into the four types of data heterogeneity. The specific objectives associated with each type are detailed in the sections below.

1.2.1 Object heterogeneity

Learning efficiency problems and imbalanced data remain the two of the greatest challenges to addressing object heterogeneity. The serial architectures most traditional approaches rely on generally require a significant amount of data preprocessing and lack the learning efficiency required to meet business demands. Additionally, in business, analysis tasks often need to focus on capturing minority classes in data with extremely skewed and imbalanced data distributions, rather than the balanced classifications associated with traditional techniques. Hence, to overcome these two main challenges, the object heterogeneity research objectives in this thesis are to devise a unified, end-to-end architecture and the corresponding learning

algorithms for mining real-world heterogeneous and imbalanced business objects.

1.2.2 Event heterogeneity

The main research objectives in improving data analysis given event heterogeneity, are to improve data processing performance while providing greater insights into the hidden patterns within the data. Combining an innovative pattern mining method with appropriate measurement functions may yield these insights. For example, in rental industry, timestamps, usage, and the effect of events, among other fleet tracking factors, are widely considered to be important features in improving the performance of fleet rental companies and predicting customer preferences. Regarding enterprise data in event heterogeneity, customer behaviour is usually linked to a range of heterogeneous factors. Hence, the research objectives associated with event heterogeneity in this thesis are to study data mining algorithms and embedding techniques that unify the formats of heterogeneous event features and to develop a data mining approach for efficiently discovering event patterns in a specific business scenario - the fleet rental industry in this case.

1.2.3 Context heterogeneity

Context heterogeneity has historically been addressed by formulating generalised triadic relationships between two or more heterogeneous contexts. Specially, enterprise requirements demand novel approaches that can generate context-aware customer recommendations. Further, these approaches need to consider contextual factors that are not IID, but rather exist in coupled relationships with other factors. Thus, the main research objective in addressing context heterogeneity is to design an optimised mining algorithm that can effectively analyse non-IID contextual relationships and reveal the latent factors that influence prediction accuracy. Efforts will focus on coupled similarity calculations for inter-item, intra-context, and inter-context correlations among items, users, and contextual factors.

1.2.4 Domain heterogeneity

Domain heterogeneity is perhaps the most well-studied aspect of enterprise data heterogeneity with the increasing cross-domain algorithms and applications. Therefore, the research objectives for this data element focus on a machine learning approach coupled with suitable attention-based mechanisms to transfer knowledge and generate aggregated market trend forecasts across heterogeneous domains.

Table 1.2 provides the specific objectives associated with each type of heterogeneity. When combined, each contributes to a full and complete research objectives for mining heterogeneous enterprise data.

Table 1.2 : Research objectives for mining heterogeneous enterprise data

Element	**Research objective**
Object heterogeneity	* Develop a machine learning architecture that can efficiently analyse diverse data types and formats in large-scale complex data with object heterogeneity in the same epoch. * Study enhanced optimisation algorithms to develop solutions that can classify minor objects in real-world heterogeneous data to greatly improve the learning efficiency of business optimisation models.

Element	Research objective
Object heterogeneity	* Design an HNN that can analyse heterogeneity in a range of objects at both the data level and the structural level. Then, validate the effectiveness of the developed approach in enterprise datasets.
Event heterogeneity	* Analyse the problems with discovering utility-based event patterns in enterprise data given event heterogeneity. * Study data mining algorithms and embedding techniques to unify heterogeneous event formats. Then, develop usage-based similarity functions to measure the events. * Develop a data mining approach to discover efficient event patterns given multiple business scenarios.
Context heterogeneity	* Address context heterogeneity by formulating generalised triadic relationships among heterogeneous contexts in enterprise data. * Study optimised mining algorithms to effectively analyse the non-IID relationships between real-world heterogeneous business contexts. * Develop an approach to generate context-aware recommendations using real-world heterogeneous contextual features for business.
Domain heterogeneity	* Design a machine learning approach to analyse complex cross-domain correlations given domain heterogeneity using transfer learning techniques.

Element	Research objective
Domain heterogeneity	* Study suitable attention mechanisms to analyze heterogeneous domain data and generate more accurate predictions by capturing time-series, inner-domain, and cross-domain correlations.

1.3 Thesis organisation

The remainder of this thesis is divided into six parts. The next section contains a review of relevant literature followed by five chapters. The next four chapters each present a novel solution to a specific type of heterogeneity in turn: objects, events, context, and domains, in that order. A brief discussion concludes the thesis. A detailed roadmap of each chapter follows.

- *Chapter 2 Literature Survey*: provides a necessary background on previous research and studies on mining heterogeneous enterprise data.

- *Chapter 3 Object Heterogeneity*: presents a novel deep learning approach to solve the problems associated with analysing heterogeneous business objects at the data and structural levels. The approach involves a Cs-HNN to handle heterogeneous data that consists of both description and sequence structures. The network operates within a unified parallel architecture that aggregates different types of neural networks into the same epoch, which greatly improves learning efficiency in real-world heterogeneous and imbalanced datasets.

- *Chapter 4 Event Heterogeneity*: presents a practical processing approach to solving the problems associated with measuring the effectiveness of heterogeneous events given business demand. The algorithms derive valuable patterns from utility-based sequences in a fleet rental enterprise. Real-world industry datasets verify the effectiveness of the approach.

- *Chapter 5 Context Heterogeneity*: introduces a couple collaborative filtering model for context-aware recommendations given context heterogeneity. The model is able to analyse non-IID relationships among heterogeneous contextual features, such as items and locations.

- *Chapter 6 Domain Heterogeneity*: combines multi-task and transfer learning with a bespoke neural network to analyse the nonlinear relationships and transfer knowledge between heterogeneous business domains. A suitable cross-domain deep learning method with an attention mechanism is designed to construct effective representations of the complex correlations between financial domains by capturing time-series, inner-domain and cross-domain correlations.

- *Chapter 7 Conclusion*: summarises the contents and contributions of this thesis along with recommendations for future research.

Chapter 2

Literature Survey

The focus of this thesis is heterogeneous enterprise data mining. Heterogeneous data has the ability to preserve rich and diverse information and, when analysed, can provide businesses with deep insights into the dynamics and circumstances surrounding business activities [16] [18] [48] [157]. Traditional machine learning methods work well with homogeneous data, assuming the dataset has been carefully prepared in the first steps of the knowledge discovery process [89] [180]. However, enterprise data typically contains a diverse range of data types, formats, and sources. Using a single learning method to extract patterns from heterogeneous data tends not to produce the accurate and comprehensive learning results, as heterogeneous data usually occupy different positions in the data space [16] [24]. Directly mining heterogeneous data often leads to an in-depth understanding of the relationships and patterns in and between different types of objects, events, contexts, and domains, which yields fruitful results [51] [52] [68] [118] [157]. In the thesis, we investigated recent advancements in deep learning and other machine learning approaches to solve the problems in mining heterogeneous enterprise data.

Sun and Han proposed the concept of using heterogeneous information networks to model real-world data [157] [158] [159]. These initial studies structured objects and their interactions into different types and introduced some general principles and methodologies for systematically mining such networks. Since then, heterogeneous network analysis has rapidly become a hot topic in the data mining, database, and information retrieval fields. Yu introduced latent features based on different types of

meta-paths to represent the connectivity between users and items [189] [190]. Burke then extended the concept of meta-paths and presented an approach to recommendations that incorporates multiple relations in a weighted hybrid approach [14]. However, as the scale and complexity of enterprise data has grown, modelling information using a graph-based network has become a less and less straightforward process. Further, the time-complexity and computation costs have soared to prepare the huge graph-based structure in data preprocessing phase [24] [48].

Departing from many existing information network models that view interconnected data as homogeneous or heterogeneous graphs, the series of approaches, frameworks and algorithms in this thesis leverage the heterogeneity of objects, events, contexts, domains, and their nonlinear complex relationships in a range of diverse applications for business. The following literature review provides an overview of the relevant studies in each of these areas.

2.1 Object heterogeneity

2.1.1 Heterogeneous business objects

In real-world enterprise data, object heterogeneity is composed of complex business objects that have different feature types at the data level and may stem from different data sources at the structural level [38] [62] [128] [166]. Data-level heterogeneity means that the data contains a mixture of data value types, e.g., text mixed the simple data types like integers, floats, characters and etc. Most traditional data analysis methods focus on a simple format of data. Accordingly, when faced with heterogeneous data in big data era, these methods will only process a subset of the available information due to the limits on the number and/or types of features that can be modelled. As a consequence, objects have incomplete descriptions, important information is lost, and both the quantity and quality of the data analysis degrades. To overcome these problems, some studies have used one-hot encoding to transform

a categorical variable into a numeric representation on the assumption that each observation has been generated according to a set of conditional Bernoulli distributions [111] [141]. Given most of the entries are zero, the embedding model typically down-weights the contribution of the zeros in the objective function. However, Mikolov used negative sampling [111], where a subset of the zero observations are chosen at random in embedding model. These approaches correspond to a biased estimate of the gradient in an exponential Bernoulli family embedding model [141]. In statistics, embedding heterogeneous objects is performed using correspondence analysis a variant of canonical correlation analysis for count data [53]. Each method relies on Euclidean distances and constrains the embeddings to normalised values based on the assumption that distance reflects probabilistic relationships between objects [52].

More recent studies have extensively investigated an approach called extreme learning machines (ELM), which is based on training single-hidden layer feedforward neural networks [64] [65]. ELMs are very interesting form of heterogeneous data integration because of their structural simplicity, their performance, and their speed. However, the theoretical basis of ELMs is widely debated, as many scholars consider them to be similar to, or special cases of, radial basis function networks, random vector functional link, least squares support vector machines (LS-SVM), or reduced SVM [63] [124] [177]. Although it has been found that ELMs do not perform well in applications characterised by noisy data, like image recognition, some most recent ELM algorithms appear to be overcoming this problem and are providing excellent generalisation and classification performance [63] [177].

Many studies have also addressed object heterogeneity problems at the structural level. These studies use a reference data model to reason about the heterogeneity of objects in a formal way. Levy provide a good definition of a reference data model [93]. This model is based on a relational model augmented with basic object-oriented

features and provides the minimal core language required for interoperability and data exchange between relational and object-oriented data sources [18].

Let $O = \{o_1, o_2, \cdots, o_N\}$ be a collection of N heterogeneous business objects. An object $o_i \in O$ is defined as a triple

$$o_i =< n(o_i), SP(o_i), DP(o_i) >, \tag{2.1}$$

where $n(o_i)$ is a unique identity for the object within its collection. $SP(o_i) = \{sp_1, sp_2, \cdots, sp_q\}$ is the set of structural properties for o_i, that is, the properties that describe o_i's dynamic features. $DP(o_i) = \{dp_1, dp_2, \cdots, dp_r\}$ is the set of descriptive properties for o_i, that is, the properties that describe o_i's static features. In the following learning process, we attach to this formal object model in the data embedding, which supports diverse information integration of object heterogeneities at both the data level and the structural level.

2.1.2 Hybrid learning approach

As previously mentioned, using any one particular algorithm alone does not yield proper results in heterogeneous enterprise data mining. Hence, several hybrid learning approaches have been developed by combining or merging several algorithms to improve results, including cascading supervised techniques, and combining supervised and unsupervised techniques.

Cascading supervised techniques: The most popular supervised algorithms include neural networks, Bayesian networks, and decision trees. Scholars have presented many and varied combinations of these algorithms. Chan used naive Bayes, C4.5, CART, and Ripper as base classifiers, combining them in a stacked fashion [20]. Phua combined backpropagation neural networks, naive Bayes, and C4.5 as base classifiers using data partitions derived from minority oversampling with replacements [132]. Farid used a combination of naive Bayes and decision tree algorithms to

improve the performance of a naive Bayesian classifier and the ID3 algorithm, along with a hybrid approach that merges decision trees with SVM [40]. Peddabachigari added a hybrid DT-SVM classifier with weights to the decision tree/SVM ensemble [129]. As demonstrated above, lots of combinations of supervised algorithms are possible, and have been presented.

Combined supervised and unsupervised techniques: More recently, scholars have begun to combined supervised algorithms with unsupervised techniques. Yasami and Mozaffari combined k-means and ID3 in a computer protocol to distinguish between normal and anomalous data traffic [187]. Agarwal and Mittal developed a hybrid approach that combines the entropy of network features and SVM, with results that show the hybrid technique outperforms each single approach on its own [2]. This is a common finding. The hybrid approaches tend to yield better results because combining different techniques can help to overcome the drawbacks associated with each individual technique, resulting in higher accuracy.

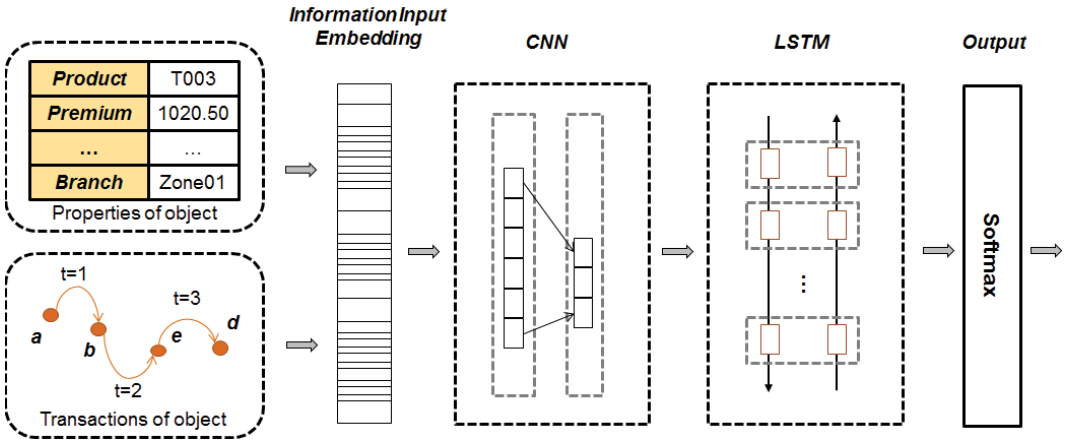

Figure 2.1 : : An example of an HNN with a two-step serial framework

As a relative newcomer to the world of heterogeneous data mining, deep learning approaches have made huge strides in period. Deep learning has the advantage of dealing with complex input-output relations and accumulated heterogeneous da-

ta analysis. HNNs combine the strengths of various neural networks and have, therefore, received great attention in some fields, especially computer vision and natural language processing [98] [174] [175] [186]. Niu and Suen introduced a hybrid classification system for objection recognition by integrating the synergies between convolutional neural networks (CNNs) and SVM with results that show improved classification accuracy [117]. Liu combined a CNN with a conditional random field. The CNN extracts the features, and the conditional random field handles classification [101]. Extensive experiments on several datasets show better segmentation performance with the hybrid structure than other methods. Tang's hybrid structure integrates a deep neural network (DNN) with an ELM to detect ships in images of space [162]. A DNN is used to process high-level feature representations and the classifications, while ELM provides effective feature pooling and does the decision making.

The main drawback with most current HNNs is that they usually solve classification problems through a two-step serial framework. For example, CNN-RNN frameworks learn image captioning through a CNN, then feed the results into an RNN as the inputs for next model training. This approach requires a significant amount of pre-processing time and data training that is seldom appropriate in real-world cases [174] [175]. Efficient ways to exploit hybrid deep learning architectures have not been well-studied, particularly with real-world business data or with parallel architectures, which may improve the strength of various neural networks.

2.1.3 Minority classification

Extant research on minority classification often considers the imbalance value from 10% to 20% (here an imbalance value of 10% means that the proportion of minority class to majority class is 10%). However, in reality, datasets can be far more imbalanced than this. For example, only about 2% of credit card

accounts are subject to fraud each year. Previous research on class imbalance problems has mainly centred on data sampling techniques and algorithm optimisation [7] [51] [70] [140] [181] [197]. A brief discussion on the different research efforts for minority classification follows.

Data sampling approaches: Data sampling manipulates the class representations in the original dataset by either oversampling the minority classes or undersampling the majority classes to ensure the resulting data distribution is balanced [22]. However, these techniques change the original distribution of the data and, consequently, introduce problems. Undersampling can result in the loss of useful information about the majority classes while oversampling artificially increases the size of the training set and often results in a computational burden. Further, when exact copies of the minority class are replicated randomly, oversampling is prone to overfitting [22] [113]. Chawla introduced a method, called SMOTE, to address this overfitting problem, where new instances are generated through a linear interpolation of closely-lying instances of the minority class [22]. However, these synthetically generated instances may lie inside the convex hull of the majority class instances, a phenomenon known as overgeneralisation. Hence, to combat overgeneralisation, several variants of the SMOTE algorithm have been developed [176]. For example, Borderline-SMOTE only oversamples the minority class instances that lie close to the class boundaries [55]. Safe-level SMOTE carefully generates synthetic instances in the so-called 'safe regions', i.e., where the majority and minority class regions do not overlap [13]. The local neighbourhood SMOTE considers neighbouring majority class instances when generating the synthetic minority class instances and reports better performance than previous variants of SMOTE [107]. Combining both undersampling and oversampling procedures to balance the training data has also been shown to perform well [7] [70] [140]. However, one drawback with these approaches is the increased computation costs required for both data pre-processing and training

the classification model. The computation costs are especially high with real-world heterogeneous enterprise data given its complexity and heterogeneous correlations.

Algorithm optimisation: Approaches that rely on algorithm optimisation directly modify the learning procedure to improve the sensitivity of the classifier toward minority classes. In Zhang and Wang's approach, the data is divided into smaller balanced subsets before intelligent sampling [199]; then CoSen SVM learning addresses problems with imbalance. Gao et al. introduced a neuro-fuzzy modelling procedure to perform leave-one-out cross-validation on imbalanced datasets [49]. Zhang et al. used a scaling kernel along with the standard SVM to improve the generalisability of the learned classifiers for skewed data sets [198]. Li emphasises the minority class samples by setting weights with Adaboost during the training of an ELM, while, Wang and Japkowicz create an ensemble of soft-margin SVMs via boosting, which performs well on both the majority and the minority classes [94] [168]. Each of these studies hints at the use of distinct costs for different training examples to improve the learning algorithm's performance. However, none addresses the type of imbalanced class learning associated with supervised classification, recognition, and segmentation problems that have recently emerged in computer vision [51] [181] [197]. Further, most of these approaches are limited to solving binary class problems [67] [168]; they do not perform joint feature and classifier learning; nor do they explore computer vision tasks, which have inherently imbalanced class distributions. In addition, each of these techniques relies on a modified loss function with a DNN learning algorithm, which usually leads to high computation costs and less flexibility limiting their application to real-world problems [29][138].

2.2 Event heterogeneity

2.2.1 Event heterogeneity and utility-based pattern mining

Two types of sequential data that describe events are commonly found in real-world enterprise data - time-series and sequences - and businesses often rely on both [56] [194]. A time-series is an ordered list of numbers, for example, amounts of money, stock prices, temperature readings, or electricity consumption readings. A sequence is an ordered list of nominal values, for example, a series of letters, the sentences of a text (sequences of words), the items purchased by customers in retail stores, or the web pages visited by users. Heterogeneous event mining involves data that comprises both types of sequential data.

Sequential pattern mining tasks are effectively enumeration problems. The aim is to enumerate all patterns of subsequences that have a support level of no less than the minimum support threshold set by the user. Numerous algorithms have been designed to discover sequential patterns. Some of the most popular are GSP, Spade, PrefixSpan , Spam , Lapin , CM-Spam and CM-Spade [5] [45] [130] [154] [184] [193]. To reduce the number of sequential patterns found and find more interesting patterns, researchers have also integrated constraints into sequential pattern mining [131]. Weighted sequential pattern mining is an extension of sequential pattern mining where weights (generally assumed to be in the [0,1] interval) are associated with each item to indicate their relative importance [21] [191]. The goal of weighted sequential pattern mining is to the find sequential patterns with the minimum weights.

High-utility sequential pattern mining (HUSPM) is an extension of weighted sequential pattern mining that considers both the item weights and the item quantities [3] [88] [188]. The goal of HUSPM is to find all the sequential patterns that have a 'utility' value of greater than or equal to the minimum utility threshold in the

database. The utility of a sequential pattern is the sum of the maximum utility generated by the pattern for each sequence it appears in [3] [88] [188]. HUSPM is quite challenging as, unlike the support measure traditionally used in sequential pattern mining, the utility measure is neither monotone nor antimonotone. Thus, it cannot be directly used to prune the search space. To address this issue, some HUSPM algorithms incorporate upper-bounds on the utility of sequential patterns to prune the search space, such as the monotone SWU measure [3]. A major challenge in HUSPM has been to develop tighter upper-bounds on the utility measure. These upper-bounds allow a larger portion of the search space to be pruned, which improves performance [3] [88] [188]. HUSPM is a very active research topic, and the extensions of the HUSPM problem have been studied. For example, hiding high-utility sequential patterns in databases to protect sensitive information or discovering high-utility sequential rules [134] [204].

Another interesting extension of the problem of sequential pattern mining is multi-dimensional sequential pattern mining [46] [133] [153]. These approaches consider an extended type of sequence database where each sequence can be annotated with symbolic values representing dimensions. For example, in association rules (or often called as 'Market Basket Analysis'), a dataset of customer purchase event sequences might be annotated with three dimensions: gender, education level, and income. Then, a multi-dimensional sequential pattern mining algorithm can be used to discover the sequential patterns that are common to various dimension values. For example, a pattern could be discovered with the dimension values (male; university; xxxx.xx) indicating that the pattern is common to male customers with a university degree but with any income. To mine multi-dimensional patterns, there are two main approaches: mining the dimensions using an itemset mining algorithm, then mining the patterns; or mining the sequential patterns, then the dimensions [46] [133] [153].

2.2.2 Attention neural networks

Strong and recent developments in deep learning technologies have firmly established RNNs, long short-term memory (LSTM) [60], and gated RNNs [28], in particular, as state-of-the-art approaches in sequence modelling and encoder-decoder architectures. Attention mechanisms have become an integral part of the compelling sequence models and transduction models for various tasks because dependencies can be modelled without regard to their distance in the input or output sequences [6] [82] [144] [145] [146]. Attention neural networks have also been proposed as a way of computing the alignment scores between elements from two different sources. Specifically, given the token embeddings of a source sequence $x = [x_1, x_2, ..., x_n]$ and the vector representation of a query q, attention computes the alignment score between x_i and q using a compatibility function $f(x_i; q)$, which measures the dependency between x_i and q, or the attention of q to x_i. A softmax function then transforms the scores $f(x_i; q); i = 1, ..., n$ into a probability distribution $p(z|x, q)$ by normalising all the n tokens of x. Here, z is an indicator for which token in x is important to q in the task; a large $p(z = i|x; q)$ means x_i contributes important information to q. The above process can be summarised by the following equations.

$$a = [f(x_i, q)]_{i=1}^{n}, \tag{2.2}$$

$$p(z|x, q) = softmax(a) \tag{2.3}$$

Specifically,

$$p(z = i|x, q) = \frac{\exp(f(x_i, q))}{\sum_{i=1}^{n} \exp(f(x_i, q))} \tag{2.4}$$

The output of this attention mechanism is a weighted sum of embeddings for all tokens in x, where the weights are given by $p(z|x, q)$. The process places a large weight on tokens important to q and can be written as the expectation of a token sampled according to its importance, i.e.,

$$s = \sum_{i=1}^{n} p(z = i|x, q)x_i = \mathbb{E}_{i \sim p(z|x,q)}(x_i), \tag{2.5}$$

where s can be used as the sentence encoding of x.

