Deep Learning for Uncertainty Measurement

Doctoral Thesis

to acquire the academic degree of
doctor rerum politicarum
(Doctor of Economics and Management Science)

submitted to

School of Business and Economics
Humboldt-Universität zu Berlin

by

M.Sc. Alisa Kim

President of Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dean of the School of Business and Economics:
Prof. Dr. Daniel Klapper

Reviewers:
1. Prof. Dr. Stefan Lessmann
2. Prof. Dr. Wolfgang K.Härdle

Date of Colloquium: 19 Okt 2020

Abstract

In 2016 the media announced the beginning of the "era of uncertainty" with the newly elected president Trump and years to follow brought on a surge of nationalism, trade wars, and civil unrest. As the pandemic of COVID-19 unraveled, the term "chronic economic uncertainty" came into play (Cassidy, 2020), manifesting uncertainty as to the "new normal." Economists refer to such long-lasting state as Knightian uncertainty, after Frank Knight, the noted Chicago economist of the early twentieth century. Knight Frank (1921) distinguished between risks that can be calculated, such as the chances of rolling two sixes or winning the lottery, and risks that are so complex and hard to decipher that they "are not susceptible to measurement." Economists, finance experts, and scholars have been designing the tools to combat the former, while the looming shade of the "non-quantifiable" drove the adoption of new methods. With volatility being a traditional "go-to" tool for financial markets, macro uncertainty quantification presents a complex task with persistent limitations, with Doran (1999) offering a thorough argumentation towards non-linearity of events development as a significant impediment to forecasting powers of uncertainty modeling.

This thesis focuses on solving the problem of uncertainty measurement and its impact on business decisions while pursuing two goals: first, develop and validate accurate and robust models for uncertainty quantification, employing both the well established statistical models and newly developed machine learning tools, with particular focus on deep learning. The second goal revolves around the industrial application of proposed models, applying them to real-world cases when measuring volatility or making a risky decision entails a direct and substantial gain or loss.

This thesis started with the exploration of implied volatility (IV) as a proxy for investors' perception of uncertainty for a new class of assets - crypto-currencies. They represent a compelling case given high velocity and a growing rate of adoption with the absence of a developed derivative market that usually supplies the IV measurement from derivative prices. The newly constructed VCRIX index captured the investor sentiments and exposed excessive volatility that presumably stems from the behavioral component of option pricing.

The second paper focused on methods to identify risk-loving traders and employed the DNN infrastructure for it to investigate further the risk-taking behavior of market actors that both stems from and perpetuates uncertainty. The results demonstrated that improvements in forecast accuracy translate into sizable increases in operating profit and confirmed the proposed DNN to effectively support (hedging) decision making and grasp the behavioral component from data.

The third paper addressed the challenging endeavor of fraud detection and offered the decision support model that allowed a more accurate and interpretable evaluation of financial reports submitted for audit. The findings indicated that the DL model is well-suited to correctly identify fraudulent cases, specifically in the highly unbalanced case of fraud detection.

The insight NLP backed by DL could distill from textual input motivated the development of the fourth paper that concludes this thesis to find a way to quantify uncertainty on a macro level and analyze its drivers. Following the importance of risk assessment and agents' expectations in economic development and building on the existing works of Baker et al. (2016) and their economic policy uncertainty (EPU) index, it offered a novel DL-NLP-based method for the quantification of economic policy uncertainty. The approach showed pathways towards capturing economic policy uncertainty over long periods while keeping track of changes in the way that news and uncertainty are reported.

In summary, this thesis offers insights that are highly relevant to both researchers and practitioners. The new deep learning-based solutions exhibit superior performance to existing approaches to quantify and explain economic uncertainty, allowing for more accurate forecasting, enhanced planning capacities, and mitigated risks. Deep Learning component makes these solutions "future-proof" by offering tools to deal with ever-increasing amounts of data and various data types. The offered use-cases provide a road-map for further development of the DL tools in practice and constitute a platform for further research.

Keywords: deep learning, NLP, uncertainty, volatility

Zusammenfassung

2016 kündigten die Medien mit dem neu gewählten Präsidenten Trump den Beginn der "Ära der Unsicherheit" an, und die folgenden Jahre führten zu einem Anstieg des Nationalismus, Handelskriegen und Unruhen. Als die COVID-19 Pandemie began, kam der Begriff "chronische wirtschaftliche Unsicherheit" ins Spiel parencite newyorker und etablierte Unsicherheit als „neue Normalität". Wirtschaftswissenschaftler bezeichnen einen so lang anhaltenden Zustand als knightianische Unsicherheit nach Frank Knight, dem bekannten Chicagoer Ökonomen des frühen 20. Jahrhunderts. Knight Frank (1921) unterschied zwischen Risiken, wie z. B. die Chancen, zwei Sechser zu würfeln und Risiken, die so komplex zu entziffern sind, dass sie "nicht messbar sind".

Wirtschaftswissenschaftler, Finanzexperten und Wissenschaftler haben Instrumente zur Bekämpfung der ersteren entwickelt, während der sich abzeichnende Schatten des "nicht quantifizierbaren" die Einführung neuer Methoden vorantreibt. Da die Volatilität ein traditionelles "go-to" -Instrument für die Finanzmärkte ist, stellt die Quantifizierung der Makrounsicherheit eine komplexe Aufgabe mit anhaltenden Einschränkungen dar. Doran (1999) bietet eine gründliche Argumentation für die Nichtlinearität der Ereignisentwicklung als erhebliches Hindernis für die Prognosefähigkeit der Unsicherheitsmodellierung.

Diese Arbeit konzentriert sich auf die Lösung des Problems der Unsicherheitsmessung und ihrer Auswirkungen auf Geschäftsentscheidungen, wobei zwei Ziele verfolgt werden: Erstens die Entwicklung und Validierung robuster Modelle zur Quantifizierung der Unsicherheit, wobei insbesondere sowohl die etablierten statistischen Modelle als auch neu entwickelte maschinelle Lernwerkzeuge zum Einsatz kommen.

Das zweite Ziel dreht sich um die industrielle Anwendung der vorgeschlagenen Modelle. Die Anwendung auf reale Fälle bei der Messung der Volatilität oder bei einer riskanten Entscheidung ist mit einem direkten und erheblichen Gewinn oder Verlust verbunden.

Diese These begann mit der Untersuchung der impliziten Volatilität (IV) als Proxy für die Wahrnehmung der Unsicherheit von Anlegern für eine neue Klasse von Vermögenswerten - Kryptowährungen. Sie stellen angesichts der hohen Geschwindigkeit und der wachsenden Akzeptanzrate einen überzeugenden Fall dar, da kein entwickelter Derivatemarkt vorhanden ist, der normalerweise die IV-Messung aus Derivatpreisen liefert. Der neu erstellte VCRIX-Index hat die Anlegerstimmung erfasst und eine übermäßige Volatilität aufgedeckt, die vermutlich auf die Verhaltenskomponente der Optionspreise zurückzuführen ist.

Das zweite Papier konzentriert sich auf Methoden zur Identifizierung risikofreudiger Händler und nutzt die DNN-Infrastruktur, um das Risikoverhalten von Marktakteuren, das auf Unsicherheit beruht und diese aufrechterhält, weiter zu untersuchen. Die Ergebnisse zeigten, dass Verbesserungen der Prognosegenauigkeit zu erheblichen Steigerungen des Betriebsgewinns führen, und bestätigten die vorgeschlagene DNN-Infrastruktur, um die Entscheidungsfindung effektiv zu unterstützen (abzusichern) und die Verhaltenskomponente aus Daten zu erfassen.

Das dritte Papier befasste sich mit dem herausfordernden Bestreben der Betrugserkennung 3 und bot das Entscheidungshilfe-modell, das eine genauere und interpretierbarere Bewertung der zur Prüfung eingereichten Finanzberichte ermöglichte. Die Ergebnisse zeigten, dass das Deep Learning-Modell gut geeignet ist, betrügerische Fälle korrekt zu identifizieren, insbesondere im sehr unausgewogenen Fall der Betrugserkennung.

Die von Deep Learning unterstützte Erkenntnis NLP könnte aus Texteingaben destillieren und die Entwicklung eines vierten Papiers motivieren, das diese These abschließt, um einen Weg zu finden, die Unsicherheit auf Makroebene zu quantifizieren und ihre Treiber zu analysieren. Angesichts der Bedeutung der Risikobewertung und der Erwartungen der Agenten für die wirtschaftliche Entwicklung und des Aufbaus der bestehenden Arbeiten von Baker et al. (2016) und ihres Index der wirtschaftspolitischen Unsicherheit (EPU) bot es eine neuartige DL-NLP-basierte Methode zur Quantifizierung der wirtschaftspolitischen Unsicherheit. Der Ansatz zeigte Wege auf, um die wirtschaftspolitische Unsicherheit über lange Zeiträume hinweg zu erfassen und gleichzeitig Änderungen in der Art und Weise zu verfolgen, in der Nachrichten und Unsicherheiten gemeldet werden.

Zusammenfassend bietet diese Arbeit Erkenntnisse, die sowohl für Forscher als auch Praktiker von hoher Relevanz sind. Die neuen Deep-Learning-basierten Lösungen bieten eine überlegene Leistung gegenüber bestehenden Ansätzen zur Quantifizierung und Erklärung wirtschaftlicher Unsicherheiten und ermöglichen genauere Prognosen, verbesserte Planungskapazitäten und geringere Risiken. Die Deep Learning-Komponente macht diese Lösungen "zukunftssicher", indem sie Tools für den Umgang mit immer mehr Datenmengen und verschiedenen Datentypen bietet. Die angebotenen Anwendungsfälle bieten einen Fahrplan für die Weiterentwicklung der DL-Tools in der Praxis und bilden eine Plattform für die weitere Forschung.

Schlüsselwörter: deep learning, NLP, Unsicherheit, Volatilität

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Dr. Stefan Lessmann, for inspiration, guidance, and support. Working with him taught me to uphold high standards while always striving to innovate.

Prof. Dr. Wolfgang K.Härdle had a great impact on my journey at Humboldt University, I am most grateful for the opportunity to learn from him.

I was blessed with a marvelous team at IRTG and at Wirtschaftsinformatik Chair, they made this journey worthwhile, and I can not thank them enough. My special gratitude goes to Dr Alona Zharova - her kind advice helped me navigate the seas of academia more than once.

I am endlessly grateful to my family, and of course, to my wonderful husband Min-Sung, who knew not to close my computer while it was running the models through the night.

Contents

List of Figures

Abbreviations

ANN	Artificial Neural Network
AUC	Area under the Receiver-Operating-Characteristic-Curve
BBD	Baker, Bloom and Davis
CC	Crypto-currency
CNN	Convolutional Neural Network
CRIX	Crypto-currency Index
DL	Deep Learning
DNN	Deep Neural Network
EPU	Economic Policy Uncertainty
LASSO	Least absolute shrinkage and selection operator
LR	Linear Regression
LSTM	Long Short Term Memory Cell
ML	Machine Learning
NLP	Natural Language Processing
P&L	Profit and Loss
RF	Random Forest
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
SMOTE	Synthetic minority oversampling technique
STX	name of an anonymous trading company
SVM	Support Vector Machine
VCRIX	Volatility crypto-currency index

1 Introduction

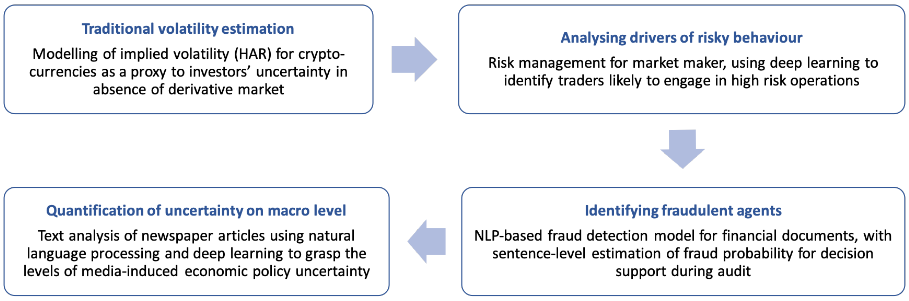
The Greek philosopher Thales of Miletus is told to be the first documented person ever to benefit from the economic uncertainty consciously. Crawford and Sen (1996) reference Aristotle by telling the story of Thales using a sort of an ancient derivative on olive harvest to gain profits. Instability has been the one guaranteed event for centuries, and people studied it to enhance survival chances. With introduction of democratic regimes, mitigation of unstable environment becomes more of state concern. Back at the beginning of the 20th century Lavington (1912) pointed out that "incessant change is followed by incessant readaptation, and the cost of imperfect foresight is continuous maladjustment of resources - a continuous social waste". The effects of economic uncertainty stretch wide, affecting vital processes from unemployment (Parker, 1996) to fertility rates (Kohler & Kohler, 2002).

In 2016 the media announced the beginning of the "era of uncertainty" with the newly elected president Trump and years to follow brought on a surge of nationalism, trade wars, and civil unrest. As the pandemic of COVID-19 unraveled, the term "chronic economic uncertainty" came into play (Cassidy, 2020), manifesting uncertainty as the "new normal". Economists refer to such long-lasting state as Knightian uncertainty, after Frank Knight, the noted Chicago economist of the early twentieth century. Knight Frank (1921) distinguished between risks that can be calculated, such as the chances of rolling two sixes or winning the lottery, and risks that are so complex and hard to decipher that they "are not susceptible to measurement." Economists, finance experts, and scholars have been designing the tools to combat the former, while the looming shade of the "non-quantifiable" drove the adoption of new methods.

Gollier (2018) offered a major research review focusing on the economics of risk and time, in particular decisions under uncertainty and asset pricing. Most of the theories in the field rely on the same principle - the utility maximization (Hey, 1996). The concept of utility, however, remains dynamic on its own. The derivation of profit is a classic approach, putting volatility at the center of financial modeling under uncertainty (Markowitz, 1952; Odean, 1998). A higher-level economic perspective considers other indicators worth optimising for, from unemployment (Caggiano et al., 2014) and demographic development (Bohn, 2001) to innovation (Teece et al., 2016) and environmental impact (Freel, 2005). All of the mentioned papers agree that uncertainty quantification presents a complex task with persistent limitations, with Doran (1999) offering a thorough argumentation towards non-linearity of events development as a major impediment to forecasting powers of uncertainty modeling. Smith (2013) provided a methodological overview of uncertainty quantification efforts, revealing an expected domination of the traditional statistical models. One of the goals of this thesis is to explore and prove the capacity of the novel machine learning, particularly deep learning (DL) methods to capture and robustly quantify the concept of uncertainty.

DL is a subset of machine learning primarily based on the hierarchical approach, where each step converts information from the previous step into more complex representations of the data (Goodfellow et al., 2016). DL (also defined as a multiple-layered Artificial Neural Network (ANN) (L. Deng & Yu, 2014)) methodology aims at learning multiple levels of representations from data, with higher levels reflecting more abstract concepts, thus capturing the complex relations between the data set features (A. Kim et al., 2020). This ability made DL a popular solution for a wide range of modeling tasks. However, the adoption of DL methods in scientific areas like economics was limited by the necessary computational capacities and interpretability issues. Neural networks notoriously represent a 'black box' - a shortcoming originating of its inherent internal complexity (Gilpin et al., 2018). Regardless these limitations DL has been extensively applied in most research areas: finance (J. Heaton et al., 2016), education (Warburton, 2003), policy evaluation (J. Sirignano et al., 2016), economic indicators (Siami-Namini & Namin, 2018) and ecology (Wang et al., 2017) to name a few.

This thesis focuses on solving the problem of uncertainty measurement and its impact on business decisions. It pursues two goals: first, develop and validate accurate and robust models for uncertainty quantification, employing both the well established statistical models and newly developed machine learning tools, with particular focus on deep learning. The second goal revolves around the industrial application of proposed models, applying them to real-world cases when measuring volatility or making a risky decision entails a direct and substantial gain or loss. The thesis is composed of 4 papers that approach the matter from different angles: financial, behavioral, economic, as well as from different perspectives: economic agent, market maker, regulator, macro-level (see Figure 1).



Traditional volatility estimation
Modelling of implied volatility (HAR) for crypto-currencies as a proxy to investors' uncertainty in absence of derivative market

Analysing drivers of risky behaviour
Risk management for market maker, using deep learning to identify traders likely to engage in high risk operations

Identifying fraudulent agents
NLP-based fraud detection model for financial documents, with sentence-level estimation of fraud probability for decision support during audit

Quantification of uncertainty on macro level
Text analysis of newspaper articles using natural language processing and deep learning to grasp the levels of media-induced economic policy uncertainty

Figure 1: Four papers, exploring the topic of economic uncertainty quantification

The first paper (in co-authorship with Simon Trimborn and Wofgang K.Härdle) focuses on a form of uncertainty quantification traditional to the finance market (Ederington & Lee, 1996) - implied volatility. The novel asset type - crypto-currencies (CC) - was selected for the analysis, as it exhibits extreme levels of volatility (Katsiampa, 2017) and still doesn't have an established derivative market (with an exception for Bitcoin). Capturing the CC market (represented by CRIX) through the construction of an implied volatility proxy in the absence of the derivatives for the majority of CC posed a challenge. The "fear index" of the American stock market - VIX - was selected as guidance and benchmark. Analysis of the relationships between VIX and volatility of the underlying assets provided an insight for the selection of a mentioned proxy - the historical rolling volatility of SPY. Following this finding, the rolling volatility of log-returns of CRIX was calculated. The HAR model proved to be best for estimating the daily volatility of CRIX log-returns, offering the MSE of 0.03. This model was further tested in a simulation, where it was used to estimate VIX. An impressive 89% correlation was achieved, thus proving the selected methodology's fitness to the announced goal. The established VCRIX provides a daily forecast for the mean annualized volatility of the next 30 days. The model was backtested for its forecasting power, resulting in low MSE performance and further examined by VIX's simulation (resulting in a correlation of 78% between the actual VIX and VIX estimated with the VCRIX model). A trading strategy using VCRIX outperformed the benchmark strategy for 68% of the tested period. VCRIX provides forecasting functionality and serves as a proxy for the investors' expectations in the absence of the developed derivatives market. These features provide enhanced decision-making capacities for market monitoring, trading strategies, and potentially option pricing. The VCRIX exposed excessive volatility that is captured by derivative-based indices like VIX and presumably stems from the behavioral component of option pricing.

To further investigate the risk-taking behavior of market actors that both stems from and perpetuates uncertainty, the second paper (in co-authorship Y. Yang, S. Lessmann, T. Ma, M.-C. Sung, and J.E.V. Johnson) focused on methods to identify risk-loving traders and employed the DNN infrastructure for it. The results obtained throughout several experiments confirmed the ability of a proposed autoencoders-based DL model to extract informative features automatically and exhibit higher accuracy in identifying high-risk traders than benchmark classifiers' forecasts. The results demonstrated that improvements in forecast accuracy translate into sizable increases in operating profit and confirmed the proposed DNN to effectively support (hedging) decision making and grasp the behavioral component from data. The application may extend to cases like increasing the likelihood of consumers' responding to promotion by studying clients' buying behaviors. E-commerce companies can dynamically adjust website layouts according to visitor preferences. Banks can enhance their risk control and make sensible credit approval decisions by analyzing clients' credit repayment behavior.

The opportunity to mitigate the volatile business climate is often on the side of corporate and governmental actors (Bachmann et al., 2013b), particularly regulators. The former, however, also remain under pressure of decision-making under uncertainty. The third paper (in co-authorship with Patricia Craja and Stefan Lessmann) addressed the challenging endeavor of fraud detection and offered the decision support model that allowed more accurate and interpretable evaluation of financial reports submitted for audit. Minimal research has been conducted on the subject of methods that combine the analysis of financial and linguistic information, and no studies were discovered on the application of text representation based on DL to detect financial statement fraud. In addition to quantitative data, we investigated the potential of the accompanying text data in annual reports, and have emphasized the increasing significance of textual analysis for the detection of signals of fraud within financial documentation. The proposed HAN method concentrates on the content as well as the context of textual information by capturing semantic associations and discerning the meanings of different word and phrase combinations. The results have shown that the DL model achieved considerable improvement in AUC compared to the benchmark models. The findings indicate that the DL model is well-suited to correctly identify fraudulent cases, specifically in the highly unbalanced case of fraud detection. We conclude that the textual information of the MD&A section extracted through HAN has the potential to enhance the predictive accuracy of financial statement fraud models, particularly in the generation of warning signals for the fraudulent behavior that can serve to support the decision making-process of stakeholders. The distorted word order handicaps the ability of the BOW-based ML benchmarks to offer a concise indication of the "red-flags". We offered the decision support solution to the auditors that allows a sentence-level indication of text fragments that trigger the classifier to treat the submitted case as fraudulent.

The insight NLP backed by DL could distill from textual input motivated the development of the fourth paper that concludes this thesis to find a way to quantify uncertainty on a macro level and analyze its drivers. Following the importance of risk assessment and agents' expectations in economic development and building on the existing works of Baker et al. (2016) and their economic policy uncertainty (EPU) index, we offered a novel DL-NLP-based method for the quantification of economic policy uncertainty. The method is applied to the corpus of articles from ten major USA newspapers, identifying them as containing or not containing the EPU. The proposed model showcased several valuable properties. Its predictive performance on the existing index outperformed the available benchmarks with an AUC of 0.96 and an F1-score of 0.65. The model remained robust in 10-fold cross-validation. Additionally, it offered high interpretability and adaptability, which was demonstrated by analyzing the top ten words responsible for EPU over time. We exposed a definite change of agenda in the newspaper articles. The first part of the sample, from Jan 2006 until Dec 2014, did not feature the word "trump". Starting in Jan 2015 until the end of our sample in Apr 2019, the word "trump" always featured in the top ten. These shifts show the necessity to adapt to changing political and economic trends when trying to capture economic uncertainty from newspaper articles. Our uncertainty index based on DL-NLP had superior forecasting ability for two out of five macroeconomic indicators, like unemployment, which resulted in lower RMSE for all variables. This way, the proposed method proved its fitness to better deal with the change in the newspaper agenda than the methodology of Baker et al. (2016). Our approach showed pathways towards capturing economic policy uncertainty over long periods while keeping track of changes in the way that news and uncertainty are reported. Two recent examples that changed newspaper reporting are the Trump presidency and the recent COVID-19 pandemic. The approach might prove especially useful for governments and institutions in countries with scarce up-to-date information sources on the level of uncertainty in the economy, as newspaper articles are widely available over time and therefore represent a feasible alternative data source to assess economic policy uncertainty.

Table 2: Overview of the four papers constituting the thesis

	Section 2	Section 3	Section 4	Section 5
Title	VCRIX - a volatility index for crypto-currencies	Can Deep Learning Predict Risky Retail Investors? A Case Study in Financial Risk Behavior Forecasting	Deep Learning application for fraud detection in financial statements	Quantification of Economic Uncertainty: a deep learning application
Contribution	Estimation of the risk measurement for the crypto-currency index (CRIX)components and delivery of market status information, analogous to implied volatility indices that capture investors expectations	Implementation of DL to spot risky traders opens ways to advance the behavioral research and employ the novel machine learning method to improve understanding of clients' actions.	First application for the state-of-the-art deep learning NLP technics to evaluate both textual and financial components company reports. The offered solution does not only provide improved accuracy but sentence-level interpretation for the decision making of the auditor. It can serve to faster and more accurate processing of audited documentation.	Introduction of a state-of-the-art deep learning model for textual analysis with improved predictive power and adaptability features in the setting of the changing newspaper rhetoric. Economic and financial institutions may use the reconstructed uncertainty index for the evaluation and forecasting of economic behavior, business cycles, as well as the assessment of effects of monetary policy and political decisions. Additionally, an analysis of word-level rhetoric is offered, enabling a deeper understanding of the uncertainty topic shift
Key findings	Historical rolling volatility is the best proxy for the implied volatility. The HAR model proved to be best for the estimation of the daily volatility of CRIX log-returns. The established VCRIX provides a daily forecast for the mean annualized volatility of the next 30 days, offering the MSE of 0.03 in back-test forecasting.	Extraction of important features done by the DL model offers improved predictive accuracy and constitutes a superior hedging strategy for trading risk evaluation, which translates into higher financial gains (around 800 GBP extra gained per trader).	The DL model is well suited to identify the fraudulent cases correctly, the textual information of the MD&A section extracted through HAN has the potential to enhance the predictive accuracy of financial statement fraud models, particularly in the generation of warning signals for the fraudulent behavior that can serve to support the decision making-process of stakeholders.	The proposed NLP method allowed to avoid dictionary-based labeling while retaining predictive powers over some of the macro variables. Additionally, it demonstrated better adaptability than the original EPU index and offered insight into the changing rhetorics of the newspapers' coverage of economic policy in the USA.
Data	CRIX index values and daily prices of crypto-currency constituents.	Details on financial trading activities from a market-maker	MDA section of financial statements submitted by USA companies for audit	Newspaper articles on the topic of economics from 10 major USA media
Approach	Heterogeneous Autoregressive model (HAR) with LSTM as closest competitor	Deep Learning with autoencoders	Deep Learning with LSTM and Hierarchical Attention Networks	Deep Learning with GRU and GPR-2 pertained embeddings
Comments	Currently in revision after the "Reject and Resubmit" from Journal of Empirical Finance	Published in European Journal of Operational Research	Submitted to Decision Support Systems	Submitted to International Journal of Forecasting

In summary, this thesis offered several solutions to approach and quantify a complicated matter of uncertainty and showcased the potential of DL models to offer accurate estimation and interpretation capabilities in different business scenarios. The results indicated that the state-of-the-art DL NLP methods could provide significant reinforcement to the forecasting, risk assessment, decision support, and economic policy quantification, without suffering the usual drawbacks of interpretability.

2 VCRIX - a volatility index for crypto-currencies

A.Kim, S.Trimborn, W.K.Härdle

2.1 Abstract

Public interest, explosive returns, and diversification opportunities gave stimulus to the adoption of traditional financial tools to crypto-currencies. While the CRIX offered the first scientifically-backed proxy to the crypto-market (analogous to S&P 500), the introduction of Bitcoin futures by Cboe became the milestone in the creation of the derivatives market for crypto-currencies. Following the intuition of the "fear index" VIX for the American stock market, the VCRIX volatility index was created to capture the investor expectations about the crypto-currency ecosystem. VCRIX is built based on CRIX and offers a forecast for the mean annualized volatility of the next 30 days, re-estimated daily. The model was back-tested for its forecasting power, resulting in low MSE performance and further examined by the simulation of VIX (resulting in a correlation of 78% between the actual VIX and VIX estimated with the VCRIX model). A trading strategy with the use of VCRIX outperformed the benchmark strategy for 68% of the tested period. VCRIX provides forecasting functionality and serves as a proxy for the investors' expectations in the absence of a developed crypto derivatives market. These features provide enhanced decision making capacities for market monitoring, trading strategies, and potentially option pricing.

2.2 Introduction

Introduction of BTC futures by the CME and Chicago Board Options Exchange (Cboe) on December 18, 2017 reinforced the positions of CC as a new asset class. The emergence of the derivatives market signaled the need for solid pricing strategies and a reliable (and stable) risk measure. The paper on pricing CC by Hou et al. (2020) addressed this issue by employing a Stochastic Volatility with a Correlated Jumps model (Duffie et al., 2000) and using insights on implied volatility dynamics by Fengler et al. (2003) in order to match non-stationarity and local heterogeneity phenomena of CRIX returns.

Industry demand and research revealed the necessity to explore the behavior of the CC volatility further, to provide the final ingredient - a proxy for implied volatility. In traditional markets, implied volatility is measured by volatility indices which can be considered a traditional financial tool. At the end of the 20th century, financial markets of the USA and Europe aimed to capture the global measure of volatility in the respective market, which led to the introduction of VIX or VDAX. The index providers settled on the model most appropriate for the specifics of the behavior of the corresponding derivative. Given the absence of a developed derivatives market, we have to infer the characteristics of the implied volatility from the CC market behavior. The specifics of the latter (high volatility and low liquidity) triggered the development of new investment methods, see Trimborn et al. (2019), further justifying the need for a volatility index, that would capture the unique specifics of CC as an asset class and provide a reliable indicator for the continuously unstable market.

Our research aims to create a VCRIX - a volatility index especially designed for markets akin to the CC ecosystem, see Subsection 2.4.1. The goal of the proposed VCRIX is the estimation of the risk measurement for the CRIX components and delivery of market status information, analogous to implied volatility indices that capture investors expectations.

Section 4.5 offers an overview of the used data sets for both traditional and CC markets. Section 2.4 provides a detailed explanation of the methodology used, including a brief revision of CRIX which was selected as an equivalent for the S&P 500, a note on the existing implied volatility indices and VIX methodology in particular (Subsection 2.4.2). Subsection 2.4.3 contains the details on the implied volatility proxy estimation, followed by Subsection 2.4.4 that clarifies VCRIX model selection and back-testing. Methodological results, details of the VIX simulation conducted to test the selected methodology and final time series are showcased in Section 2.5. Applications of the proposed volatility index are further explored in Section 2.6, which contains an example of the trading implementation of VCRIX. Additional observations and a summary of the conducted research are provided in Sections 5.6 and 2.8.

2.3 Data

This research employs CRIX values and traditional financial data, namely S&P 500 index values and VIX, which is the volatility index of Cboe based on the S&P 500. The daily historical closing values of CRIX for the period from Sep 2014 - the emergence of CRIX - to December 2018 (1583 observations, including weekends) were sourced from thecrix.de and converted to log-returns.

The daily historical closing prices of the S&P 500 and VIX from 2000 to the end of 2018 (4780 observations) were sourced from finance.yahoo.com. It must be pointed out that SPY (ETF on S&P 500 index) has closer relations to VIX by design, as clarified in Subsection 2.4.3, however, the log-returns of S&P 500 and SPY reveal no difference and thus could be interchangeable for the conducted analysis. The S&P 500 time series were converted to log-returns, VIX values remained as is.

2.4 Methodology

Implied volatility became a subject of academic research with the development of the derivatives market in the last quarter of the 20th century. The Black and Scholes (1976) model yields implied volatility as a volatility measure because, by definition, the implied volatility is the future volatility expected by the market. However, the market crash of October 1987 that bent the volatility surface of index options into a skewed "volatility smile", motivated an alternative solution that would provide a more accurate fit to market conditions. Bakshi et al. (1997) provide an extensive overview of the further developments in this field, including the stochastic interest rate option models of Merton et al. (1973), the jump - diffusion/pure jump models of Bates (1991), the stochastic volatility models of Heston (1993) and others. While acknowledging the diversity of options pricing models, authors agree on the necessity of matching the selection of one to the goals at hand.

The goal of VCRIX is to capture the expectations of the CC market, much like VIX is offering an uncertainty measurement with regard to the American stock prices. In simplified terms, VIX "predicts" the mean annualized volatility of the S&P 500 for the next 30 days in the future, that is in turn derived from the implied volatility extracted from the S&P 500 ETF swap prices. Absence of a CC analog calls for an alternative solution for VCRIX. In the absence of intrinsic predictive power, VCRIX would also have to be forward-looking, providing a valid estimation of the CC market volatility in the future. The selection of the new methodology thus includes two tasks: estimation of the best implied volatility proxy and further search for the model to exhibit the most consistent predictive performance.

2.4.1 CRyptocurrency IndeX

S&P 500 and DAX serve as indicators of the current state of American and German markets by aggregating the weighted performance of the most significant listed companies. CRIX, developed by Trimborn and Härdle (2018), plays a similar role for the CC market, providing a statistically-backed market measure, which distinguishes it from other CC indices like Crypto20, CCi30, WorldCoinIndex. At the core of CRIX lies the idea that a fixed number of constituents (as in case of S&P 500) may be a good approach for relatively stable markets, however, with the ever-growing number of CC, practical implementation would demand a filter that keeps out the noise, while preserving the information about the market dynamics. CRIX employs Akaike Information Criterion (AIC, Akaike (1987)) that determine the number of constituents quarterly according to the explanatory power each CC has over the market movements. CRIX was used as a proxy to the CC market before in research papers by Elendner et al. (2018), Klein et al. (2018), Mihoci et al. (2019), and was adopted as a benchmark by commercial projects like Smarter Than Crypto, Crypto20, F5 Crypto Index, and also used by the European Central Bank as a market indicator in the report dedicated to understanding the "crypto-asset phenomenon" (Chimienti et al., 2019). These use cases confirm the applicability of CRIX as an appropriate basis for VCRIX.

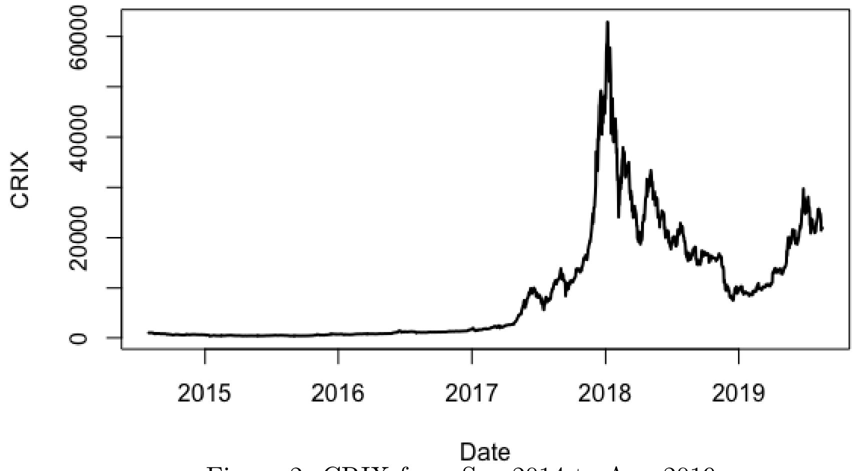


Figure 2: CRIX from Sep 2014 to Aug 2019

CRIXcode

Consequently, the index rules will have a significant impact on the behavior of VCRIX. The initial paper by Härdle and Trimborn (2015) defines CRIX as a Laspeyres index, taking the value of a k asset basket and comparing it against the base period, as indicated in Equation 1:

$$CRIX_t(k) = \frac{\sum_{i=1}^{k} P_{it} Q_{i,t_l^-}}{Divisor(k)_{t_l-}} \tag{1}$$

with P_{it} the price of asset i at time t and Q_{i,t_l^-} the quantity of asset i at time t_l^-(the last time point when Q_{i,t_l^-} was updated). Monthly re-balancing accounts for the changes in the market capitalization of a CC and the number of index components, the *Divisor* ensures that this procedure does not affect the value of CRIX, rather only price changes in its constituents shall be of effect.

2.4.2 Implied volatility indices

Consideration of the existing volatility indices would constitute a logical step towards the selection of the appropriate solution. As observed by Siriopoulos and Fassas (2009) recent decades saw the rise of the model-free indices (based on model-free implied volatility (MFIV)) that were made possible by highly liquid options markets and readily available model-free implied variances (France, Germany, Japan, Switzerland, the U.K., and the U.S). Major alternatives to the "model-free" approaches are the Black-Scholes (BS) implied volatility and statistical models such as GARCH (Bollerslev, 1986). While MFIV is extracted from the corresponding set of current option prices without the need to assume any specific pricing model, this approach comes along with a range of methodological issues. For example, Biktimirov and Wang (2017) tested both approaches on the subject of forecasting accuracy, and BS implied volatility came out superior both in terms of in-sample "encompassing" models that include several forecasts in the same combined specification and also in out-of-sample forecasting. We consider model-free and model-based methodologies given the available data and above mentioned empirical results.

Introduction of XBT-Cboe BTC Futures by the Cboe in 2017 became the first step in the establishment of the CC derivatives market, thus approaching the possibility of the model-free implied volatility index construction. However BTC futures were not considered for this research due to several reasons: officially listed (Cboe and CME Group) futures do not provide insight into implied volatility of the underlying like option prices do by design, existing data for options is so far only available for BTC from commercial providers like Deribit (2019), not for the broader CC market. Most importantly, the goal of the VCRIX is to grasp the investors' expectations of the whole CC market. As Figure 3 shows, the weight of BTC in CRIX has been remaining below 0.6 most of the time, and thus BTC and its options cannot be considered sufficiently representative.

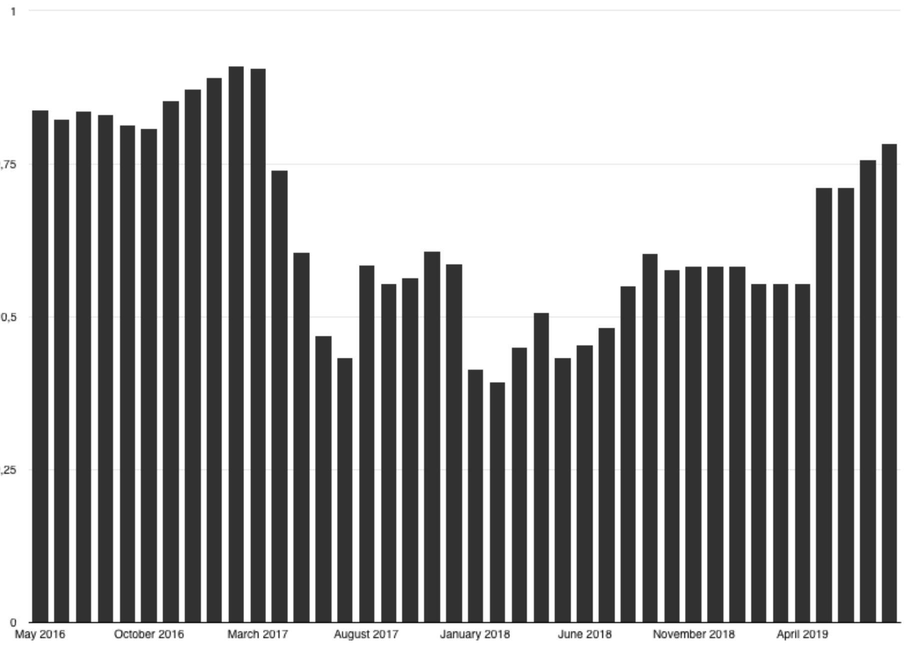


Figure 3: Weight of BTC as a constituent of the CRIX over time

Given the outlined limitations of the CC derivatives market, we settle for a model-based index, that is capable of capturing the predictive power of a traditional volatility index. The VIX by Cboe for the US market was selected as a guidance and benchmark. VIX is acknowledged by the established CC players as a standard for the implied volatility modeling: in 2019 one of the biggest CC derivative trading platforms Ledger X - a US company regulated by CFTC (United States Commodity Futures Trading Commission) - introduced an implied volatility index for BTC called LXVX (Cointelegraph, 2019), announcing its inheritance to VIX (LXVX, 2019).

The current VIX methodology was developed based on the pioneering research of Whaley (1993), Neuberger (1994), Madan et al. (1998), Demeterfi et al. (1999) and Britten-Jones and Neuberger (2000) among others. It estimates the implied volatility of option prices on the S&P 500 by taking strikes and option prices as inputs. With exchange-traded S&P 500 variance swap rate as its underlying, VIX became a proxy for market volatility (Cboe, 2009):

$$\sigma^2 = \frac{2}{T}\sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i) - \frac{1}{T}\left[\frac{F}{K_0} - 1\right]^2 \tag{2}$$

$$VIX = \sigma * 100, \tag{3}$$

where T is time to expiration, F is a forward index level from index option prices, K_0 is a first strike price below F, K_i is a strike price of the ith OTM option (on average the range of i is between 1 and 500, reflecting the composition of the S&P 500) , $Q(K_i)$ is the midpoint of the bid-ask spread for each option with strike K_i, ΔK_i is an interval between strike prices (half the difference between the strike on either side of K_i) and R the risk-free interest rate to expiration.

2.4.3 Implied volatility proxy

VCRIX is designed to measure and proxy the lacking implied volatility in the CC market, hence it has to be based on a model, capable of capturing the predictive power of a traditional implied volatility index like VIX. In order to select an appropriate proxy for VIX, one has to check the dynamics of the underlying, in particular the annualized historical rolling volatility of SPY log-returns over 30 days (VIX measures how much the market thinks the S&P 500 Index will fluctuate in the 30 days from the time of each tick, according to Cboe (2009)). Equation 4 displays the rolling volatility method (r_t being a daily return of an asset on day t and $\hat{\mu}$ an estimated mean daily return over the 30 day period). In case of historical volatility, the σ would define the volatility of the last day of the month, while for forward volatility the same calculation will account for the volatility of the first day of the month. It should be pointed out that we are not using the notion of forward volatility as in Taleb (1997), namely, how implied volatility differs for related financial instruments with different maturities. In this case, the "forward" part only bears the idea of adjusting the time span of the traditional rolling volatility measure to be forward-looking (results are displayed in Figure 5).

$$\sigma_t = \sqrt{\frac{1}{30}\sum_{i=t-30}^{t-1} (r_i - \hat{\mu})^2} * \sqrt{252} * 100 \tag{4}$$

2.4.4 Model selection and back-testing

The dataset of CRIX log-returns was transformed into annualized daily volatility based on 30-day rolling windows (CC are traded everyday, unlike traditional securities). We considered both univariate and multivariate models, however, the latter did not prove superior in approximating the selected time series and for the sake of brevity this case will not be described in this paper. Thus the choice was made in favor of univariate models. 273 values of the dataset were set aside for back-testing, which corresponds to 20% of the dataset. We considered the following models that describe the volatility dynamics:

1. GARCH family (tested by Hansen and Lunde (2005), French et al. (1987), Antoniou and Holmes (1995)
   - GJR
   - EGARCH
   - EWMA
2. Heterogeneous Auto-Regressive (HAR) model (introduced by Corsi (2009) and tested by Chiriac and Voev (2011), Busch et al. (2011), Patton and Sheppard (2015))
3. neural network-based Long short-term memory cell (LSTM) models (Hochreiter & Schmidhuber, 1997b)

The latter represents a comparatively new approach to volatility modeling. The LSTM architecture belongs to the Recurrent Neural Networks family and has been extensively used (together with Gated Recurrent Units) for the modeling of sequential data like text or time series. Its complex architecture provides interesting forecasting opportunities that have been explored and proven useful by Kong et al. (2017), Pichl and Kaizoji (2017), H. Y. Kim and Won (2018a), R. Luo et al. (2018). Figure 4 provides a visual comparison of the 3 best-performing models: HAR (specified in Equations 9-11, EWMA model (specified in Equation 5, where $\sigma^2_{i,t+1}$ is the variance of CRIX log-returns ($r_{i,t}$) in the next period and the decay factor λ=0.96) and LSTM model (15 epochs, 3 layers of 365 neurons, specified in Equation 6 in its simplified form, where $\hat{\theta}$ signifies the complex set of parameters that are optimized during the training of the neural network).

$$\sigma^2_{i,t+1} = \lambda\sigma^2_{i,t} + (1-\lambda)r^2_{i,t} \tag{5}$$

$$\sigma^2_{i,t+1} = f_{\hat{\theta}}(\sigma^2_{i,t}) \tag{6}$$

As can be observed from Figure 4, all three models learn to anticipate the behaviour of the 30-day rolling volatility of CRIX quite well, however, the similar peaks from August to October expose their limited ability to timely reflect a sudden splash in the CC market. LSTM proves to be particularly vulnerable in its predictive capacity. This could be further remedied by the more complex architecture and increased training time, making the modeling more computationally costly. Given the non-substantial role of LSTM in the further implementation of VCRIX and the fact that the detailed explanation of the LSTM methodology with regards to financial forecasting has been provided previously in papers by K. Chen et al. (2015), J. Heaton et al. (2016), Fischer and Krauss (2018a), we omit the detailed explanation of the LSTM application.

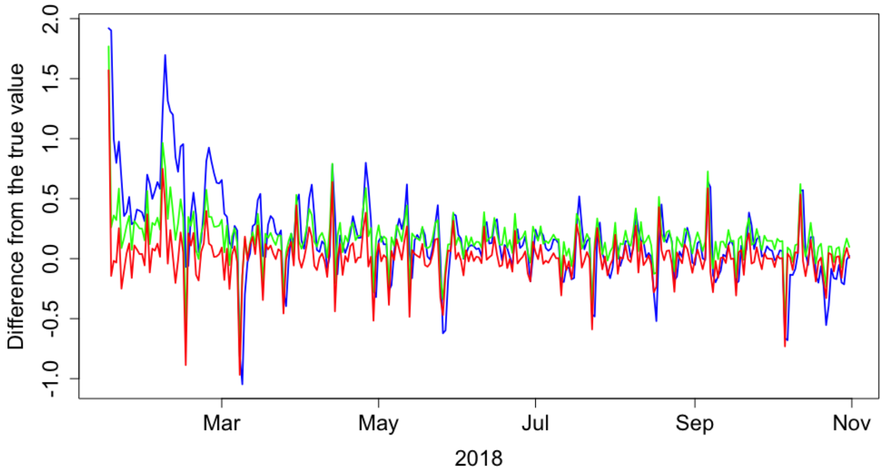


Figure 4: Difference between the true (30-day rolling volatility of CRIX) and the HAR, EWMA and LSTM models

Metric	HAR	EWMA	LSTM
Correlation	0.99	0.99	0.97
MSE	0.03	0.06	0.16
MAE	0.11	0.19	0.30
Mincer Zarnowitz R-adj	0.98	0.98	0.94

Table 3: Evaluation of the predicted values of 30-day annualized rolling volatility of log-returns on CRIX (daily re-estimation)

2.5 Simulation and assessment

During the model back-testing, the HAR and the EWMA models performed very closely. EWMA consistently underestimated the volatility but registered the up and down shifts faster. The LSTM frequently overestimated the volatility, which is coherent with the higher values that are picked up by VIX in comparison to the rolling volatility as showcased in Figure 4.

According to the results in Table 3, the HAR model was selected as the best predictive performer with correlation 0.99, MSE 0.03, and MAE 0.11. It should be specified that the original HAR model, Corsi (2009), is built on the premise that traders conduct their activities according to the strategies based on different frequencies (high-frequency trading, daily traders, weekly, monthly), which in turn affects the overall market volatility at certain points in time. As the CC market is young and presumably still dominated by sporadic non-expert traders (due to the pseudo-anonymity of most CC, justification of this assumptions remains challenging), presenting an informed judgment at this stage is rendered impossible by the implicit anonymity of most CC and its users. The recent analysis for potential herding behavior by Bouri et al. (2018) and da Gama Silva et al. (2019) touches on this topic, without providing actual analysis of the traders' practices.

In the absence of data on CC traders' behavior, we have made the assumption that the traditional practices could potentially be applied for the CC case. This led us to make two adjustments to the original HAR model. 30-day historical rolling volatility (annualized, as shown in Equation 7 was used instead of realized volatility (it was selected as a most representative to proxy VIX).

$$RV_t^d = \sigma_t = \sqrt{\frac{1}{30} \sum_{i=t-30}^{t-1} (r_i - \hat{\mu})^2 * \sqrt{365} * 100} \tag{7}$$

Similarly to Equation 4, r_t is a daily return of CRIX on day t and $\hat{\mu}$ an estimated mean daily return over the past 30 days (we keep the span to 30 days as CC are traded without the weekends), meanwhile, the number of days was changed to 365 for the same reason. Further on we will refer to σ_t^2 as daily realized volatility RV_t^d to maintain the usual HAR notation.