Additive attention (or multi-layer perceptron (MLP) attention) [143] [6] and multiplicative attention (or dot-product attention) [167] [156] are the two most commonly-used attention mechanisms. They share the same unified form of attention introduced above but are different in terms of the compatibility function $f(x_i; q)$. Additive attention is associated with

$$f(x_i, q) = \omega^T \sigma(W^{(1)} x_i + W^{(2)} q), \tag{2.6}$$

where $\sigma(\cdot)$ is an activation function, and $\omega \in \mathbb{R}^{d_e}$ is a weight vector, ω^T means the transpose of ω. Whereas, multiplicative attention uses the inner product/cosine similarity for $f(x_i; q)$, i.e.,

$$f(x_i, q) =< W^{(1)} x_i, W^{(2)} q > . \tag{2.7}$$

In practice, additive attention often produces higher-quality predictions than multiplicative attention, but multiplicative attention is faster and more memory-efficient due to its optimised matrix multiplication. Self-attention is a special case of the at-

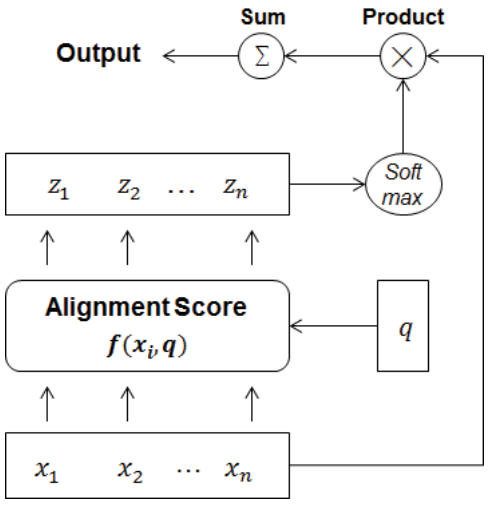

Figure 2.2 : Traditional attention mechanism

tention mechanisms introduced above, where q is replaced with a token embedding

x_j from the source input itself. Self-attention relates elements at different positions to a single sequence by computing the attention between each pair of tokens x_i and x_j. It is a very expressive and flexible approach for both long-range and local dependencies, which used to be modelled by RNN and CNN, respectively. Moreover, self-attention has a much smaller computational complexity and fewer parameters than RNN/CNN. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstract summarisation, textual entailment, and learning task-independent sentence representations [27] [99] [126] [127]. Recent studies have also witnessed its success in a variety of natural language processing tasks, such as reading comprehension [61] and machine translation [167].

2.3 Context heterogeneity

2.3.1 Context-aware algorithms

Context provides a background to the environments or situations that surround business objects, activities and behaviours. With context, businesses can respond to the customers more accurately and proactively with better understanding their circumstances and requirements, which makes contextual heterogeneity an important consideration in any heterogeneous data mining approach for business.

Context-aware algorithms can be built with three distinct techniques [1] [9]: contextual prefiltering, where a separate model is learned for every context type; contextual postfiltering, where adjustments are performed after a general context-unaware model is built; and contextual modelling, where context becomes an essential part of the training process. The first two techniques are prone to information loss about the interrelations within a context. However, contextual modelling extends the dimensionality of the problem to highlight the multi-relational aspects of context. Therefore, context-aware algorithms are likely to generate more accurate results for business requirements [80]. Contextual modelling can be formalised as

follows:

$$F_C : g(Object) \times h(Event) \times Context_1 \times \cdots \times Context_N \rightarrow RelevanceScore, \quad (2.8)$$

where $g(Object)$ means the contribution from objects and their correlations, $h(Event)$ represents the contribution from events and their correlations, and $Context_i$ denotes one of N heterogeneous contexts.

As context-aware insights are semantic abstractions from low-level contextual cues, human knowledge and their interpretations of the world must be integrated into contextual model representations. Most approaches focus on classifying basic human activities or scenarios without considering richer contextual descriptions [115]. However, some studies have attempted to acquire high-level contextual insights by incorporating context-aware algorithms. Clarkson proposed a wearable system capable of distinguishing coarse locations and user situations [30]. Each user's locations and situations are isolated and then recognised through a clustering process that groups audio and video recordings. McCowan takes Clarkson's approach a step further with a two-layered framework that models then recognises individual and group actions in meetings [109]. Brdiczka doubles the number of layers with a four-layered framework for situation learning [12]. Brdiczka's framework captures different aspects of a situation, namely the situations and various roles, with different levels of supervision. Depending on the number and type of observations to be recognised, each of these learning-based approaches is able to correctly recognise situations with an accuracy rate of over 85%.

The recognition of human behaviour is highly dependent on perception, context, and prior knowledge of the most recent event patterns. Research has shown that real-world decision making in business is contingent on context, such as where the customer shop located or whether the feedback came from social networking [1] [125]. Moreover, studies on recommendation systems have also shown that

extracting items of interest to users based on relevant contextual information using context-aware algorithms provides more accurate predictions in real-world cases [23] [149] [202] [203] [195] [74].

2.3.2 Non-independent and identical distribution

In practice, contextual information is usually linked to other entities through latent non-IID relationships that can indicate a customer's true preferences [15] [16]. However, when business behaviours only occur in specific circumstances, considering objects, events, and contexts independently may not capture these complex relationships well enough to produce accurate predictions. Most current algorithms do not thoroughly consider this problem, as most operate on the assumption of IID data. Further, most context-aware algorithms model contextual factors independently and are only designed to operate in fairly simple IID environments.

An increasing number of researchers have pointed out that the assumption of independence often leads to massive information loss [15] [170] [171]. Models designed for IID environments essentially ignore or simplify complex relationships, including co-occurrence, neighbourhoods, dependencies, linkages, correlations, and causality, among other poorly explored and unquantified relationships. By contrast, non-IID models are better able to capture the complexity of all these interactions, as shown in Fig. 2.3.

Wang presented a coupled nominal similarity measure to examine both the intra-coupling and inter-coupling of categorical features [170]. Liu incorporates the couplings between objects, features, and feature values into the classification of class-imbalanced [100]. Shi builds several non-IID representations of original features by various graph kernel functions and automatically learns these metric from the combined non-IID representations [148]. Yet, despite these efforts, analysing non-IID relationships remains a very challenging undertaking as both the explicit and

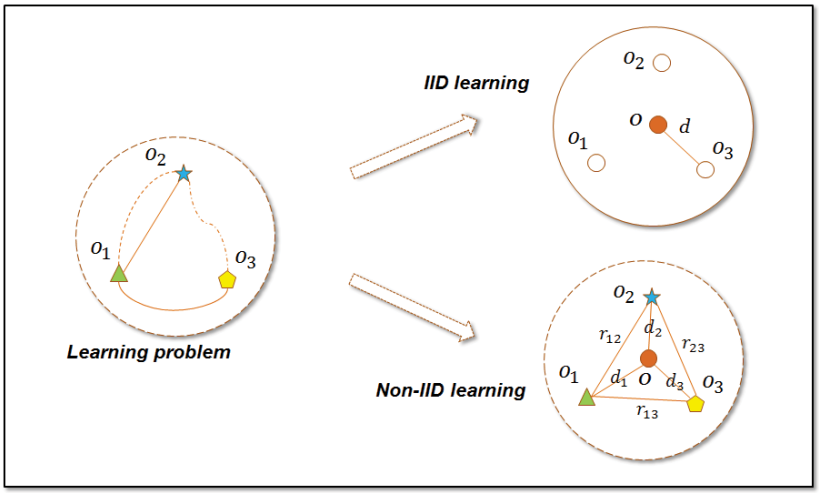

Figure 2.3 : IID data vs non-IID data in real-world data mining

implicit couplings between context heterogeneities need to be considered.

2.4 Domain heterogeneity

2.4.1 Cross-domain algorithms

Traditional data mining usually draws on data from a single domain. However, in business, the desired outcome from a data analysis task more commonly requires data from different sources and heterogeneous domains [75] [195]. Cross-domain learning algorithms can capture insights from each domain and can identify the relationships between features across different domains, which can be classified into four groups:

Multi-view-based methods simply treat different domains as different views of an object or an event. Features are fed into a suite of models to describe each object from a different perspective. The results are subsequently merged to mutually reinforce the findings from all models. Co-training is an example of this category.

Similarity-based methods leverage the underlying correlations between different objects in heterogeneous domains. CCF, also known as context-aware CF, is a

typical example of this method, where different domains are modelled by different matrices using common dimensions. Once decomposed, these multiple matrices produce better results than solely factorising a single matrix.

Probabilistic dependency methods model the probabilistic dependencies between different domains using graphs as representations. Bayesian network and Markov random field are representative of this method. These models denote features extracted from heterogeneous domains as graph nodes.

Transfer learning methods transfer knowledge extracted from one or more source domains to train a model or improve performance in a target domain. One fundamental assumption in traditional transfer learning is that the training and testing data must be sampled from identical distributions [123] [150] [151] [183]. However, this assumption is not always valid in real-world business cases. More often, the only available data for the source domain follows a different distribution than the limited data available in the target domain, so traditional learning algorithms are inapplicable. Hence, effective ways to transfer knowledge between heterogeneous domains has been a highly-explored, yet persistent, obstacle. Some of the strategies with distinctive intuitions include: sample selection bias correction [39] [66] [68], which uses a reweighting method to generate an approximately unbiased distribution for learning; a self-taught learning approach that finds new feature representations to improve learning performance in the target domain [102] [136]; and a shared latent space or common prior distributions to transfer the knowledge across domains [106] [137].

2.4.2 Multi-task and transfer learning

Traditional machine learning methods only work well when the training and the test data have been drawn from the same feature space and share a common distribution. Once the distribution changes, most statistical models must be rebuilt from

scratch using newly collected training data. In practice, such a process is usually either too expensive or collecting the required training data is impossible, which means the model cannot be rebuilt. Hence, multi-task learning and transfer learning have become desirable tools for dealing with domain heterogeneity, due to valuable business insights they can draw from heterogeneous domains [120] [121] [75]. Multi-task learning (MTL) is a subfield of machine learning where models solve multiple learning tasks simultaneously while exploiting the similarities and differences between tasks in a unified framework [17] [116]. In addition, deep learning has become more and more popular for the applications due to its ability to learn nonlinear features, including applications as basic models for learning tasks in MTL. Most MTL methods assume that different tasks share the first several hidden layers, but then use their own specific parameters to generate outputs [95] [103] [196] [200]. In contrast to these deep MTL methods, Misra et al. presented a cross-stitch network to learn task relations according to hidden feature representations – a similar approach to learning the relationships between tasks [112].

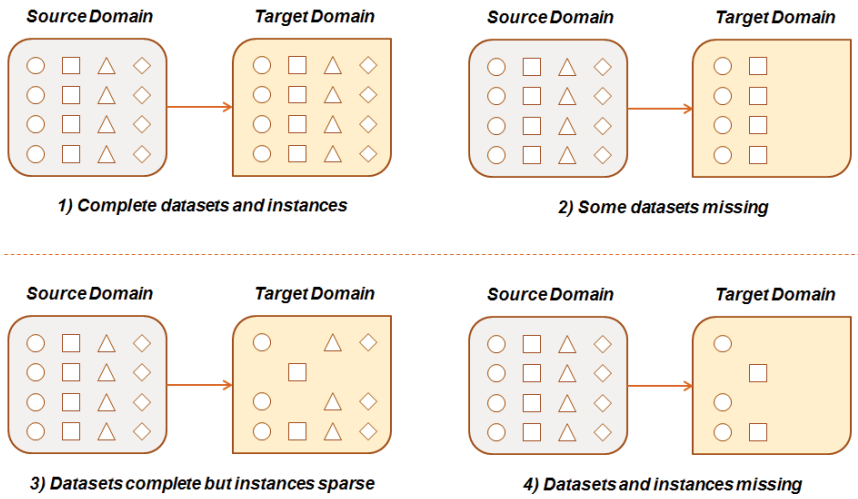

Figure 2.4 : Knowledge transfer between heterogeneous domains

In the context of MTL problems, transfer learning improves model performance

by using a related task that has already been learned to support the learning process for a new task [8] [91]. Several studies show that this approach can result in improved learning efficiency and prediction accuracy for task-specific models, when compared to training the models separately [8] [17] [91] [104].

Transfer learning approaches can be divided into four broad categories. Instance-based transfer learning assumes that certain parts of the data in the source domain can be reweighting for learning in the target domain [32] [33] [73] [135] [147] [169] [192]. Instance reweighting and importance sampling are two major techniques in these approaches. Feature-representation transfer learning encodes the knowledge to be transferred to the target domain into featured representations learned in the source domain for significantly improved target task performance [10] [11] [31] [81] [105] [139]. Parameter-based transfer learning assumes that the source tasks and the target tasks share some parameters or prior distributions of the model's hyper-parameters [69] [90] [142] [179]. The transferred knowledge is encoded into the shared parameters or priors and, through a discovery process, knowledge can be transferred across heterogeneous domains. Relational-knowledge transfer learning deals with transfer learning for relational domains [101]. The basic assumption behind these approaches is that some relationships between the data in the source and target domains are similar; therefore, these relationships are transferred [34] [86] [110].

Attention neural networks are a relatively recent addition to the deep learning approaches in the multi-task and transfer learning fields. Bahdanau et al. was the first to apply this technique to a machine language translator [6]. An encoder-decoder framework and an attention mechanism are used to select reference words in the original language before translating the required text into the foreign language. Subsequent uses of the attention mechanism include parsing, natural language question answering, and image question answering [59] [156] [160] [185]. Unlike these studies, the research in this thesis explores attention mechanisms for their power to

transfer inner-domain and cross-domain correlations to support multi-task predictions in heterogeneous financial domains.

In summary, the scope of all existing heterogeneous data mining methods has the limitations, narrowing complex data into a homogeneous, a balanced, or a simple-learning data space, with an inefficient process that requires a significant amount of pre-processing. Mining heterogeneous enterprise data faces the challenges from huge volume, complex heterogeneity, and diverse business requirements, which requires us to solve the problems of inefficient learning architectures, multi-task leaning across multiple domain, Non-IID data, imbalanced classifications, etc. In this thesis, the unique challenges of mining heterogeneous enterprise data are considered by investigating a series of algorithms and methods associated with each type of heterogeneity in Chapter 3, 4, 5, and 6.

Chapter 3

Object Heterogeneity

Objects represent the entities and instruments involved in business activity, for example, the organisations, departments, customers, or employees involved in business transactions, or the contracts, products, and services a business provides. Previous research on business objects defines object heterogeneity at two levels: the data level and the structural level [62] [128]. Data-level heterogeneity means that business objects are described using different types of values, e.g., integers, floats, characters. Structural-level heterogeneity is defined as a combination of different data formats, i.e., descriptions and sequences [76] [77]. Descriptions contain the static attributes of the object that do not change over time. Sequences trace the dynamic attributes of the transactions that might change over the timeline of the object's lifecycle.

Further, the most recent efforts to address object heterogeneity tend to rely on hybrid neural networks (HNNs) in the two-step serial architectures and focus on classifying the features that reflect most of the objects in the data, i.e., the majority classes. These techniques perform relatively well when the data is characterised by fairly simple linear relationships. But with the data complexity increasing, they require significantly more data pre-processing time and thus hardly meet the efficiency demands of real-world business competitions.

In the real world, the properties of enterprise data tend to be interconnected nonlinearly via explicit or implicit relationships, and increasing business classification tasks focus on minority classes, such as incidents of fraud or outlying customers. Unlike traditional approaches, deep learning techniques have the ability to extract

features and model nonlinear relationships without the need for prior human assumptions. However, the efficiency challenges associated with processing large-scale data and the technical challenges associated with identifying minority classifications in complex imbalanced heterogeneous data remain largely unaddressed in the literature.

Hence, the approach presented in this chapter provides an alternative solution to the serial architectures associated with traditional techniques, a suite of novel learning algorithms, and a novel approach to classifying minority features. The approach comprises a unified, end-to-end Cs-HNN that learns real-world heterogeneous data via a parallel network architecture. A specifically-designed cost-sensitive matrix automatically generates a robust model for learning minority classifications. And the parameters of both the cost-sensitive matrix and the HNN are alternately, yet jointly, optimised during the training process. The results of comparative experiments on two real-world business cases - insurance fraud detection and demographic classifications for mobile phone users - indicate that the proposed approach demonstrates superior performance over baseline procedures even at extreme levels of imbalance.

The remainder of this chapter was published as "Cost-sensitive hybrid neural networks for heterogeneous and imbalanced data" at the 2018 *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2018 by the attributed authors: X. Jiang, S. Pan, G. Long, J. Chang, J. Jiang, and C. Zhang. The full paper was accepted on 15 March 2018. Amendments have been made to improve the clarity of the language in the originally published version and to better suit the context of this thesis.

3.1 Introduction

The current nature of globalisation and competition stress the importance of leveraging valuable knowledge from the vast stores of accumulated data businesses

are now collecting. Given that DNNs deal with complex input-output relationships, scholarly attention has recently turned to applying these machine learning techniques to different business applications. However, while these efforts have resulted in major advancements to the analysis and classification of business objects, real-world classification tasks are not as straightforward when the data are heterogeneous and unbalanced. Traditionally, machine learning algorithms work well with homogeneous and balanced datasets that have been carefully prepared in the first steps of the knowledge discovery process. However, real-world datasets are usually derived from information systems that integrate collections of people, products, and processes. Such datasets are widely characterised as heterogeneous and imbalanced.

Real-world data objects roughly fall into two categories descriptions and sequences as shown in as in Fig. 3.1. Descriptions define the properties of an object that do not change over time. Sequences record transactions that occur over an object's lifecycle. Heterogeneous data contains objects in both categories.

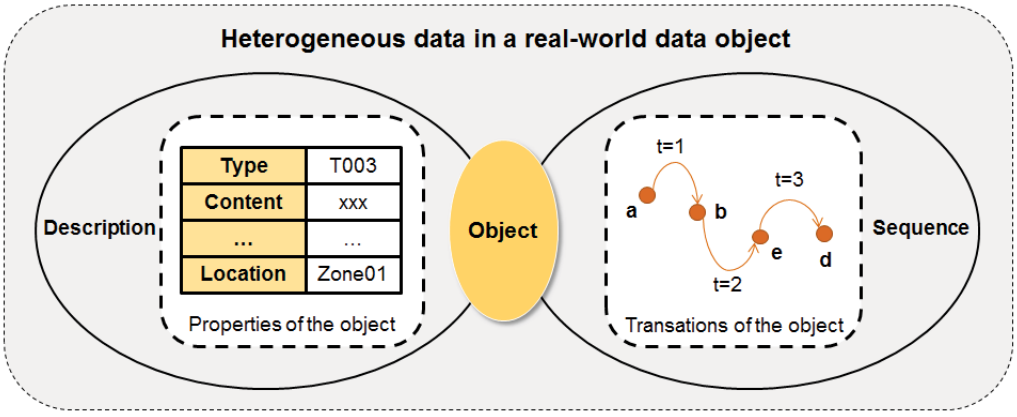

Figure 3.1 : An example of a real-world heterogeneous dataset

In addition, heterogeneous data usually occupy different positions in the data space, so using a single learning method or one projection to extract patterns from heterogeneous data does not produce a comprehensive learning result. Moreover, real-world business tasks often focus on minority classes, rather than the balanced

classifications associated with traditional techniques. As a result, using traditional techniques with data that have imbalanced class distributions usually produces outcomes that are extremely skewed. Take fraud detection as an example. Most often, there are significantly fewer fraudulent transactions in a dataset than normal transactions. Hence, fraudulent transactions become the minority class and are severely under-represented compared to the other classes.

Despite the recent advancements in deep learning with real-world heterogeneous and imbalanced datasets, applying DNNs to business applications still faces a lot of challenges. HNNs leverage the advantages of several different types of neural networks and have, therefore, received increasing interest in computer vision and natural language processing. However, while suitable for some learning areas, most current HNNs attempt to solve classification problems via a two-step serial framework and do not tend to consider which type of data is the most suitable for processing. Further, in practical data analysis, serial frameworks usually demand more data pre-processing and lack the learning efficiency required to meet real-world business demands. As yet, hybrid architectures have not been well-studied with real-world heterogeneous and imbalanced data.

Therefore, to add to the growing researches on heterogeneous data analysis, this chapter focuses on minority classifications and trend forecasting with imbalanced data using a deep network solution. Our approach involves learning descriptions and sequences in heterogeneous data and adjusting the cost-sensitive matrix of imbalanced data through an end-to-end Cs-HNN. To avoid the heavy demand for pre-processing and the inefficiency problems associated with training heterogeneous data in traditional serial HNN architectures, the Cs-HNN integrates an MLP and LSTM within a new parallel architecture that trains descriptions and sequences within the same epoch during training. To address the problem of imbalanced data, previous studies on DNNs tend to disturb the data distribution in the training set to

obtain better classifiers. Whereas, we have directly modified the learning procedure to incorporate class-dependent costs during the training process.

To this end, we introduce a Cs-HNN specifically designed for heterogeneous and imbalanced datasets. The key contributions of this chapter are:

- a novel HNN to handle heterogeneous business objects that consist of both description and sequence data. The network operates within a unified parallel architecture that aggregates two types of neural networks, MLP and LSTM, into the same epoch to greatly improve learning efficiency in complex real-world data analysis.
- an algorithm that jointly optimises the HNN's parameters and the cost-sensitive matrix to solve data imbalance problems in DNNs, along with an analysis of the effect of the modified loss functions by deriving relations for propagated gradients.
- an empirical study based on real-world datasets to validate the effectiveness of the presented approach.

3.2 Preliminaries

This statement addresses the problems with analysing real-world datasets in two respects: heterogeneous input representations and the cost-sensitive classification of imbalanced outputs.

Let $X = \{D, S\}$ represent the descriptions and the transactions of the heterogeneous data inputs, where $D = \{A_1, \cdots, A_n\}$ is a set of n-dimensional attributes that specify the object's characteristics, and $S = \{T_1, \cdots, T_m\}$ is the set of m transactions that records the object's activities over its lifecycle within the time-series sequence S. Each transaction T_m consists of $(t_1, \cdots, t_j)$, where t_j is the jth feature that describes the context-aware information of the mth occurring transaction.

Table 3.1 : Descriptive statistics for the heterogeneous and imbalanced datasets

Dataset	Heterogeneous inputs						Imbalanced output
	Descriptions			Sequences			
XID	A_1	**...**	A_n	T_1	**...**	T_m	**Y**
X_1	2	...	Z01	(a,6)	...	(e,1)	No
X_2	6	...	Z02	(b,7))	...	(f,2)	No
X_3	9	...	Z03	(c,8)	...	(a,5)	Yes
X_4	3	...	Z04	(a,2)	...	(d,8)	No

Our goal is to take a heterogeneous dataset $X = \{D, S\}$, with a suitable data structure, feed the data into a novel unified neural network through a weight and bias optimisation, and demonstrate that this approach produces superior performance with real-world heterogeneous and imbalanced datasets.