The change of 5 (weekly) and 21 (monthly) trading frequencies to 7 and 30 days respectively is reflected in the calculation of weekly and monthly volatilities, Equations 9 and 8.

$$RV_t^w = \frac{1}{7}(RV_t^d + RV_{t-1}^d + ... + RV_{t-6}^d) \tag{8}$$

$$RV_t^m = \frac{1}{30}(RV_t^d + RV_{t-1}^d + ... + RV_{t-29}^d) \tag{9}$$

The final version of VCRIX is forward-looking and offers a forecast of the mean annualized daily volatility for the next 30 days. The index is re-estimated daily based on the realized daily volatility. The Equations 10 and 11 offer the actual methodology where the forecast - RV_{t+1}^d - is estimated with a regression given the daily RV_t^d (initially estimated with 30-day rolling window), weekly RV_t^w and monthly RV_t^m volatilities that are recalculated daily.

$$RV_{t+1}^d = \alpha + \beta^d RV_t^d + \beta^w RV_t^w + \beta^m RV_t^m + \omega_{t+1} \tag{10}$$

$$VCRIX_t = \frac{RV_{t+1}^d}{Divisor} \tag{11}$$

The initial value of VCRIX is set to 1000, following the convention set by CRIX. A *Divisor* is introduced in order to account for the jumps that might occur due to the change in the number of constituents every month. The *Divisor* is set to a certain value on the first day to transform the estimated volatility to 1000 points of VCRIX. *Divisor* remains the same over the month. Every month the constituents can change. In this case, the value of VCRIX from the last day of the month will be transferred to the first day of the next month, after that the *Divisor* will be reevaluated in order to reflect the value for transformation.

In order to provide an additional justification for the selected methodology, a VIX simulation was performed. It comprised the application of the selected HAR model to log-returns of the S&P 500 instead of CRIX.

After establishing the CRIX as the underlying for VCRIX and selecting VIX as a benchmark for the evaluation of the CC volatility index, we proceeded with selection of the appropriate implied volatility proxy in the absence of CC derivatives market. The time series (Figure 5) analysis showed the correlation of 0.89 between VIX and historical volatility, while the correlation between VIX and forward-looking volatility was 0.78. Given the scale of the differences, it is obvious that both historical and forward-looking volatilities fail to grasp the exact variation of VIX. This gap grows in crisis periods (as it can be seen for 2009) but shrinks back during market cool-down.

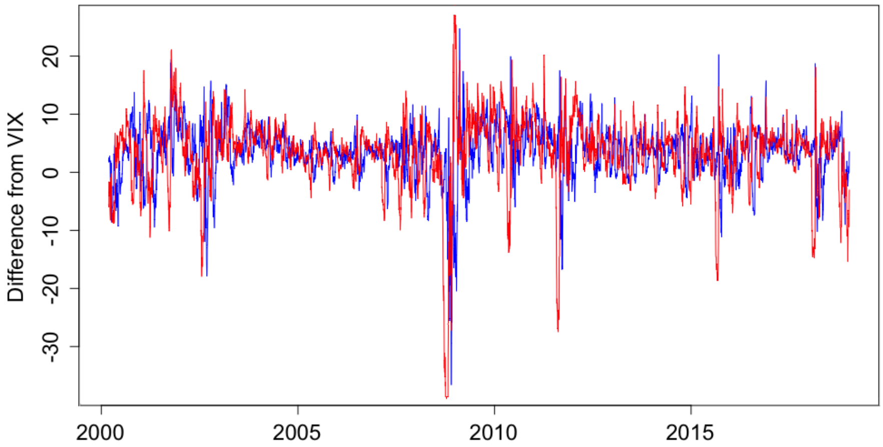


Figure 5: Difference between VIX and historical and forward-looking volatilities (30 calendar days)

Further analysis with linear regression showed that historical volatility could explain 80% of the VIX variance. Thus the historical 30-day rolling volatility of S&P 500 log-returns was selected as the best proxy for VIX. Following this decision and the goal of granting VCRIX predictive capabilities, the time series of 30-day historical rolling volatility of CRIX log-returns was constructed and used as a true value in back-testing of several predictive models that were estimating the annualized volatility one day ahead. According to the evaluation metrics, as shown in Table 3, the HAR model was selected as a basis for VCRIX. Further on this model was tested in the simulation of actual VIX using the S&P 500 log-returns instead of CRIX log-returns. The resulting pair of time series showcased the correlation of 89%, thus justifying the model selection.

Days of lag	**Correlation**	**MDA**
Day-on-day	0.89	51%
21 days	0.89	64%
42 days	0.87	73%

Table 4: Evaluation of the simulation of VIX using VCRIX methodology, comparison of true and simulated values

The simulation of VIX exhibited correlation of 89% and a Mean Directional Accuracy (MDA) of 51% rising to 64% in case lag of 21 days is considered, as indicated in Table 4. Figure 6 and Figure 7 showcase the difference between the estimated values and actual VIX. These results led us to believe that the chosen methodology does indeed provide a solid estimation of the implied volatility in the absence of the derivatives market.

Figure 8 displays the time series of VCRIX from Jan 2015 to Aug 2019 and the smoothed conditional means (LOESS) red line with a span of 0.5, it is added to offer a long term review on volatility.

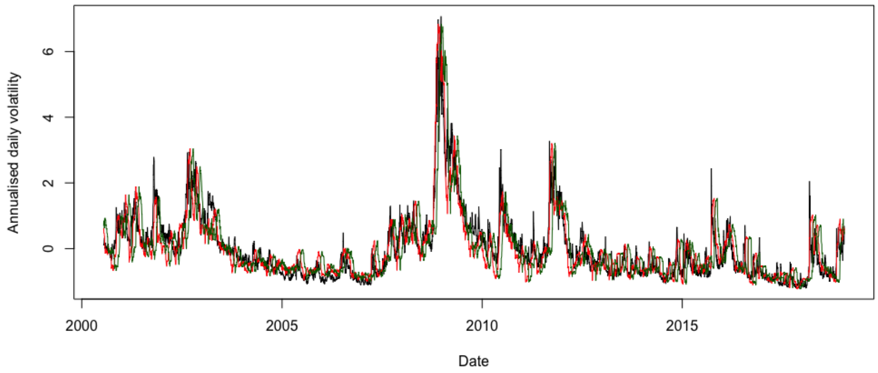


Figure 6: VIX estimated with HAR model on scaled daily volatility of SPY log-returns, VIX estimated with HAR with 21 days lag and true **VIX** values from 2000 to 2019

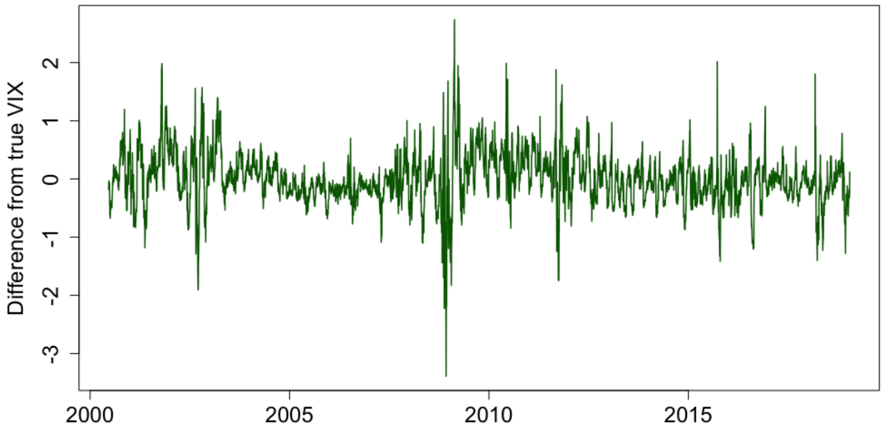


Figure 7: Difference between true and estimated VIX, values from 2000 to 2019. One can observe that the proposed model lags in catching the big spikes but performs well when market volatility is lower.

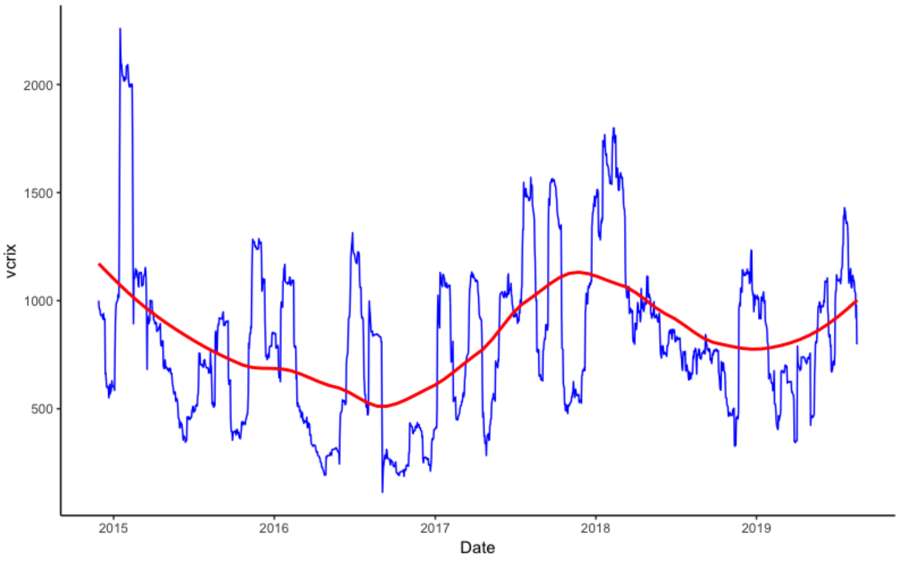


Figure 8: VCRIX and LOESS-smoothed mean (span=0.5)

VCRIX

2.6 Trading implementation

As the CC market develops and new financial instruments based on CC appear, VCRIX can become increasingly employed in trading strategies. As one of the examples, an inverse volatility ETF is a financial product that allows investors to gain exposure to volatility, and thus hedge against portfolio risk, without having to buy options.

Regardless of the absence of the above mentioned derivative instruments, volatility-based trading strategies may still be employed and tested. Conventional short-term reversal strategies have been explored and perfected by scholars and industry practitioners (Blitz et al., 2013; Jegadeesh, 1990; Lehmann, 1990) over the years. We have employed a number of modified reverse volatility trading strategies with an example provided below. As an input, we employ VCRIX for daily volatility estimation and LOESS of VCRIX (as a variation of MA, different spans represented in Figure 9) as a benchmark.

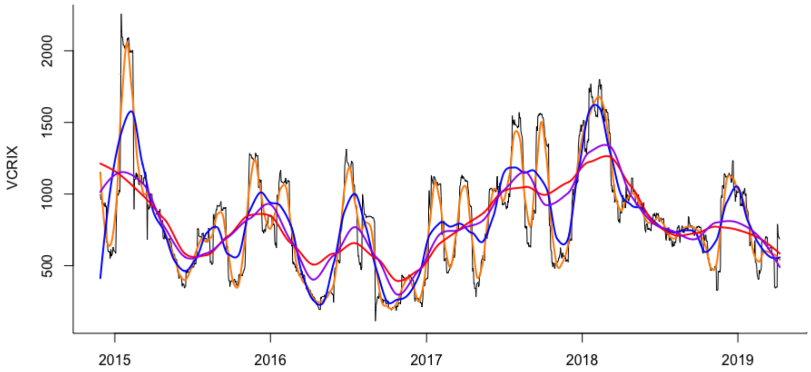


Figure 9: VCRIX and the LOESS-smoothed mean of VCRIX, with span=0.05, span=0.1, span=0.2, span=0.25

VCRIXloess

LOESS is a non-parametric operator that yields a smooth function by locally minimizing the variance of the residuals or prediction error (Cleveland, 1979). For each value of x, the value of $f(x)$ is estimated by using its neighboring sampled (known) values (quite similarly to a knn algorithm). In the case of LOESS, the tunable parameter is the span that will determine the smoothness of the resulting estimate, with a broader span resulting in higher bias and narrower span offering higher variance.

Figure 10 provides an illustration of a trading strategy that is based on long-cash signals generated by the relationships between the daily VCRIX value and its two LOESS curves (span=0.25 and span=0.20). In further notation we indicate the span with the subscripts, as in Figure 10, constructed with the use of $LOESS_{0.25}$ and $LOESS_{0.20}$.

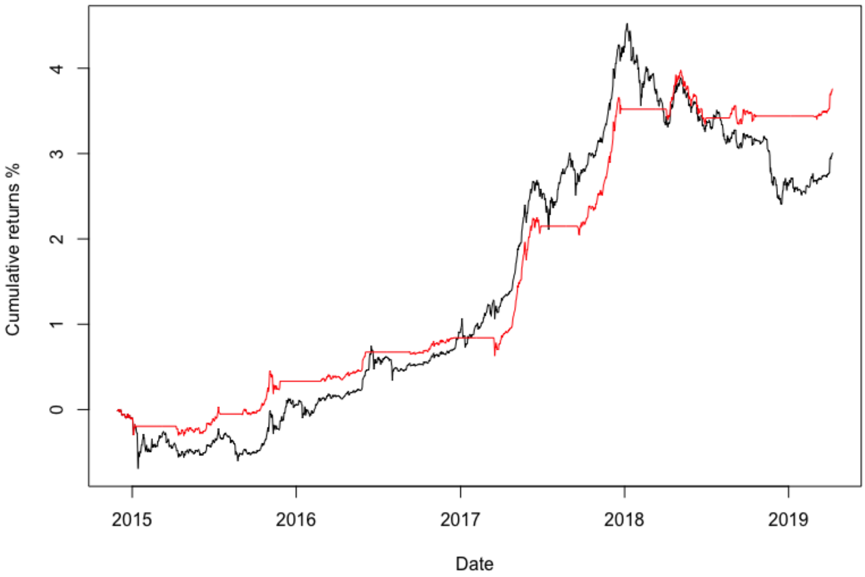


Figure 10: Cumulative returns of the trading strategy with $LOESS_{0.25}$ and $LOESS_{0.20}$ versus the cumulative **returns on CRIX**

VCRIXtrading

The strategy gets its signals from the LOESS-smoothed mean of VCRIX. The trading strategy, Algorithm 1, dictates to go long in cash when the volatility measured by VCRIX is high and go long in an ETF on CRIX when the volatility measured by VCRIX is low. We compare if the volatility is high or low by the LOESS-smoothed mean of VCRIX. A LOESS with a broad span gives a long term smoothed average for VCRIX, whereas a LOESS with smaller span gives the short term average. In particular we go Long in a CRIX ETF when the short term volatility is low compared to the long term one, $LOESS_i \geq LOESS_j$, and vice versa go Long in cash when the short term volatility is comparably high, $LOESS_i < LOESS_j$, see Algorithm 1.

Algorithm 1: : Trading strategy

```
1 [1] $i, j \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$, $i > j$ LOESS$_i$, LOESS$_j$, CRIX ETF Investment
  product $y$ LOESS$_i \geq$ LOESS$_j$ $y =$ CRIX ETF  LOESS$_i <$ LOESS$_j$ $y =$ Cash
```

By construction the choice of the span of LOESS is critical for the performance of the trading strategy. We construct the LOESS for the spans 0.05, 0.1, 0.15, 0.2, 0.25, and compare the results with the following measures:

1. **cumul.returns**: the aggregate gain over the observed time period up to the final day of trading.
2. **mean.returns**: the mean of the daily trading strategy returns.
3. **takeover.days**: the percentage of days when the cumul.returns are higher for the trading strategy than for CRIX.

4. **Sharpe.ratio**: compares the mean of the returns of the trading strategy over the standard deviation of the returns of the trading strategy, reflecting extra return per unit of increase in risk.

The results are presented in Table 5. The rows are named by the two LOESS-smoothed means involved in the trading strategy. CRIX returns are offered for reference. The left LOESS measures the long term VCRIX volatility and the right one the shorter-term one, in Algorithm 1 indicated as i and j respectively.

	cumul.returns	mean.returns	takeover.days	Sharpe.ratio
CRIX	3.00%	0.19%	NA	0.0484
$\text{LOESS}_{0.10} \sim LOESS_{0.05}$	0.83%	0.05%	27.24%	0.0202
$\text{LOESS}_{0.15} \sim LOESS_{0.05}$	2.18%	0.14%	45.64%	0.0583
$\text{LOESS}_{0.20} \sim LOESS_{0.05}$	2.36%	0.15%	45.89%	0.0661
$\text{LOESS}_{0.25} \sim LOESS_{0.05}$	2.96%	0.19%	55.05%	0.0810
$\text{LOESS}_{0.15} \sim LOESS_{0.10}$	3.43%	0.22%	75.52%	0.0948
$\text{LOESS}_{0.20} \sim LOESS_{0.10}$	2.95%	0.19%	63.97%	0.0867
$\text{LOESS}_{0.25} \sim LOESS_{0.10}$	3.41%	0.21%	66.92%	0.1009
$\text{LOESS}_{0.20} \sim LOESS_{0.15}$	3.51%	0.22%	57.88%	0.1013
$\text{LOESS}_{0.25} \sim LOESS_{0.15}$	3.58%	0.23%	49.15%	**0.1039**
$\text{LOESS}_{0.25} \sim LOESS_{0.20}$	**3.76%**	**0.24%**	**68.05%**	0.1029

Table 5: Comparison of trading strategies with several LOESS-smoothed means of VCRIX.

We observe the Sharpe ratio is best when we measure the long term volatility over a longer window, meaning for higher values of LOESS-smoothed means, e.g., $i = 0.20$ and $i = 0.25$. We found the best results, in terms of the Sharpe ratio, for the pair of LOESS spans 0.25 and 0.15, as well as 0.25 and 0.20. The second pair performs best, followed by spans 0.25 and 0.15 in terms of cumulative returns as well as takeover days (these trading strategies are more often above the one for a CRIX ETF). The trading strategy, see Algorithm 1, works in this case in the following way: We go long in a CRIX ETF when $LOESS_{0.25} > LOESS_{0.10}$ and long in cash when $LOESS_{0.25} < LOESS_{0.10}$. Similarly, for 0.25 and 0.20, the trading strategy receives signals if: We go long in a CRIX ETF when $LOESS_{0.25} > LOESS_{0.20}$ and long in cash when $LOESS_{0.25} < LOESS_{0.20}$.

As it can be observed from the graph, Figure 10, and Table 5, for 68% of the days the strategy outperforms the benchmark. As an additional benefit for the portfolio balancing, the variation of the trading strategy is lower than one of CRIX returns. Regardless of the downturn that takes place during the 2017 boom, the results after the cool-down remain superior to the plain CRIX returns, which suggests the viability of VCRIX as a trading tool.

2.7 Discussion

From the beginning, one of the biggest complexities in crypto-trading came from the absence of clear pricing strategies: what is BTC worth? How do we estimate the value of new coins? Are coins under- or over-appreciated? (Yermack, 2015). While mechanics and potential implications of CC in financial economics are being explored Härdle et al. (2020), there is still no established consensus over the evaluation methods. Nowadays agents are often left with nothing but the information on the overall market "feeling" about the CC, which is communicated by the rise and fall of the price, in other words, It is volatility.

VCRIX captures the volatility jumps that correspond to the development of the CC-ecosystem and can tell a story of the CC adoption (Figure 12). We observe spikes of interest in BTC in 2015, winter and summer of 2016 when BTC was slowly making its way to the attention of the general public. The large scale swings in price would not constitute a significant shock in absolute values, but when something that was still considered a digital maverick rose in value from roughly 400 USD to 1000 USD within a year (Business Insider, 2016, n.d.), investors noticed. VCRIX further captures the beginning of the first massive growth wave (also captured well by the CRIX in Figure 11) and development of altcoins (ETH, LTH, and others).

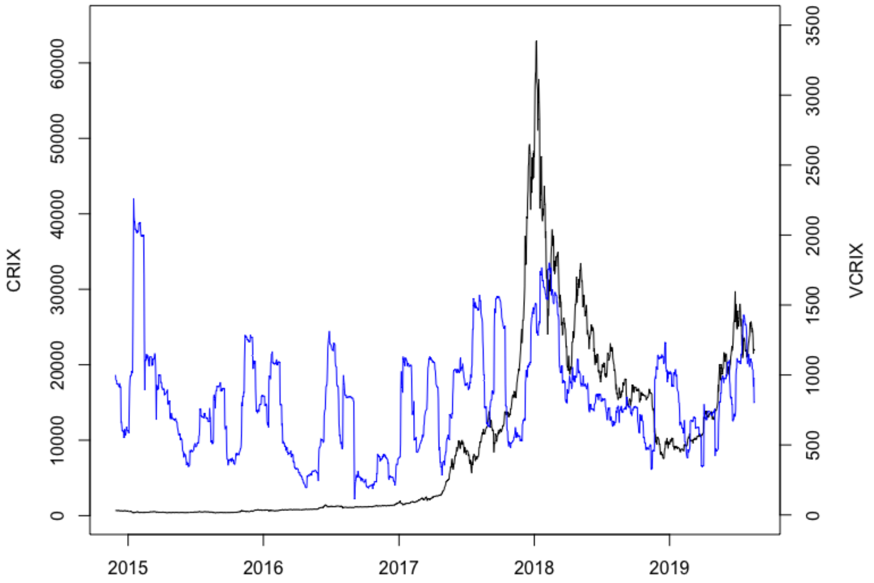


Figure 11: CRIX and VCRIX

2017 became the year of massive volatility (VCRIX showcases the values that can be interpreted as daily volatility of 140%). These levels of uncertainty were largely caused by the major legislative shifts that were happening in countries-juggernauts of CC movement: China, Korea, Japan, and the USA. Additionally, BTC was going through the heated debates on the SegWit (Segregated Witness) fork that was supposed to improve the speed and cost of BTC transactions. The fork was implemented in August, 2017 and led to the emergence of BTC Cash due to a certain number of big miners disagreeing with the implementation. These volatility spikes yet proved to be minor in comparison with the major market meltdown that happened at the beginning of 2018, when prices of most currencies on average suffered an 80% drop (CoinMarketCap, 2018). 2018 was considered to be a stabilization period when governments and financial corporations were getting on-board, however, the end of 2018 saw another volatility spike, majorly driven by the "holiday race" and uncertainty driven by "Constantinople fork" that is expected from Ethereum at the beginning of 2019.

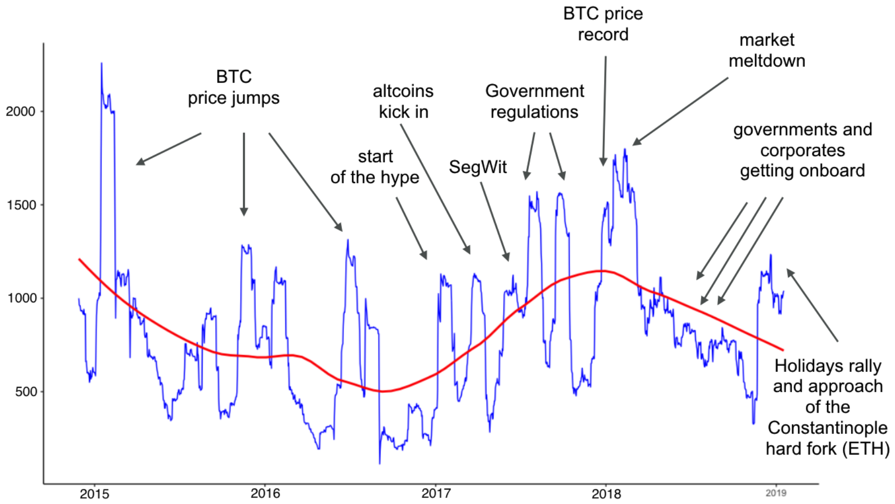


Figure 12: VCRIX interpretation

Pattern analysis of the VCRIX graph allows to distinguish a pattern that could be allegedly interpreted as a signal to large volatility spikes. Volatility clusters take the "triple spike" shape with the first spike indicating the upcoming large wave - this structure can be observed throughout 2016 and 2017, with the biggest wave at the end of 2018, taking a form of a tall "triple spike". This structure fades throughout 2018 during the settle-down, however, one may expect that the spike at the beginning of 2019 may be interpreted as the signal to a large wave of volatility coming during summer and autumn of 2019 (this prognosis was made during the writing of the paper in Spring of 2019). As of August 2019, this forecast proved correct (Figure 13), although the interpretation requires further economic investigation and cannot be used as a forecasting tool without additional scrutiny.

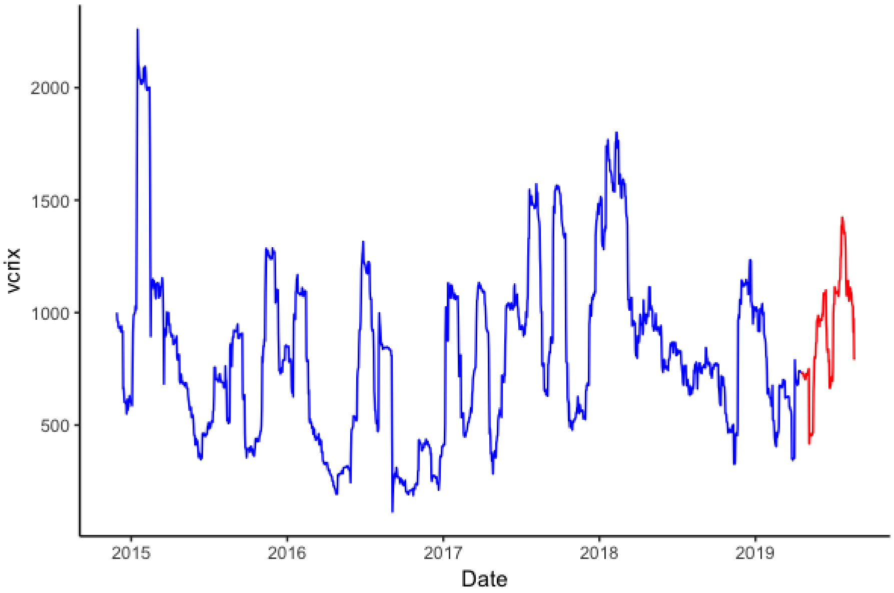


Figure 13: VCRIX and realization of the forecasted volatility spike

The search for an implied volatility proxy performed in Section 2.4.3 showed that VIX tends to overestimate the realized volatility. As it would seem, there is some information about market expectations that is not explained by the historical volatility. The excessive uncertainty would be expected to have strong relationships with returns that happen at the point of the highest delta. Given the design of VIX, one may expect it to contain additional signal about the emotional status of the market that tends to overreact in times of uncertainty. Interestingly enough, the LSTM predictive model also tends to overestimate the volatility. The neural network-based models are known for the capability to pick up underlying trends that are omitted in traditional financial models, however, the "black box" nature of models render clear interpretation complicated.

2.8 Conclusion

We have set the goal of capturing the expectations on the CC market (represented by CRIX) through the construction of an implied volatility proxy in the absence of the derivatives for the majority of CC. The "fear index" of the American stock market - VIX - was selected as guidance and benchmark. Analysis of the relationships between VIX and volatility of the underlying assets provided an insight for the selection of a mentioned proxy - the historical rolling volatility of SPY. Following this finding, the rolling volatility of log-returns of CRIX was calculated. The HAR model proved to be best for the estimation of the daily volatility of CRIX log-returns, offering the MSE of 0.03 and a 99% correlation with the 30 day-rolling volatility of CRIX log-returns. This model was further tested in a simulation, where it was used to estimate VIX. An impressive 89% correlation was achieved, thus proving the fitness of the selected methodology to the announced goal. The established VCRIX provides a daily forecast for the mean annualized volatility of the next 30 days, and showcases the observed excessive volatility that is captured by derivative-based indices like VIX and presumably stems from the behavioral component of option pricing.

3 Can Deep Learning Predict Risky Retail Investors? A Case Study in Financial Risk Behavior Forecasting

A.Kim, Y.Yang, S.Lessmann, T.Ma, M.-C. Sung, J,.E.V. Johnson

3.1 Abstract

The paper examines the potential of deep learning to produce decision support models from structured, tabular data. Considering the context of financial risk management, we develop a deep learning model for predicting whether individual spread traders are likely to secure profits from future trades. This embodies typical modeling challenges faced in risk and behavior forecasting. Conventional machine learning requires data that is representative of the feature-target relationship and relies on the often costly development, maintenance, and revision of handcrafted features. Consequently, modeling highly variable, heterogeneous patterns such as the behavior of traders is challenging. Deep learning promises a remedy. Learning hierarchical distributed representations of the raw data in an automatic manner (e.g. risk taking behavior), it uncovers generative features that determine the target (e.g., trader's profitability),

avoids manual feature engineering, and is more robust toward change (e.g. dynamic market conditions). The results of employing a deep network for operational risk forecasting confirm the feature learning capability of deep learning, provide guidance on designing a suitable network architecture and demonstrate the superiority of deep learning over machine learning and rule-based benchmarks.

3.2 Introduction

The paper applies recently developed deep learning (DL) methods to forecast the behavior of retail investors in the spread-trading market. Market makers depend on accurate forecasts of traders' future success to manage financial risks. Through developing a DL-based forecasting model and confirming the profitability of a model-based hedging strategy, we provide evidence that characteristic features of DL generalize to the structured data sets commonly employed in retail finance and decision support.

DL methods operate in a stage-wise manner. For example, in a deep neural network (DNN), each layer receives an input from previous layers, learns a high-level representation of the input, and passes the representation (i.e., output) to a subsequent layer.

A popular example of the stage-wise approach is that of face recognition. To detect faces in an image, the first layers in a DNN learn low-level concepts such as lines and borders from raw pixels. Deeper layers generalize lower layer outputs to more complex concepts such as squares and triangles that eventually form a face (LeCun et al., 2015). An analogous example in decision support could be corporate credit risk modeling. Bankruptcy prediction models estimate default probabilities on the basis of ratios of accounting variables (e.g., total assets/total liabilities) (Geng et al., 2015). In a DL framework, such ratios represent a low level representation. Using the balance sheet as (raw) input, lower layers in a DNN can relate a variety of statement variables and calculate informative ratios in a data-driven manner. A higher level representation of the data could then include the trend in a financial ratio or inter-dependencies between ratio variables. The specific representation is calculated autonomously.

A hierarchical composition of representations of different complexities enable a DNN to learn abstract concepts such as that of a delinquent borrower. Representation learning also enhances the ability of a model to extract patterns that are not well represented in the training data, which is a problem for other data-driven models (Bengio, 2009). DL methods have delivered excellent results in applications such as computer vision, language processing, and many others (Schmidhuber, 2015). This success has established the effectiveness of DL-based feature learning in applications that rely on unstructured data (W. Liu et al., 2017).

Applications of conventional - 'shallow' - machine learning (ML) are manifold. Marketing decision models, for example, support all stages of the customer life cycle including response modeling, cross-/up-selling (Z.-Y. Chen et al., 2016), and churn prediction (Verbeke et al., 2012). Financial institutions use ML to anticipate financial market developments (Oztekin et al., 2016a), predict the solvency of corporate borrowers (du Jardin, 2016), or inform credit approval decisions (Paleologo et al., 2010). Such applications rely on structured data such as past customer transactions, price developments, or loan repayments. It is not obvious that the success of DL in text mining, image recognition and other tasks that involve unstructured data generalizes to decision support applications where structured data prevails. Therefore, the objectives of the paper are to examine the effectiveness of DL in decision support, to test whether its feature learning ability generalizes to the structured data sets typically encountered in this field, and to offer guidance on how to setup a DL-based decision support model.

We pursue our objectives in a financial risk management context. Using data from the spread-trading market, we predict the profitability of individual traders. The modeling goal is to identify traders that pose a high risk to the market maker, and recommend a hedging policy that maximizes the marker maker's profits. Beyond the utility of such a policy for a spread-trading company, the trader risk prediction task represents challenges that are commonly encountered in ML-based decision support.

A first challenge is class imbalance. Adverse events such as borrower default or customer churn represent minorities in their populations, and this impedes ML (Verbeke et al., 2012). A second challenge called concept drift arises in dynamic environments. ML models infer (*learn*) a functional relationship between subject characteristics (e.g., previous trades of a client) and a prediction target (e.g., trader profitability) from past data. Changes in the environment render this relationship more volatile and harder to infer. The curse-of-dimensionality is another modeling challenge. Corporate data warehouses provide a huge amount of information about modeling subjects (e.g., traders) and it is difficult to learn generalizable patterns in the presence of a large number of features (Hastie, Tibshirani, & Friedman, 2009). Finally, the success of ML depends on the availability of informative features. Feature engineering is carried out manually by domain experts. Given high labor costs, a shortage of skilled analysts and the need to revise hand-crafted features in response to external changes (e.g., in trader behavior), manual feature engineering decreases the efficiency of ML and becomes an impediment to ML-based decision support.

The common denominator of the modeling challenges is that they reduce the representativeness of the training data. Being a data-driven approach, ML suffers from reduced representativeness, which suggests that the challenges diminish the effectiveness and efficiency of data-driven ML models. Considering our application setting as an example, the representation learning ability of DL could help to identify a more generic representation of the trading profile of high-risk traders than that embodied in hand-crafted features. More generality in the inferred feature-target-relationship would offer higher robustness toward external variations in trading behavior; for example, variations introduced by changes in business cycle, market conditions, company operations, etc. Replacing the need for costly manual feature engineering would also raise the efficiency of model-based decision support.

Examining the degree to which DL remedies common modeling challenges in decision support, the paper makes the following contributions. First, it is one of the first studies to examine the effectiveness of DL in conjunction with structured, individual-level behavioral customer data. Predicting individual trader's risk taking behavior, we focus on retail finance, which is a pivotal application area for operations research (Crook et al., 2007) that, to our knowledge, no previous DL study has considered. Empirical results provide evidence that DL predicts substantially more accurately than ML methods. Second, we demonstrate the ability of DL to learn informative features from operational data in an automatic manner. Prior research has confirmed this ability for unstructured data (LeCun et al., 2015). We expand previous results to transactional and behavioral customer data. This finding is managerially meaningful because many enterprises employ structured data for decision support. Third, the paper contributes to financial risk taking forecasting practice in that it proposes a DNN-based approach to effectively manage risk and inform hedging decisions in a speculative financial market. The DL methodology that we employ in the paper is not new. However, DL and its constituent concepts such as distributed representations are rarely explained in the language of business functions. Business users can benefit from an understanding of DL concepts to enable them to engage with data scientists and consultants on an informed basis. A better understanding might also lead to more appreciation of formal, mathematical models and help to overcome organizational inertia, which is a well-known impediment to fact-based decision support (Hsinchun et al., 2012; Lilien, 2011). Against this background, a final contribution of the paper is that it increases awareness of DL in business through evidencing its potential and providing a concrete recipe for how to set up, train, and implement a DNN-based decision support approach. To achieve this, we elaborate on the methodological underpinnings of DL and the decision model we devise for trader risk classification.

3.3 Related Work

The literature on DL is growing at high pace. A comprehensive overview of the state-of-the-art is available in recent surveys (LeCun et al., 2015; W. Liu et al., 2017; Schmidhuber, 2015). We focus the review of related literature to DL applications in finance. Table 6 analyzes corresponding studies along different dimensions related to the forecasting setting, underlying data, and neural network topology.

To clarify the selection of papers, we acknowledge that DL has other applications in finance beyond forecasting including index tracking (J. B. Heaton et al., 2017) or modeling state dynamics in limit-order-book data (J. Sirignano, 2016). Furthermore, DL has been applied to generate financial forecasts from textual data (Kraus & Feuerriegel, 2017). Table 6 does not include such studies as they do not concentrate on prediction or consider a different source of data.

Finally, one may argue that a recurrent neural network (RNN) is a DNN by definition, because recurrent cells exhibit temporal depth. With the rise of DL, gated RNNs such as LSTM (long short-term memory) gained popularity and are often characterized as DNNs (Fischer & Krauss, 2018b). This is not necessarily true for their predecessors, some of which have been used in finance (Huck, 2009). Table 6 analyzes studies that used contemporary gated RNNs and omits those that use earlier types of RNNs.

Table 6 shows that the majority (roughly 60%) of previous studies forecast developments in financial markets, such as price movements (Y. Deng et al., 2017), volatility (Xiong et al., 2016) or market crashes (Chatzis et al., 2018). Applications in risk analytics such as financial distress prediction (Addo et al., 2018) or credit scoring (J. A. Sirignano et al., 2016) are also popular. Considering the objectives of forecasting, columns two and three reveal that previous studies have not considered forecasting human behavior, which is the focus of this paper.

The type of input data represents a second difference between most previous studies and this paper. DNNs that forecast financial market prices typically receive lagged prices as inputs. For example, Y. Deng et al. (2017) and Fischer and Krauss (2018b) use minute- and day-level price returns as inputs. By contrast, the risk modeling task we face consists of a dynamic regression problem with different types of predictor variables (see Section 3.6.1). The *feature* columns in Table 6 show that few prior studies mix numerical and discrete input variables.

A core feature of DNNs is the ability to automatically extract predictive features from the input data (Montufar et al., 2014). One objective of this paper is to confirm the feature learning capability in a risk management context. A substantial difference in the type of input data has implications for feature learning. It is not obvious that results observed in a time series setting generalize to a dynamic regression setting with diverse input variables. With respect to risk management, we observe from the column *profit simulation* in Table 6 that most previous work has not examined the economic implications of a DL-based risk management approach; J. A. Sirignano et al. (2016) being an exception.

In addition to the application setting and input data, a third difference between most previous work and this study concerns the architecture of the DNN. Table 6 sketches the topology of previous networks in its three rightmost columns. Given our focus on forecasting studies, every network includes a supervised learning mechanism, meaning that weights in the network are trained through minimizing the empirical loss on the training data set (Bengio et al., 2016). This is typically implemented by means of a fully-connected output layer. This layer requires only one unit with a linear or softmax activation function to solve regression and classification problems, respectively. Table 6 shows that purely supervised learning networks prevail in previous work. From this observation, we conclude that more research into networks with supervised and unsupervised layers is desirable.

In total, nine studies consider unsupervised pre-training. The majority implement pre-training using a deep belief network. Long before pre-training was popularized, a seminal study proposed self-organizing maps for unsupervised time series pattern extraction (Giles et al., 2001). Stacked denoising auto-encoders (SdA), the approach we use for feature learning, have received little attention. Evidence of their effectiveness in risk analytics is originally provided in this paper.

In summary, the contribution of our work to literature emerges through a combination of characteristics concerning the forecasting setting, the data employed, and the way in which we devise and assess the DL-based forecasting model through using state-of-the-art approaches for network training and unsupervised pre-training and evaluation of the profitability of model-based hedging decisions.

The study closest related to our work is J. A. Sirignano et al. (2016). The authors estimate a DNN from a data set of over 3.5 billion loan-month observations with 272 variables relating to loan characteristics and local economic factors to support portfolio management. To that end, (J. A. Sirignano et al., 2016) model the transition probabilities of individual loans between states ranging from current over different delayed payment states to delinquency or foreclosure. Our study differs from J. A. Sirignano et al. (2016) in terms of the application setting and DL methodology.

The DL models of J. A. Sirignano et al. (2016) consists of feed-forward networks of up to 7 layers (and ensembles thereof). Deep feed-forward networks are a generalization of the three-layer networks widely used in previous work (Oztekin et al., 2016b). The DNN architecture proposed here is different. It uses multiple layers of different types of units and relies on unsupervised pre-training to extract predictive features. Pre-training elements provide distinctive advantages and have been found effective in financial applications (J. B. Heaton et al., 2017). Consequently, we further advance the methodology of J. A. Sirignano et al. (2016).

Concerning the application, the mortgage risk modeling setting of J. A. Sirignano et al. (2016) as well as conventional credit scoring settings (Crook et al., 2007) differs substantially from trader risk prediction. A credit product can be considered a put option with the lender having the right to grant credit, but no obligation to do so. Credits may also be secured by collateral and, most importantly, it is possible to hedge risks while still earning money from commissions. However, we consider a spread-trading context where the market maker is *obliged* to accept orders from its clients. These orders are similar to futures contracts with an arbitrary strike date. In addition, unlike in the money lending business where a customer will be given a credit limit, in the spread trading market, informed traders or insiders can make unlimited profits from the market marker. Consequently, the market maker faces the risk of adverse selection. At the same time, the economics of the spread-trading market require the market maker to hedge risks very selectively (because hedging quickly reduces revenues to zero). Thus, our forecasting task is to identify those traders who pose a substantial risk to the market maker.

Table 6: Summary of Related Work on DL in Finance

	Area[1]	Subject[2]	Target	Time Series	Time Window	Obser- vations	Features[3]		Horizon	Study Design	Data part.[4]	Profit sim.	Supervised Layers[5]	Unsup. Pretrain.[5]	Architecture[6]
Giles et al. (2001)	MM	Exr	Direction	5	9/1973 - 5/1987	3,645	2	con	day	rolling window	100 / 30		RNN	SOM	x-7-2-o
Shen et al. (2015)		Exr	Return	3	1976 - 2004	$< 1,000$	5	con	week	temporal split	0.7 / 0.3		FC	DBM	x-20-20-20-o
Xiong et al. (2016)		Ind	Volatility	1	10/2004 - 7/2015	2,682	25 (t3)	con	day	temporal split	0.7 / 0.3		LSTM		x-1-o
Dixon et al. (2017)		Fut	Direction	43	3/1991 - 9/2014	50	9895	con	5 min	rolling window	25000 / 12500	yes	FC		x-1000-100-100-100-o
Y. Deng et al. (2017)		Fut	Return	4	1/2014 - 9/2015	100	150	con	minute	rolling window	15000 / 5000	yes	RLRNN		x-128-128-128-20-o
Bao et al. (2017)		Ind	Return	6	7/2008 - 9/2016	2,079	19 (t4)	con	day	rolling window	2y / 1q / 1q[7]	yes	LSTM	SdA	x-10-10-10-10-10-1-1-1-1-1-o
Krauss et al. (2017)		Sto	Better S&P500	~500	1/1992 - 10/2015	380	31	con	day	rolling window	750 / 250	yes	FC		x-31-10-5-o
Zhao et al. (2017)		Co	WTI crude oil spot price	1	1/1986 - 5/2016	365	200	con	month	temporal split	0.80 / 0.20		FC	SdA	x-100-10-o
Fischer and Krauss (2018b)		Sto	Better S&P500	~500	1/1992 - 10/2015	380	1 (t240)	con	day	rolling window	750 / 250	yes	LSTM		x-25-o
Baek and Kim (2018)		Ind	Return	2	1/2000 - 7/2017	4,3	6 (t20)	con	day	temporal split	0.45 / 0.55	yes	LSTM+ LSTM+FC		x-(x1-5-3 | x2-4-2)-2-o
Chatzis et al. (2018)		Ind	Crash	2	1/1996 - 12/2017	5,4	131	con	1 | 5 days	temporal split	0.66 / 0.33		FC		x-64-32-8-2-o
H. Y. Kim and Won (2018b)		Ind	Volatility	1	1/2001 - 1/2017	3,963	6 (t22)	con	day	temporal split	0.68 / 0.32		LSTM+FC		x-10-4-2-5-o
Huck (2019)		Sto	Better S&P	300	1/1993 - 5/2015	6300	≤ 592	con	day	rolling window	504 / 126	yes	FC	DBN	x-148-74-o
B. Ribeiro and Lopes (2011)	RA	Ent	Insolvency	N.A.	2002 - 2006	1,2	30	con	Year	random split	800 / 400		FC	DBN	x-500-500-1000-o
Yeh et al. (2015)		Ent	Insolvency	N.A.	2001 - 2011	$\sim 83,000$	180	con	Year	rolling window	04.01.2001		FC	DBN	x-1000-1000-1000-o
Lee et al. (2017)		Ent	Firm perf.	22	2000 - 2015	286	15	con	Year	temporal split	10. Mrz		FC	DBN	x-200-200-200-200-o
C. Luo et al. (2017)		Ent	Rating	N.A.	1/2016 - 12/2016	661	11	mix	N.A.	cross-val.	10 fold		FC	DBN	Not specified
J. A. Sirignano et al. (2016)		Mor	Default	N.A.	1/1995 - 6/2014	$3.5 * 10^9$	272	mix	month	temporal split	214 / 19	yes	FC		x-200-140-140-140-140-o
Addo et al. (2018)		Ent	Insolvency	286	2016 - 2017	117,019	181	con	N.A.	random split	0.6/0.2/ 0.2		FC		x-50-50-1
Jurgovsky et al. (2018)	FD	CC	Fraud	2	5/2015 - 5/2015	$1.65 * 10^7$	30 (t10)	mix	N.A.	temporal split	0.43/ 0.08/ 0.49		LSTM		x-100-100-100-o

[1] MM: financial market modeling, RA: risk analytics, FD: fraud detection

[2] Exr: exchange rate of a pair of currencies, Ind: financial market index, Fut: future contract, Sto: individual stock, Co: commodity, Ent: enterprise, Mor: mortgage, CC: credit card.

[3] Number of input features and their type using abbreviations con (continuous feature) and mix (continuous and categorical features). For studies that use LSTM networks we also report the length of a time-lagged input sequence using the notation (tl) where t means time and l is the number of lags. For example, [3] consider 25 features and feed the last three observations (days) of each feature into their LSTM.

[4] Partitioning of the data for model training, validation, and testing. Fractional numbers represent percentages with respect to the size of the data set while values greater zero depict absolute numbers of observations. The notation is training set size / validation set size / test set size. Not all studies use separate validation data. Then, the two values given in the column represent training set size / test set size.

[5] RNN: recurrent neural network, FC: fully-connected layer, LSTM: Long-short-term-memory, RLRNN: reinforcement learning RNN, SOM: self-organizing-map, SdA: stacked denoising auto-encoders, DBM: deep belief network.

[6] Symbols x and o represent the multivariate input and scalar output of the network. Numbers give the size of hidden layers. For studies that use pre-training, hidden layer sizes refer to units of the unsupervised layers (e.g., DBM, SOM, or SdA). Exceptions and special cases for complex topologies exists and we elaborate on these in the discussion of the table.

[7] The notation is slightly different from other studies. The authors use a rolling window evaluation to train, validate, and test their models using daily prices from two years, one quarter, and one quarter, respectively.

3.4 Risk Taking and Behavior Forecasting in the Spread-Trading Market

Spread trading is becoming increasingly significant. Forty per cent of the £1.2 trillion traded annually on the London Stock Exchange is equity derivative related and 25 per cent of this (£120 billion) relates to spread trading (Brady & Ramyar, 2006).

Spread trading often refers to pairs trading of stocks or to trading spreads in the futures market (Huck, 2010). However, our study focuses on the form of spread trading which relates to retail *contracts for difference (CFD).* In this market, a retail investor and a market maker enter a contract related to a specified financial instrument (e.g. a share, commodity or an index) and at the end of the contract they exchange the difference between the closing and opening price of that financial instrument. Consequently, investors trade on the direction and magnitude of movements of a financial instrument. For example, a client might place a long order on the S&P500 with stake size $10 per point. If the S&P500 rises by a particular *increment*, the client makes a profit of $\$10 * increment$; otherwise s/he loses this amount. The market maker continuously quotes bid and ask prices for marketable instruments. Unlike brokers, who help clients to trade with other investors, market makers buy or sell financial instruments from their own inventory. Provided clients meet a margin requirement, they can open and close positions at any time. The market maker is obliged to accept these orders and faces the risk of adverse selection.