Within our method, an information table is constructed that maps each description and sequence into its corresponding attribute or transaction column. The table, as illustrated by the example in Table 3.1, consists of four samples $X = \{X_1, X_2, X_3, X_4\}$ and includes: n description columns and the corresponding attributes $\{A_1 - A_n\}$; m sequence columns and the corresponding transactions $\{T_1 - T_m\}$; and a labelled classification output, column Y.

The output column $Y = \{No, No, Yes, No\}$ shows the imbalanced classifications in the input samples $X = \{X_1, X_2, X_3, X_4\}$. Given a sample X_1, the values $\{T_1, \cdots, T_m\}$ are $\{(a, 6), \cdots, (e, 1)\}$, which represents a sequence of m transactions and two of its features: name and usage. For example, $(a, 6)$ means the transaction T1's name is a and six of a have been used.

Using the constructed information table, the HNN learns a combination of both the description and sequence representations within a unified training structure, and a cost-sensitive layer solves the imbalance problems.

3.3 A Cs-HNN for heterogeneous and imbalanced data

This section presents a novel Cs-HNN for analysing heterogeneous and imbalanced datasets in the real world. The Cs-HNN's architecture appears in Fig. 3.2. It consists of an HNN that analyses heterogeneous inputs, and a cost-sensitive loss function that solves imbalanced classifications. Each of the different components is described in the following sections.

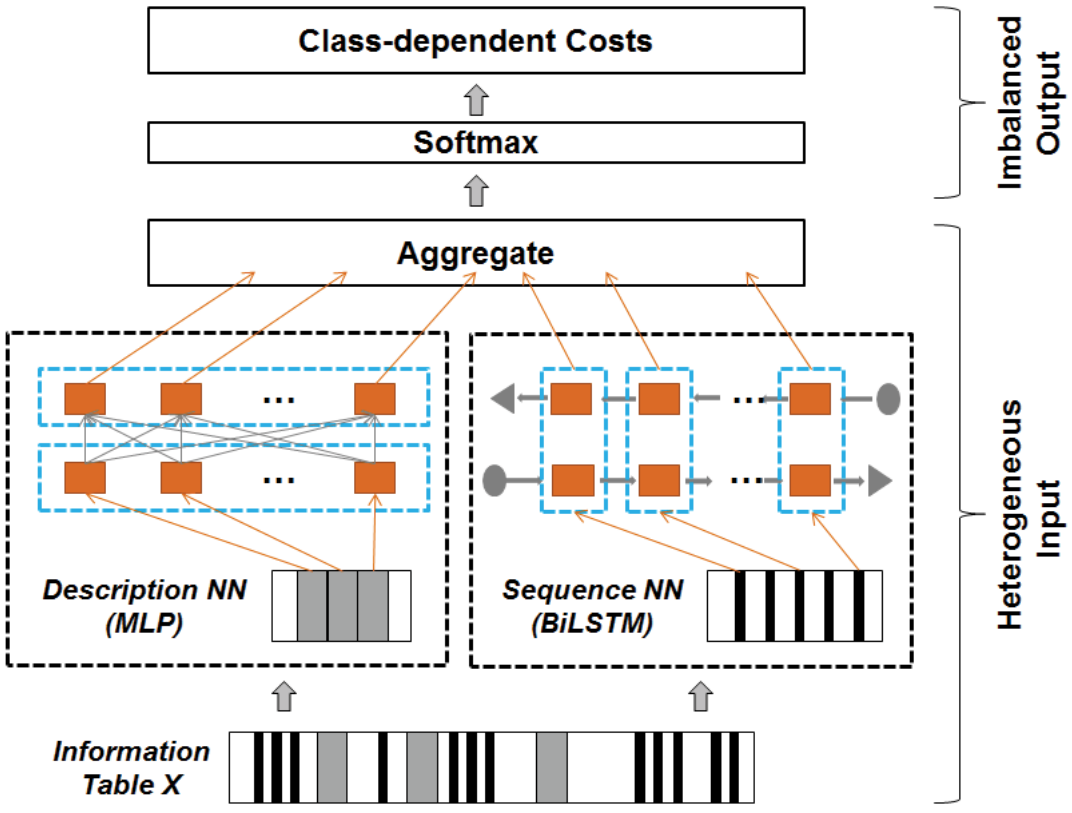

Figure 3.2 : The architecture of the Cs-HNN

3.3.1 An HNN for heterogeneous data

The heterogeneous input $X = \{D, S\}$ contains both property-based descriptions and transaction-based sequences. In most real-world cases, traditional monotonous methods produce low accuracy with such heterogeneous inputs. Hence, to solve this problem, we have developed a hybrid concept that integrates both the descriptive information and the sequence information within one training procedure. The proposed HNN contains the following components:

Heterogeneous data embedding

Given a heterogeneous data source with n attributes for description D and m transactions in sequence S, the data embedding task requires that the heterogeneous data inputs X are pre-processed into a uniform information table (see Table 3.1 for an example): $X \rightarrow \{\{A_1, \cdots, A_n\}, \{T_1, \cdots, T_m\}\}$, where $\{A_1, \cdots, A_n\}$ corresponds to the n attributes of description D and $\{T_1, \cdots, T_m\}$ corresponds to m transactions of sequence S. The categorical variables of the information table are then converted with one-hot encoding, and the numerical variables are normalised to provide better performance. The Cs-HNN then processes the heterogeneous data inputs X through two parallel networks - one for descriptions and another for sequences - with a training procedure that parses whole epochs.

Description neural network

The description neural network (DsNN) learns the description elements of the data inputs $D = \{A_1, \cdots, A_n\}$ in the constructed information table. This network is essentially an MLP with more than 3 layers $\{Input \rightarrow Hidden \rightarrow Output\}$ and several nonlinear activation functions, either *tanh* or logistic *sigmoid*. Given a 1-hidden-layer MLP, the description parameter $\alpha = \{W_1, W_2, b_1, b_2\}$. The inference

function $F(x)$ follows:

$$F(x) = \sigma(b_2 + W_2\sigma(b_1 + W_1 x)), \tag{3.1}$$

where the MLP inference function is formulated by the bias vectors b_1, b_2 and the weight matrices W_1, W_2; σ represents the sigmoid activation function.

Sequence neural network

The sequence neural network (SqNN) processes the sequences $S = \{T_1, \cdots, T_m\}$ in the constructed information table using a bi-directional LSTM (BiLSTM) for learning. The BiLSTM orders the sequential inputs in two ways, one from past to future and one from future to past. Compared to traditional unidirectional LSTMs, BiLSTM networks combine the hidden states in both directions to preserve the information for any point in time from both the past and the future. In real-world sequential cases, BiLSTM networks usually show good results as they are better at interpreting context. Through the BiLSTM, the SqNN efficiently processes the past, via forward states, and the future, via backward states, for a specific time frame as:

$$\begin{aligned} \overrightarrow{H_t} &= \overrightarrow{LSTMU}(T_t), t \in [1, m] \\ \overleftarrow{H_t} &= \overleftarrow{LSTMU}(T_t), t \in [1, m] \end{aligned} \tag{3.2}$$

where LSTMU represents a standard unit of LSTM. Given a sequence with m transactions, the hidden outputs of a given transaction input T_t, $t \in [1, M]$ are calculated from the following subfunctions:

- forget gate: $f_t = \sigma(W_f \cdot [H_{t-1}, T_t] + b_f)$
- input gate layer: $i_t = \sigma(W_i \cdot [H_{t-1}, T_t] + b_i)$
- new contribution: $\tilde{C}_t = \tanh(W_C \cdot [H_{t-1}, T_t] + b_C)$

- update cell state (memory): $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
- output gate layer: $o_t = \sigma(W_o \cdot [H_{t-1}, T_t] + b_o)$
- output to next layer: $H_t = o_t * \tanh(C_t)$

where σ represents the sigmoid activation function, and $[H_{t-1}, T_t]$ is a concatenation of H_{t-1} and T_t. The SqNN parameters $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$ are a concatenation of the forward hidden state $\overrightarrow{H_t}$ and the backward hidden state $\overleftarrow{H_t}$. $H_t = [\overrightarrow{H_t}; \overleftarrow{H_t}]$, summarises information about the sequence of transactions centred around T_t.

Neural network aggregation

To aggregate the DsNN and the SqNN into an HNN (see Fig. 3.2), the outputs of the DsNN and SqNN could be concatenated, multiplied, or averaged. In our implementation, the outputs of both the description and the sequence networks are concatenated within a softmax layer for classification:

$$Comb = \text{softmax}(W_c v + b_c), \tag{3.3}$$

where v is a high-level vector of the combined hidden outputs. $v = [H_{des}, H_{seq}]$ is the concatenation of the hidden outputs H_{des} from the description network and H_{seq} from the sequence network. A softmax function is then used on the heterogeneous dataset $X = \{D, S\}$ for data classification.

3.3.2 Imbalanced cost-sensitive classification

Class imbalance problems are addressed during training. A cost-sensitive classification function minimises the expected risk $\mathcal{R}(p|x)$, where x is an input sample, and p is the output classification of the classifier. The expected risk can be expressed

as

$$\mathcal{R}(p|x) = \sum_{q} \delta_{p,q} P(q|x) \tag{3.4}$$

where the cost matrix $\delta_{p,q}$ denotes the cost of misclassifying a sample to class q when it should belong to class p. $P(q|x)$ is the posterior probability over all possible classes given a sample x.

The cost-sensitive error function is expressed as the loss function over the training set:

$$E(\alpha, \beta, \delta) = \ell(y, \hat{y}_{(\alpha,\beta,\delta)}) \tag{3.5}$$

where $\hat{y}_{(\alpha,\beta,\delta)}$ is parameterised by the HNN, $\alpha = \{W_1, W_2, b_1, b_2\}$ are the weights and biases in the description network, $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$ are the weights and biases of the sequence network, and δ is the matrix of the class-sensitive costs. $y \in \{0, 1\}^{1\times N}$ is the desired output, and N denotes the total number of neurons in the output layer, which is equal to the number of classes. For example, in Table 3.1, $N = 2$ according to column Y. Therefore, the objective of the cost-sensitive classification optimisation is

$$(\alpha^*, \beta^*, \delta^*) = \arg\min_{\alpha,\beta,\delta} E(\alpha, \beta, \delta) \tag{3.6}$$

where the optimal parameters $(\alpha^*, \beta^*, \delta^*)$ are the objectives of the learning algorithm. These objectives are the minimum possible cost of E in Eq. (3.5). The loss function $\ell(\cdot)$ in Eq. (3.5) could be any suitable loss function. Here, we have used a cross-entropy loss function.

Cost-sensitive cross-entropy loss function: The cross-entropy loss function maximises the predictions for the desired output, formulated as

$$\ell(y, \hat{y}) = -\sum_{n} y \log \hat{y}_{(\alpha,\beta,\delta)}, \tag{3.7}$$

where y incorporates the class-dependent cost δ. The output is related to the output of the previous combination layer via the softmax function and calculates the

probability distribution of different possible outcomes.

3.3.3 Learning optimal parameters

The goal in optimising the learning parameters is to jointly learn the three types of parameters used in the Cs-HNN's functions. These parameters are the class-dependent loss function parameter δ, the description hypothesis parameter $\alpha = \{W_1, W_2, b_1, b_2\}$, and the sequence hypothesis parameter $\beta = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}$. Each is solved alternately by keeping two fixed and minimising cost with respect to the other. Stochastic gradient descent with a backpropagation error optimises α and β, while a gradient descent algorithm optimises the cost-sensitive matrix δ by calculating the direction of the steps to update the parameters.

The following cost function is used for the gradient computation to update δ, which can be understood as a squared L_2 norm of the difference between the vectors $\hbar$ and δ

$$f(\delta) = \frac{1}{2}\sum_{c}(\hbar_c - \delta_c)^2, c \in [1, N] \tag{3.8}$$

where N is the total number of distinct classes in the training set, and $\hbar$ denotes the histogram vector that encodes the distribution of classes in the training set.

The minimisation objective to find the optimal δ^* is expressed as:

$$\delta^* = \arg\min_{\delta} f(\delta), \tag{3.9}$$

The gradient descent algorithm that calculates the direction of updated steps and optimises the cost function is

$$\begin{aligned} \nabla f(\delta) =& \nabla((\hbar - \delta)(\hbar - \delta)^T) \\ =& (\hbar - \delta)J_\delta^T = -(\hbar - \delta)\mathbf{1}^T \end{aligned} \tag{3.10}$$

where J is the Jacobin matrix. To compute the dependence of $f(\delta)$ on the validation error, we take the update step only if it results in a decrease in the

Algorithm 1 Learning the optimisations for parameters (α, β, δ)

Input: Training set (X_T, Y_T), Validation set (X_V, Y_V), Max epochs (Max_{ep}), Learning rate $(\gamma_\alpha, \gamma_\beta, \gamma_\delta)$

Output: Learned parameters (α^*, β^*, δ^*)

Net $\leftarrow$ construct-Hybrid-Neural-Net()
{*Random initialization*}
α, $\beta \leftarrow$ initialize-Net(Net), $\delta \leftarrow 1$, val-err $\leftarrow 1$
{*Looping in number of epochs*}
for $e \in [1, Max_{ep}]$ **do**
  $grad_\delta \leftarrow 1$, compute-Grad $(X_T, Y_T, F(\delta))$
  $\delta^* \leftarrow$ update-CostParams $(\delta, \gamma_\delta, grad_\delta)$
  $\delta \leftarrow \delta^*$
  for $b \in [1, batchSize]$ **do**
    $out^b \leftarrow$ forwardPass$(X_T^b, Y_T^b, Net, \alpha, \beta)$
    {***Training Description of hybrid neural net***}
    $grad_\alpha^b \leftarrow$ backwardFPass$(out^b, X_T^b, Y_T^b, Net, \alpha, \beta, \delta)$
    $\alpha^* \leftarrow$ update-FNet-Params$(Net, \alpha, \beta, \gamma_\alpha, grad_\alpha^b)$
    {***Training Sequence of hybrid neural net***}
    $grad_\beta^b \leftarrow$ backwardSPass$(out^b, X_T^b, Y_T^b, Net, \alpha^*, \beta, \delta)$
    $\beta^* \leftarrow$ update-SNet-Params$(Net, \alpha^*, \beta, \gamma_\beta, grad_\beta^b)$
    α, $\beta \leftarrow \alpha^*$, β^*
  end for
  val-err$^* \leftarrow$ forwardPass$(X_V^b, Y_V^b, Net, \alpha, \beta)$
  if val-err* >val-err **then**
    $\gamma_\delta \leftarrow \gamma_\delta * 0.01$
    val-err $\leftarrow$ val-err*
  end if
end for
return (α^*, β^*, δ^*)

validation error. The complete parameter optimisation algorithm is provided below as Algorithm 1.

In the next section, we discuss the impact of the backpropagation algorithm's modified loss functions on the gradient computations.

3.3.4 Gradient computation with backpropagation

DsNN backpropagation

In the DsNN, the minimisation objective to find the optimal α^* is expressed as

$$\alpha^* = \arg\min_{\alpha} E(\alpha) \tag{3.11}$$

The loss function is represented as $\ell(y, \hat{y}) = \frac{1}{2}\sum_k (y_k - \hat{y_k})^2$ with the output as the kth neuron in the training set. Using gradient descent, the mathematical expression of the gradient at each neuron is

$$\frac{\partial \ell(y, \hat{y})}{\partial v_k} = -(y_k - \hat{y}_k)\frac{\partial \hat{y}_k}{\partial v_k} \tag{3.12}$$

where v_k is the weighted sum of the input connections. $\hat{y}_k$ in sigmoid activation function is defined as

$$\hat{y}_k = (1 + \exp(-v_k))^{-1} \tag{3.13}$$

Therefore, the partial derivation of $\hat{y}_k$ can be given as

$$\frac{\partial \hat{y}_k}{\partial v_k} = \frac{\exp(-v_k)}{(1 + \exp(-v_k))^2} = \hat{y}_k(1 - \hat{y}_k) \tag{3.14}$$

SqNN backpropagation

In the SqNN, the minimisation objective to find the optimal β^* is expressed as

$$\beta^* = \arg\min_{\beta} E(\beta) \tag{3.15}$$

The loss function is represented as $\ell(y, \hat{y}) = \frac{1}{2}\sum_k (y_k - \hat{y}_k)^2$ with the output as the kth neuron of the training set. Using gradient descent, the mathematical expression of the gradient at each neuron is

$$\frac{\partial \ell(y, \hat{y})}{\partial v_k} = -(y_k - \hat{y}_k)\frac{\partial \hat{y}_k}{\partial v_k}, \tag{3.16}$$

where v_k is the weighted sum of the concatenated inputs. The $\hat{y}_k$ in the tanh activation function is defined as

$$\hat{y}_k = \sigma(W_o \cdot [H_{k-1}, T_k] + b_o) * \tanh(C_k), \tag{3.17}$$

where σ represents the sigmoid activation function. C_k is defined as *Update Cell State* in previous section. The partial derivation of $\hat{y}_k$ can be given as

$$\frac{\partial \hat{y}_k}{\partial v_k} = \tanh(\mathcal{W}_i T_k + o_k \cdot (\mathcal{U}_i H_{k-1}) + b_i) \tag{3.18}$$

where o_k, and the parameters b_i and W_i are explained in Section IV.3.

Cost-sensitive cross-entropy backpropagation

The cost-sensitive softmax log loss function begins by calculating the partial derivative of the softmax neuron with respect to its input:

$$\frac{\partial \hat{y}_c}{\partial Comb_c} = \hat{y}_c(1 - \hat{y}_c), \tag{3.19}$$

where the output of the previous layer $Comb$ is defined in Section IV.4, $c \in [1, N]$ and N is the total number of distinct classes in the training set. The loss function is differentiated as follows:

$$\begin{aligned} \frac{\partial \ell(y, \hat{y})}{\partial Comb_c} &= -\sum_c y_c \frac{1}{y_c}\frac{\partial y_c}{\partial Comb_c} \\ &= -y_c + \sum_c y_c \hat{y}_c = -y_c + \hat{y}_c, \end{aligned} \tag{3.20}$$

Since y_c is defined as a probability distribution over all output classes, $\sum_c y_c = 1$.

$$\frac{\partial \ell(y, \hat{y})}{\partial Comb_c} = -y_c + \hat{y}_c \tag{3.21}$$

The result is the same when the cross-entropy loss function does not contain any cost-sensitive parameters. Thus, the costs affect the softmax output, but the gradient formulas remain the same.

3.4 Experiments and evaluation

The approach outlined above was evaluated on six real-world heterogeneous and imbalanced datasets. Three datasets were extracted from the Insurance-FD dataset, and three were extracted from the Mobile-CD dataset. Insurance-FD contains data on fraud detection, while Mobile-CD classifies customer demographic information. Details on each dataset and the experimental settings follow.

3.4.1 Datasets and experimental settings

Fraud detection

Insurance-FD is a real-world dataset derived from the information systems of a large Chinese life insurance company. It contains over 138,200 samples, and 411 dimensions narrowed from an original 2457 dimensions. The data objects are insurance policies, and the samples describe each policy's properties in terms of 341 attributes (descriptions) including product lists, agents, sales channels, the insured parties, and the policies' beneficiaries. The policies are also classified into seven different types of transactions. These transactions contain the sequence information, such as cooling-off periods, insurance premiums or deductions, loans, claims, surrenders, account changes, and changes to the listed parties or other information. For the purposes of this study, these sequence features are considered to belong to positive (fraudulent) and negative (non-fraudulent) classes.

Experimental setting: To evaluate the Cs-HNN on datasets of various scales, three datasets of different sizes were extracted from Insurance-FD and converted into information tables, as described in Section 3.2. To represent different degrees of imbalance in the data distribution, we reduced the representations of one of the two classes in each extracted dataset to 20%, 10%, and 5%. For instance, an imbalance value of 5% means 5% of the samples fall into the minority class compared to the majority class. The neural networks settings for Insurance-FD for each compared method are shown in Table 3.2.

Table 3.2 : Network settings for the Insurance-FD dataset

Compared Method	**Learning Rate**	**Hidden Layer Neutron Setting**
DNN: MLP	γ=0.01	$\{input \rightarrow 512 \rightarrow 256 \rightarrow output\}$
RNN: BiLSTM	γ=0.01	$\{input \rightarrow 256 \rightarrow 256 \rightarrow output\}$
HNN: Hybrid NN	γ_α=0.01 γ_β=0.01	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 256 \rightarrow 256 \rightarrow h\}$
Cs-HNN: Cost-sensitive Hybrid NN	γ_α=0.01 γ_β=0.01 γ_δ=0.0001	$input : n \sim description$ $\{n \rightarrow 512 \rightarrow 256 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 256 \rightarrow 256 \rightarrow h\}$

Demographic classification

Mobile-CD is a real-world dataset from Kaggle-TalkingData. It contains mobile user demographics and behavioural data about more than 70% of the 500 million

Table 3.3 : Network Settings for the MOBILE-CD Data Set

Compared Method	**Learning Rate**	**Hidden Layer Neutron Setting**
DNN: MLP	γ=0.01	$\{input \rightarrow 64 \rightarrow 32 \rightarrow output\}$
RNN: BiLSTM	γ=0.01	$\{input \rightarrow 1024 \rightarrow 1024 \rightarrow output\}$
HNN: Hybrid NN	γ_α=0.01 γ_β=0.01	$input : n \sim description$ $\{n \rightarrow 64 \rightarrow 32 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 1024 \rightarrow 1024 \rightarrow h\}$
Cs-HNN: Cost-sensitive Hybrid NN	γ_α=0.01 γ_β=0.01 γ_δ=0.0001	$input : n \sim description$ $\{n \rightarrow 64 \rightarrow 32 \rightarrow h\}$ $input : m \sim sequence$ $\{m \rightarrow 1024 \rightarrow 1024 \rightarrow h\}$

mobile devices that are active daily to help the companies better understand and interact with their audience. The dataset contains more than 6500 descriptions of mobile devices and app activities that are used to predict the demographic characteristics of mobile users. The descriptions and transactions in Mobile-CD have 996 dimensions, narrowed from an original 6513 dimensions. We extracted three datasets of different sizes covering two demographic groups from this dataset.

Experimental settings: To represent different levels of imbalance in the data distribution, we reduced the representative samples of one of the two classes in each dataset to 20%, 10%, and 5%. Again, a 5% imbalance value means the dataset contains 5% minority samples with the remainder as majority samples. The three datasets were then converted into information tables. Table 3.3 lists the neural

networks settings for each compared method for Mobile-CD.

Comparison baselines

The following DNN, RNN, and HNN algorithms were selected as appropriate comparisons to evaluate Cs-HNN's performance:

- DNN: an MLP network trained on the description data;
- RNN: a BiLSTM network trained on the sequence data; and
- HNN: Our proposed parallel HNN without the cost-sensitive matrix, trained on both the description and sequence data.

All neural networks were trained on one dataset during the training procedure. 80% of the samples in each dataset were used as the training set; the remaining 20% were used as the test set. Each network was evaluated on the three versions of each dataset, reflecting different levels of imbalance and used to make predictions in the subsequent testing procedure.