Forecasting which traders pose the most risk (i.e. those who are likely to make the most profit) and deciding which risks to hedge into the main market is crucial for market makers. Informed traders might take advantage of inside information and leave the market maker with positions against a market rally. In theory, the potential loss of the market maker from one trade is unbounded. For example, IG Group, the largest retail financial services provider in UK, recently lost 30 million GBP due to deficient risk control and inflexible hedging strategies. As a result of similar problems, FXCM, the largest market maker on the global spot FX market, went bankrupt[1].

The spread between quoted bid and ask prices is the main source of revenue of the market maker. For liquid markets, such as those for the S&P500 or for the USD/GBP or EUR/GDP, the spread is greater in the spread trading market than in the underlying market. However, for less liquid financial instruments (e.g. the DAX or FTSE100 index) the spread is less than that offered in the underlying market. This later situation is often faced by spread trading firms when they need to place large volume transactions into the underlying market for less liquid financial instruments. If the market maker hedges a trade, they lose the potential profit from the spread whether or not the hedging was necessary. The market maker also faces transaction costs to hedge a position, including commission and the higher spreads in some markets when they seek to hedge large volumes. Therefore, designing a predictive classification model that distinguishes A-book clients (i.e. those who pose most risk to the market maker) from B-book clients (those who pose less risk) is vital. The market maker will hedge positions from A-book

[1] See https://www.forbes.com/sites/steveschaefer/2015/01/16/swiss-bank-stunner-claims-victims-currency-broker-fxcm-bludgeoned/#7e94f5466de0

clients to protect against losses and will take the risk of the positions from B-book clients to increase profits. Typically, 90% of the total revenues come from B-book clients (Pryor, 2011).

The decision task under study is whether to hedge an incoming trade. This task translates into a classification problem, which we address through developing a DNN to predict high risk (A-book) traders. Provided the DNN learns patterns from observed trading behavior that facilitate an accurate prediction of a trader's future successes, it can assist the market maker through recommending hedging decisions and enhancing risk management in daily operations. Figure 14 illustrates the DNN-enabled hedging strategy.

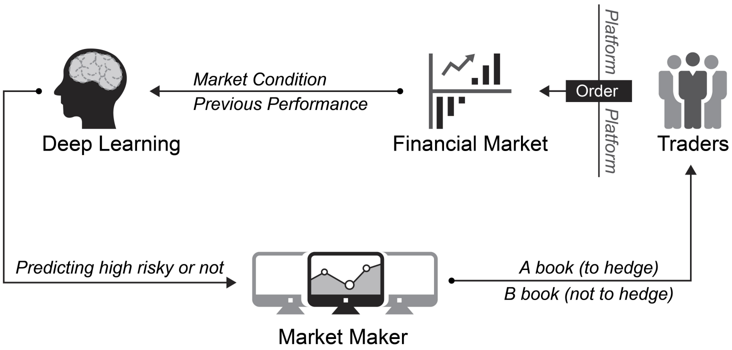


Figure 14: Workflow of how hedge strategy works for market makers

3.4.1 Trader Classification and Hedging Strategy

The definition of an A-book client is subjective and depends on the business strategy of the market maker. The company which provided the data prefers to remain anonymous (we refer to them hereafter as STX), but is a large player in the UK spread-trading market. From interviews with their front-desk dealers, who engage in day-to-day risk management, we found that STX at the time of the study, defined a client i to be a high risk trader if s/he secured a return greater than 5% from her previous 20 trades. The strategy of STX was to hedge the trades of these clients.

The deployed hedging strategy is dynamic, since STX determines the status of a client (A- or B-book) from the performance of their previous 20 trades. Therefore, client status can change due to a single trade. Accordingly, we frequently observe a situation where STX takes the risk of trade j of client i while hedging against trade $j+k$ of client i. In a speculative market, the overall return of a set of past trades can give misleading guidance to the future profitability of a trader. For example, a skilled trader, who follows a consistent strategy, shows high trading discipline, routinely uses and updates stop-loss limits, etc., can regularly lose money due to the randomness of the environment. Similarly, a poor trader, who violates all the above principles, occasionally makes a profit. This suggests that a trader's past performance is not necessarily a reliable signal of their true ability. Consequently, the goal of developing a client classification model is to generate a superior signal for hedging decisions by accounting for all other characteristics available in the data.

We develop a DNN to learn the latent nature of a trader from past trading data. The target concept, trader ability, is highly variable, corrupted by noise, and difficult to accommodate in a pre-defined, static set of trader characteristics. Therefore, it will be important for the DNN to distill, from transactional data, high-level distributed representations of the target concept, which capture the underlying generative factors that explain variations in trading behavior. In this regard, success in trader classification will evidence the ability of DL to automatically extract informative features.

3.4.2 Trader Behavior Prediction and Decision Support

It is not obvious that representation learning is effective in risk management. Applications such as, credit scoring, churn prediction or trader classification involve the forecasting of human behavior. One would expect the maximal attainable accuracy in a behavior forecasting model to be less than in, e.g., face detection. For example, the prediction target is less clearly defined (e.g., STX used a 5% threshold but this is subjective) and the feature-target relationship is typically weak. Our trader behavior forecasting study aims to clarify the potential of representation learning and DL in decision support.

We argue that the prediction task is representative of a range of modeling challenges in decision support because it exhibits several characteristics that often occur in corporate applications of data-driven prediction models. More specifically, we face challenges that diminish the representativeness of the training data. First, in response to previous gains and losses and changes in the macro-environment, the behavior of individual traders can be variable, erratic and dynamic. Second, detailed, time-ordered information about individual traders, asset prices and their underlying fundamentals and broader indicators of market sentiment (e.g., economic growth) are readily available, which leads to high dimensionality. Third, the specific way in which variables relate to each other and govern traders' profits is complex, nonlinear, and likely to evolve over time. Automatic feature extraction, if successful, is a promising way to cope with these challenges. Fourth, the spread trading setting displays class imbalance in that only a few traders succeed in securing systematic positive returns above 5%, while the vast majority of clients lose money. Last, effective risk management requires accurate predictions at the level of an individual trader. Accuracy is a general requirement in predictive decision support.

3.5 Methodology

In view of the scarcity of DL applications in the risk analytics literature, we revisit principles of DL and detail how we configure the DNN to classify spread traders into A- or B-book clients. The online Appendix elaborates on these concepts and DNN training.

3.5.1 Principles of Deep Learning

DL aims at learning multiple levels of representations from data, where higher levels represent more abstract concepts. A deep architecture with multiple layers of abstraction and its ability to learn distributed representations provides several advantages over conventional *shallow* ML methods and we discuss these below.

The deep architecture ML methods learn a functional relationship between variables, which characterize the relationship between an observation and a prediction target. High variability of this function complicates the ML approach and may lead to poor generalization. Sources of high variability include external shocks to the environment in which a decision model operates. Learning theory suggests that to represent a functional relationship, a learning machine with depth k needs exponentially more computational units than a machine with depth $k+1$ (Montufar et al., 2014). The depth of commonly-used machine learning methods is as follows (Bengio, 2009): linear and logistics regression (depth 1); boosting and stacking ensembles: depth of base learner (depth $+1$, one extra layer for combining the votes from base learners); decision trees, one-hidden-layer artificial neural networks (ANNs), support vector machines (depth 2); the visual system in the human brain (depth 5-10, (Serre et al., 2007)).

The concept of depth explains a large number of empirical findings related to, for example, ANNs or support vector machines outperforming simple regression models or ensemble classifiers outperforming individual learners (Lessmann et al., 2015). Increased depth allows these methods to implicitly learn an extra level of representation from data (Bengio, 2009). Additional levels facilitate generalization to new combinations of the features, which are less represented in the training data. Enlarged capacity also allows the learning machine to capture more variations in the target function, which discriminates classes accurately. Furthermore, the number of computational units a model can afford is severely restricted by the number of training examples. As a result, when there are variations of interest in the target function, shallow architectures need extreme complexity (large amounts of computational units) to fit the function. Consequently, they need exponentially more training examples than a model with greater depth (Bengio, 2009).

Distributed Representations DL methods learn distributed representations from data. An example of a distributed representation is principal component analysis (PCA). PCA re-orients a data set in the direction of the eigenvectors, which are ordered according to their contribution to explained variation. This is a distributed representation where the raw variables collaborate to generate a principle component. In predictive ML, principle components can replace the original variables. The functional relationship to learn is then that between the target variable and the principle components. This can simplify the learning task, increase predictive accuracy, and facilitate feature reduction (Ulaş et al., 2012). However, ML methods learn local, non-distributed representations. Using the raw variables in a data set, they partition the input space into mutually exclusive regions. For example, support vector machines infer a decision boundary from the local training examples of adjacent classes that are closest to each other.

The goal of ML is to classify novel examples, which are not part of the training set. However, the training data may lack representativeness (e.g., because of a change in the environment). An advantage of distributed representations is that they are better able to accommodate new observations that the training data does not represent well. Consider our trader classification problem as an example: Traders exhibit different trading styles; they use different strategies, follow different stop-loss rules, etc. Assume traders are split into 5 different clusters, with traders in the same cluster sharing a trading style. Using a non-distributed representation, we need 5 different features to exclusively represent each cluster, $0 = 00000, 1 = 01000, ..., 4 = 00001$. A distributed representation requires only $\lceil \log_2 5 \rceil = 3$ features to model the clusters (as a binary code), $0 = 000, 1 = 001, ..., 4 = 100$. Using three distributed features, this representation can also accommodate a new type of trader (i.e., using trading strategies that have not been employed in the training sample): $5 = 101$. This exemplifies an advantage of distributed representations, namely that the number of patterns that the representation can distinguish grows exponentially with the number of features. However, for non-distributed representations, this number grows, at best, linearly.

3.5.2 Building the Deep Neural Network

DL methods consist of multiple components with levels of non-linear transformations. A typical instance is a neural network with several hidden layers (Krauss et al., 2017; J. A. Sirignano et al., 2016). Training a DNN requires solving a non-convex optimization problem, which is difficult because of the vanishing gradient problem (Bengio et al., 2007). Gradient vanishing prohibits propagating error information from the upper layer back to lower layers in the network, so that connection weights in lower layers cannot be adapted (Larochelle et al., 2009). As a result, the optimization will often terminate in poor local minima. Remedies to this problem include unsupervised pre-training, parametric Rectifier Unit (ReLu), Xavier initialization, dropout, and batch normalization. We take advantage of these techniques to develop a trader classification DNN for risk management. Below, we introduce pre-training and dropout. Interested readers find a similar description of the other concepts in the online Appendix.

Unsupervised Pre-Training The goal of pre-training is to find invariant, generative factors (i.e., distributed representations), which explain variations in the data and amplify those variations that are important for subsequent discrimination. Through a sequence of non-linear transformations, pre-training creates layers of inherent feature detectors without requiring data labels. This facilitates a local learning of connection weights. Avoiding a propagation of error information through multiple layers of the network, pre-training helps to overcome the vanishing gradient problem. Stacking multiple layers of progressively more sophisticated feature detectors, the DNN can be initialized to sensible starting values. After discovering a structural relationship in the data, one can then add a supervised learning technique (e.g., logistic regression) on top of the pre-trained network and tune parameters using back-propagation. Unsupervised pre-training, where the use of the target label is postponed until the fine-tuning stage, is especially useful in management decision support where class imbalance is a common problem (Paleologo et al., 2010).

Two classical implementations of pre-training are deep belief networks (DBN), which are pre-trained by restricted Boltzmann machine (G. Hinton et al., 2006), and stacked denoising autoencoders (SdA), which are pre-trained by the autoencoder (Bengio et al., 2007). Both strategies minimize an approximation of the log-likelihood of a generative model and, accordingly, typically show similar performance (Bengio & Delalleau, 2009; Vincent et al., 2008). This, together with the fact that deep belief networks have already received some attention in the risk analytics literature (see Table 6), led us to use the framework of the stacked denoising autoencoder (Vincent et al., 2008).

Denoising Autoencoder The denoising autoencoder (dA) learns a distributed representation (namely the "code") from input samples. Suppose we have N samples and each sample has p features. Receiving an input $\mathbf{x} \in \mathbb{R}^p$, the learning process of a dA includes four steps:

Step 1: *Corruption.* The dA first corrupts the input $\mathbf{x}$. By sampling from the Binomial distribution $(n = N, p = p_q)$, (where the corruption rate q_p is a hyper parameter that needs tuning outside the model), the dA randomly corrupts a subset of the observed samples and deliberately introduces noise. For example, if the input features a binary, corruption corresponds to flipping bits.

Step 2: *Encoder.* The dA deterministically maps the corrupted input $\widetilde{\mathbf{x}}$ into a higher-level representation (the code) $\mathbf{y} \in \mathbb{R}^k$. The encoding process is conducted via an ordinary one-hidden-layer neural network (the number of hidden units k is a hyper parameter that needs tuning outside the model). With weight matrix W, biases b, and encoding function $h(\cdot)$, e.g., *sigmoid* function, $\mathbf{y}$ is given as:

$$\mathbf{y} = h(W \cdot \widetilde{\mathbf{x}} + b) \tag{12}$$

Step 3: *Decoder.* The code $\mathbf{y}$ is mapped back by a decoder into the reconstruction $\mathbf{z}$ that has the same shape as the input $\mathbf{x}$. Given the code $\mathbf{y}$, $\mathbf{z}$ should be regarded as a prediction of $\mathbf{x}$. Such reconstruction represents a *denoising* process; it tries to reconstruct the input from a noisy (corrupted) version of it. Similar to the encoder, the decoder has the weight matrix $\widetilde{W}$, biases $\widetilde{b}$, and a decoding function $g(\cdot)$. The reconstruction $\mathbf{z}$ is:

$$\mathbf{z} = g(\widetilde{W} \cdot \mathbf{y} + \widetilde{b}) \tag{13}$$

Step 4: *Training.* Optimizing the parameters of dA (W, b, $\widetilde{W}$, $\widetilde{b}$) involves minimizing the reconstruction error $L_{(\mathbf{x},\mathbf{z})}$; achieved by letting the code $\mathbf{y}$ learn a distributed representation that captures the main factors of variation in $\mathbf{x}$. Theoretically, if we use the mean squared error ($L_{H(\mathbf{x},\mathbf{z})} = ||\mathbf{x} - \mathbf{z}||^2$) as the cost function and linear functions as both encoder $h(\cdot)$ and decoder functions $g(\cdot)$, the dA is equivalent to PCA; the k hidden units in code $\mathbf{y}$ represent the first k principal components of the data. The choice of cost function depends on the distributional assumptions of input $\mathbf{x}$. In this paper, we measure the reconstruction error by the cross entropy loss function, as most of our features are probabilities $\mathbf{x} \in [0, 1]^p$. In addition, we incorporate an L2 penalty (also called weight decay (Krogh & Hertz, 1992)). This is equivalent to assuming a

Gaussian prior over the weights and a common approach to encourage sparsity among weights and improve generalization. The regularization parameter λ captures the trade-off between reconstruction error and model complexity. The parameter needs tuning outside the model and offers a way to protect against overfitting. Higher values of λ penalize model complexity more heavily and, ceteris paribus, reduce the risk of overfitting. The final cost function is:

$$L(\mathbf{x}, \mathbf{z}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{p}[x_{ik}\log z_{ik} + (1 - x_{ik})\log(1 - z_{ik})] + \lambda \left\| W \right\|_2 \tag{14}$$

Several solvers (e.g., stochastic gradient descent) are available to carry out the optimization.

$$\arg \min_{w,\widetilde{w},b,\widetilde{b}} L(\mathbf{x}, \mathbf{z} \mid \Theta) \tag{15}$$

Step 5: *Stacking.* Once a dA has been trained, one can stack another dA on top. Layers are organized in a feed-forward manner. The second dA takes the encoded output of the previous dA (the code $\mathbf{y}$) as its new inputs $\mathbf{x}$. Each layer of dA is trained locally, finding its own optimal weights regardless of the other layers. Iteratively, a number of dAs can be stacked upon each other to construct a stacked denoising autoencoder (SdA). The encoding weights of each dA can then be treated as initializations for the network in the next step. Figure 15 illustrates the working flow of dA.

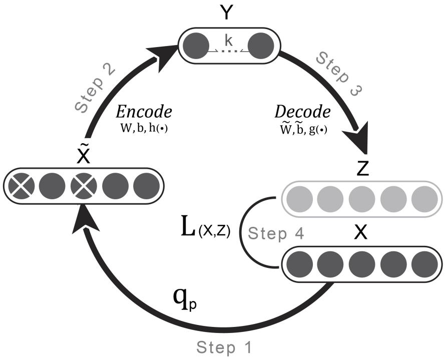


Figure 15: Architecture of denoising auto-encoder.

Supervised fine-tuning The SdA can be trained in a feed-forward, layer-wise manner. To employ the network for prediction, network training continues with supervised fine-tuning that teaches the DNN which types of trading behaviors (in the form of distributed representations) identify A-book clients. To that end, we add a *softmax* regression on top of the SdA. This way, we solve a supervised learning problem using the distributed representation of the raw input as features (which the SdA output embodies), and a binary indicator variable as target, which indicates whether a trade should be hedged. Formally, with parameter weight W and bias b, the probability that a trade $\mathbf{x}$ belongs to class i is:

$$
\begin{aligned}
P(Y = i|x, W, b) &= \text{softmax}_i(Wx + b) \\
&= \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}
\end{aligned} \tag{16}
$$

We employ the negative log-likelihood as cost function in supervised fine-tuning. Suppose $y^{(i)}$ is the true class for the input $x^{(i)}$, the cost function then states:

$$
L(W, b, \mathbf{x}) = -\sum_{i=1}^{N} \log(P(Y = y^{(i)}|x^{(i)}, W, b)) \tag{17}
$$

Protecting Against Overfitting Using Dropout Regularization Neural networks are vulnerable to overfitting. To prohibit the DNN emphasizing idiosyncratic patterns of the training data and protect against overfitting, our DNN includes a dropout layer behind each hidden layer. Figure 16 depicts the concept of dropout. During DNN training, hidden layer neurons and their corresponding connection weights are removed from the network. This is done for each batch of training samples in an iteration. The gradients contributed by that batch of samples also bypass the dropped-out neurons during back-propagation (see the online Appendix for a detailed explanation of DNN training). The probability of a hidden neuron being dropped out follows a Bernoulli distribution with a given dropout rate.

A DNN trained with dropout mimics the behavior of an ensemble model. When calculating predictions, the DNN considers all hidden layer neurons but multiplies the connecting weights of each hidden neuron by the expectation of the Bernoulli distribution. This way, although training a single DNN with N hidden neurons, the prediction of the DNN implicitly integrates predictions of 2^N candidate networks with different combinations of hidden neurons. More formally, dropout simulates a geometric model averaging process; each possible combination of hidden neurons is considered, which is the extreme case of bagging. Model combination is known to increase predictive accuracy (Lessmann et al., 2015; Verbeke et al., 2012).

Dropout also acts as a regularizer in that it effectively removes random weights from training, which prevents hidden neurons from co-adapting to each other. Moreover, model averaging reduces variance, which, via the bias-variance decomposition, reduces forecast error. Controlling the complexity of a DNN, dropout helps to protect against overfitting. Theoretical details on dropout and how it prevents overfitting can be found in Srivastava et al. (2014).

Recall that we augment dropout through using an $L2-$penalty during SdA training to increase the robustness of the DNN toward overfitting. For the same reason, we make use of early-stopping

Network Training and Configuration Our DNN involves unsupervised pre-training using SdA. We tune weights in a layer-wiser manner and then fine-tune the DNN as a whole in a supervised way, with each hidden layer followed by a dropout layer. In addition, we use several other DL

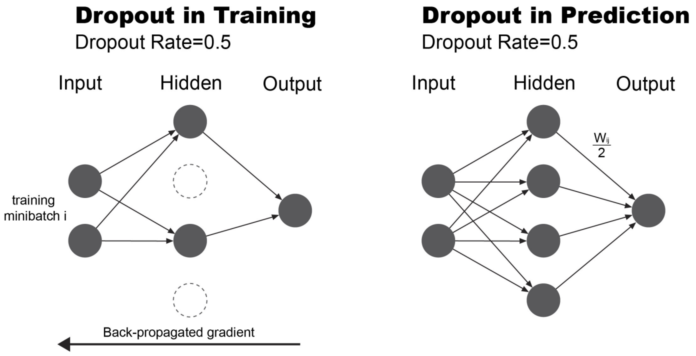


Figure 16: Principle of dropout in training and predicting.

concepts to protect against overfitting and simplify the network training process including Xavier's initialization, batch normalization layers and using ReLU as the activation function. Previous work on DL has elaborated on these concepts and established their utility (Bengio et al., 2016), and we detail them in the online Appendix. Figure 17 summarizes the overall architecture of the DNN employed for trader risk classification.

The parameters to determine in the pre-training stage are the weight matrix and bias in each dA (both the encoder and the decoder). The parameters in the supervised fine-tuning stage are the weight matrix and bias in each encoder of SdA and in the *softmax* regression. We use stochastic gradient descent with momentum and a decaying learning rate for DNN training. The online Appendix provides an explanation of these concepts and motivates our choices . In particular, Algorithm 1 in the online Appendix provides a fully-comprehensive description of network training using pseudo-code. Section 2 of the online Appendix also elaborates on our approach to decide on DNN hyper-parameters such as the number of hidden layers in SdA, and how we tune these using random search (Bengio, 2012).

The techniques we employ are available in DL software packages, which facilitate defining the topology of a DNN, provide routines for numerical optimization to train the DNN, and offer the functionality to apply a trained model for forecasting. We use the Python library *Theano*, which is a GPU-based library for scalable computing. The GPUs used for experiments were *Nvidia Tesla K20m* with 2496 cores and 5GB GDDR5 high bandwidth memory each. We observe this infrastructure to provide a 10-15 times improvement in speed over training a DNN using traditional CPUs for DNN training (which equates to reducing run-times from more than a week to 1-2 days). In appraising these figures it is important to note that i) large run-times result from the size of the data set, and that ii) training complex ML models may be as costly. For example, depending on the specific configuration, training a random forest classifier on the spread trading data can easily require more than 3 days.

3.6 Experimental Design

The following sub-sections describe the spread-trading data set and elaborate on the definition of A-book clients, and introduce model evaluation criteria and benchmark classifiers.

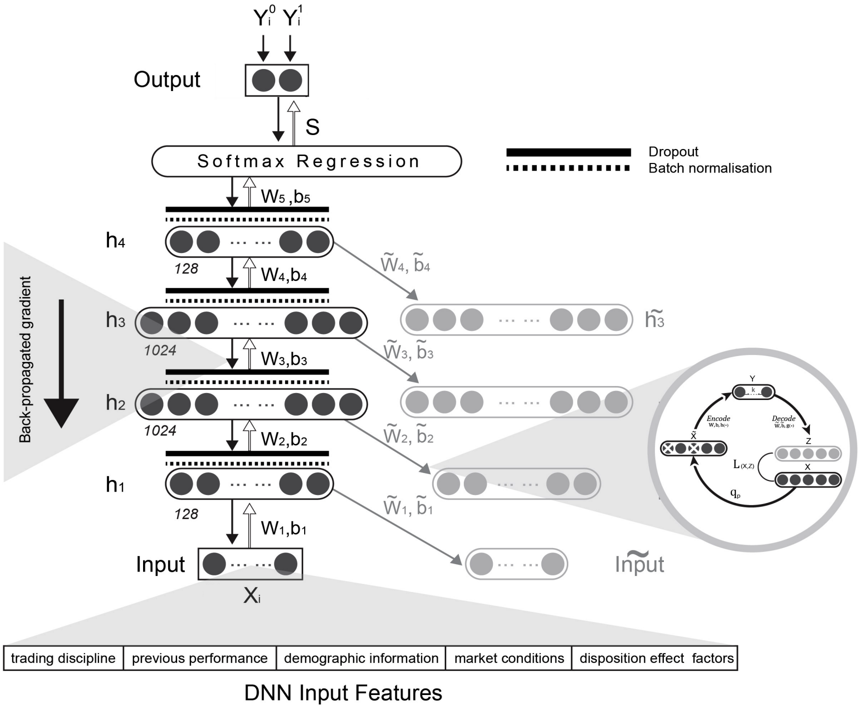


Figure 17: Topology of the deep network employed in this study. Stacked denoising auto-encoder with 4 hidden layers with $128, 1024, 1024, 128$ hidden units each. The output layer predicts class membership probabilities based on the output of the last dropout layer using the softmax function.

3.6.1 Dataset and Target Label Definition

STX provided 11 years of real-life trading data for the period November 2003 to July 2014. Overall, the data includes the trades of $25,000$ active traders (over 30 million trades across 6064 different financial instruments). To prepare the data for analysis, we replaced missing values using EM imputation and *Chebyshev*'s method for outlier treatment (Hastie, Tibshirani, & Friedman, 2009).

Supervised learning requires a labeled data set $D = \{y_j, x_j\}_{j=1\ldots n}$, where x_j is a vector of features that characterize trade j, $n = 30$ million is the total number of trades in the data, and y_i denotes the target variable. However, data characterizing an individual trade is limited. Relating trades to their corresponding traders facilitates enriching the set of features by using information from previous trades $j - k$ to decide on trade j.

The decision task of STX is whether to hedge trade j. Therefore, we consider a binary target:

$$y_{ij} = \begin{cases} +1 \rightarrow \text{hedge} & , \text{ iif Return}_i \geq 5\% \\ -1 & , \text{ otherwise} \end{cases} \tag{18}$$

$$\text{with} \tag{19}$$

$$\text{Return}_i = \frac{\sum_{20<j\leq 100} \text{P\&L}_{i,j}}{\sum_{20<j\leq 100} \text{Margin}_{i,j}}$$

where i, j index trader i and trade j, respectively, $P\&L$ is the profit and loss of trade j, and *Margin* is the amount of money required by the market maker in order to place the order, which normally equals the stake size times the margin requirement. To label trade j, we determine the status of trader i at the time of issuing that trade. We define trader i to be an A-book client if s/he secures a return above 5% from her next hundred trades subsequent to j. Recall that the 5% threshold mimics the current policy of STX. We also sustain the STX approach to hedge all trades from A-book clients. However, our method to define the client status and label their trades is forward looking whereas STX considers the past profits of trader i. Our target label definition is also dynamic in that the trader status can change with every trade. According to that definition, 6.43% of the trades in the data set come from A-book clients and should be hedged.

Of course, at the time when STX receives trade j, the future profits of trader i are unknown. Therefore, we develop a prediction model to forecast y_{ij} from the information the company can observe at the time when trade j is made. The feature vector x_{ij} includes demographic information of the client making trade j and information concerning the client's trading behavior for the 20 trades prior to trade j. The decision to consider the past 20 trades is based on the hedging policy of STX, which uses a rolling window of the 20 trades prior to trade j to decide on the status of the client.

3.6.2 Trader Characteristics and Feature Creation

We create variables for trader classification based on interviews with experienced members of the dealing desk of STX. A first round of interviews was aimed at identifying risk factors that domain experts deem indicative of good/bad traders. Based on corresponding results, we developed a semi-structured survey that was presented to seven members of the dealing desk in a second round of interviews. The survey asked participants to evaluate behavioral traits, which emerged from the first round, on a Likert Scale from 1-7, where values of 1 and 7 represent a strong indication for bad and good trading behavior, respectively. After completing the survey, we asked participants to suggest strategies they would apply if trading the FTSE100 index and a single stock from the FTSE100, respectively. This was to gather ideas for novel factors not yet covered in the survey. The results of the interviews guided the feature engineering. A non-disclosure agreement with STX prohibits formally defining all features. However, the following description provides a comprehensive overview of the type of features and how they have been created. The features reflect the specific situation of STX. Risk analysts may find the following description useful to inform feature engineering in related applications. However, since our study focuses on the application of a DL methodology, it does not warrant claims related to generalizability of features. In general, features split into five groups. The first group comprises trader *demographics* such as age, country of origin, post code, employment status and salary group. Features of this group are nominal and enter ML models in the form a dummy codes. STX employs a range of socio- and micro-geographic data to cluster post codes. They follow a similar logic to cluster countries. [2]

Features of the second group capture the *past performance* of traders. We use aggregations such as the mean and standard deviation to calculate corresponding features over a rolling window of 20 previous trades relative to the focal trade. The choice of a window size of 20 follows STX's hedging policy at the time of the study. In addition to profitability, we compute a set of related performance indicators such as the average win rate, average number of points in profit, whether a client has been in profit, etc. We also consider the risk adjusted return (i.e., Sharpe ratio (Dowd, 2000) and features related to the number and sizes of past withdrawals and deposits).

A third group of features describes traders' preferences related to *markets* and *channels*. For example, one feature simply counts the number of markets in which a trader invests while another encodes whether traders showed a strong preference for a specific market in their previous 20 trades. Using this information, we create features describing the most popular market cluster in a trader's full history and last 20 trades, respectively. The subgroup of channel preferences includes features that count the number of opening and closing trades made through the STX web front-end and mobile app, respectively, as well as ratios derived from these counts.

Results of the survey identified the *disposition effect* as a relevant factor to detect poor traders. The disposition effect (Weber & Camerer, 1998) describes the phenomenon that investors tend to quickly sell trades that are in profit but are reluctant to sell trades in loss. Features of the fourth group strive to capture the disposition effect. We determine per trader the average amount and time s/he leaves winning and loosing positions open, and calculate their ratio. We also consider sums instead of averages and window lengths of the previous 20 and all previous trades.

Another discriminating factor that emerged from the interviews concerns *trading discipline*. Members of the dealing desk pointed out that good traders display a tendency to set manual limits (stop losses and profit levels) and when making profits to move these with the market. The fifth group of features captures signals concerning the consistency of a trader's strategy. The variation index of stake sizes exemplifies corresponding features. We also consider simpler features such as the standard deviation of stake sizes and features that capture the frequency of trades as well as their variation. Other features in this group relate to the tendency of clients to trade within/outside of normal trading hours (e.g., number and share of corresponding trades), which we consider an indicator of traders' professionalism. The degree to which traders partially close trades may also signal expertise and hence traders posing a higher risk. Hence, we create a feature measuring the share of trades that have been closed in the previous 20 trades. The previous examples sketch the type of features we employ. Using operations such as varying window sizes, aggregation functions, creating dummy features through comparing a feature to a threshold (e.g., whether any of the last 20 trades has been closed using the mobile app), and

[2] STX has not revealed details of their cluster mechanisms to us. However, they assured us that the clustering does not employ any information of trader profits, which might otherwise introduce a look-ahead bias through leaking information from the prediction target to the features.

considering bi-variate interactions, we obtain a collection of close to 100 features. An objective of the paper is to test whether the DNN can learn predictive higher-level features automatically. For example, the discussion on feature engineering suggests multicollinearity among features, which feature selection could remedy. However, a sub-goal following from our objective is to test how effectively the DNN automatically discards redundant and irrelevant features. Therefore, we do not perform feature selection.

3.6.3 Exploratory Data Analysis and Feature Importance

To shed light on how A-book and B-book clients differ across the features, we report results of an exploratory data analysis. Table 7 reports descriptive statistics for the ten most informative features for A-book and B-book clients, respectively. We select these features according to the Fisher score (Verbeke et al., 2012). Features with the suffix 20 are calculated over a window of 20 past trades relative to a focal trade. For example, given a trade j (equivalent to one observation in the data set) from a trader i, we consider the $j-1, j-2, ...j-20$ trades of trader i and calculate their mean, standard deviation, etc. We use all available trades of a trader if s/he has less than 20 trades. In interpreting the results of Table 7 it is important to note that STX rescaled numeric features to the zero-one interval using min/max scaling. Rescaling is a common data preprocessing approach and ensures comparability of feature values. In addition, it helps to protect the confidentiality of the data.

Table 7 reveals that differences between the client groups in the means of variable values are small. This indicates that good and bad traders cluster in the behavioral space spanned by these features and that a classification of traders using these features will be challenging. To support this view, we estimate a logistic regression model on the training set using the features of Table 7 and observe a McFadden R^2 close to zero. Considering standard deviations, Table 7 suggests that the trading behavior of B-book clients is slightly more volatile compared to A-book clients, which supports findings from the interviews that good traders follow a consistent strategy. Table 7 also emphasizes the disposition effect as a potentially discriminating factor. Several of the top ten features aim at capturing the disposition effect through contrasting the duration with which traders keep winning versus losing positions. Last, the third and fourth moment of the feature distributions hint at some differences between good and bad traders. However, as shown by the failure of the logistic regression, translating these differences into a classification rule is difficult and may be impossible with a linear model.

To further inspect the origin of close to random performance of logistic regression (on all features) and to gain more insight into the feature-response relationship, we also examine feature importance using a random forest (RF) classifier. Feature importance scores extracted from tree-based ensemble classifiers are a popular way to quantify the relative impact of features on the response variable (Hastie, Tibshirani, & Friedman, 2009). Figure 18 depicts the distribution of RF-based normalized importance scores for the first fifty features (ordered in terms of importance); the remainder being omitted to ensure readability. We highlight those features that have previously been identified as important by the Fisher-score.

Table 7: Descriptive Statistics of Top-Ten Features

Feature	**Mean**		**Std.Dev.**		**Skew**		**Description**
	A-book	**B-book**	**A-book**	**B-book**	**A-book**	**B-book**	
ProfitxDur20	0.325	0.332	0.172	0.178	0.994	0.962	Interaction of ProfitRate20 and DurationRate20
SharpeRatio20	0.443	0.446	0.081	0.085	1.097	1.131	Mean/st.dev. of returns
ProfitRate20	0.496	0.504	0.241	0.248	0.346	0.328	Average profit rate of client
WinTradeRate20	0.621	0.626	0.203	0.207	-0.203	-0.210	Client's average winning rate
AvgOpen	0.534	0.539	0.218	0.228	-0.345	-0.311	Average of the P&L among trader's first 20 trades
DurationRate20	0.319	0.322	0.119	0.121	-0.148	-0.161	Average time client leaves winning vs losing position open
PerFTSE20	0.251	0.244	0.357	0.353	1.151	1.197	Share of trades placed in the FTSE100
DurationRatio20	0.127	0.128	0.067	0.070	3.398	3.812	Mean trade duration (mins) / std.dev. trade duration
AvgShortSales20	0.487	0.482	0.269	0.274	-0.027	-0.018	Share of short positions
PassAvgReturn20	0.502	0.503	0.052	0.057	-0.295	0.065	Average return up to the last 20 trades

Figure 18 reveals differences between the variance adjusted comparison of group means, which the Fisher-score embodies, and the RF-based ranking. For example, the strongest feature according to Table 7, *ProfitxDur20*, does not appear in Figure 18 and the highest rank that a feature of Table 7 achieves in Figure 18 is ten as observed for the feature capturing a trader's average over the last twenty trades prior to the decision point. Interestingly, this feature, *PassAvgReturn20*, is the one that STX use in their hedging policy.

RF generates importance scores through comparing (out-of-bag) classification performance on the original data and that data after corrupting one feature through adding random noise. The magnitude of the performance decrease captures the importance of the corrupted feature (Hastie, Tibshirani, & Friedman, 2009). This implies that RF assesses the importance of one feature vis-a-vis all other features, whereas the Fisher-score assesses one feature at a time. Given the different mechanism to measure importance, some differences between the RF and Fisher-score ranking are to be expected. It is still surprising that the most important features of the latter receive relatively low ranks in Figure 18. An interpretation of this result is that it evidences intricate dependencies between the binary response and features, which the Fisher-score does not capture. This interpretation agrees with the poor performance of the logit model. As detailed in Section 3.7.1, the performance of the logit model improves but remains inferior to more expressive nonlinear classifiers after accounting for multicollinearity.

With respect to the complexity of the feature-response relationship, the distribution of importance scores in Figure 18 may be considered evidence of a set of three to four features being particularly strongly related to the response. We caution against this interpretation. The distributional shape is a consequence of the scaling of the y-axis, to ensure readability of the figure. The magnitude of importance scores is small, even for the left-most features. Therefore, importance differences between features (e.g., feature four and five) appear more substantial than they are. Recall that the scores capture the degree to which RF performance decreases if

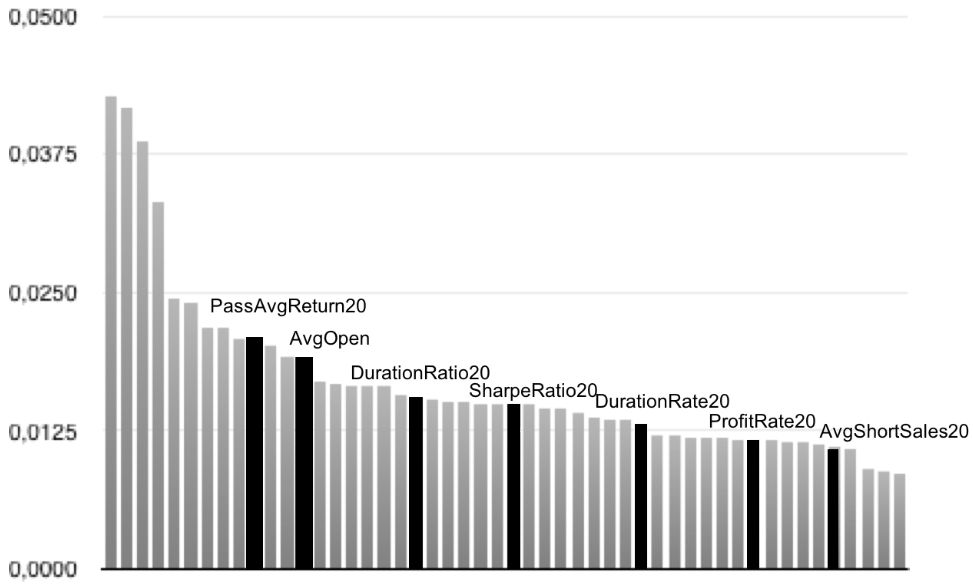


Figure 18: Normalized variable importance scores based on RF-classifier for the top 50 features. Dark color identifies features that also appear in the Fisher-score ranking (Table 7)

we corrupt one feature. Given the magnitude of importance scores, we interpret the results of Figure 18 as evidence of a low signal between the raw features and the future success of a trader. This emphasizes the trader classification task to be challenging. Even a powerful RF classifier, often observed to predict accurately (Krauss et al., 2017; Lessmann et al., 2015; Verbeke et al., 2012), fails to identify strong dependencies among the raw features and the target. Low importance scores also question representativeness of the training data. This motivates our analysis whether a DNN, equipped with higher depth than RF, is able to learn more abstract, latent, features that enable predicting traders' future performance more accurately than conventional 'shallow' learners.

We complete the analysis of feature importance by aggregating importance scores across the main feature groups in Figure 19. The analysis offers insight as to the relative importance of different types of trader characteristics. The results displayed in Figure 19 agree with the views of STX dealing desk members. We find trader demographics and features in the markets & channels category to carry least weight, which reinforces the view that propensity for risk taking may be attributed to the competence and trading style rather than particular country of origin or gender. Past performance and trading discipline are most important for high risk trader identification, substantiating the claim that features capturing the professional behavior of traders are of primary value for the task at hand.

3.6.4 Data Organization, Evaluation Criteria and Benchmark Classifiers

Testing the predictive performance of ML models requires assessing the accuracy of their forecasts on hold-out data not used during training. Several strategies for data organization exists. We employ n-fold cross-validation, which involves randomly partitioning the data into n folds

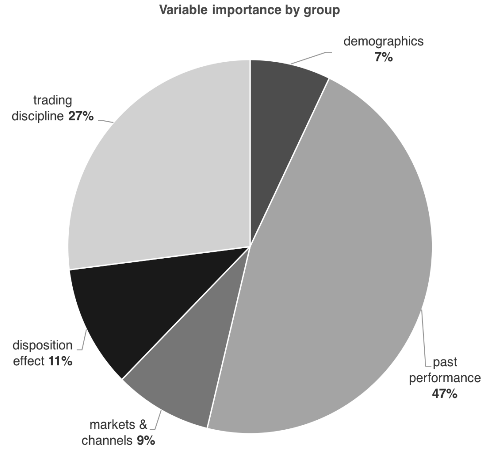


Figure 19: Analysis of group-level feature importance. The aggregation is performed by adding up the RF-based importance scores of all features belonging to the same group and normalizing group-level scores to sum to unity.

of approximately equal size, training a model on the union of $n-1$ of these folds, and assessing the performance of the resulting model through comparing actual classes to model-based class probability predictions on the remaining fold. Repeating model building and assessment n times increases the robustness of results compared to a single partitioning of the data into one training and one test set. We consider settings of $n = 10$ and $n = 5$ in subsequent comparisons. [3]

The client classification problem exhibits class imbalance and asymmetric error costs. Hedging a trade that eventually leaves the trader with a loss diminishes the profit margin of the market maker. Failing to hedge a high risk trade is far more severe and may leave the market maker with a very large loss. To reflect this asymmetry, we evaluate a classification model in terms of the profit or loss (P&L) that results from hedging trades according to model predictions.

The P&L assesses classification performance in that it is based on discrete class predictions. Taking cost asymmetry into account, it is a more suitable performance indicator than conventional metrics such as classification error, the F-measure, or others. However, to augment the P&L-based evaluation, we also assess classification models in terms of the area under a receiver-operating-characteristics curve (AUC). The AUC is equivalent to the Mann-Whitney-Wilcoxon U statistic. Considering a randomly chosen A-book client and a randomly chosen B-book client, the AUC approximates the probability that a classifier assigns a higher score to the A-book client (Hand, 2009). In this interpretation, we use the term *score* to refer to the classifier-estimated probability of a client being a high risk trader. A notable feature of the AUC is that it captures the discriminatory ability of a classifier to rank order cases in the right order; for example, assigning higher (lower) probabilities to A-book (B-book) clients. The evaluation is independent from a classification threshold and the degree to which probabilistic

[3] The computationally simpler train-test split setup was considered in preliminary experiments to identify suitable benchmark classifiers and examining the impact of class imbalance on these classifiers and the DNN. Interested readers find corresponding results in the online Appendix.

predictions are well-calibrated (Beque et al., 2017). Hence, the AUC assesses the model from a different angle than the P&L.

To compare the performance of our DNN to benchmarks, we select four ML classifiers as benchmarks, including logistic regression, ANNs, RF, and adaptive boosting. A comprehensive description of the classifiers is available in, e.g., Hastie, Tibshirani, and Friedman (2009). We report the hyper-parameter settings that we consider during model selection in Section 2 of the online Appendix, where we also elaborate on hyper-parameter tuning.

3.7 Empirical Results

The empirical analysis compares the proposed DNN to benchmark classifiers and rule-based hedging strategies that embody domain knowledge.

3.7.1 Predictive Accuracy of the DNN and ML-based Benchmark Classifiers

We first present results concerning the predictive performance of different classifiers in Table 8. The AUC assesses models in terms of their ability to discriminate A- and B-book clients whereas P&L values capture the profitability of a model-based hedging policy. To ensure comparability across folds, we normalize the total P&L observed in one cross-validation fold by the number of traders in that fold. For example, a value of 296 GBP in the first fold in the base scenario where STX does not hedge against any trade indicates that the average client loses this amount of money from trading with STX, which is equivalent to the profit of STX.

Table 8: DNN Performance vs. Benchmarks in terms of P&L and the AUC

	Cross-validation folds										
	1	2	3	4	5	6	7	8	9	10	mean
	Average P&L per trader in GDP										
no hedging	2268	2130	1601	2122	2230	2536	1785	1938	1870	2306	2079
DNN	2245	2906	3490	2863	2245	3536	2679	3536	3014	2792	2931
Logit	3021	2281	2901	2707	3413	2643	2443	2358	2452	2679	2690
ANN	2619	2624	2207	2614	3002	2515	2594	2503	2691	2920	2629
RF	2745	2539	2255	2559	2574	3198	2624	2295	2578	2451	2582
AdaBoost	2402	2756	1869	1857	2435	2956	2215	2482	2672	2741	2439
SVM	1511	2656	1382	1278	1140	1796	835	3149	312	2402	1646
	Area Under Receiver-Operating Characteristics Curve (AUC)										
DNN	0.816	0.806	0.809	0.802	0.826	0.804	0.842	0.816	0.782	0.832	0.814
ANN	0.633	0.643	0.645	0.638	0.625	0.645	0.640	0.645	0.618	0.616	0.635
Logit	0.708	0.698	0.716	0.699	0.690	0.704	0.735	0.692	0.708	0.701	0.705
RF	0.714	0.734	0.726	0.728	0.736	0.721	0.710	0.717	0.684	0.730	0.720
AdaBoost	0.647	0.631	0.637	0.618	0.648	0.656	0.635	0.650	0.625	0.632	0.638
SVM	0,688	0,887	0,692	0,691	0,794	0,664	0,695	0,586	0,625	0,592	0,690

Table 8 reveals variations in model performance across different folds, which emphasizes the merit of comparing models using cross-validation. Considering P&L, we observe the DNN to provide the best performance in seven out of ten folds. Accordingly, the DNN also achieves the highest average P&L and outperforms benchmarks by a sizeable margin. For example, the average P&L per trader (across folds) of the DNN is 2,931 GBP compared to 2,079 GBP of a hypothetical base setting in which STX would not hedge any trade. Compared to the second highest average P&L of 2,690, which comes from the logit model, the DNN provides a nine percent improvement. The P&L is informative for risk managers as it estimates the value of model-recommended hedging decisions. The AUC offers an additional perspective on model performance. Unlike P&L, which depends on the specific trades against which a model recommends hedging, the AUC emphasizes a model's discriminatory ability, that is whether it assigns higher risk scores to actual A-book clients. The AUC supports the appealing performance of the DNN. It achieves the highest performance in each fold and performs substantially better than the benchmarks in the comparison. For example, the second best benchmark in terms of the AUC is the RF classifier, which produces an average AUC of 0.720 c.f. an average AUC of 0.814 for the DNN).

3.7.2 Antecedents of DNN Forecast Accuracy

Table 8 evidences the superiority of the DNN over ML-benchmarks for the specific data set used here. To examine the robustness of model performance, it is important to clarify the antecedents of DNN success. One characteristic feature of the DNN is its multilayered - deep - architecture. Previous research establishes a connection between the depth of a model and its expressive capacity (Montufar et al., 2014), which suggests depth to be a determinant of predictive accuracy. Another characteristic that distinguishes the proposed DNN from ML benchmarks is its use of unsupervised pre-training. Aiding model training through finding more abstract, generative features, we expect predictive accuracy to benefit from pre-training. A third factor of interest is class imbalance. Skewed class distributions are a well-known impediment to classification and only seven percent of the traders in the data are A-book clients. Therefore, the fact that the DNN is more robust toward class imbalance than the ML benchmarks could also explain the results of Table 8. In the following, we examine the influence and importance of these three factors.

The Deep Architecture The DNN generates predictions in the last layer, where the last layer output neuron receives the combined input from multiple previous layers of SdAs and translates these signals into class probability predictions using the softmax function. This network configuration is equivalent to running logistic regression on the output of the hidden layers. To shed light on the value of the deep architecture, we compare DNN predictions to predictions from an ordinary logistic regression with the original features as covariates. The logistic model represents an approach which takes away the deep hidden layers from the DNN and only sustains the last layer. This is useful for appraising the merit of the distributed representations, which the deep hidden layers extract from the raw features.