Evaluation metrics

Two commonly-used classification metrics were used to evaluate prediction performance: F-measure and area under curve (AUC) [41].

3.4.2 Experimental results

In general, accurately classifying the minority class rather than the majority class is more important when the data is imbalanced. Without loss of generality, we mainly focused on classification performance in the minority class the positive class in these experiments.

The results of the experiments on the three Insurance-FD datasets and the three Mobile-CD datasets are shown in Tables 3.4 and 3.5, respectively.

Table 3.4 : Evaluation with the Insurance-FD dataset

Datasets	**F-measure**			
Experimental settings	***DNN***	***RNN***	***HNN***	***Cs-HNN***
Insur + Imb. level 20%	0.67778	0.68493	0.72922	0.74487
Insur + Imb. level 10%	0.47099	0.57576	0.58084	0.67085
Insur + Imb. level 5 %	0.46258	0.46953	0.56410	0.60645
Datasets	**AUC**			
Experimental settings	***DNN***	***RNN***	***HNN***	***Cs-HNN***
Insur + Imb. level 20%	0.77104	0.78183	0.84283	0.82634
Insur + Imb. level 10%	0.67185	0.76185	0.83767	0.84204
Insur + Imb. level 5 %	0.67377	0.64297	0.74412	0.75761

Imb. denotes imbalance

With each dataset, we observed that the more imbalanced the data, the worse the classification performance, as illustrated by the general downward trend in terms of both F-measure and AUC as the degree of imbalance increased. However, and more importantly, both the Cs-HNN and the HNN network based on Cs-HNN but without the cost-sensitive matrix did show better classification accuracy on most of the datasets than the baseline DNN and RNN networks at the same levels of imbalance. Additionally, the Cs-HNN showed obvious improvements in terms of F-measure and AUC in datasets with an extreme imbalance of 5%. This is a promising result for effectively classifying imbalanced datasets.

We also tested Cs-HNN in terms of the loss value trend when training and testing the three Insurance-FD datasets at an imbalance level of 5%, as shown in Fig. 3.3. Even with highly imbalanced heterogeneous datasets, Cs-HNN significantly

Table 3.5 : Evaluation with the Mobile-CD dataset

Datasets	**F-measure**			
Experimental settings	***DNN***	***RNN***	***HNN***	***Cs-HNN***
Mobil + Imb. level 20%	0.63917	0.21052	0.95302	0.94891
Mobil + Imb. level 10%	0.30769	0.12500	0.72000	0.86957
Mobil + Imb. level 5 %	0.24590	0.25882	0.42307	0.50909
Datasets	**AUC**			
Experimental settings	***DNN***	***RNN***	***HNN***	***Cs-HNN***
Mobil + Imb. level 20%	0.73485	0.50374	0.97251	0.96569
Mobil + Imb. level 10%	0.59090	0.48649	0.81067	0.94412
Mobil + Imb. level 5 %	0.57537	0.59973	0.67332	0.71560

Imb. denotes imbalance

decreased the loss value trend during both the training and testing procedures as the number of epochs increased. This result empirically verifies the theoretical analysis in the previous sections, demonstrating that Cs-HNN can deliver effective classification performance on real-world heterogeneous and imbalanced datasets.

In summary, our investigation into the problems associated with object heterogeneity when mining heterogeneous real-world enterprise data explored an alternative solution to the serial architectures traditionally used with most existing HNNs. We argue that the serial architectures require more data pre-processing and lack the learning efficiency demanded by real-world business needs. Additionally, real-world business tasks often focus on minority classes, not the majority classifications existing techniques are designed to provide. The solution presented in this chapter relies on a novel, unified, end-to-end architecture and learning algorithms that are

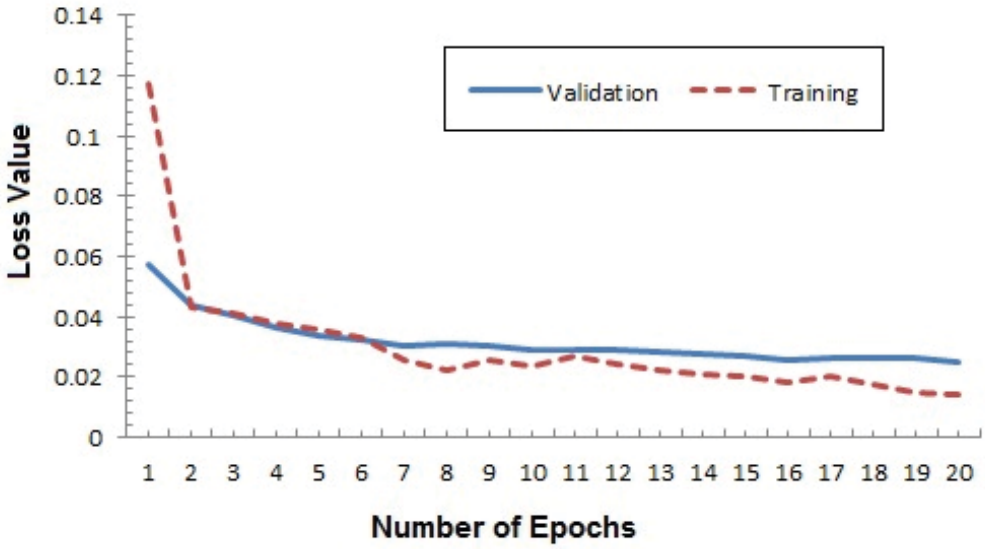

Figure 3.3 : Cs-HNN loss values with an increase in the number of epochs using the Insurance-FD dataset and an imbalance level of 5%, Cs-HNN effectively decreases the loss value.

designed to operate within a Cs-HNN. The Cs-HNN feeds a heterogeneous information table into a network architecture that processes descriptions and sequences in parallel. Alternating optimisation algorithms then efficiently learn the resulting imbalanced cost-sensitive matrices along with the Cs-HNN's parameters at the epoch level. The results of the experiments with six real-world heterogeneous and imbalanced datasets demonstrate that this approach is able to generate the types of object classifications commonly required by real-world businesses with superior performance over other baseline neural networks.

Chapter 4

Event Heterogeneity

In this chapter, we explore and analyse event heterogeneity through effective and efficient pattern mining models that consider event heterogeneity. Due to the natural complexity of heterogeneous enterprise data, traditional pattern mining approaches tend to be inefficient and inaccurate. Most are based on the assumption of homogeneous data, whereas heterogeneous event-related features carry critical information that influence both the accuracy and relevance of the insights derived from data analysis tasks.

As one of the most well-known methods of customer analysis, general sequential pattern mining methods mostly focus on major events to provide deep insights into the activities and behaviours that characterise a business. However, in the real-world business events are often linked to many other factors [78] [178]. Hence, this chapter presents a practical and efficient sequential pattern mining approach to discover valuable event patterns by mining events with heterogeneous features using two different measuring algorithms, each providing specific insights into important aspects of business activity.

The work in this chapter was supported by the Australian Research Council Linkage grant (LP120100566). This investigation of event heterogeneity focuses on a real-world case study in the fleet rental industry. In fleet tracking, vehicle location and usage time are widely considered to be important features for increasing operational efficiency, improving the quality of customer service, and predicting future customer requirements. The solutions presented in this chapter, therefore,

focus on identifying frequent location and usage patterns in fleet tracking data. Experimental results on real-world datasets verify the effectiveness of this approach.

The remainder of this chapter was published as Discovering sequential rental patterns by fleet tracking at the International Conference on Data Science. Springer, 2015 pp. 4249 by the attributed authors: X. Jiang, X. Peng, and G. Long. Amendments have been made to improve the clarity of the language in the originally published version and to better suit the context of this thesis.

4.1 Introduction

In fleet rentals, the quality of a company's products and services depends on how successful they are at getting vehicles to the place a customer requires and at the time the customer needs [79] [152] [164]. Fleet tracking information, such as the locations of all the vehicles and the operating hours of the business, has been widely recognised as an important part of improving work performance, predicting customer preferences, and improving a company's competitive advantage [4] [152] [164].

As one of the most well-known methods of customer behaviour analysis, sequential pattern mining reveals frequent subsequences in a given database and represents them as patterns [114] [133] [193]. In fleet rentals, these patterns, when combined with fleet tracking information, are an effective way to identify high-utility vehicles or highlight highly-profitable services.

Fig. 4.1(a) illustrates an example of general sequential pattern mining. Here, the sequence comprises vehicles rented by different customers. The identified rental pattern in the rectangle (red, grey, yellow) has the highest frequency but has ignored the actual vehicles used at different customer sites. In reality, the vehicle rented by one customer may service two different work sites. Hence, analysing this pattern as a single sequence is likely to decrease the accuracy of the results. Fig. 4.1(b)

demonstrates an example of location-based sequential pattern mining. The sequence is divided into two different locations A and B. Here, two different rental patterns replace the single pattern identified by general pattern mining techniques with more accuracy a (red, grey) pattern is discovered for location A and a (yellow, green) pattern at location B. Fig. 4.1(c) depicts usage-based sequential pattern mining. This type of analysis mines patterns associated with vehicle use, according to time in this case, e.g. daily (yellow, red, green). Detecting high-utility vehicles by the amount of time they have been used can identify patterns with even more accuracy and far deeper insights to support prediction about future rental behaviour.

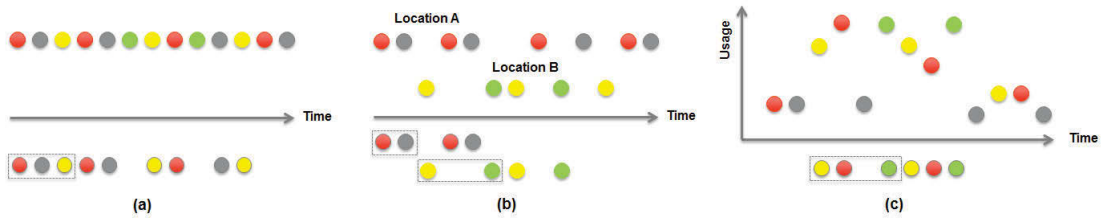

Figure 4.1 : Three sequential pattern mining approaches (a) general (b) location-based (c) usage-based

With the aim of improving the accuracy, efficiency, and the potential insights provided by sequential pattern mining for the fleet rental industry, this chapter presents a novel sequential pattern mining framework and two corresponding modelling algorithms. The algorithms identify frequent itemsets in fleet tracking information according to location and use. Further, the performance of the approach is verified through experiments on real-world fleet tracking datasets. The specific contributions of this chapter are:

- an analysis of the problems associated with discovering sequential rental patterns in fleet tracking data;
- a sequential pattern mining framework to efficiently discover rental patterns;

- two algorithms that reveal location-based and usage-based sequences, respectively;
- comparative experiments with real-world fleet tracking data that verify the effectiveness of the approach.

4.2 Preliminaries

4.2.1 Basic concepts and definitions

Several sequential pattern mining algorithms have been proposed to support the discovery of frequent patterns in customer purchasing behaviour, such as Spade, Prefixspan, and SPAM [5] [57] [193]. These algorithms typically result in the identified patterns, but most are either incomprehensible or irrelevant to businesses. While those that are relevant to a business's needs, but have a lower frequency than the given support threshold, are ignored [50] [114] [131] [165] [201]. To overcome these problems, many studies have explored ways to improve sequential pattern discovery. For example, Kumar used multiple minimum supports to enhance performance [84] [85]. Liao proposed a depth-first spelling algorithm for biological sequences analysis [97]. And Lan applied a utility measure to discover patterns with high utility [88]. Each of these endeavours has improved the great potential of sequential pattern mining in various applications.

However, in terms of the fleet rental industry, the problem definitions and the way information is analysed and used in the above studies have some major shortcomings. The most significant of these is a lack of understanding about what information the customer actually needs to provide quality products and services, such as high-quality fleet tracking patterns from real-world data [4] [164] [165].

Sequential pattern mining. Given a database of sequences gathered by an IoT-enabled device and a minimum support threshold, the main objective of a se-

quential rental pattern mining task is to find a complete set of sequential rental patterns.

Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of all items, with an itemset representing a subset of these items. A sequence is an ordered list of itemsets. A sequence s is denoted by $< s_1 s_2 \cdots s_l >$, where s_j is an itemset, i.e., $s_j \subseteq I$ for $1 \leq j \leq l$. s_j also represents an element of the sequence, denoted as $(x_1 x_2 \cdots x_m)$, where x_k is an item, i.e., $x_k \subseteq I$ for $1 \leq k \leq m$. The number of instances of items in a sequence is referred to as the length of the sequence. A sequence with length l is denoted as l-sequence.

A sequence database S is a set of tuples $< sid, s >$, where sid is the sequence-id and s is the sequence. A tuple $< sid, s >$ is said to contain a sequence α, if α is a subsequence of s, i.e., $\alpha \sqsubseteq s$. Support for a sequence α in a sequence database S is measured by the number of tuples in the database containing α, i.e., $support_s(\alpha) =\mid \{< sid, s > \mid (< sid, s > \in S) \bigwedge (\alpha \sqsubseteq s)\} \mid$. Given a positive integer ξ as the minimum support threshold, a sequence α is identified as a sequential pattern in database S if the sequence contains at least ξ tuples, i.e., $support_s(\alpha) \geq \xi$. A sequential pattern of length l is denoted as l-pattern.

Fleet tracking. With the integration of devices, sensors, information, software instructions, and communications technologies, fleet tracking data contains the connections between machines and business transactions [4]. Further, in the fleet rental industry, fleet tracking has attracted considerable interest as a way to evaluate how well a company does business. It can reflect how well a company has optimised its services or how much value it is bringing to its customers. By knowing the location and use of every vehicle in a fleet, a company can manage their vehicles in a more efficient and effective manner. The standard features in fleet tracking data include: vehicle location with time and date; the total operating hours of the vehicle, again with time and date; and the amount of fuel used by the vehicle during the 24 hours

prior to a specific time and date [164] [4].

4.2.2 Technical framework

The proposed framework combines sequential pattern mining techniques with fleet tracking information. A general framework, comprising four steps, is shown on the left of Fig. 4.2. These steps are data acquisition, item detection, pattern mining, and prediction. Once complete, these steps can be re-executed using the resulting evaluation and updated model parameters to provide higher performance for future predictions. The components on the right side of the graph provide more specific details on how to undertake pattern mining with fleet tracking data.

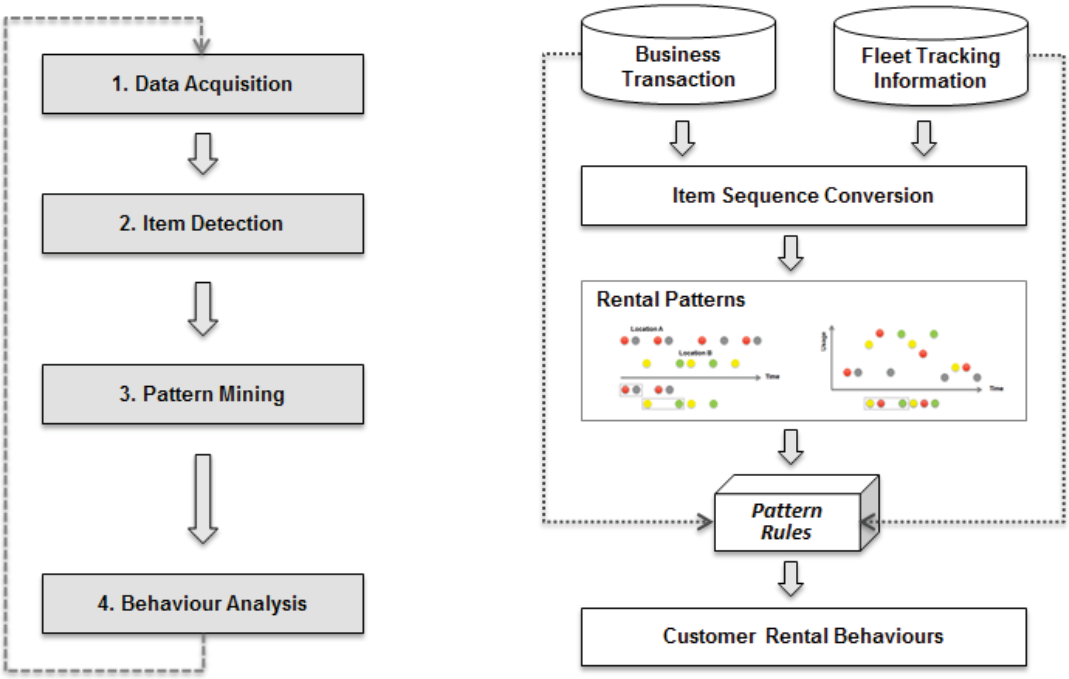

Figure 4.2 : A rental pattern mining framework with fleet tracking data

The transaction dataset in the above framework above is a set of fleet tracking records derived from business transactions, and the item sequence conversion is designed to operate these events assuming relevant corresponding tracking information. Each event (itemset) comprises a set of items within the fleet tracking data.

The pattern algorithms identify the subsets of items within each itemset that meet a given threshold according to a set of parameters. The discovered rental patterns are stored as prediction rules in a knowledge base, which can be used to predict future rental behaviours using current business fleet tracking data as inputs.

4.3 Rental event pattern mining

Event pattern mining discovers frequent subsequences in a sequence database and represents them as patterns. This technique has proven to be a useful tool for analysing order-based businesses and customer behaviours. An assumption that transactional data reflecting past customer behaviour effectively reflects future customers behaviour lies at the core of this premise. Hence, to avoid capturing the patterns with low utility typically derived from traditional sequential mining methods, our approach relies on both transactional data and fleet tracking information to focus on high-performance events and identify the most valuable customer behaviours in real fleet rental circumstances.

4.3.1 Data acquisition and item detection

The sequence data is composed from two sources of information: business transactions and fleet tracking data. The data acquisition process samples then convert real-world business transactions and vehicle information into a series of database tables that can be manipulated during the item detection phase. Each step in the data acquisition process is managed by a series of extract, transform, and load programs developed using various general-purpose programming languages.

The importance of location. Traditional sequential item detection techniques focus on the rental behaviour of a customer. However, in practice, a customer's fleet requirements depend on the characteristics of the project the vehicle is required for, and these characteristics tend to be location-specific. Therefore, determining where

a vehicle is used will often capture specific features that improve the accuracy and potential insights of predictions.

The importance of use. Vehicle use sits at the very core of fleet rental companies and is perhaps the most important feature for identifying high-utility vehicles. As maintaining high utilisation is the key to business stability and profitability, analysing customer rentals for high-utility vehicles is a crucial requirement for most fleet rental businesses.

4.3.2 Event pattern mining and behaviour analysis

The problem objective As mentioned above, the overall aim of this approach is to discover high-utility sequential rental patterns. However, this objective needs to be defined in more detail, and suitable parameters need to be defined to fulfil the algorithm. In the case of fleet rentals, these objectives are: 1)Identify the vehicles customers prefer to hire with high frequency in sequential order. 2)Predict the vehicles customers are likely to be interested in hiring in the near future.

Pattern mining algorithm The event pattern mining algorithm is presented in the Algorithm 2. Given a sequence $s =< s_1 s_2 \cdot \cdot \cdot \cdot_k >$, a k-sequence represents a sequence with k items. L_k is a set of frequent k-sequences, while C_k is the set of candidate k-sequences. The goal is to generate a candidate set of all frequent k-sequences, given the set of all frequent $(k-1)$-sequences.

Behaviour analysis In this setting, learning is performed off-line. The outcome is a set of pattern rules that provide information about the probability of specific rental behaviours occurring when certain preconditions are satisfied. Business transaction data and fleet tracking information form the inputs for the algorithm. The pattern rules and these two data sources are used to determine the likelihood of future rental behaviours.

Algorithm 2 The event pattern mining algorithm

Generate the candidate sequences in C_1

Save the frequent sequences in L_1

Iteratively find the sequences with kth pass:

Generate the candidate sequences in C_k from the frequent sequences in L_{k-1}.

{*Join Phase:*}

Join L_{k-1} with L_{k-1}

Join if (s_1 first item) is the same as (s_2 last item), s_1 join with s_2.

{*Prune Phase:*}

Delete candidate sequences C_k that have a contiguous $(k-1)$ subsequence whose support count is less than the minimum support.

Terminated until: No more frequent sequences L_k are found. No candidate sequences C_k are generated.

4.4 Testing and results

Extensive experiments were conducted to verify the ability of this framework to produce the desired results and to evaluate four different item detection strategies in terms of computational cost, memory usage, number of patterns, and length of patterns.

The data was provided by a real-world fleet rental company and contains 180,613 customer transactions with related fleet tracking data between January and December 2014. The four different detection strategies, DS1 to DS4, are described as follows.

- $DS1$ - solely mined customer rental transactions;
- $DS2$ - mined customer rental transactions and fleet tracking data with location information;

- $DS3$ - mined customer rental transactions and fleet tracking data with usage information; and
- $DS4$ - mined customer rental transactions and fleet tracking data with both location and usage information.