Figure 20 displays the receiver-operating-characteristics (ROC) and a Precision-Recall (PR) curve for the DNN and a simple logit model. The plot emphasizes that the deep architecture substantially improves the network's discriminative ability. The performance of the logit model on raw features is almost random. The AUC value of 0.812 for the DNN suggests that performing the same regression on the high level representations, which the DNN learns from the raw features, facilitates a reliable detection of the positive class. Consequently, the DNN succeeds in extracting predictive features from the input data. In appraising Figure 20 it is important to note that the logit model is not meant to contribute a strong benchmark. As shown in Table 8, a regularized logit model with feature selection performs better than random. The purpose of Figure 20 is to evaluate the overall effect of the deep architecture compared to using the raw features as is, which motivates using the ordinary logit model for this comparison. The overall conclusion emerging from the analysis is that the deep architecture affects the predictive performance of the DNN.

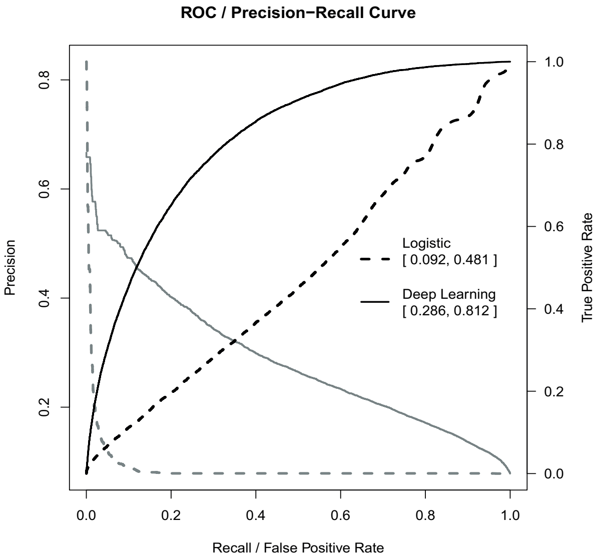


Figure 20: ROC (black), Precision-Recall Curve (grey) of deep learning and logistic regression. Results are based on a DNN model estimated from the first 70% of the data and applied to predict risk scores for the remaining 30% of trades. Curves depict model accuracy across these 30% trades.

Unsupervised Pre-Training The proposed DNN uses unsupervised pre-training for representation learning and feature extraction. To confirm the merit of pre-training, we examine the discriminative strength of each neuron in the unsupervised pre-training stage. We aim to check whether DNN learns distributed representations that help differentiate A- and B-book clients from *unlabeled* data. To that end, Figure 21 provides the histograms of activation values for neurons in the first dA layer of the DNN. The histograms show that activation values tend to be less than 0.4 when receiving a trade from a B-book client. Trades from A-book clients typically result in an activation value of 0.4 and above. While the magnitude of the activation values is irrelevant, the discrepancy of activation values for trades from different types of clients illustrates that - even with unlabeled data - the neurons in the first dA layer differentiate A- from B-book client trades. The intricate non-linear transformation between layers prohibit a replication of this analysis for higher layers because the relationship between activation values and input signals is no longer monotone. However, Figure 21 provides preliminary evidence that the spread trading data facilitates the extraction of higher-level generative features using

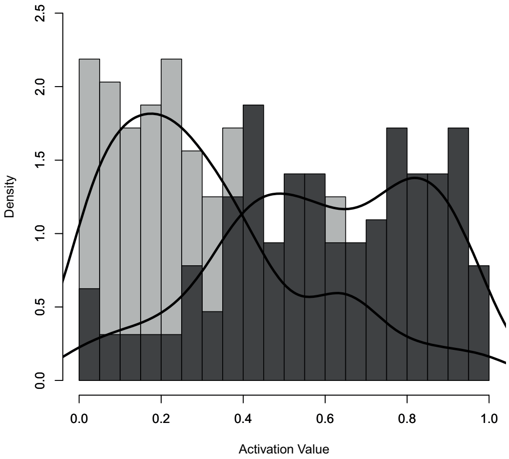


Figure 21: Histogram of activation values of neurons in the first dA layer for A-book (deep color) and B-Book (light color) client trades. The test set is re-sampled such that the ratio between high risk and normal traders is one.

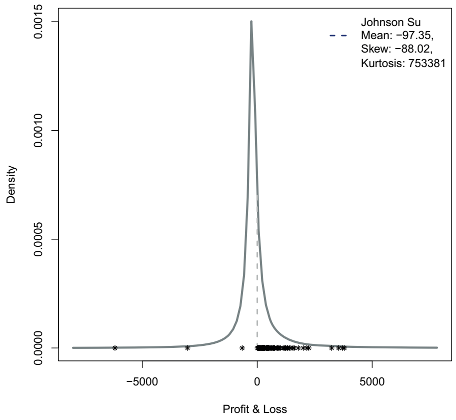


Figure 22: Top 100 stimuli of the best neuron from the test set

pre-training.

To substantiate the analysis and gain more insight into the link between neuron activation values and trades from different types of traders, we examine whether trades that trigger high activation values in a neuron are indeed worth hedging. To that end, we first calculate the maximum and minimum activation values for every neuron of the first layer, and 20 equally spaced threshold values between these boundaries. Subsequent analysis is based on a single neuron. We chose the neuron and corresponding threshold that give the purest separation between A- and B-book client trades (see Figure 21) upon manual inspection. Using this neuron, we find the 100 trades in the test set that activate the neuron the most. Figure 22 plots these trades on the overall P&L distribution. The results illustrate that, with a few false negatives, 97% of the trades that maximally activate the neuron end in profit and leave the market maker with a loss. Hedging against these trades, as indicated by the neuron's activation levels, is economically sensible. Although the eventual hedging decisions are based on the prediction of the DNN as a whole, the single neuron analysis provides further evidence of unsupervised pre-training of SdA layers to extract patterns that are indicative of a trade's risk. This confirms that the DNN learns distributed representations from the input data, which eventually help to distinguish high risk traders from other clients.

Analysis of the Class Imbalance Effect A growing body of literature on deep imbalanced learning indicates that DL models inherit vulnerability toward class imbalance from their ML ancestors (Johnson & Khoshgoftaar, 2019). However, it seems plausible that the DNN is more robust toward the adverse effect of imbalance than the ML benchmarks due to pre-training. Pre-training is carried out in an unsupervised manner. Therefore, class imbalance cannot occur. Figure 21 indicates that, without having access to class labels, pre-training has extracted patterns that help to differentiate high risk traders and B-book clients. Only the DNN has access to this information, which might give it an advantage over the ML benchmarks in the comparison of Table 8. To test this, we repeat the comparison after addressing class imbalance using the SMOTE (synthetic minority class oversampling technique) algorithm. SMOTE remedies class skew through creating artificial minority class examples in the neighborhood of actual minority class cases (He & Garcia, 2009).

Table 3 in the online Appendix reports detailed results of classifiers after applying SMOTE. Given that oversampling increases the number of observations and, in turn, the time to train different learning algorithms, we reduce the number of cross-validation folds and estimate performance using 5-fold cross-validation. More specifically, we configure the SMOTE algorithm such that it produces artificial A-book clients until both classes are balanced. Figure 23 summarizes corresponding results through depicting the average cross-validation performance for each learning algorithm before and after applying SMOTE in terms of P&L and the AUC.

Figure 23 reveals that SMOTE consistently improves the predictive performance for all models. P&L and AUC are substantially higher after addressing class imbalance, which reemphasizes the adverse effect of the latter. We also observe that the margin with which the DNN outperforms ML benchmarks decreases. For example, the strongest benchmarks after oversampling in terms of P&L and the AUC are the logit and ANN benchmark, respectively. The DNN performs 6 (9) and 4 (13) percent better than these competitors, where numbers in brackets denote the corresponding percent performance improvement in the original (i.e., imbalanced) data. A first interpretation of this result is that Table 8 gives an optimistic picture of DNN performance. The accuracy gap between the DNN and the ML benchmarks is less than Table 8 suggests if ML benchmarks receive auxiliary tuning in the form of addressing class imbalance. In addition, Figure 23 also confirms the DNN to be more robust toward class imbalance. While benefiting from SMOTE, its ability to identify high risk traders accurately is less dependent on oversampling the minority class compared to the ML benchmarks. This agrees with results of Figure 21 concerning the merit of unsupervised pre-training.

3.7.3 Implications for Risk Management

A model-based hedging policy comprises hedging the trades of clients classified as A-book by the model and taking the risk of all other trades. To clarify the managerial value of the proposed DNN, we compare the P&L of a DNN-based hedging strategy against that of rule-based strategies. One rule-based approach is the current policy of STX, which involves hedging trades of clients who secured a return above five percent in their previous 20 trades. In addition, we develop three custom hedging heuristics. Our first policy, *Custom 1*, relies on the Sharpe Ratio and singles out traders who achieve a higher than average Sharpe ratio in their past 20 trades. We suggest that securing risk-adjusted returns above the average indicates trader expertise. Since professionalism is only one reason for a successful trading history, *Custom 2* heuristic addresses another group of traders, which we characterize as overconfident. Such traders may display higher yields than other market participants and exhibit aggressive trading

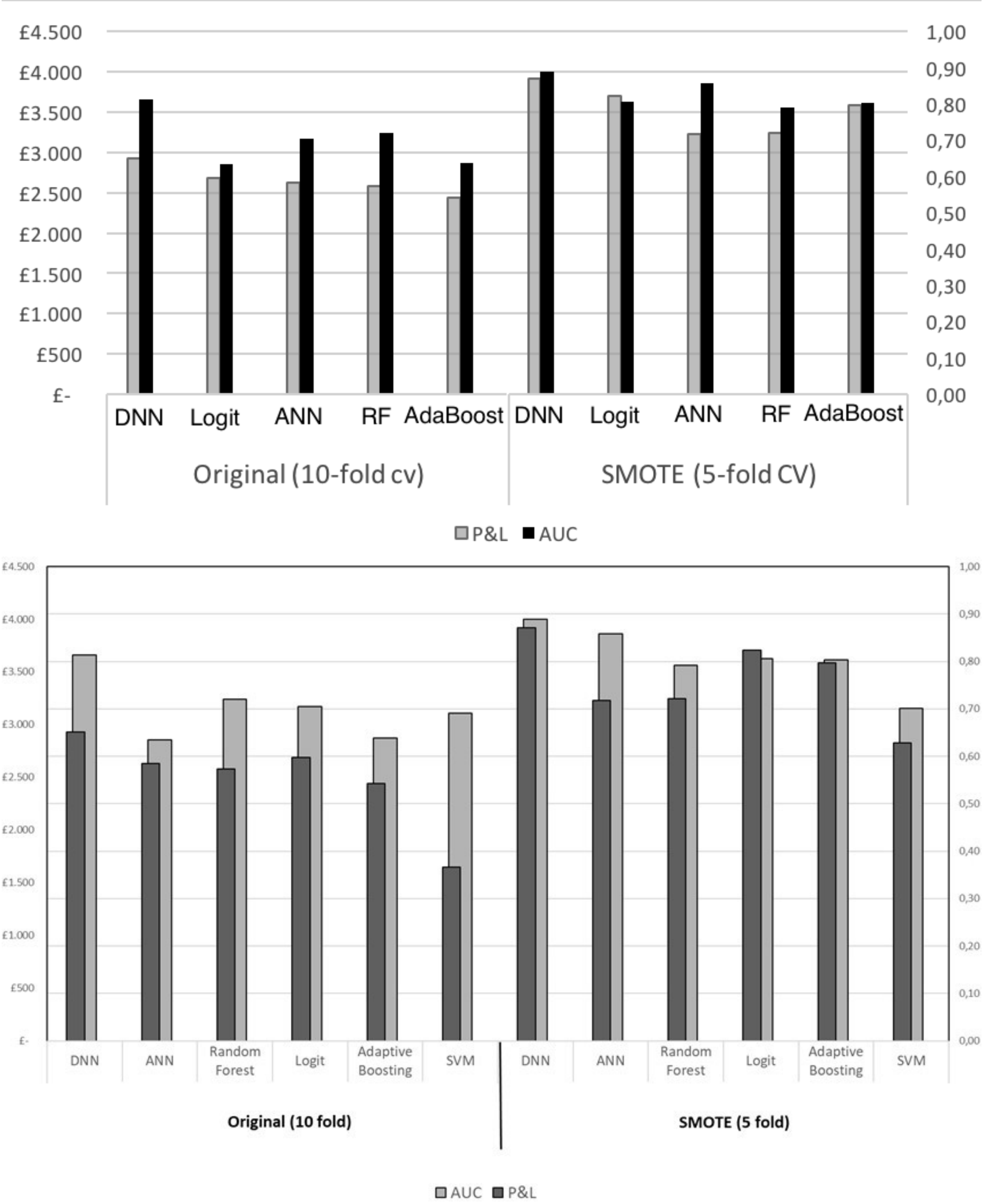


Figure 23: Cross-validation performance in terms of P&L before and after SMOTE.

behavior, manifesting itself through bigger lot sizes, higher frequency and shorter time interval trades (Benos, 1998). The Custom 2 heuristic thus considers the average trade duration and number of trades to deduce traders who may pose a greater risk. The third strategy, *Custom 3*, hedges trades from clients with a positive track record since trading with STX. The rationale is that traders who are unsuccessful in their early experiences might quit. Traders with a longer track record are either truly successful (and should be hedged against) or gamblers with a negative expected value (and should not be hedged against). Following this line of thinking, the most important risk STX is facing comes from *new* A-book clients. Comparisons to Custom 3 shed light on the ability of the DNN to identify such new A clients, as improvement over Custom 3 signals the DNN recognizing high risk traders that the track record-based logic of Custom 3 fails to capture.[4] We also consider an ensemble of the custom rule-based heuristics, constructed by means of majority voting. Drawing on domain knowledge, the rule-based strategies adopt a deductive approach. To complement the analysis, we also add one inductive rule-based approach in the form of a classification tree. Trees are regarded as interpretable classifiers. However, the degree to which decision makers can understand trees depends on their depth. In the interest of interpretability, we consider a classification tree (ctree) with two levels.

Table 9: Average P&L per trader in GBP of the DNN-Based and Rule-Based Hedging Strategies

	Cross-validation folds										
	1	2	3	4	5	6	7	8	9	10	mean
no hedge	2.268	2.130	1.601	2.122	2.230	2.536	1.785	1.938	1.870	2.306	2.079
DNN	**2.245**	**2.906**	**3.490**	**2.863**	**2.245**	**3.536**	**2.679**	**3.536**	**3.014**	**2.792**	**2.931**
STX	2.014	2.229	2.155	2.238	1.878	1.913	1.872	2.190	2.937	1.811	2.124
custom 1	1.534	1.549	1.236	1.550	1.417	1.488	1.392	1.618	1.986	1.341	1.511
custom 2	1.969	1.831	1.486	1.440	2.299	1.724	2.076	1.594	1.594	2.046	1.806
custom 3	1.736	1.825	1.579	1.965	1.865	1.937	2.000	1.785	2.402	1.750	1.884
ensemble	1.869	1.950	1.530	2.151	1.897	1.852	1.784	1.656	2.480	1.693	1.886
ctree	2.163	1.544	2.799	2.154	2.089	2.299	1.972	2.051	2.215	1.974	2.126

Table 9 reveals the current policy of STX to be the most suitable deductive hedging strategy. The logic of Custom 1 - 3 draws on financial theory. However, each of the three approaches, as well as an ensemble of them, performs worse than a hypothetical baseline setting in which STX would not hedge any trade. Observing Custom 1 - 3 to perform worse than this baseline supports the view that the focal trader classification task represents a challenging problem. Following this line of reasoning, Table 9 also emphasizes the soundness of the STX policy.

Unlike the deductive STX approach, the tree-based heuristic learns from past data. More specifically, the tree uses three features for splitting the data: a trader's average P&L in their initial 20 trades with STX, the minimum number of minutes until closing a losing position in their last 20 trades, and the average Sharpe ratio over their last 20 trades. These features display similarity with the custom heuristics. For example, considering a trader's initial performance follows the logic of Custom 3 while account for risk-adjusted returns is similar to Custom 1. Finally, considering a trader's reaction toward losses, the tree uses one of the variables to capture the disposition effect. We observe the two-level tree to produce slightly larger P&L than the

[4] We are grateful to an anonymous reviewer who suggested the logic of the Custom 3 heuristic.

STX heuristic. This suggest that a trader's average past performance, embodied in the STX approach, approximates the more complex rule set of the tree with some accuracy. Although the criticality of accurate hedging in the spread trading market suggests a revision of the STX approach with a tree-based approach, another finding from Table 9 is that implementing a DNN-based hedging strategy enables STX to further improve P&L compared to its current policy and the other rule-based hedging strategies we consider. Compared to the STX heuristic, the DNN raises per trader profits by 2,931 - 2,124 = 807 GBP, which implies a substantial, managerially meaningful improvement when considering the total number of clients of STX. For example, the data set used here comprises roughly 25K active traders.

The STX heuristic represents an established business practice at the partner company and reflects many years of industry experience. Moreover, the heuristic is extremely fast to execute and completely transparent. The situation for the DNN is far different. Classifying incoming trades more accurately, a DNN-based hedging policy is more profitable than rule-based approaches. The main cost of accuracy and profitability improvements is the black-box character of the corresponding risk management system. The client classification rules from the DNN originate from automatically extracted distributed representations of high risk traders. The business logic encapsulated in these rules is not interpretable for decision-makers, which also prohibits testing the agreement of these rules with domain knowledge.

Improved performance of the DNN leaves risk managers with the task to decide whether performance improvements are large enough to compensate the opaqueness of DNN and associated disadvantages, such as a lack of justifiability, higher computational requirements, etc. In the case of STX, we expect the imperative to hedge trades accurately and the magnitude of the performance improvement observed on their data to justify the adoption of a sophisticated DNN-based hedging strategy. The same might be true for other the spread-trading companies, although these would first need to replicate the results of this study to confirm the effectiveness of the DNN. A detailed description of the DNN configuration in the paper and especially the online Appendix will hopefully simplify this task. A more strategic consideration is that reluctance to adopt a new technology such as a sophisticated DNN-based hedging policy might also harm the competitive position of STX if competitors deploy corresponding solutions and use them to offer better prices to retail investors. At the same time, we caution against an overly optimistic view toward advanced DL-based decision aids. The empirical results observed in this study come from a single data source, which, although large in size, reflects the peculiarities of the market position and client structure of STX, and require a replication with different data in future research to raise confidence in the superiority of DL that we observe here. Given that the main disadvantage that we associate with the DNN is opaqueness, we conclude this chapter with acknowledging that DL and other complex ML models are not incomprehensible per se. An approach called information fusion-based sensitivity analysis provides insight into the relationship between model inputs (i.e. features) and outputs (i.e., forecasts) in any type of ML-based prediction model, including DNNs (Oztekin et al., 2013). Previous finance applications of this approach (Oztekin et al., 2016a; Sevim et al., 2014) demonstrate how it enables interpreting black-box ML models.

3.8 Discussion

The empirical results suggest that the DNN approach outperforms rule-based and ML benchmarks. It identifies high risk traders more accurately than other classifiers and provides higher financial gains when used for hedging decisions.

Predicting traders' risk taking behavior and future profitability under dynamic market conditions is challenging. Traders differ in their characteristics and trading behavior, and both are likely to change over time. Identifying unskilled traders is especially difficult due to the high variation in both behavior (input) and performance (output). Compared to genuine good traders, it is harder to identify uniform trading patterns for poor traders. Interviews with STX's staff hint at skilled traders sharing certain characteristics such as the ability to capture market rallies, following a consistent strategy, setting and adjusting limits, etc. On the other hand, there are countless ways in which poor traders lose money, including ignoring any of the above rules. In the high dimensional behavioral space, the number of potential variations of poor traders is innumerable. This contradicts the prior assumption of ML methods that the distribution $P(label|features)$ is smooth and well represented in the training data. Consequently, conventional ML faces difficulties in profiling trading patterns. The deep architecture equips DNN with higher expressive capacity to store the large number of variations of trading behaviors. Complexity theory shows a function that can compactly be represented by an architecture of depth k to require an exponentially growing number of computational units to represent the same function with smaller depth (Bengio, 2009). This suggests that increased depth enables the DNN to profile new combinations of behavioral variations and generalize to new trading patterns less represented in the training data.

Furthermore, the chance of making profit in the spread-trading market is highly noisy. Even poor traders can, by luck, win money. In fact, Figure 24 reveals that most of the clients who trade with STX have a greater than 50% win/lose ratio. However, even though traders win money on more than 50% of their trades, Figure 24 shows that average losses exceed average winnings by a large margin. Therefore, it is often sensible to classify a trader as a B-client and refrain from hedging their trades, even if many of their previous trades ended in profit.

Although based on an economic rationale, input features relating to past risk-adjusted return, trading frequency, etc. do not facilitate an accurate discrimination of spurious from genuine good traders. This arises because several feature values may coincide. The entanglement of spurious and genuine good traders in the behavioral feature space of trader characteristics further complicates the trader classification problem and harms conventional ML methods. The DNN draws upon the raw features and creates sensible abstractions from these features that exhibit a stronger connection with the target.

A specific DNN component we employ for trader classification is unsupervised pre-training. Observed results confirm that pre-training enables the DNN to construct layers of feature detectors that capture underlying generative factors, which explain variations across different trading behaviors. Stacking multiple layers of progressively more sophisticated feature detec-

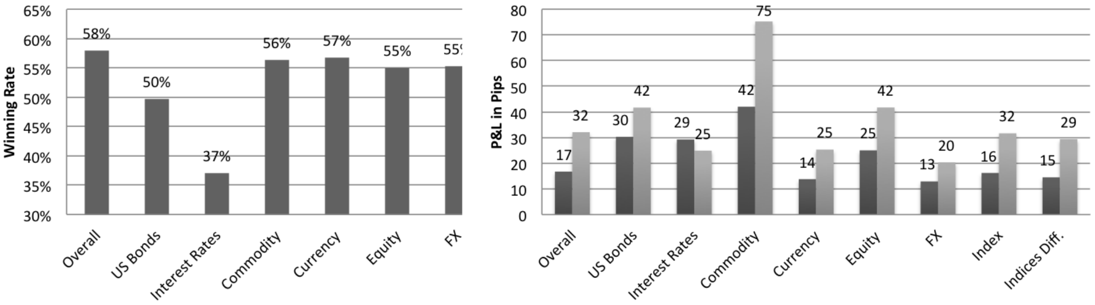


Figure 24: Retail traders' average winning ratio and average P&L points (profit in dark, loss in grey) on different categories of investments on the spread trading market.

tors, the DNN learns to disentangle these factors from the input distribution. Variations that are important for subsequent discrimination are amplified, while irrelevant information within the input data is suppressed (Erhan et al., 2010). We examine this ability in Figure 20, 21 and 22. After pre-training, the higher levels of the feature hierarchy store robust, informative, and generalizable representations that are less likely to be misled - and, thus, invariant to - the entangling of trading patterns in the input-space.

3.9 Conclusions

We set out to examine the effectiveness of DL in management support. Corresponding applications often involve developing normative decision models from structured data. We focus on financial risk taking behavior prediction and develop a DNN-based risk management system.

The results obtained throughout several experiments confirm the ability of DL, and the specific architecture of the DNN we propose, to extract informative features in an automatic manner. We also observe DNN-based predictions of trader behavior based on these features to be substantially more accurate than the forecasts of benchmark classifiers. Finally, our results demonstrate that improvements in forecast accuracy translate into sizable increases in operating profit. This confirms the ability of the proposed DNN to effectively support (hedging) decision making in this risk management case study.