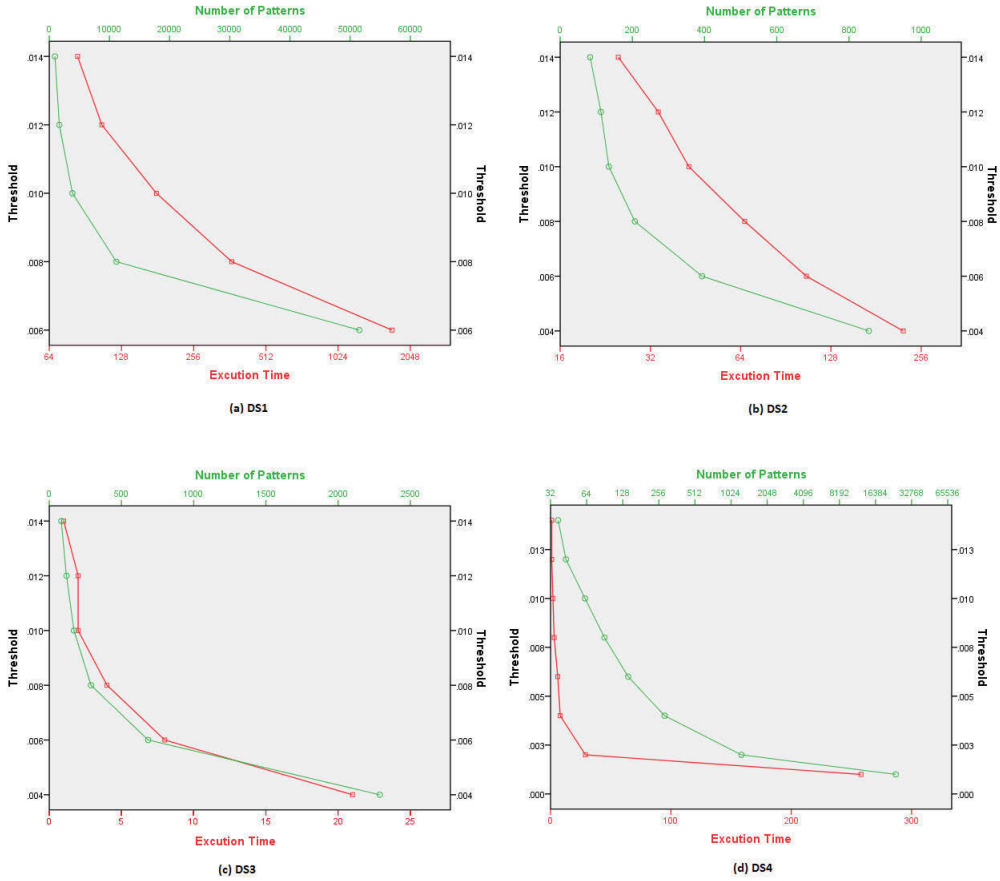

Figure 4.3 : Execution time and number of patterns for the four detection strategies

Fig. 4.3 shows the execution times and number of patterns discovered for each detection strategy. As illustrated in the figure, by considering the fleet tracking factors with rental pattern mining, more useful event patterns are discovered under same conditions. Especially for DS4, with the combination of multiple heterogeneous factors, which takes more execution time tracking with both location and usage as the minimum threshold decreased. However, the results show that it was able to

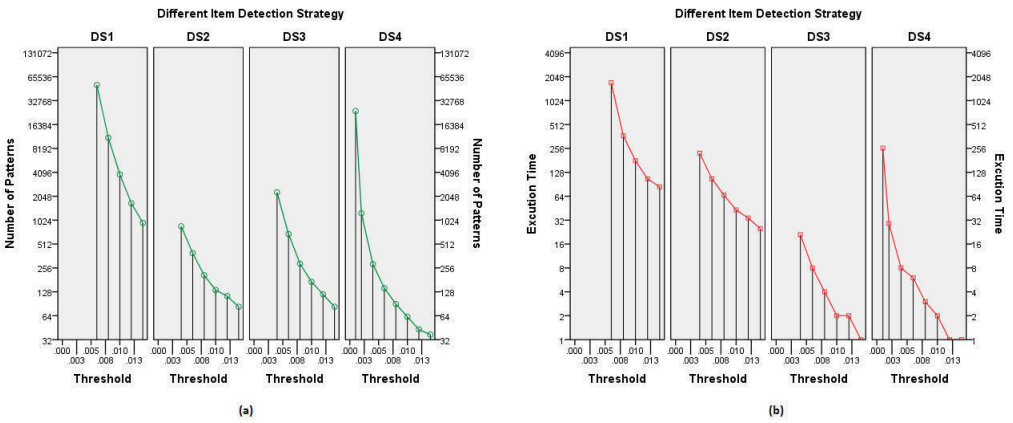

Figure 4.4 : Comparison of the four detection strategies (a) number of patterns (b) execution time

extract more sequential rental patterns in less execution time. The results also show that DS2, DS3, and DS4 took less time to execute given the same threshold as DS1, which did not use any fleet tracking data. For example, DS1 took 1719 seconds, while DS2 only took 106 seconds.

Fig. 4.4 shows the effectiveness of each strategy. We observe that the execution time dramatically decreased with fleet tracking approaches as ($DS2 \sim DS4$), yet the number of patterns detected, given a suitable threshold, stayed a high level. For instance, DS1 took 1719 seconds to execute and discovered around 25,000 patterns, while DS4 took 258 seconds to execute and discovered 24,196 patterns. The results also show that incorporating multiple fleet tracking features into the analysis yielded better execution times and discovered more patterns than either of the single-featured approaches.

To further verify the results of these experiments, we asked several fleet rental industry experts to evaluate the results. Each expert randomly chose a selection pattern identified through each of the strategies ($DS1 \sim DS4$) grouped in lengths of ($2 \sim 6$) and evaluated the results according to their own domain knowledge. They

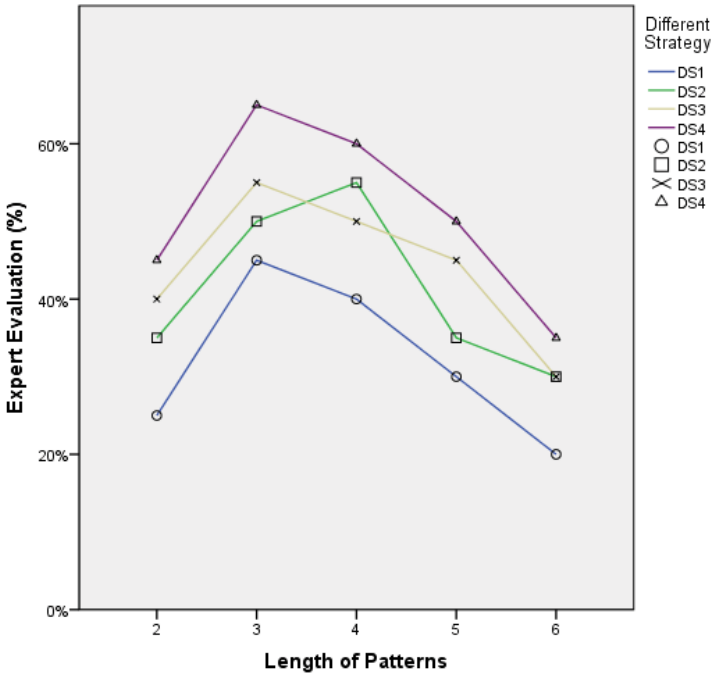

Figure 4.5 : Evaluation of the four detection strategies with length of patterns

then selected the patterns they deemed to be of high utility (more important event patterns) from each group. Fig. 4.5 shows the results of these expert evaluations as the percentage of qualified patterns selected from each group. Although the percentages vary somewhat by pattern length, each of the strategies that incorporated fleet tracking data resulted in a higher percentage of valuable patterns for every group. These results further support the effectiveness of the fleet tracking strategies.

In summary, this chapter demonstrates a novel and practical data mining method for learning complex heterogeneous events in real-world business. We argue that the traditional sequential pattern mining approaches tend to require more data processing and lack accuracy or deep insights into business activities as they ignore important heterogeneous event-related features and information. The practical and efficient sequential pattern mining approach presented in this chapter is able to discover valuable event patterns by combining heterogeneous enterprise data with

measurement algorithms using real-world business transactions and fleet tracking data as evidenced by the results from experiments with real-world enterprise data and the supporting evaluations by industry experts. By effectively mining heterogeneous event data, this pattern mining approach is able to reveal valuable patterns about customer service requirements that will help fleet rental companies ensure they can deliver vehicles to customers when and where they are needed while maintaining high vehicle utilisation to enhance stability and improve profitability.

Chapter 5

Context Heterogeneity

This chapter explores context heterogeneity. Heterogeneous contextual features have been widely recognised as important factors in many enterprise applications. These factors are non-IID and are usually linked to other factors by coupled relationships that have a latent influence on customer behaviours and business activities. Ignoring these coupled relationships can significantly detract from the accuracy of recommendations and predictions, which could otherwise provide deep insights for business.

Hence, the data analysis models presented in this chapter reflect a novel approach to address context heterogeneity in one of the most popular enterprise applications – recommender systems. Collaborative filtering is a fundamental part of the traditional recommender systems. However, integrating contextual information into recommendations is not straightforward with most existing techniques. Therefore, to improve recommendations with contextual features, the model presented here is based on coupled collaborative filtering and measures heterogeneous contextual information through non-IID relationships. Coupled similarity computations are calculated based on the inter-item, intra-context and inter-context correlations between items, users, and contexts. Comparative experiments with different types of collaborative filtering models demonstrate the effectiveness of this approach.

The remainder of this chapter was published as Coupled collaborative filtering for context-aware recommendation at AAAI, 2015, pp. 41724173 by the attributed authors: X. Jiang, W. Liu, L. Cao, and G. Long. Amendments have been made to

improve the clarity of the language in the originally published version and to better suit the context of this thesis.

5.1 Introduction

In recent years, information technology has led to dramatic changes in the business world. Products and services have been reshaped and, more importantly, the nature of competition has changed. Recommender systems analyse patterns of user interest in products or services and then, based on that analysis, provide personalised recommendations. Recommender systems are now widely accepted as an effective means of predicting consumer preferences and enhancing a business's competitive edge [1] [155]. Traditionally recommender systems only take users and items into consideration when making predictions [155]. However, research on consumer behaviour has shown that customer purchasing behaviours are contingent on the context in which decisions are made [1]. Contextual information, such as time, location, and social networking, can influence user preferences, which in turn affects the accuracy of predictions [1] [23] [149] [202]. Therefore, personal recommendations might be significantly improved by incorporating relevant contextual information into a recommender system.

As the most successful approach to building recommender systems, collaborative filtering uses a matrix of items rated by users to predict other topics or products those users might like. Generally, collaborative filtering recommender systems only rely on two types of entities for recommendations, users and items, while ignoring the contextual factors that drive a user's preferences. Further, these two entities are typically contained in one data source, which is a set of user ratings for other items, as shown in Fig. 5.1(a). The context-aware recommender system shown in Fig. 5.1(b) considers user preferences for items given the context of location. This type of system provides more information for us to understand what the real pref-

erence of customers in the context they rate. Current contextual recommendation methods typically treat users, items, and locations as independent and identically distributed (IID). However, in practice, contextual information is usually linked to other entities by non-IID coupled relationships and latently act on customer intention prediction [15], as shown in Fig. 5.1(c). Thus, in certain circumstances, simply considering users, items, and locations independently may not provide appropriate recommendations. For example, a person's preferred drink at an airport may be very different from their choice at a vacation spot, underlying the reasons of items, users and contextual may affect each other and drive user preferences latently.

In this chapter, we address the issues associated with recommendations based on context from a new perspective. The approach designed here is based on the assumption that product ratings are determined by both personal factors and the coupled contextual information. Fig. 5.2 illustrates an example of our approach using the context of 'location': the non-IID coupled relationships between users, items, and locations, are modelled and analysed through three interaction formulations: inter-item, intra-context, and inter-context. The intra-context formulation is used to measure the correlations between user ratings for different items in the same location, the inter-item formulation is used to measure the correlations between user ratings for items in different locations, and the inter-context formulation is used to measure the complex correlations between all items in different locations.

These three formulations are modelled using Pearson's correlation coefficient [47], which measures the extent to which two variables linearly relate to each other. To generate context-aware recommendations, novel contextual weighting and prediction functions are used to calculate the coupled similarities between different items. The specific contributions of this chapter are:

- a formulation for a generalised triadic relation between users, items, and con-

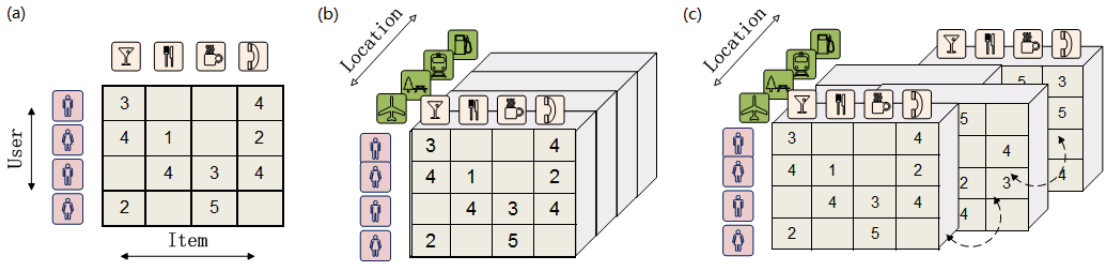

Figure 5.1 : Collaborative filtering (a) User ratings for items. (b) An example of context-aware recommendation. (c) Non-IID coupled relationships with contextual information.

texts to address the problems associated with coupled context-aware relationships;

- a CCF approach that incorporates couplings between and within users, items, and contexts;
- a CCF weighting algorithm for item-based context-aware recommendation; and
- comparative experiments to evaluate the ability of these algorithms and models to generate accurate, insightful recommendations.

5.2 Preliminaries

Research on customer behaviour has shown that customer purchase behaviours are contingent on the context in which a decision is made [1]. With the aim of identifying items of interest to users based on relevant contextual information [23] [149] [202], context-aware recommender systems usually provide more accurate predictions in real-world situations. Chen [23] was among the first to use Pearson's correlation coefficient to extend collaborative filtering models for context-aware recommendations. However, Chen's model is based on the assumption of independent

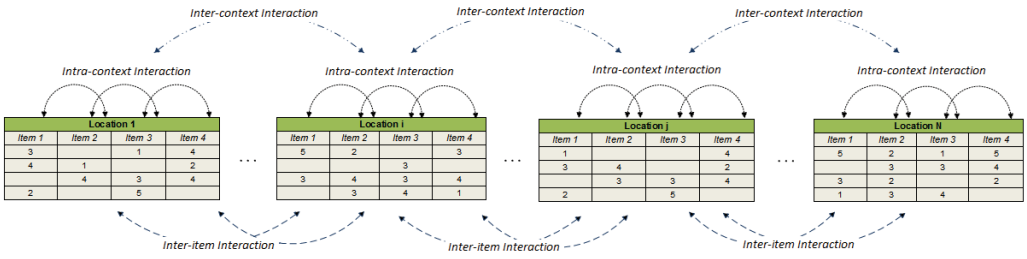

Figure 5.2 : Contextual coupled similarity where the context is 'location'.

contextual factors, which is only applicable to straightforward non-IID environments. Further, an increasing number of researchers argue that the assumption of independence often leads to massive information loss [15] [171]. More recently, Wang et al. [171] proposed the concept of using coupled similarity to assist classification and clustering tasks in unsupervised settings, with some success. Yet, despite these advancements, no reported study has systematically taken coupled relationships into account for context-aware recommendations.

5.3 CCF for context-aware recommendations

Calculating the similarities between items and users is a critical step in collaborative filtering algorithms. In our approach, coupled similarity is calculated in terms of inter-item, intra-context, and inter-context interactions regarding non-IID relationship among users, items and contexts. Each of these three interaction formulations and the coupled similarity calculation, are formalised below.

A common method of exploring similarity is to calculate a Pearson correlation coefficient [47] [23] as a measure of the extent to which two variables linearly relate to each other. Taking an item-based collaborative filtering algorithm as an example, the Pearson correlation between items p and q is

$$Cor_{p,q} = \frac{\sum_{u \in U}(r_{u,p} - \overline{r}_p)(r_{u,q} - \overline{r}_q)}{\sqrt{\sum_{u \in U}(r_{u,p} - \overline{r}_p)^2}\sqrt{\sum_{u \in U}(r_{u,q} - \overline{r}_q)^2}} \tag{5.1}$$

where $u \in U$ denotes the set of users that rated both items p and q, $r_{u,p}$ is user u's rating for item q, and r_q is the average rating of the qth item by all users in U.

Following traditional similarity calculation techniques, the non-contextual information is first arranged into an initial user rating matrix ($L \times K$), which represent L different items rated by K different users. The contextual information contains N different values. Hence, for context-aware recommendations, the user rating matrix can be represented as an ($L \times K \times N$) extended preference cube tensor. The correlations between each pair of users and item ratings can then be calculated for each contextual feature to reflect the globally coupled relationships given context.

5.3.1 Coupled similarity calculations

This subsection introduces the fundamental definitions used to calculate coupled similarity given contextual features.

Definition 1 The inter-item interaction measures the correlations between context i and j is quantified as as an $L \times L$ matrix $M^{Ie}(i,j)$, in which the (p,q) entry represents the correlation between each pair of item ratings $\langle r_i \rangle^p$ and $\langle r_j \rangle^q$.

$$M^{Ie}(i,j) = \begin{bmatrix} m^{Ie}_{11}(i,j) & m^{Ie}_{12}(i,j) & \cdots & m^{Ie}_{1L}(i,j) \\ m^{Ie}_{21}(i,j) & m^{Ie}_{22}(i,j) & \cdots & m^{Ie}_{2L}(i,j) \\ \vdots & \vdots & \ddots & \vdots \\ m^{Ie}_{L1}(i,j) & m^{Ie}_{L2}(i,j) & \cdots & m^{Ie}_{LL}(i,j) \end{bmatrix} \tag{5.2}$$

where $m^{Ie}_{pq}(i,j) = Cor(\langle r_i \rangle^p, \langle r_j \rangle^q)$ is the Pearson's correlation coefficient between values $\langle r_i \rangle^p$ and $\langle r_j \rangle^q$ between different items p and q.

Taking the numbers of rating matrix as an example in Fig.5.2, the inter-item

below shows the correlation between each item pair of Location 1 and Location n.

$$M^{Ie}(1,n) = \begin{bmatrix} 1 & 0 & -0.327 & -1 \\ 0 & -1 & 0 & -1 \\ -1 & 0.866 & 1 & -1 \\ 0 & -1 & -10 & -0.189 \end{bmatrix} \tag{5.3}$$

Thus, the inter-item relationship between Location 1 and Location n is captured as a correlation coefficient between the items in different contexts.

Definition 2 The intra-context interaction within a context i is represented as an $L \times L$ matrix $M^{Ia}(i)$, in which the (p, q) entry measures the correlation between the values $\langle r_i \rangle^p$ and $\langle r_i \rangle^q$.

$$M^{Ia}(i) = \begin{bmatrix} m_{11}^{Ia}(i) & m_{12}^{Ia}(i) & \cdots & m_{1L}^{Ia}(i) \\ m_{21}^{Ia}(i) & m_{22}^{Ia}(i) & \cdots & m_{2L}^{Ia}(i) \\ \vdots & \vdots & \ddots & \vdots \\ m_{L1}^{Ia}(i) & m_{L2}^{Ia}(i) & \cdots & m_{LL}^{Ia}(i) \end{bmatrix} \tag{5.4}$$

where $m_{pq}^{Ia}(i) = Cor(\langle r_i \rangle^p, \langle r_i \rangle^q)$ is the Pearson's correlation coefficient between the ratings of items p and q in context i.

Again, based on Location 1 data in Fig. 5.2, the intra-context measurement between Item 1 and Items 2, 3, and 4) would be

$$M^{Ia}(c_1) = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \tag{5.5}$$

where c_1 represents the context 'Location 1'. We then capture the intra-context

relationships within Location 1 as a correlation coefficient between items in the same context.

Based on previous intra-context calculation result, the inter-context interaction aims to measure inter correlation among different contexts.

Definition 3 The inter-context interaction between context i and j is represented as an $L \times L$ matrix $M^{Ia}(i,j)$, in which the (p,q) entry describes the correlation between the intra-context of $M^{Ia}(i)$ and $M^{Ia}(j)$. $\langle m^{Ia}(i)\rangle^p$ represents the vector of $M^{Ia}(i)$ for item p. While $\langle m^{Ia}(j)\rangle^q$ is for the vector of $M^{Ia}(j)$ for item q.

$$M^{Ia}(i,j) = \begin{bmatrix} m^{Ia}_{11}(i,j) & m^{Ia}_{12}(i,j) & \cdots & m^{Ia}_{1L}(i,j) \\ m^{Ia}_{21}(i,j) & m^{Ia}_{22}(i,j) & \cdots & m^{Ia}_{2L}(i,j) \\ \vdots & \vdots & \ddots & \vdots \\ m^{Ia}_{L1}(i,j) & m^{Ia}_{L2}(i,j) & \cdots & m^{Ia}_{LL}(i,j) \end{bmatrix} \tag{5.6}$$

where $m^{Ia}_{pq}(i,j) = Cor(\langle m^{Ia}(i)\rangle^p, \langle m^{Ia}(j)\rangle^q)$ is the Pearson's correlation coefficient for $m^{Ia}(i), m^{Ia}(j)$ between different items p and q. And

$M^{Ia}(i) = \begin{bmatrix} m^{Ia}_{p1}(i) & m^{Ia}_{p2}(i) & \cdots & m^{Ia}_{pL}(i) \end{bmatrix}$, $M^{Ia}(j) = \begin{bmatrix} m^{Ia}_{q1}(j) & m^{Ia}_{q2}(j) & \cdots & m^{Ia}_{qL}(j) \end{bmatrix}$.

5.3.2 Coupled similarity integrated-weight

With above three definitions, the coupled similarity integrated-weight is designed to measure the total weight $w_{i,j,v}$ of correlations between two different contexts i and j for a specific item v.

Definition 4 The coupled similarity integrated-weight between context i and j is represented as:

$$w_{i,j,v} = \frac{\|M^{Ie}_v(i,j)\|_F + \lambda \|M^{Ia}_v(i,j)\|_F}{1+\lambda} \tag{5.7}$$

where $\|M_v^{Ia}(i,j)\|_F = \sqrt{\sum_{k\in L}(m_{vk}^{Ia}(i,j))^2}$, reflecting the inter-item relationship between context i and j,

$\|M_v^{Ie}(i,j)\|_F = \sqrt{\sum_{k\in L}(m_{vk}^{Ie}(i,j))^2}$, reflecting the inter-context relationship between context i and j.

Further, the coupled similarity integrated-weight is incorporated into a prediction function reflecting the couplings within and between different contexts.

5.3.3 Prediction calculation

Definition 5 In CCF, coupled similarities are used to predict contextual ratings. Here, we use a simple weighted average to predict the rating P_{u,v,c_i} for user u on item v given the context c_i [155].

$$P_{u,v,c_i} = \frac{\sum_{c_j\in N} r_{u,v,c_j} w_{c_i,c_j,v}}{\sum_{c_j\in N} \mid w_{c_i,c_j,v} \mid} \tag{5.8}$$

where the summations are over all other rated contexts $c_j \in N$ on item v by user u. $w_{c_i,c_j,v}$ is the weight between the contexts c_i and c_j for item v, and r_{u,v,c_j} is user u's rating for item v given the context c_j.

5.4 Experiments

The experiments were conducted on a real-world GPS location dataset recorded from April 2007 to October 2009 in Beijing [202]. CCF was compared to several other approaches to evaluate its performance.

5.4.1 Data preparation

The datasets were recorded by GPS devices covering a total of 139,310 square kilometres and divided into five different types of activities: 'Food & Drink', 'Shopping', 'Movies & Shows', 'Sports & Exercise', and 'Tourism' across 168 locations for

the purposes of this study. Since frequency indirectly reflects the user preferences, we normalised the frequency counts for each user and activity.

5.4.2 Metrics and comparison methods

To measure the quality of the rating predictions, we used the most widely-used evaluation metrics, mean absolute error (MAE) and root mean squared error (RMSE) [155]:

$$MAE = \frac{\sum_{i,j}|p_{i,j}-r_{i,j}|}{n} \ , \ RMSE = \sqrt{\frac{\sum_{i,j}(p_{i,j}-r_{i,j})^2}{n}} \ ,$$

where n is the total number of ratings over all users, $p_{i,j}$ is the predicted rating of user i for item j, and $r_{i,j}$ is the actual rating given.

The following approaches were selected as comparisons.

- **UBCF**: a traditional recommender system based on a user collaborative filtering method [155]. The five closest users were considered to be the neighbourhood.
- **UBCF**: a traditional recommender system based on an item collaborative filtering method [155]. Again, the five closest users were considered to be the neighbourhood.
- **MF**: The most well-known matrix factorization. This method minimises the squared error with stochastic gradient descent [83].

5.4.3 Results

Table 5.1 and Fig. 5.3 show the results of all comparison methods in a context-aware recommendation task.

As Table 5.1 shows, the CCF method delivered much better performance than the traditional collaborative filtering methods in terms of RMSE. To verify the

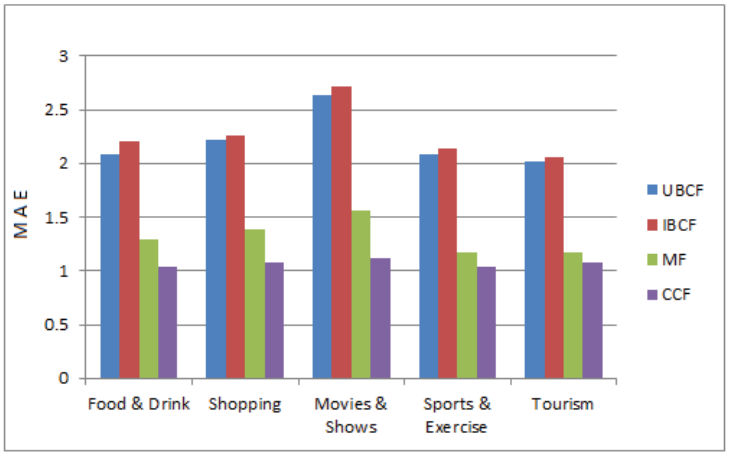

Figure 5.3 : MAE comparison for all models

significance of the performance improvements, we conducted paired t-tests. The low p-values shown in the bottom of Table 5.1 demonstrate the significant advantages of our method. Fig. 5.3 plots the results on bar charts, illustrating that CCF better captures the contextual factors for use in rating predictions. In terms of MAE, our proposed model resulted in a minimum 12% improvement over the UBCF, IBCF, and MF models in all test cases.

Table 5.1 : RMSE comparison for all models

Model	*UBCF*	*IBCF*	*MF*	*CCF*
Food & Drink	2.50835	2.60468	1.71146	1.33885
Shopping	2.70816	2.75476	1.79297	1.42321
Movies & Shows	3.06439	3.15677	1.87627	1.61924
Sports & Exercise	2.47597	2.50778	1.46167	1.21474
Tourism	1.65684	2.46739	1.27661	1.03623
T-test	*0.0112*	*0.0001*	*0.0005*	*Base*

In summary, this chapter investigates context heterogeneity by formulating generalised triadic relationships between contexts in heterogeneous real-world enterprise data. We argue that existing contexts-aware methods do not provide a straightforward way of integrating non-IID heterogeneous contextual relationships into the algorithms from a business perspective. Whereas, this novel CCF approach provides context-aware recommendations by fully leveraging the value of heterogeneous contextual information for real-world business scenarios. Heterogeneous context-aware factors provide valuable information for analysing customer behaviour. By measuring the non-IID relationships between items, customers, and contexts, the proposed CCF describes inter-item, intra-context, and inter-context correlations through coupled similarity calculations to make context-aware predictions about which products or services customers are likely to prefer. Experiments comparing different types of collaborative filtering models demonstrate the effectiveness of our design.

Chapter 6

Domain Heterogeneity

This chapter explores domain heterogeneity in the context of financial markets. Globalisation has resulted in a steady increase in cross-border financial flows around the world, and the complexity and nonlinearity of these flows is challenging for traditional machine learning approaches to model. However, recent developments in deep learning are opening up new opportunities for much more detailed data analyses of complex relationships, such as those found in financial markets.

The cross-domain deep learning approach (Cd-DLA) presented in this chapter is designed for forecasting and predicting market trends in multiple financial markets. An abstract representation of real-world financial market landscapes is built by identifying and structuring three different types of correlations between homogeneous (e.g., the stock markets in two different countries) and heterogeneous (e.g., a stock market and a currency market) markets. The mining architecture integrates the times-series, inner-domain, and cross-domain features and relations to compose multi-task predictions by transferring the heterogeneous domain knowledge. An RNN captures the time-series correlations, while two attention mechanisms identify the inner-domain and cross-domain correlations, respectively. Once captured, all the correlations are aggregated within a parallel deep learning multi-task framework that uses transfer learning to generate market trend forecasts. Experimental results with 10 years of financial data on the currency and stock markets in three countries demonstrate the effectiveness of the design.

The remainder of this chapter was published as "Cross-domain deep learning ap-

proach for multiple financial market prediction" at the 2018 *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2018 by the attributed authors: X. Jiang, S. Pan, J. Jiang, and G. Long. The full paper was accepted on 15 March 2018. Amendments have been made to improve the clarity of the language in the originally published version and to better suit the context of this thesis.

6.1 Introduction

Financial analysis is used to evaluate economic trends, set financial policy, build long-term plans for business activity, and identify projects or companies for investment. Successful predictions about future financial market trends have the potential to yield significant profit and may also help to overcome many other business challenges as well. The global financial system is a structured worldwide framework of legal agreements, institutions, and economic actors, all of which facilitate the international flow of financial capital for the purposes of investment and trade financing. The financial analysis of global markets is extremely complex as multiple factors interact across mutiple markets, each influencing the others. For example, a real estate bubble in the US triggered the financial crisis of 2007 and 2008. However, that bubble was financed by foreign capital from many different countries. In the wake of the crisis, the total volume of world trade in goods and services fell 10% from 2008 to 2009, with much of this loss concentrated in emerging markets.

Fig. 6.1 illustrates some of the complex relationships in global financial markets. In addition to other factors, the movement of stock markets is affected by three major types of correlations: inner-domain correlations, which are the interrelationships between homogeneous markets, for example, the UK stock market and US stock market; cross-domain correlations, which are the interrelationships between heterogeneous markets, such as the US currency market and the US stock market; and time-series correlations, which represent the transitional influences in different

time periods of the markets. Such correlations are embedded within and between all financial markets and countries and all need to be considered for accurate financial market forecasts.

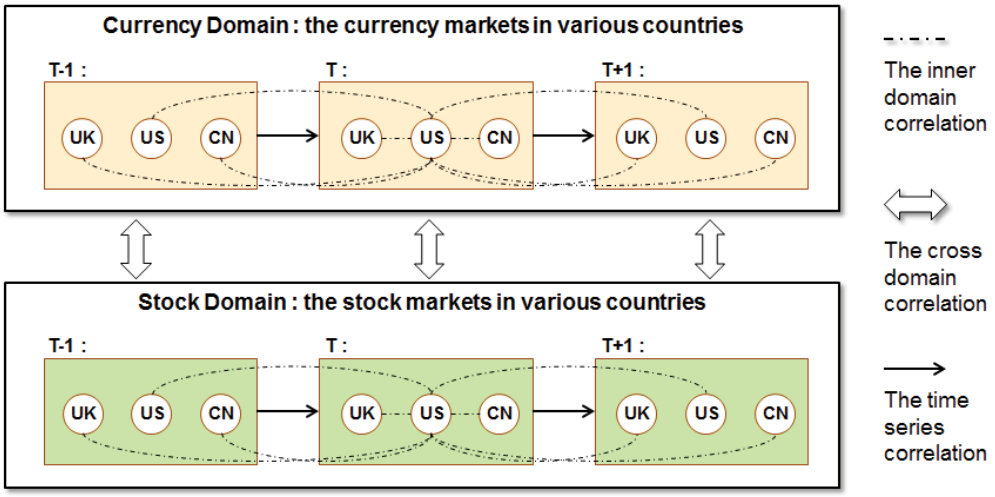

Figure 6.1 : Complex correlations among multiple financial markets

However, analysing the complex correlations between multiple financial markets comes with some significant challenges. First, these correlations are driven by hidden features that are not directly observable from market data. Methods are needed to model factor-driven influences so these hidden factors can be discovered and analysed. Second, the three major types of correlations illustrated in Fig. 6.1 represent highly nonlinear and dynamic movements that fundamentally challenge existing approaches to financial market prediction and substantially increase the learning difficulty for models.

Traditionally, financial analysis has been approached with two main methods: time-series methods and model-based methods. Both focus on historical data from the market itself and, hence, both suffer from several shortcomings associated with data nonlinearity and feature limitations [19]. Time-series methods, such as logistic regression [87] [44], use historical data to infer market trends, but they are limited

to capturing linear relationships between markets. Model-based methods, such as artificial neural networks and hidden Markov models (HMMs) [58] [54], are able to forecast market movements based on hidden factors and the models can learn non-linear relationships. However, most existing model-based methods can only process data from a single market and struggle when modelling the complex correlations in heterogeneous data.

Compared to traditional approaches, deep learning approaches have the advantage of extracting features and modelling nonlinear correlations without relying on econometric assumptions or human expertise. For these reasons, they have recently attracted growing interest in financial market prediction [37] [43] [35] [36]. CNNs [37] use convolution instead of general matrix multiplication to extract the most representative global financial features. RNNs [36] and LSTM [43] have both been used in financial analysis due to their good performance in sequential applications. However, in analysing the complex inner-domain, cross-domain, and time-series correlations between financial markets, transfer learning [35] [8] and attention mechanisms [108] [6] are among the few methods that allow knowledge to be aggregated and transferred from a related task that has already been learned in a deep learning framework.

In this chapter, we propose a novel cross-domain deep learning approach for prediction across multiple financial markets. The approach builds on a parallel MTL framework. First, using an RNN, we capture the time-series correlations along with the hidden features for all markets. These correlations and hidden features are then fed as inputs into a designed attention mechanism and the inner-domain and cross-domain correlations are modelled. The resulting trained model can then be used to forecast trends across multiple homogeneous and heterogeneous markets. The proposed approach was evaluated with 10 years of financial data on the currency and stock markets of three countries - the US, China, and India. The key contributions

of this chapter are:

- a novel cross-domain deep learning approach (Cd-DLA) based on a deep learning network that operates within a unified parallel architecture that models and analyses the complex inner-domain and cross-domain correlations of multiple homogeneous and heterogeneous markets in the same epoch, e.g., currency and stock markets in the US and India;
- a specifically-designed attention mechanism for constructing effective representations of the complex interactions in global financial markets by capturing inner-domain, cross-domain, and time-series correlations;
- a joint optimisation algorithm for learning the deep learning network parameters and attention weights given inner-domain, cross-domain, and time-series influences for multiple financial market prediction; and
- an empirical study using real-world financial datasets that validates the effectiveness of our proposed approach.

6.2 Preliminaries

This section introduces some of the concepts used in this chapter and formalises the complex correlations in financial markets, followed by a brief background on financial market analysis and attention mechanisms, which are key components of our model.

6.2.1 Problem formalisation

Suppose there are J countries, and each country has I financial market domains. Let m_{ij} represent the observation values from the market domain i in country j.

In this chapter, we focus on representing three types of correlations: inner-domain, cross-domain, and time-series correlations. The corresponding definitions follow.

Definition 1. Inner-domain correlation. These are the correlations between the homogeneous domains of all countries. Formally, an inner-domain correlation with respect to market i is represented as

$$\delta_i = \{Idc_{j=1}^{J}(m_{ij})\} \tag{6.1}$$

where Idc denotes a correlation between homogeneous markets with respect to domain i.

Definition 2. Cross-domain correlation. These are the correlations between the heterogeneous domains of all countries. Cross-domain correlations are represented as

$$\eta = \{Cdc_{i=1}^{I}(\delta_i)\} \tag{6.2}$$

where Cdc denotes a correlation between heterogeneous markets with respect to heterogeneous domains.

Definition 3. Time-series correlation. These correlations influences are derived from historical data. The representation of n time-series correlations with respect to δ_i is given by

$$\delta_{i,t}|\{m_{ij,t-n,t-1}\}_{j=1}^{J} \tag{6.3}$$

which denotes a representation of an inner-domain correlation at time t influenced by the past period from $t-n$ to $t-1$.

The representation of n-order time-series correlation with respect to η is

$$\eta_t | \{\delta_{i,[t-n,t-1]}\}_{j=1}^{J} \tag{6.4}$$

which denotes the representation of a cross-domain correlation at time t influenced by the past period from $t-n$ to $t-1$.

6.2.2 Financial market analysis

Financial market analysis has been studied in a variety of fields, but most approaches are highly dependent on historical observations and input variables. Limited studies can be found that address the underlying complex correlations between market indicators that fundamentally drive global market movements.

Machine learning methods have been increasingly explored for financial market analysis. Typical models include the logistic method and HMM, which check for any systematic patterns in the time-series data, then use those patterns to make predictions [58] [87]. However, these methods suffer from several shortcomings due to the difficulty of capturing the nonlinear relationships that characterise complex financial markets. But the recent emergence of the concept of multi-layered networks now allows DNNs to be used as prediction tools [25] [96] [119] [122] [172] [182]. For example, stacked restricted Boltzmann machines have been used as autoencoders to extract features [161] [173]. RNNs have shown promising results in a variety of sequential applications [160] [163]. LSTM, as a class of RNNs with sophisticated recurrent hidden and gated units, have been particularly successful due to their ability to learn hidden long-term sequential dependencies [25] [28]. Yet, although these neural network techniques have explored temporal and hidden correlations, few studies have effectively addressed the inner-domain and cross-domain correlations among homogeneous and heterogeneous markets for making predictions about multiple financial markets.

6.2.3 Attention mechanisms

In recent work on deep learning, attention mechanisms have been proposed as a way of computing the alignment scores between elements from two different sources. As an example, consider a source sequence $x_t^{ij} = [x_1^{ij}, x_2^{ij}, ..., x_n^{ij}]$ from market i in country j and a vector representation of a source influence g from another market. An attention mechanism computes the alignment score between x_t^{ij} and g with a compatibility function $f(x_t^{ij}, g), t = 1, ..., n$, which measures the dependency between x_t^{ij} and g. A softmax function then transforms the scores into a probability distribution $p(a|x^{ij}, g)$ by normalising all the n sequence of x^{ij}. Here, a is an indicator of which sequence in x^{ij} is important to g in the task. The above process is illustrated in Fig. 6.2 and summarised by the following formulations.

$$p(a|x^{ij}, g) = softmax([f(x_t^{ij}, g)]_{i=1}^{n}) \tag{6.5}$$

Specifically,

$$p(a = t|x^{ij}, g) = \frac{\exp(f(x_t^{ij}, g))}{\sum_{i=1}^{n} \exp(f(x_t^{ij}, g))} \tag{6.6}$$

The output of this attention mechanism is the weighted sum of all time-series in x^{ij}, where the weight assigned to each value is calculated with a compatibility function for sequence x_t^{ij} in market i in country j and another correlation to market g. In this way, the domain heterogeneity could be learned across heterogeneous markets.

6.3 A cross-domain deep learning approach for multiple financial market prediction

In this section, we focus on analysing the three types of correlations in global financial markets, as formalised in Definitions 1, 2, and 3 in Section 6.3. We propose a novel cross-domain deep learning approach (Cd-DLA) to model these complex correlations.

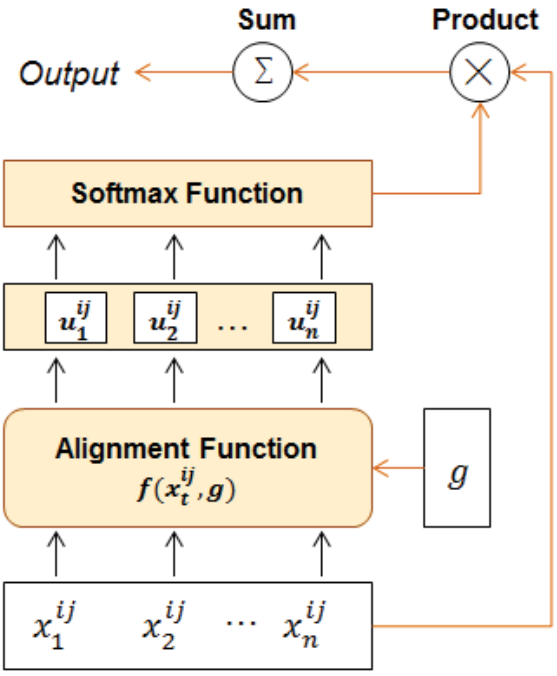

Figure 6.2 : The attention mechanism u_t^{ij} denotes alignment score $f(x_t^{ij}, g)$.

6.3.1 Financial market prediction framework

This section demonstrates an example of how the Cd-DLA framework analyses the three types of correlations between currency markets and stock markets in multiple countries. Fig. 6.3 illustrates the overall architecture of the Cd-DLA framework.

To analyse the time-series correlations between the temporal data inputs across all markets, Cd-DLA leverages a recurrent neural network (RNN). An inner-domain attention neural network (Id-ANN) captures the inner-domain correlations for homogeneous markets, and a cross-domain attention neural network (Cd-ANN) models the cross-domain correlations for heterogeneous markets. Market forecasting is handled in the prediction layer. Each of these components is described in more detail in the following sections.

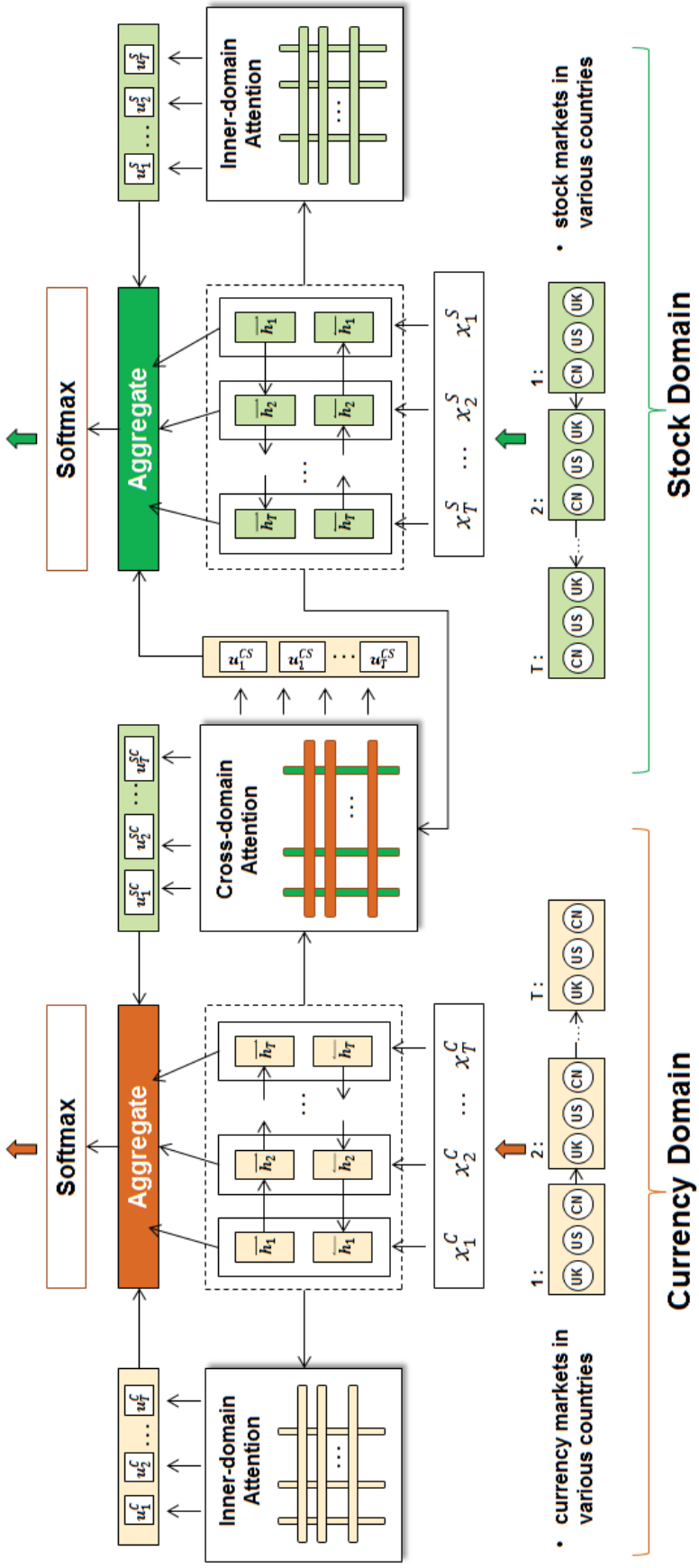

Figure 6.3 : The architecture of the cross-domain deep learning approach in currency and stock markets prediction.

Global financial data embedding

Suppose there are two types of global financial markets, for example, a currency domain C and a stock domain S, and any of these markets may be located in any country. Each market has t sequences of market values in all the countries. The objective of a data embedding task is to pre-process the complex data inputs X of the global financial markets with a uniform expression: $X \rightarrow \{\{x_1^C, \cdots, x_t^C\}, \{x_1^S, \cdots, x_t^S\}\}$, where $\{x_1^C, \cdots, x_t^C\}$ corresponds to t sequences in the currency market C and corresponds to t sequences in the stock market S, respectively.

The items $x_t^C =< x_t^{uk}, x_t^{us}, \cdots, x_t^{au} >^C$ and $x_t^S =< x_t^{uk}, x_t^{us}, \cdots, x_t^{au} >^S$ represent the vectors of the countries' market values in the tth time period for each type of market. Once the data embedding process is complete, Cd-DLA can process the data inputs X for each financial market through the RNN and the attention mechanisms to reveal the complex correlations as a whole epoch training procedure.

A neural network for time-series correlations

To model the nonlinear time-series correlations between market values, we rely on a deep learning approach with a bi-directional RNN and an LSTM unit. The bi-directional RNN orders the sequential inputs in two ways, from past to future and from future to past. Unlike traditional unidirectional RNNs, a bi-directional RNN combines both directions of the hidden states to preserve the information for any point in time from either the past or the future. This means the RNN can efficiently process the past via forward states, and the future via backward states, for a specific time t as:

$$\begin{aligned} \overrightarrow{h_t} &= \overrightarrow{LSTMU}(x_t), t \in [1, T] \\ \overleftarrow{h_t} &= \overleftarrow{LSTMU}(x_t), t \in [1, T] \end{aligned} \tag{6.7}$$

where LSTMU represents a standard unit of long short-term memory. Given a sequence with T items, the hidden outputs of a given transaction inputX_t, $t \in [1, T]$ are produced from the following subfunctions:

- forget gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$,
- input gate layer: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$,
- new contribution: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$,
- update cell state (memory): $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$,
- output gate layer: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$, and
- output to next layer: $h_t = o_t * \tanh(C_t)$

where σ represents the sigmoid activation function, and $[h_{t-1}, x_t]$ is a concatenatation of h_{t-1} and x_t. The parameters are a concatenation of the forward hidden state $\overrightarrow{h_t}$ and the backward hidden state $\overleftarrow{h_t}$. $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$, summarises the sequence information for all transactions centred around x_t.

A neural network for inner-domain correlations

To model the inner-domain correlation within homogenous markets, we add inner-domain attention to the previous output $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. The inner-domain neural network (IdNN) includes an attention function $f_{inner}(h_t) \rightarrow f_{inner}([\overrightarrow{h_t}, \overleftarrow{h_t}])$, where $\overrightarrow{h_t} = \{\overrightarrow{h_t^{uk}}, \overrightarrow{h_t^{us}}, \cdots, \overrightarrow{h_t^{au}}\}$ and $\overleftarrow{h_t} = \{\overleftarrow{h_t^{uk}}, \overleftarrow{h_t^{us}}, \cdots, \overleftarrow{h_t^{au}}\}$. These functions are responsible for linking and fusing the hidden temporal outputs from the RNN network, and a feature vector $u^{(inner)}$ summarises the attention weights. The result A_{inner} is obtained by aggregating the feature vector $u^{(inner)}$ and the previous hidden output h_t. The IdNN's attention mechanism and aggregation functions are explained in Section 6.2.

A neural network for cross-domain correlations

To model the cross-domain correlations between heterogeneous markets, we add cross-domain attention to the previous outputs. For instance, the heterogeneous currency market C and stock market S shown in Fig. 6.3. The cross-domain neural network (CdNN) includes an attention function $f_{cross}(h_t^C, h_t^S)$ for linking and fusing the temporal outputs from multiple financial markets, where $h_t^C = [\overrightarrow{h_t}, \overleftarrow{h_t}]^C$ represents the currency market and $h_t^S = [\overrightarrow{h_t}, \overleftarrow{h_t}]^S$ represents the stock market. Then, the feature vectors $u^{(cross)}$, like the vectors u^{SC} and u^{CS} in Fig. 6.3, are used to summarise the cross-domain attention weights in two directions, where u^{SC} focuses on the influence direction from the stock market to the currency market, and u^{CS} focuses on the influence direction from the currency market to the stock market. The result A_{cross} in the CdNN is obtained by aggregating the feature vector and the previous hidden output. CdNN's attention mechanism and aggregation functions are explained in Section 6.2.

Multiple financial market prediction

To generate predictions for multiple markets, each pair of outputs from IdNN and CdNN is concatenated $[A_{inner}, A_{cross}]$, followed by a softmax layer to predict whether the trend in each financial market will increase or decrease in the next trading window:

$$Pred = \text{softmax}(W_p v + b_p), \tag{6.8}$$

where v is a high-level vector that represents the aggregated hidden outputs A_{inner} and A_{cross}. The inner-domain attention outputs A_{inner} from the IdNN network, and the cross-domain attention outputs A_{cross} from the CdNN network. Each heterogeneous market has its own softmax function to predict future trends in mar-

ket value. In the complex financial dataset $X \rightarrow \{\{x_1^C, \cdots, x_t^C\}, \{x_1^S, \cdots, x_t^S\}\}$, two prediction functions are then used to generate market predictions for the currency and stock domains, as shown in Fig. 6.3. The vector v that aggregates A_{inner} and A_{cross} is explained in Section 6.2.

6.3.2 Using attention in a learning approach

To effectively model the complex interactions between multiple financial markets, Cd-DLA incorporates three attention mechanisms: one for inner-domain attention, which captures the interactions between homogeneous markets; one for cross-domain attention, which captures the interactions between heterogeneous markets; and an overarching Cd-DLA attention mechanism, which combines the inner-and cross-domain attentions to model transitional influences over different time periods.

Inner-domain Attention

This section explains how the Cd-DLA framework captures correlations in homogeneous markets using currency markets as an example. In the neural network for time-series correlations, we use hidden outputs from the RNN to learn the time-series correlation between markets. However, even within a domain, financial markets are usually influenced by each other. However, not all hidden values contribute equally to these influences. Hence, an inner-domain attention mechanism is introduced to extract hidden temporal values, which can then be used to learn the level of influence each factor has on a specific country. Specifically, an aggregated representation of those hidden temporal values is used to construct an inner-domain feature vector.

$$f_{inner}(h_t^{(1)}) = W_{inner}^T \tanh(W_h \cdot h_t^{(1)} + b_h) + b_{inner} \tag{6.9}$$

where $\tanh(\cdot)$ represents the activation function, and $h_t^{(1)}$ represents the hidden outputs from the RNNs for all countries. The function $f_{inner}(h_t^{(1)})$ measures

the attention of other countries to a specific country. Then, the following formula transforms the scores $f_{inner}(h_t^{(1)}), t = 1, \cdots, T$ into a probability distribution:

$$u_t^{(inner)} = \frac{\exp(f_{inner}(h_t^{(1)}))}{\sum_t \exp(f_{inner}(h_t^{(1)}))} \tag{6.10}$$

The inner-domain attention A_{inner} is calculated as a weighted probability vector $u_t^{(inner)}$ for each hidden output h_t:

$$A_{inner} = \sum_t u_t^{(inner)} \cdot h_t^{(1)} \tag{6.11}$$

To compute the inner attention for stock markets, $h_t^{(1)}$ is simply replaced with the hidden values h_t^C from the stock domain outputs.

Cross-domain attention

Again using currency markets as an example, an attention mechanism is used to capture the correlations between heterogeneous domains, e.g., stock markets. A cross-domain feature vector measures the importance of each hidden value in each market:

$$f_{cross}(h_t^{(1)}, h_t^{(2)}) = W_{cross}^T \tanh(W_h^{(1)} h_t^{(1)} + W_h^{(2)} h_t^{(2)}) + b_{cross} \tag{6.12}$$

where $\tanh(\cdot)$ represents the activation function, $h_t^{(1)}$ represents the hidden currency market values for all countries, and $h_t^{(2)}$ denotes the hidden stock market values for all countries. $f_{cross}(h_t^{(1)}, h_t^{(2)})$ measures the cross-domain attention from the stock market to the currency market for all countries. Then, a softmax formula transforms the scores $f_{cross}(h_t^{(1)}, h_t^{(2)})$ into a probability distribution $u_t^{(cross)}$ as follows:

$$u_t^{(cross)} = \frac{\exp(f_{cross}(h_t^{(1)}, h_t^{(2)}))}{\sum_t \exp(f_{cross}(h_t^{(1)}, h_t^{(2)}))} \tag{6.13}$$

Cross-domain attention A_{cross} is calculated as a weighted probability vector $u_t^{(cross)}$ for each hidden value $h_t^{(1)}$.

$$A_{cross} = \sum_t u_t^{(cross)} \cdot h_t^{(1)} \tag{6.14}$$

To calculate cross-domain attention from stock markets to currency markets, $h_t^{(1)}$ is replaced with the hidden values h_t^C from the currency domain.

Cd-DLA's attention mechanism

Global financial markets usually work together and influence each other through nonlinear relationships. To capture these influences over time, an overarching Cd-DLA attention mechanism aggregates the components of the inner-domain attention A_{inner} and the cross-domain attention A_{cross} explained in the previous sections using the following fusion functions F:

$$F = \sigma(W^{(f1)} A_{inner} + W^{(f2)} A_{cross} + b^{(f)}) \tag{6.15}$$

$$v = F \odot A_{inner} + (1 - F) \odot A_{cross} \tag{6.16}$$

where σ represents a sigmoid active function, and $W^{(f1)}$,$W^{(f2)}$ and $b^{(f)}$ are the learnable parameters of the aggregation layer.

6.3.3 Learning optimal parameters

The parameters for the learning optimisation algorithm are divided into two categories: α and β. The parameters for α govern currency domains and include:

- RNN: $\alpha_{rnn} = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}^C$

Algorithm 3 Learning optimisation for parameters (α, β)

Input: Training set (X_T, Y_T), Validation set (X_V, Y_V), Max epochs (Max_{ep}), Learning rate $(\gamma_\alpha, \gamma_\beta)$

Output: Learned parameters (α^*, β^*)

Net $\leftarrow$ construct-Attention-Neural-Net()
{*Random initialization*}
α, $\beta \leftarrow$ initialize-Net(Net)
val-err $\leftarrow 1$
{*Looping in number of epochs]*}
for $e \in [1, Max_{ep}]$ **do**
  for $b \in [1, batchSize]$ **do**
    $out^b \leftarrow$ forwardPass$(X_T^b, Y_T^b, Net, \alpha, \beta)$
    {***Training currency market of attention neural net***}
    $grad_\alpha^b \leftarrow$ backwardFPass$(out^b, X_T^b, Y_T^b, Net, \alpha, \beta, \delta)$
    $\alpha^* \leftarrow$ update-CNet-Params$(Net, \alpha, \beta, \gamma_\alpha, grad_\alpha^b)$
    {***Training stock market of attention neural net***}
    $grad_\beta^b \leftarrow$ backwardSPass$(out^b, X_T^b, Y_T^b, Net, \alpha^*, \beta, \delta)$
    $\beta^* \leftarrow$ update-SNet-Params$(Net, \alpha^*, \beta, \gamma_\beta, grad_\beta^b)$
    α, $\beta \leftarrow \alpha^*$, β^*
  end for
  val-err$^* \leftarrow$ forwardPass$(X_V^b, Y_V^b, Net, \alpha, \beta)$
  val-err $\leftarrow$ val-err*
end for
return (α^*, β^*)

- IdNN: $\alpha_{inner} = \{W_{inner}, W_h, b_h, b_{inner}\}^C$
- OdNN: $\alpha_{cross} = \{W_{cross}, W_h^{(1)}, W_h^{(2)}, b_{cross}\}^C$
- MMP: $\alpha_{pred} = \{W^{(f1)}, W^{(f2)}, b^{(f)}\}^C$

While the parameters for β govern the stock domain and include:

- RNN: $\beta_{rnn} = \{W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o\}^S$
- IdNN: $\beta_{inner} = \{W_{inner}, W_h, b_h, b_{inner}\}^S$
- OdNN: $\beta_{cross} = \{W_{cross}, W_h^{(1)}, W_h^{(2)}, b_{cross}\}^S$
- MMP: $\beta_{pred} = \{W^{(f1)}, W^{(f2)}, b^{(f)}\}^S$

The goal in optimising these parameters is to jointly learn both groups of parameters using stochastic gradient descent with backpropagation errors.

6.4 Experiments and evaluation

The proposed approach was evaluated for its ability to generate accurate predictions about multiple financial markets on two real-world datasets. The datasets were extracted from the International Monetary Fund and Yahoo Finance. One dataset spans the financial crisis period (2007-2009); the other spans a non-crisis period (2010-2017). The details of each dataset and the experimental settings used follow.

6.4.1 Datasets and experimental settings

Financial crisis period (2007-2009)

This dataset contains historical prices for various market indexes in three countries: the Unites States(US), China, and India. These countries were chosen because,

according to the International Monetary Fund, they account for more than 40% of the world's total GDP. Two types of financial domains for each country were included, the stock market and the currency market. The data comprises the daily closing prices from Jan 2007 to Dec 2009, and those prices decoded into returns according to $RI_t = \frac{PI_t - PI_{t-1}}{PI_{t-1}}$, where RI_t and PI_t are the return and closing prices at time t, respectively. Given that different markets may trade on different days, we only included data for the days on which all markets traded.

Non-crisis period (2010-2017)

This dataset covers the same markets and countries as the crisis period dataset but contains the daily closing prices for the period Jan 2010 to Dec 2017, along with the decoded prices returns according to $RI_t = \frac{PI_t - PI_{t-1}}{PI_{t-1}}$, where RI_t and PI_t are the return and the closing prices at time t, respectively. Again, only data for the days where all markets traded were included.

Specific details on the stock and currency markets used in both datasets are provided in Table 6.1.

Table 6.1 : Data preparation: trading indexes

Domain	**United States**	**China**	**India**
Currency Markets	SDR/USD	SDR/CNY	SDR/INR
Stock Markets	∧DJI	∧SSEC	∧BSESN

Comparison methods

The following RNN, Id-ANN, and Cd-ANN algorithms were selected as appropriate comparisons to evaluate Cd-DLA's performance.

- RNN - a BiLSTM network trained on sequence data from homogeneous markets;
- Id-ANN - an inner-domain attention network trained on the sequence data from homogeneous markets; and
- Cd-ANN - a cross-domain attention network trained on the sequence data from heterogeneous markets.

Experimental settings

Each record in each dataset was converted into an information input X as outlined in 6.3. All neural networks were trained on each dataset during the training procedure. 80% of the samples in each dataset were used as the training set, with the remaining 20% used as the testing set. The neural network settings for each compared method appears in Table 6.2.

Evaluation Metrics

Two commonly-used classification metrics were used to evaluate the prediction performance: F-measure and area under curve (AUC) [41].

6.4.2 Experiment results

The results of the experiments conducted on the crisis and non-crisis datasets are shown in Tables 6.3 and 6.4, respectively.

With each dataset, Cd-DLA outperformed all comparison methods in terms of both F-measure and AUC. We also observed that the two attention-based methods, Id-ANN and Cd-ANN, showed better prediction performance on most of the datasets than the baseline RNN network for all homogeneous and heterogeneous markets. Additionally, Cd-DLA showed obvious improvements in terms of F-measure and AUC with the dataset spanning the financial crisis period. This effectively reflects

Table 6.2 : Network settings for the financial data sets

Compared Method	**Lrn. Rate**	**Hidden Layer Neutron Setting**
RNN	γ=0.01	$\{input \rightarrow 256 \rightarrow 256 \rightarrow output\}$
Id-ANN	γ=0.01	$\{input \rightarrow 256 \rightarrow 256 \rightarrow h\}$
		$\{h \rightarrow 512 \rightarrow 64 \rightarrow u_{(inner)}\}$
Cd-ANN	γ_α=0.01	$input : c \sim$ currency domain
		$\{c \rightarrow 256 \rightarrow 256 \rightarrow h^\alpha\}$
	γ_β=0.01	$input : s \sim$ stock domain
		$\{s \rightarrow 256 \rightarrow 256 \rightarrow h^\beta\}$
		$\{h^\alpha, h^\beta \rightarrow 512 \rightarrow 64 \rightarrow u^\alpha_{(cross)}, u^\beta_{(cross)}\}$
Cd-DLA	γ_α=0.01	$input : c \sim$ currency domain
		$\{c \rightarrow 256 \rightarrow 256 \rightarrow h^\alpha\}$
		$input : s \sim$ stock domain
	γ_β=0.01	$\{s \rightarrow 256 \rightarrow 256 \rightarrow h^\beta\}$
		$\{h^\alpha \rightarrow 512 \rightarrow 64 \rightarrow u^\alpha_{(inner)}\}$
		$\{h^\beta \rightarrow 512 \rightarrow 64 \rightarrow u^\beta_{(inner)}\}$
		$\{h^\alpha, h^\beta \rightarrow 512 \rightarrow 64 \rightarrow u^\alpha_{(cross)}, u^\beta_{(cross)}\}$

Table 6.3 : Evaluation on financial crisis data set

Experiment Setting		**F-measure**			
Domain	***Country***	***RNN***	***Id-ANN***	***Cd-ANN***	***Cd-DLA***
Currency	US	0.4427	0.04233	0.5039	0.6111
	China	0.5241	0.5161	0.5655	0.6052
	India	0.4640	0.4444	0.5390	0.5714
Stock	US	0.4962	0.4671	0.5625	0.6154
	China	0.4748	0.5263	0.6000	0.6154
	India	0.5067	0.4615	0.4932	0.6329
Experiment Setting		**AUC**			
Domain	***Country***	***RNN***	***Id-ANN***	***Cd-ANN***	***Cd-DLA***
Currency	US	0.5142	0.5413	0.5881	0.6090
	China	0.5715	0.5654	0.6276	0.6367
	India	0.5409	0.5649	0.6149	0.7718
Stock	US	0.5344	0.5420	0.5021	0.5663
	China	0.5388	0.5457	0.5305	0.5465
	India	0.5122	0.5414	0.5072	0.5589

the practical influence between the markets of different domains during periods of significant global financial crisis.

We also tested Cd-DLA in terms of loss values with the financial crisis dataset, as shown in Fig. 6.4. Cd-DLA significantly decreased the loss values during both the training and testing procedures as the number of epochs increased. This result empirically verifies the theoretical analysis in the previous sections, demonstrating that Cd-DLA can deliver effective classification performance on real-world datasets

Table 6.4 : Evaluation on non-crisis data set

Experiment Setting		**F-measure**			
Domain	***Country***	***RNN***	***Id-ANN***	***Cd-ANN***	***Cd-DLA***
Currency	US	0.4040	0.4204	0.5161	0.5372
	China	0.3724	0.4986	0.5159	0.5269
	India	0.5239	0.5361	0.5299	0.5398
Stock	US	0.4115	0.4741	0.4916	0.5668
	China	0.4500	0.5108	0.6694	0.6958
	India	0.4104	0.4250	0.5927	0.6322
Experiment Setting		**AUC**			
Domain	***Country***	***RNN***	***Id-ANN***	***Cd-ANN***	***Cd-DLA***
Currency	US	0.5160	0.5468	0.5341	0.5528
	China	0.5268	0.5357	0.5459	0.6052
	India	0.5170	0.5278	0.5197	0.5960
Stock	US	0.5345	0.5541	0.5025	0.5643
	China	0.5163	0.5253	0.5085	0.5352
	India	0.5319	0.5486	0.5062	0.5573

in multiple financial markets.

In summary, this chapter presented a data mining algorithm and associated techniques for analysing domain heterogeneity in real-world financial markets. We argue that the existing machine learning approaches require more data pre-processing and lack the efficiency to model the complicated and nonlinear relationships associated with the world's financial markets, particularly the financial flows between heterogeneous domains. We proposed a novel cross-domain deep learning approach (Cd-

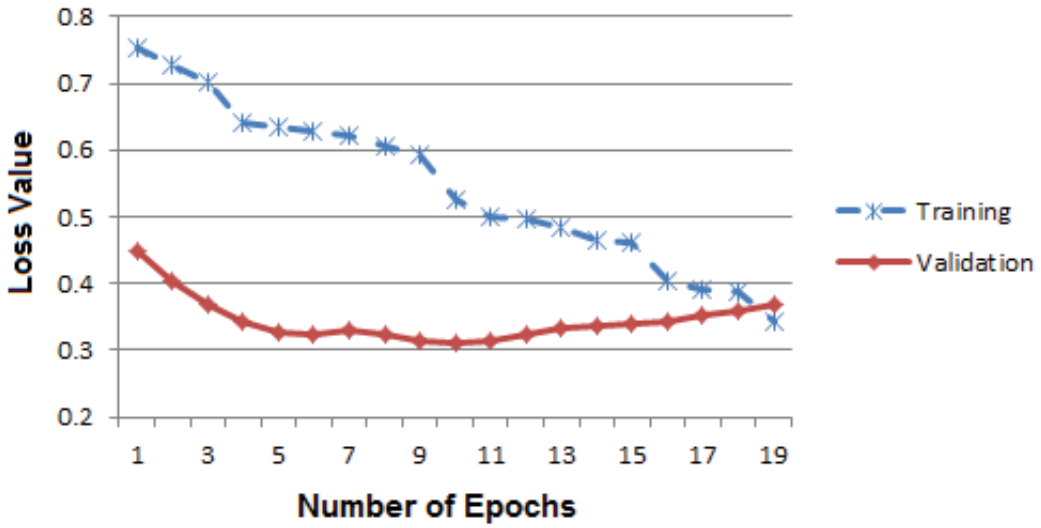

Figure 6.4 : Cd-DLA's number of epochs: Cd-DLA effectively decreases the loss value with an increase in the number of epochs on the financial crisis dataset of the multiple financial markets from currency and stock market domains, regarding forecasts for the stock market in the United States.

DLA) with a parallel MTL architecture and several attention neural networks model three types of complex correlations for multi-domain financial forecasting. The results of the experiments with two real-world financial datasets desmonstrated that Cd-DLA demonstrates superior performance over other baseline neural networks.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Effectively deriving meaningful insights from complex enterprise data and converting knowledge into business strategy remain challenging problems. First, huge volumes of data are being amassed from an increasing number of sources. Further, emerging applications, such as e-commerce and mobility, and the nature of different business processes results in a diverse range of data representations. Both the scale and the diversity of today's data are presenting significant challenges to traditional analysis methods. Hence, new methods that can overcome these issues are required to efficiently and effectively leverage information. Second, data is becoming more and more heterogeneous. In a variety of data domains, such as social networks and internet of things, enterprise data can no longer be represented by a few instance-feature tables. Rather, today's enterprise data contains complex structures that reflect dependencies, correlations, and implicit relationships. Third, learning tasks in real-world business applications has become more and more complicated, with a range of constraints on the number of labelled data, the type of class distributions that can be analysed, or the number of learning tasks that can be handled, and so on. Given these challenges, this thesis has presented a range of novel learning methods and algorithms for mining heterogeneous enterprise data with four specific types of heterogeneity: object heterogeneity, event heterogeneity, context heterogeneity, and domain heterogeneity.

Chapter 3 introduces a Cs-HNN for learning heterogeneous objects within a par-

allel network architecture, and a robust model that is generated with a specifically-designed cost-sensitive algorithm for minority classification. Experiments with real-world data demonstrate that the proposed approach demonstrates superior performance over baseline procedures.

In Chapter 4, event heterogeneity is explored through a sequential pattern mining approach that considers heterogeneous event-related factors. By combining heterogeneous fleet tracking features with several purpose-built measurement algorithms, this approach represents a practical and efficient solution for deriving valuable insights in the fleet rental industry.

Chapter 5 investigates context heterogeneity in enterprise data given the assumption of non-IIDness. The CCF approach presented in this chapter is able to generate context-aware recommendations by measuring non-IID relationships in heterogeneous business contexts.

Finally, in Chapter 6, the research focus turns to the complex and nonlinear relationships between heterogeneous domains. The correlations within and between time-series and static data across multiple domains are learned through a novel cross-domain deep learning approach (Cd-DLA) in a parallel MTL architecture. An attention mechanism, based on an RNN, then analyses these complex domain-related correlations to forecast market trends for the next trading window in multiple financial markets.

Each substantive chapter of this thesis is supported by at least one accepted or published conference/journal paper. More encouragingly, several of the approaches proposed in this thesis have been successfully applied to real-world business cases, such as insurance operation optimisation, fraud detection, customer segmentation, device management, recommendation systems, and multiple financial market analysis, with relevant papers recognised by research peers. Therefore, the contributions

in this thesis to research related to heterogeneous data mining are of great significance, and the practical implementations of this work validate its potential for real-world business applications.

7.2 Future work

All the proposed algorithms and techniques in this thesis are based on solving the challenges associated with mining heterogeneous enterprise data. However, applying data mining methods to complex, dynamic, and diverse enterprise tasks is far from straightforward and dramatically challenging. In reality, real-world enterprise data have complicated characteristics, and the level of complexity is heavily dependent on how heterogeneous objects, events, contexts, and domains are assessed and characterised. There is much work still to be done. The ongoing work and future directions for the approaches presented in this thesis, and for heterogeneous enterprise data mining in general, follows. These endeavours fall roughly into three categories: business understanding and data pre-processing, data learning and model evaluation, and knowledge presentation.

7.2.1 Business understanding and data processing

Each of the approaches presented target real-world business requirements for heterogeneous data analysis: classifying minority classes, improving learning efficiency, discovering and transferring knowledge across domains, or finding hidden patterns in nonlinear data. Investigations currently in progress focus on enriching the parallel architectures used to support these approaches for minority classifications with semi-structured non-IID data. Future work will seek to extend this work to classification tasks, such as fraud detection, with real-world Big Data.

7.2.2 Data learning and model evaluation

A range of machine learning techniques and deep learning algorithms were applied to heterogeneous data mining in new ways in this thesis, including neural networks, sequential pattern mining, attention mechanisms, MTL, transfer learning, non-IID algorithms, and nonlinear algorithms. Each of these techniques provided some measure of improvement to dynamic data-driven business processes but, more importantly, new insights into innovative ways of expanding and repurposing traditional techniques to new problems. Hence, future efforts will extend this range of techniques to include multi-modality and large-scale reinforcement learning to further improve business outcomes and performance.

7.2.3 Knowledge presentation

The results and insights derived from the approaches in this thesis are currently presented in relatively simplistic formats, such as reports, tables, and graphs. While these presentation styles meet some business requirements, there are some possibilities for improving the ease, acuity, and interpretability of the knowledge gained. There are also lots of opportunities to develop more sophisticated tools to explore and present insights from heterogeneous enterprise data, such as intelligent platforms that can interact autonomously with diverse business systems, explainable AI, and general AI with a focus on enterprise data-driven insights.

Bibliography

[1] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 191–226.

[2] B. Agarwal and N. Mittal, "Hybrid approach for detection of anomaly network traffic using data mining techniques," *Procedia Technology*, vol. 6, pp. 996–1003, 2012.

[3] C. F. Ahmed, S. K. Tanbeer, and B.-S. Jeong, "A novel approach for mining high-utility sequential patterns in sequence databases," *ETRI journal*, vol. 32, no. 5, pp. 676–686, 2010.

[4] J. E. Andriesson and R. A. Roe, *Telematics and work*. Psychology Press, 2013.

[5] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential pattern mining using a bitmap representation," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 429–435.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[7] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[8] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 17–36.

[9] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.

[10] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 440–447.

[11] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2006, pp. 120–128.

[12] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models for providing context-aware services," in *International Conference on Universal access in human-computer interaction*. Springer, 2007, pp. 23–32.

[13] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2009, pp. 475–482.

[14] R. Burke, F. Vahedian, and B. Mobasher, "Hybrid recommendation in heterogeneous networks," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2014, pp. 49–60.

[15] L. Cao, "Non-iidness learning in behavioral and social data," *The Computer Journal*, vol. 57, no. 9, pp. 1358–1370, 2013.

[16] ——, "Data science: Challenges and directions," *Communications of the ACM*, vol. 60, no. 8, pp. 59–68, 2017.

[17] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[18] S. Castano and V. De Antonellis, "Global viewing of heterogeneous data sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 2, pp. 277–297, 2001.

[19] R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, J. P. Nobrega, and A. L. Oliveira, "Computational intelligence and financial markets: A survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[20] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[21] J. H. Chang, "Mining weighted sequential patterns in a sequence database with a time-interval weight," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 1–9, 2011.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[23] A. Chen, "Context-aware collaborative filtering system: Predicting the users preference in the ubiquitous computing environment," in *International*

Symposium on Location-and Context-Awareness. Springer, 2005, pp. 244–253.

[24] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: literature review and challenges," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 431047, 2015.

[25] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2823–2824.

[26] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.

[27] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[29] Y.-A. Chung, H.-T. Lin, and S.-W. Yang, "Cost-aware pre-training for multiclass cost-sensitive deep learning," *arXiv preprint arXiv:1511.09337*, 2015.

[30] B. P. Clarkson, "Life patterns: structure from wearable sensors," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[31] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 210–219.

[32] ——, "Transferring naive bayes classifiers for text classification," in *AAAI*, vol. 7, 2007, pp. 540–545.

[33] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 193–200.

[34] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 217–224.

[35] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2017.

[36] L. Di Persio and O. Honchar, "Recurrent neural networks approach to the financial forecast of google assets," *International journal of Mathematics and Computers in simulation*, vol. 11, 2017.

[37] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction." in *Ijcai*, 2015, pp. 2327–2333.

[38] F. Emmert-Streib, R. de Matos Simoes, G. Glazko, S. McDade, B. Haibe-Kains, A. Holzinger, M. Dehmer, and F. C. Campbell, "Functional and genetic analysis of the colon cancer network," *BMC bioinformatics*, vol. 15, no. 6, p. S6, 2014.

[39] W. Fan, I. Davidson, B. Zadrozny, and P. S. Yu, "An improved categorization of classifier's sensitivity on sample selection bias," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 4–pp.

[40] D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining naive bayes and

decision tree for adaptive intrusion detection," *arXiv preprint arXiv:1005.4496*, 2010.

[41] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[42] U. Fayyad and R. Uthurusamy, "Evolving data into mining solutions for insights," *Communications of the ACM*, vol. 45, no. 8, pp. 28–31, 2002.

[43] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, 2017.

[44] R. D. Foreman, "A logistic analysis of bankruptcy within the us local telecommunications industry," *Journal of Economics and Business*, vol. 55, no. 2, pp. 135–166, 2003.

[45] P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, "Fast vertical mining of sequential patterns using co-occurrence information," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 40–52.

[46] P. Fournier-Viger, R. Nkambou, and E. M. Nguifo, "A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems," in *Mexican International Conference on Artificial Intelligence*. Springer, 2008, pp. 765–778.

[47] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. Siam, 2007, vol. 20.

[48] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[49] M. Gao, X. Hong, and C. J. Harris, "Construction of neurofuzzy models for imbalanced data classification," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1472–1488, 2014.

[50] M. N. Garofalakis, R. Rastogi, and K. Shim, "Spirit: Sequential pattern mining with regular expression constraints," in *VLDB*, vol. 99, 1999, pp. 7–10.

[51] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[52] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Embedding heterogeneous data using statistical models," in *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, vol. 21, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1605.

[53] M. J. Greenacre, "Theory and applications of correspondence analysis," 1984.

[54] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10 389–10 397, 2011.

[55] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International Conference on Intelligent Computing*. Springer, 2005, pp. 878–887.

[56] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[57] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *proceedings of the 17th international conference on data engineering*, 2001, pp. 215–224.

[58] M. R. Hassan and B. Nath, "Stock market forecasting using hidden markov model: a new approach," in *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on.* IEEE, 2005, pp. 192–196.

[59] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[61] M. Hu, Y. Peng, and X. Qiu, "Mnemonic reader for machine comprehension," *arXiv preprint arXiv:1705.02798*, 2017.

[62] R. Hu, C. P. Yu, S.-F. Fung, S. Pan, H. Wang, and G. Long, "Universal network representation for heterogeneous information networks," in *Neural Networks (IJCNN), 2017 International Joint Conference on.* IEEE, 2017, pp. 388–395.

[63] G.-B. Huang, "Reply to comments on the extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1495–1496, 2008.

[64] G.-B. Huang, L. Chen, C. K. Siew *et al.*, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[65] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[66] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2007, pp. 601–608.

[67] K. Huang, H. Yang, I. King, and M. R. Lyu, "Learning classifiers from imbalanced data based on biased minimax probability machine," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–II.

[68] Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5081–5090.

[69] T. N. Huy, H. Shao, B. Tong, and E. Suzuki, "A feature-free and parameter-light multi-task clustering framework," *Knowledge and information systems*, vol. 36, no. 1, pp. 251–276, 2013.

[70] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Classification of imbalanced data by combining the complementary neural network and smote algorithm," in *International Conference on Neural Information Processing*. Springer, 2010, pp. 152–159.

[71] J. Jiang, J. Lu, G. Zhang, and G. Long, "Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop," in *Services*, 2011, pp. 490–497.

[72] ——, "Optimal cloud resource auto-scaling for web applications," in *Ieee/acm International Symposium on Cluster, Cloud and Grid Computing*, 2013, pp. 58–65.

[73] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 264–271.

[74] X. Jiang, W. Liu, L. Cao, and G. Long, "Coupled collaborative filtering for context-aware recommendation." in *AAAI*, 2015, pp. 4172–4173.

[75] X. Jiang, S. Pan, J. Jiang, and G. Long, "Cross-domain deep learning approach for multiple financial market prediction," in *Neural Networks (IJCNN), 2018 International Joint Conference on*. IEEE, 2018, pp. 1–8.

[76] X. Jiang, S. Pan, G. Long, J. Chang, J. Jiang, and C. Zhang, "Cost-sensitive hybrid neural networks for heterogeneous and imbalanced data," in *Neural Networks (IJCNN), 2018 International Joint Conference on*. IEEE, 2018, pp. 1–8.

[77] X. Jiang, S. Pan, G. Long, F. Xiong, J. Jiang, and C. Zhang, "Cost-sensitive parallel learning framework for insurance intelligence operation," *Transactions on Industrial Electronics*, pp. 1–11, 2018.

[78] X. Jiang, X. Peng, and G. Long, "Discovering sequential rental patterns by fleet tracking," in *International Conference on Data Science*. Springer, 2015, pp. 42–49.

[79] D. I. Kaplan and C. Rieser, *Service Success! Lessons from a Leader on How to Turn Around a Service Business*. John Wiley & Sons, 1994.

[80] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware

collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[81] Z. Khorshidpour, J. Tahmoresnezhad, S. Hashemi, and A. Hamzeh, "Domain invariant feature extraction against evasion attack," *International Journal of Machine Learning and Cybernetics*, pp. 1–12, 2017.

[82] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," *arXiv preprint arXiv:1702.00887*, 2017.

[83] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[84] K. Kumar, P. Srinivas, and C. R. Rao, "Sequential pattern mining with multiple minimum supports by ms-spade," *International Journal of Computer Sciences*, vol. 9, no. 5, pp. 61–73, 2012.

[85] K. M. Kumar, P. Srinivas, and C. R. Rao, "Sequential pattern mining with multiple minimum supports in progressive databases," *International Journal of Database Management Systems*, vol. 4, no. 4, p. 29, 2012.

[86] R. Kumaraswamy, P. Odom, K. Kersting, D. Leake, and S. Natarajan, "Transfer learning via relational type matching," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 811–816.

[87] E. K. Laitinen and T. Laitinen, "Bankruptcy prediction: Application of the taylor's expansion in logistic regression," *International review of financial analysis*, vol. 9, no. 4, pp. 327–349, 2000.

[88] G.-C. Lan, T.-P. Hong, V. S. Tseng, and S.-L. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5071–5081, 2014.

[89] D. T. Larose and C. D. Larose, *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.

[90] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 65.

[91] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[92] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[93] A. Levy, A. Rajaraman, and J. Ordille, "Querying heterogeneous information sources using source descriptions," Stanford InfoLab, Tech. Rep., 1996.

[94] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted elm for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[95] S. Li, Z.-Q. Liu, and A. B. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 482–489.

[96] Y. Li and W. Ma, "Applications of artificial neural networks in financial economics: a survey," in *Computational Intelligence and Design (ISCID), 2010 International Symposium on*, vol. 1. IEEE, 2010, pp. 211–214.

[97] V. C.-C. Liao and M.-S. Chen, "Dfsp: a depth-first spelling algorithm for sequential pattern mining of biological sequences," *Knowledge and information systems*, vol. 38, no. 3, pp. 623–639, 2014.

[98] T. Lin, T. Guo, and K. Aberer, "Hybrid neural networks over time series for trend forecasting," 2017.

[99] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[100] C. Liu, L. Cao, and S. Y. Philip, "Coupled fuzzy k-nearest neighbors classification of imbalanced non-iid categorical data," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 1122–1129.

[101] F. Liu, G. Lin, and C. Shen, "Crf learning with cnn features for image segmentation," *Pattern Recognition*, vol. 48, no. 10, pp. 2983–2992, 2015.

[102] T. Liu, Q. Yang, and D. Tao, "Understanding how feature structure transfers in transfer learning," in *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, 2017, pp. 2365–2371.

[103] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, "Multi-task deep visual-semantic embedding for video thumbnail selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3707–3715.

[104] G. Long, L. Chen, X. Zhu, and C. Zhang, "Tcsst: transfer classification of short and sparse text using external data," in *ACM International Conference on Information and Knowledge Management*, 2012, pp. 764–772.

[105] M. Long, J. Wang, J. Sun, and S. Y. Philip, "Domain invariant transfer kernel learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1519–1532, 2015.

[106] A. J. Ma, J. Li, P. C. Yuen, and P. Li, "Cross-domain person reidentification using domain adaptation ranking svms," *IEEE transactions on image processing*, vol. 24, no. 5, pp. 1599–1613, 2015.

[107] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of smote for mining imbalanced data," in *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*. IEEE, 2011, pp. 104–111.

[108] A. K. McCallum *et al.*, "Learning to use selective attention and short-term memory in sequential tasks," in *From animals to animats 4: proceedings of the fourth international conference on simulation of adaptive behavior*, vol. 4. MIT Press, 1996, p. 315.

[109] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang, "Automatic analysis of multimodal group actions in meetings," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 3, pp. 305–317, 2005.

[110] L. Mihalkova and R. J. Mooney, "Transfer learning by mapping with minimal target data," in *Proceedings of the AAAI-08 workshop on transfer learning for complex tasks*, 2008.

[111] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[112] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[113] R. Mollineda, R. Alejo, and J. Sotoca, "The class imbalance problem in pattern classification and learning," in *II Congreso Español de Informática*

(CEDI 2007). ISBN, 2007, pp. 978–84.

[114] C. H. Mooney and J. F. Roddick, "Sequential pattern mining–approaches and algorithms," *ACM Computing Surveys (CSUR)*, vol. 45, no. 2, p. 19, 2013.

[115] M. Muhlenbrock, O. Brdiczka, D. Snowdon, and J.-L. Meunier, "Learning to detect user activity and availability from a variety of sensor data," in *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on.* IEEE, 2004, pp. 13–22.

[116] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[117] X.-X. Niu and C. Y. Suen, "A novel hybrid cnn–svm classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[118] D. L. Olson and D. D. Wu, "Data mining models and enterprise risk management," in *Enterprise Risk Management Models.* Springer, 2017, pp. 119–132.

[119] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder," *arXiv preprint arXiv:1802.04407*, 2018.

[120] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Boosting for graph classification with universum," *Knowledge and Information Systems*, vol. 50, no. 1, pp. 1–25, 2017.

[121] ——, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 1–15, 2017.

[122] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Network*, vol. 11, no. 9, p. 12, 2016.

[123] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[124] Y. Pao, "Adaptive pattern recognition and neural networks," 1989.

[125] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2017.

[126] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.

[127] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[128] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, "Gene functional classification from heterogeneous data," in *Proceedings of the fifth annual international conference on Computational biology*. ACM, 2001, pp. 249–255.

[129] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of network and computer applications*, vol. 30, no. 1, pp. 114–132, 2007.

[130] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Transactions on knowledge and data engineering*, vol. 16, no. 11, pp. 1424–1440, 2004.

[131] J. Pei, J. Han, and W. Wang, "Constraint-based sequential pattern mining: the pattern-growth methods," *Journal of Intelligent Information Systems*, vol. 28, no. 2, pp. 133–160, 2007.

[132] C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: classification of skewed data," *Acm sigkdd explorations newsletter*, vol. 6, no. 1, pp. 50–59, 2004.

[133] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, "Multi-dimensional sequential pattern mining," in *Proceedings of the tenth international conference on Information and knowledge management.* ACM, 2001, pp. 81–88.

[134] M. N. Quang, T. Dinh, U. Huynh, and B. Le, "Mhhusp: An integrated algorithm for mining and hiding high utility sequential patterns," in *Knowledge and Systems Engineering (KSE), 2016 Eighth International Conference on.* IEEE, 2016, pp. 13–18.

[135] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning.* The MIT Press, 2009.

[136] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning.* ACM, 2007, pp. 759–766.

[137] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proceedings of the 23rd international conference on Machine learning.* ACM, 2006, pp. 713–720.

[138] V. Raj, S. Magg, and S. Wermter, "Towards effective classification of imbalanced data with convolutional neural networks," in *IAPR Workshop on*

Artificial Neural Networks in Pattern Recognition. Springer, 2016, pp. 150–162.

[139] M. Rajesh and J. Gnanasekar, "Annoyed realm outlook taxonomy using twin transfer learning," *International Journal of Pure and Applied Mathematics*, vol. 116, pp. 547–558, 2017.

[140] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory," *Knowledge and information systems*, vol. 33, no. 2, pp. 245–265, 2012.

[141] M. Rudolph, F. Ruiz, S. Mandt, and D. Blei, "Exponential family embeddings," in *Advances in Neural Information Processing Systems*, 2016, pp. 478–486.

[142] A. Schwaighofer, V. Tresp, and K. Yu, "Learning gaussian process kernels via hierarchical bayes," in *Advances in neural information processing systems*, 2005, pp. 1209–1216.

[143] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[144] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," 2017.

[145] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang, "Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling," 2018.

[146] T. Shen, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Bi-directional block self-attention for fast and memory-efficient sequence modeling," 2018.

[147] Q. Shi, B. Du, and L. Zhang, "Domain adaptation for remote sensing image classification: A low-rank reconstruction and instance weighting label propagation inspired algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 10, pp. 5677–5689, 2015.

[148] Y. Shi, W. Li, Y. Gao, L. Cao, and D. Shen, "Beyond iid: Learning to combine non-iid metrics for vision tasks." in *AAAI*, 2017, pp. 1524–1531.

[149] D. Shin, J.-w. Lee, J. Yeon, and S.-g. Lee, "Context-aware recommendation by aggregating user context," in *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on.* IEEE, 2009, pp. 423–430.

[150] S. Si, D. Tao, and K.-P. Chan, "Evolutionary cross-domain discriminative hessian eigenmaps," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1075–1086, 2010.

[151] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.

[152] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A survey of the connected vehicle landscape–architectures, enabling technologies, applications, and development areas," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[153] P. Songram and V. Boonjing, "Closed multidimensional sequential pattern mining," *International Journal of Knowledge Management Studies*, vol. 2, no. 4, pp. 460–479, 2008.

[154] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *International Conference on Extending Database Technology.* Springer, 1996, pp. 1–17.

[155] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.

[156] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[157] Y. Sun and J. Han, "Meta-path-based search and mining in heterogeneous information networks," *Tsinghua Science and Technology*, vol. 18, no. 4, pp. 329–338, 2013.

[158] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[159] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 2009, pp. 565–576.

[160] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[161] L. Takeuchi and Y.-Y. A. Lee, "Applying deep learning to enhance momentum trading strategies in stocks," in *Technical Report*. Stanford University, 2013.

[162] J. Tang, C. Deng, G.-B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1174–1185, 2015.

[163] P. Tino, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.

[164] T. E. Trimble, D. S. Bowman *et al.*, "Market guide to fleet telematics services: Creating a consumer's guide to currently available aftermarket solutions," Virginia Tech. Virginia Tech Transportation Institute, Tech. Rep., 2012.

[165] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L. T. Yang *et al.*, "Data mining for internet of things: A survey." *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[166] C. Turkay, F. Jeanquartier, A. Holzinger, and H. Hauser, "On computationally-enhanced visual analysis of heterogeneous data and its application in biomedical informatics," in *Interactive knowledge discovery and data mining in biomedical informatics*. Springer, 2014, pp. 117–140.

[167] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[168] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and information systems*, vol. 25, no. 1, pp. 1–20, 2010.

[169] B. Wang, M. Liakata, A. Zubiaga, R. Procter, and E. Jensen, "Smile: Twitter emotion classification using domain adaptation," in *25th International Joint Conference on Artificial Intelligence*, 2016, p. 15.

[170] C. Wang, L. Cao, M. Wang, J. Li, W. Wei, and Y. Ou, "Coupled nominal similarity in unsupervised learning," in *Proceedings of the 20th ACM*

international conference on Information and knowledge management. ACM, 2011, pp. 973–978.

[171] C. Wang, Z. She, and L. Cao, "Coupled attribute analysis on numerical data," in *International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2013.

[172] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "Mgae: Marginalized graph autoencoder for graph clustering," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 889–898.

[173] ——, "Mgae: Marginalized graph autoencoder for graph clustering," in *ACM on Conference on Information and Knowledge Management*, 2017, pp. 889–898.

[174] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 2285–2294.

[175] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2016, pp. 225–230.

[176] K.-J. Wang, B. Makond, K.-H. Chen, and K.-M. Wang, "A hybrid classifier combining smote with pso to estimate 5-year survivability of breast cancer patients," *Applied Soft Computing*, vol. 20, pp. 15–24, 2014.

[177] L. P. Wang and C. R. Wan, "Comments on "the extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1494–1495, 2008.

[178] S. Wang, X. Li, X. Chang, L. Yao, Q. Z. Sheng, and G. Long, "Learning multiple diagnosis codes for icu patients with local disease correlation mining," *Acm Transactions on Knowledge Discovery from Data*, vol. 11, no. 3, p. 31, 2017.

[179] S. Wang and Z. Li, "A new transfer learning boosting approach based on distribution measure with an application on facial expression recognition," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 432–439.

[180] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[181] G. Wu and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," in *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, 2003, pp. 49–56.

[182] Y. Yan, Z. Xu, I. W. Tsang, G. Long, and Y. Yang, "Robust semi-supervised learning through label aggregation," in *AAAI*, 2016.

[183] X. Yang, T. Zhang, and C. Xu, "Cross-domain feature learning in multimedia," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 64–78, 2015.

[184] Z. Yang and M. Kitsuregawa, "Lapin-spam: An improved algorithm for mining sequential pattern," in *Data Engineering Workshops, 2005. 21st International Conference on*. IEEE, 2005, pp. 1222–1222.

[185] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[186] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[187] Y. Yasami and S. P. Mozaffari, "A novel unsupervised classification approach for network anomaly detection by k-means clustering and id3 decision tree learning methods," *The Journal of Supercomputing*, vol. 53, no. 1, pp. 231–245, 2010.

[188] J. Yin, Z. Zheng, and L. Cao, "Uspan: an efficient algorithm for mining high utility sequential patterns," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2012, pp. 660–668.

[189] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining.* ACM, 2014, pp. 283–292.

[190] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *Proceedings of the 7th ACM conference on Recommender systems.* ACM, 2013, pp. 347–350.

[191] U. Yun and J. J. Leggett, "Wspan: Weighted sequential pattern mining in large sequence databases," in *Intelligent Systems, 2006 3rd International IEEE Conference on.* IEEE, 2006, pp. 512–517.

[192] B. Zadrozny, "Learning and evaluating classifiers under sample selection

bias," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 114.

[193] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine learning*, vol. 42, no. 1-2, pp. 31–60, 2001.

[194] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[195] Q. Zhang, J. Wu, H. Yang, W. Lu, G. Long, and C. Zhang, "Global and local influence-based social recommendation," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1917–1920.

[196] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," *IEEE Transactions on Big Data*, 2016.

[197] Y. Zhang, P. Fu, W. Liu, and G. Chen, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 927–935, 2014.

[198] ——, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 927–935, 2014.

[199] Y. Zhang and D. Wang, "A cost-sensitive ensemble method for class-imbalanced datasets," in *Abstract and applied analysis*, vol. 2013. Hindawi, 2013.

[200] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by

deep multi-task learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 94–108.

[201] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: A survey," *ITechnical Report CAIS Nayang Technological University Singapore*, vol. 1, p. 26, 2003.

[202] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach." in *AAAI*, vol. 10, 2010, pp. 236–241.

[203] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 58, 2015.

[204] S. Zida, P. Fournier-Viger, C.-W. Wu, J. C.-W. Lin, and V. S. Tseng, "Efficient mining of high-utility sequential rules," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2015, pp. 157–171.