Our findings pave a way to approach other behavior forecasting problems using DL. For example, direct marketers can increase the likelihood of consumers' responding to a promotion by studying clients' buying behaviors. Banks can enhance their risk control and make sensible credit approval decisions by analyzing clients' credit repayment behavior. E-commerce companies can dynamically adjust website layouts according to visitor preferences. These are only a few examples out of the vast space of tasks in decision support which generate large amounts of structured data and are routinely supported by ML. We provide evidence that the methodology reported here offers potentially significant gains in forecasting accuracy. Reappraising these gains in the scope of other business applications is essential to confirm that the appealing performance of the DNN that we observe is not specific to this case study.

4 DL application for fraud detection in financial statements

P.Craja, A.Kim, S.Lessmann

4.1 Abstract

Financial statement fraud is an area of significant consternation for potential investors, auditing companies, and state regulators. Intelligent systems facilitate detecting financial statement fraud and assist the decision-making of relevant stakeholders. Previous research detected instances in which financial statements have been fraudulently misrepresented in managerial comments. The paper aims to investigate whether it is possible to develop an enhanced system for detecting financial fraud through the combination of information sourced from financial ratios and managerial comments within corporate annual reports. We employ a hierarchical attention network (HAN) with a long short-term memory (LSTM) encoder to extract text features from the Management Discussion and Analysis (MD&A) section of annual reports. The model is designed to offer two distinct features. First, it reflects the structured hierarchy of documents, which previous models were unable to capture. Second, the model embodies two different attention mechanisms at the word and sentence level, which allows content to be differentiated in terms of its importance in the process of constructing the document representation. As a result of its architecture, the model captures both content and context of managerial comments, which serve as supplementary predictors to financial ratios in the detection of fraudulent reporting. Additionally, the model provides interpretable indicators denoted as “red-flag” sentences, which assist stakeholders in their process of determining whether further investigation of a specific annual report is required. Empirical results demonstrate that textual features of MD&A sections extracted by HAN yield promising classification results and substantially reinforce financial ratios.

4.2 Introduction

Fraud is a global issue that concerns a variety of different businesses, with a severe negative impact on the firms and relevant stakeholders. The financial implications of fraudulent activities occurring globally in the past two decades are estimated to amount up to $5.127 trillion, with associated losses increasing by 56% in the past ten years (Gee & Button, 2019). Nevertheless, the actual costs of fraud are potentially greater, particularly if one also considers the indirect costs, including harm to credibility and the reduction in business caused by the resultant scandal.

The Association of Certified Fraud Examiners (ACFE), the world’s largest anti-fraud organization, recognizes three main classes of fraud: corruption, asset misappropriation, and fraudulent statements (Singleton & Singleton, 2011). All three have specific properties and successful fraud detection requires comprehensive knowledge of their particular characteristics. This study concentrates on financial statement fraud and adheres to the definition of fraud proposed by Nguyen (1995), who stated that it is “the material omissions or misrepresentations resulting from an intentional failure to report financial information in accordance with generally accepted accounting principles“. For this study, the terminology “financial statement fraud“, “fraudulent financial reporting“, and “financial misstatements“ are used interchangeably and are distinguished from different factors that cause misrepresentations within financial statements, such as unintended mistakes.

The Center for Audit Quality indicted that managers commit financial statement fraud for a variety of reasons, such as personal benefit, the necessity to satisfy short-term financial goals, and the intention to hide bad news. Fraudulent financial statements can be manipulated so that they bear a convincing resemblance to non-fraudulent reports, and they can emerge in various distinct types (Huang et al., 2014). Examples of frequently used methods are net income over- or understatements, falsified or understated revenues, hidden or overstated liabilities and expenses, inappropriate valuations of assets, and false disclosures (Singleton & Singleton, 2011). Authorities directly reacted to the increased prevalence of corporate fraud by adopting new standards for accounting and auditing. Nevertheless, financial statement irregularities are frequently observed and complicate the detection of fraudulent instances.

Detecting financial statement abnormalities is regarded as the duty of the auditor (Dyck et al., 2010). Despite the existing guidelines, the detection of indicators of fraud can be challenging. A 2018 report revealed that only a limited number of cases of fraud were identified by internal and external auditors, with rates of 15% and 4%, respectively (ACFE, 2019). Hence, there has been an increased focus on automated systems for the detection of financial statement fraud (US Securities and Exchange Comission, 2019). Such systems have specific importance for all groups of stakeholders: for investors - to facilitate qualified decisions, for auditing companies - to speed up and improve the accuracy of the audit, and for state regulators - to concentrate their investigations more effectively (Abbasi et al., 2012; Albrecht et al., 2008). Therefore, efforts have been made to develop smart systems designed to detect financial statement fraud to generate early warning indicators (red-flags) that facilitate stakeholders' decision-making processes. We aim to contribute to the development of decision support systems for fraud detection by offering a state-of-the-art deep learning model for screening submitted reports based on a combination of financial and textual data. The proposed method exhibits superior predictive performance and allows the identification of "red-flags" on both the word- and sentence-level for the facilitation of the audit process. Additionally, we showcase the results of comparative modeling on different data types associated with financial reports and offer the alternative performance metrics that are centered around the cost imbalance of miss-classification errors.

4.3 Research design and contributions

In line with the above goals, we pose three research questions (RQ) that frame our research:

- **RQ 1**: What is the most informative data type for fraud detection? Can it benefit from the novel combination of financial and text data (FIN+TXT)?
- **RQ 2**: Can a state-of-the-art deep learning (DL) model be developed, that can detect indications of fraud from the textual information contained in financial statements? If yes, how effective does the DL approach perform as compared to the bag-of-words (BOW) ap-

proach for textual feature extraction in combination with quantitative financial features?

- **RQ 3**: In addition to predictive performance, can the proposed DL model assist in interpreting textual features signaling fraud? Given that the Hierarchical Attention Network (HAN) provides both word and sentence-level interpretation, is it possible to derive preliminary judgment on what level of granularity is more informative for practical application?

To determine answers to these research questions, we select an array of classification models and task them to perform fraud detection on different combinations of data. The choice is based on previous studies and recently developed methods that proved efficient for similar classification tasks. The classic statistical models include Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF), from recent Machine Learning (ML) models Extreme gradient Boosting (XGB) and Artificial Neural Network (ANN) algorithms were selected. Additionally, a novel DL method named Hierarchical Attention Network (HAN) is offered for consideration and advocated to be the most efficient fraud statement classifier. In line with previous research, this paper concentrates on the MD&A sections of annual reports filed by firms within the United States with the Securities and Exchange Commission (SEC), which are referenced as annual reports on form 10-K. All selected models are trained on five different combinations of data types contained in the statements submitted for audit: financial indicators (FIN), linguistic features (LING) of an MD&A text, financial and linguistic features (FIN + LING), the full text of an MD&A (TXT), full text and the financial indicators (FIN + TXT). We compare the predictive performance of the models with regard to their ability to distinguish fraud cases, for which we use traditional metrics like Accuracy and area under the Receiver Operating Curve (AUC). We also provide the analysis of metrics that account for the error cost imbalance, namely Sensitivity, F1-score, and F2-scores. This allows us to bring the existing state of research closer to the industrial setting. The provided analysis contributes to the field of fraud detection not only with the comparative study insights but offers previously unexplored data combinations and new DL methods, displaying superior results and additional interpretative features.

Following the RQ 3, we offer a novel fraud detection method that provides signaling tools for scholars and practitioners. We perform a comparative analysis of words considered "red-flags" by the RF feature importance method and the HAN attention layer output. We argue that the use of words for signaling may be the subject of manipulation and offer a remedy in the shape of sentence-level importance indicators. We further demonstrate how the latter may be applied for decision support in the audit process.

4.4 Decision support for fraud detection

Previous studies proposed fraud detection systems and offered systematic literature reviews on fraud detection approaches (Pourhabibi et al., 2020; West & Bhattacharya, 2016). Table 10 depicts the status-quo in the field of financial fraud detection along four dimensions: the technique utilized, the type of data, the country of study, and the predictive performance in terms of classification accuracy and other metrics. Much research focused on the financial variables and applied a wide range of modeling techniques, from LR to DL. Several authors experimented with linguistic variables, however, the majority of those have solely examined the relation between linguistic aspects and fraudulent actions. Only Hajek and Henriques (2017) combined them with financial data and showed that although financial variables are essential for the detection of fraud, it is possible to enhance the performance through the inclusion of linguistic data. At least two attempts to apply natural language processing (NLP) techniques focusing on the textual content have been undertaken. Nevertheless, to the best of our knowledge, this is the first study to apply DL models that allow for contextual information to be extracted from the text.

Study	Data (fraud / no fraud)	Country	Features	Classifiers	Used metrics
Hajek and Henriques (2017)	311/311	US	FIN+ LING	BBN(90.3), DTNB(89.5), RF(87.5), Bag(87.1), JRIP(87.0), CART(86.2), C4.5(86.1), LMT(85.4), SVM(78.0), MLP(77.9), AB(77.3), LR(74.5), NB(57.8)	Acc, TPR, TNR, MC, F-score, AUC
Y. J. Kim et al. (2016)	788/2156	US	FIN	LR (88.4), SVM (87.7), BBN (82.5)	Acc, TPR, G-mean, Cost Matrices
Goel and Uzuner (2016)	180/180	US	LING+POS tags	SVM(81.8)	Acc,TPR,FPR,Precision,F-score
L. Purda and Skillicorn (2015)	1407/4708	US	TXT (BOW), top 200 RF words	SVM (AUC 89.0)	AUC, Fraud Probability
Throckmorton et al. (2015)	41/1531	US	FIN+ LING from Conference Calls	GLRT (AUC 81.0)	AUC
Goel and Gangolly (2012)	405/622	US	LING	χ^2 statistics	χ^2 statistics
Dechow et al. (2011)	293/79358	US	FIN	LR(63.7)	Acc, TPR, FPR, FNR, min F-Score
Humpherys et al. (2011)	101/101	US	LING	C4.5 (67.3), NB (67.3), SVM (65.8)	Acc, Precision, Recall, F-score
Glancy and Yadav (2011)	11/20	US	TXT (BOW)	hierarchical clustering (83.9)	TP, TN, FP, FN, p-value
Perols (2011)	51/15934	US	FIN	SVM(MC 0.0025), LR(0.0026), C4.5 (0.0028), bagging(0.0028), DNN(0.0030)	Fraud Probability and MC
Cecchini et al. (2010)	61/61	US	LING	SVM (82.0)	AUC, TPR, FPR, FNR
Goel et al. (2010)	126/622	US	LING+TXT (BOW)	SVM(89.5), NB(55.28%)	Acc, TPR, FPR, Precision, F-score
Lin et al. (2015)	127/447	Taiwan	FIN	DNN (92.8), CART (90.3), LR (88.5)	Acc, FPR, FNR, MC
Ravisankar et al. (2011)	101/101	China	FIN	PNN (98.1), GP (94.1), GMDH (93.0), DNN (78.8), SVM (73.4)	Acc, TPR, TNR, AUC

Table 10: Analysis of classifier comparisons in financial statement fraud detection

[5] IN – financial data, LING – linguistic data (word category frequency counts, readability and complexity scores, etc.), TXT - text data, BOW–bag-of-words, POS – part of speech tags (nouns, verbs, adjectives), BBN – Bayesian belief network, NB-Naive Bayes, DTNB - NB with the induction of decision tables, CART–classification and regression tree, LMT - logistic model trees, MLP - multi-layer network, Bag-Bagging, AB - AdaboostM1, GMDH – group method data handling, GP–genetic programming, GLRT - generalized likelihood ratio test, LR–logistic regression, DNN – deep neural network, PNN – probabilistic neural network, RF – random forest, SVM – support vector machine, Acc - Accuracy, AUC – area under the ROC curve, MC - misclassification cost, TPR - true positive rate, TNR - true negative rate, FPR - false-positive rate, FNR - false-negative rate.

The majority of existing research measured performance in terms of accuracy. Some studies also considered precision and recall. Additionally, most of the reported studies neglected the interpretability which is a crucial aspect to facilitate decision support for fraud detection. This paper adds to the literature by offering an integrated approach for processing both textual and financial data using interpretable state-of-the-art DL methods. Furthermore, we provide a comprehensive evaluation of different modeling techniques using cost-sensitive metrics to account for the different severities of false alarms versus missed fraud cases.

4.4.1 Text-based indicators

Textual analysis is frequently employed for the examination of corporate disclosures. Linguistic features have been utilized in the analysis of corporate conference calls (Larcker & Zakolyukina, 2012), earnings announcements (Davis et al., 2012), media reports (Tetlock, 2007) and annual reports (Brown & Tucker, 2011; T. I. M. Loughran & Mcdonald, 2011). Multiple researchers have specifically concentrated on the MD&A section to examine the language used in annual reports (Cecchini et al., 2010; Feldman et al., 2010; Humpherys et al., 2011). The MD&A has a particular relevance as it offers investors the possibility of reviewing the performance of the company as well as its future potential from the perspective of management. This part also provides scope for the management's opinions on the primary threats to the business and necessary actions. It is interesting to note that as suggested by social psychology research, the emotions and cognitive processes of managers who intend to conceal the real situation could indicate specific linguistic cues that can facilitate the identification of fraud (DePaulo et al., 1982). Therefore studies have emphasised the increasing significance of textual analysis of financial documentation.

As stated in Li (2010a) literature review, research that analyzes the use of language within annual reports usually adopts one of two strategies. The first strategy is primarily based on past research into linguistics and psychology and is dependent on pre-determined lists of words that have an association with a specific sentiment, like negativity, optimism, deceptiveness, or ambiguity. T. I. M. Loughran and Mcdonald (2011) (L&M) demonstrated that if these lists are adapted to the financial domain, it is possible to determine relationships among financial-negative, financial-uncertain, and financial-litigious word lists and 10-k filing returns, trading volume, return volatility, fraud, material weakness, and unexpected earnings. As it was developed for analyzing 10-K text, the L&M sentiment word lists have been broadly employed in fraud-detection research (Hajek & Henriques, 2017). This research applies the L&M word lists for the extraction of sentiment features from the MD&A section of 10-Ks in the benchmark models. Other researchers based their approaches for detecting fraud on word lists that indicate positive, negative or neutral emotions (Goel & Gangolly, 2012; Humpherys et al., 2011) or more specifically anger, anxiety, and negativity according to the definitions supplied by the Linguistic Inquiry and Word Count dictionary (Humpherys et al., 2011; Larcker & Zakolyukina, 2012; Pennebaker et al., 2003).

The second strategy relies on ML to extract informative features for automatic differentiation between fraudulent and non-fraudulent texts. Li (2010a) contended that this method has various benefits compared with predetermined lists of words and cues, including the fact that no adaptation to the business context is required. ML algorithms have been used in the detection of financial statement fraud by various researchers, such as Cecchini et al. (2010), Glancy and Yadav (2011), Goel et al. (2010), Goel and Uzuner (2016), Hajek and Henriques (2017), Humpherys et al. (2011), L. Purda and Skillicorn (2015). Some attempts to integrate different types of data have also been made. L. Purda and Skillicorn (2015) compared a language-based method to detect fraud based on SVM to the financial measures proposed by Dechow et al. (2011), and concluded that these approaches are complementary. The methods displayed low forecast correlation and identified specific types of fraud that the other could not detect. This finding motivates the present research to combine Dechow et al. (2011) financial variables with linguistic variables to complement each other in the detection of fraud in financial statements.

The study of Hajek and Henriques (2017) is closest to this work as they combined financial ratios with linguistic variables from annual reports of US firms and employed a variety of classification models, as shown in Table 10. Despite these similarities, the study by Hajek and Henriques (2017) was not targeted at evaluating the textual content of corporate annual reports. Hence, it did not include modern NLP approaches such as deep learning-based feature extraction.

4.4.2 Methods and evaluation metrics

Prior work has tested a variety of statistical fraud detection models including ANNs, Decision Trees (DT), SVM, evolutionary algorithms, and text analysis (Hajek & Henriques, 2017). The BOW technique was frequently adopted for the extraction of the linguistic properties of financial documentation. The BOW approach represents a document by a vector of word counts that appear in it. Consequently, the word frequency is used as the input for the ML algorithms. This method does not consider the grammar, context, and structure of sentences and could be overly simple in terms of uncovering the real sense of the text (Larcker & Zakolyukina, 2012). A different technique for analyzing text is DL. Deep ANN are able to extract high-level features from unstructured data automatically. Textual analysis models based on DL can "learn" the specific patterns that underpin the text, "understand" its meaning and subsequently output abstract aspects gleaned from the text. Hence, they resolve some of the problems associated with the BOW technique, including the extraction of contextual information from documents. Due to their capacity to deal with sequences with distinct lengths, ANN have shown excellent results in recent studies on text processing. Despite their achievements in NLP, there has been limited focus on the application of state-of-the-art DL methods to the analysis of a financial text. For an effective adoption in practice, the models should not only be precise, but also interpretable (Huang et al., 2014). However, the majority of systems designed to detect fraud reported by researchers aim to maximise the prediction accuracy, while disregarding how transparent they are (Hajek & Henriques, 2017). This factor has particular significance as the development of interpretable models is critical for supporting the investigation procedure in auditing.

4.5 Data

Fraud detection is a challenging task because of the low number of known fraud cases. A severe imbalance between the positive and the negative class impedes classification. For example, the proportion of statements that were fraudulent and non-fraudulent in the annual reports submitted to the SEC for the period from 1999 to 2019 was 1:250. In past research, the number of firms that committed fraud contained in the data varied between 12 and 788 (Y. J. Kim et al., 2016; Throckmorton et al., 2015). The data used here consists of 208 fraudulent and 7 341 non-fraud cases, making it the most significant data set with a textual component so far (c.f., Table10).

The data set consists of US companies' annual financial reports, referred to as 10-K filings, that are publicly available through the EDGAR database of the SEC's website [6] and quantitative financial data, sourced from the Compustat database [7].

4.5.1 Labeling

Companies submit yearly reports that undergo an audit. Labeling these reports requires several filtering decisions: when can a report be considered fraudulent and what type of fraud should we consider. To address the first question, we follow the approach of Hajek and Henriques (2017), Humpherys et al. (2011), L. Purda and Skillicorn (2015), and consider a report as "fraudulent" if the company that filed it was convicted. The SEC - a source widely used by the previous research (Gray & Debreceny, 2014) - publishes statements, referred to as the "Accounting and Auditing Enforcement Releases" (AAER) that describe financial reporting related enforcement actions taken against companies that violated the reporting rules [8]. SEC concentrates on the cases with the highest importance (Karpoff et al., 2014) and applies enforcement actions where the evidence of manipulation is sufficiently robust (Hajek & Henriques, 2017), which provides a high degree of trust to this source. Labeling reports based on the AAER offers simplicity and consistency with easy replication, allowing to avoid possible bias related to subjective categorization. Following the filtering criteria offered by L. Purda and Skillicorn (2015), we select the AAERs concerning litigations issued during the period from 1999 to 2019 with identified manipulation instances between the year 1995 and 2016 that discuss the words "fraud", "fraudulent", "anti-fraud" and "defraud" as well as "annual reports" or "10-K". Addressing the second question, we follow Cecchini et al. (2010), Goel et al. (2010), Hajek and Henriques (2017), Humpherys et al. (2011), L. Purda and Skillicorn (2015) and focus on binary fraud classification. This implies that we do not distinguish between truthful and unintentionally misstated annual reports. The resulting data set contains 187 869 annual reports filed between 1993 and 2019, with 774 firm-years subject to enforcement actions. However, due to missing entries and mismatches in existing CIK indexation, the final data set is reduced to 7 757 firm-year observations with 208 fraud and 7 549 non-fraud filings. Further, we perform the extraction of text and financial data.

[6] The SEC is the preeminent financial supervisory organisation that is responsible for monitoring financial reports of firms listed on the US stock exchange: www.sec.gov/edgar.shtml

[7] Compustat is a database of financial, statistical and market information on companies throughout the world

[8] https://www.sec.gov/divisions/enforce/friactions.shtml

4.5.2 Text data

The retrieved reporting forms 10-K (Securities and Exchange Commission, 2019b) contain the MD&A section. The segment commonly called "Management's Discussion and Analysis of Financial Condition and Results of Operations" (Item 7) constitutes the primary source of raw text data. In addition, nine linguistic features are utilized as predictors (described in the online appendix). The selection of these features is influenced by the past studies, that demonstrated several patterns of fraudulent agents, like an increased likelihood of using words that indicate negativity (Newman et al., 2003; Throckmorton et al., 2015), absence of process ownership implying lack of assurance, thus resulting in statements containing less certainty (Larcker & Zakolyukina, 2012) or an average of three times more positive sentiment and four times more negative sentiment in comparison to honest reports Goel and Uzuner (2016). Additionally, the general tone (sentiment) and the proportion of constraining words, were included by Bodnaruk et al. (2015), Hajek and Henriques (2017), T. I. M. Loughran and Mcdonald (2011). Lastly, the average length of sentence, the proportion of compound words, and fog index are incorporated as measures of complexity and legibility and calculated based on formulas presented by Humpherys et al. (2011) and Li (2008), who concluded that reports produced by misstating firms had reduced readability.

4.5.3 Quantitative data

Along with text features, we used 47 quantitative financial predictors (described in the online appendix), which are capable of capturing financial distress as well as managerial motivations to misrepresent the performance of the firm. Past studies have presented robust theoretical evidence supporting the utilization of financial variables (Abbasi et al., 2012; Gaganis, 2009; Hajek & Henriques, 2017; Richardson et al., 2005). Following the guidelines of existing research, the financial ratios and balance sheet variables presented in the online appendix are extracted from Compustat, based on formulas presented by Dechow et al. (2011) and Beneish (1999). Financial variables include indicators like total assets (adopted as a proxy for company size (Bai et al., 2008; Throckmorton et al., 2015)), profitability ratios (Hajek & Henriques, 2017), accounts receivable and inventories as non-cash working capital drivers (Abbasi et al., 2012; Cecchini et al., 2010; Persons, 2011). Additionally, a reduced ratio of sales general and administrative expenses (SGA) to revenues (SGAI) is found to signalize fraud (Abbasi et al., 2012). Missing values are imputed using the RF algorithm. However, observations with more than 50% of the variables missing are excluded.

4.5.4 Imbalance treatment

The majority of previous research has balanced the fraud and non-fraud cases in a data set using undersampling (Hajek & Henriques, 2017; Humpherys et al., 2011; Ravisankar et al., 2011). We follow this approach and consider a fraud-to-non-fraud-ratio of 1:4, which reflects the fact that the majority of firms have no involvement in fraudulent behaviour. Both year and sector are utilized for balancing, in order to take into account different economic conditions, change in regulation, as well as to eradicate any differences across distinct sectors (Humpherys et al., 2011; Y. J. Kim et al., 2016). The latter is extracted with the SIC code (Securities and Exchange Commission, 2019a) and is of particular importance for text mining, as the utilization of words within financial documentation could differ according to the sector. The resulting balanced data set consists of 1 163 reports, out of which 201 are fraudulent, and 962 are non-fraudulent annual reports.

In the years 2002 to 2004 more financial misstatements than in other years can be observed. This could be attributed to the tightened regulations after the big fraud scandals in 2001 and the resulting implementation of SOX in 2002. Also, fewer misstatements are noted in recent years since the average period between the end of the fraud and the publication of an AAER is three years (L. D. Purda & Skillicorn, 2012).

4.6 Methodology

The objective of this study is to devise a fraud detection systems that classifies annual reports. While financial and linguistic variables represent structured tabular data and require no extensive preprocessing, the unstructured text data has to be transformed into a numeric format, which preserves its informative content and facilitates algorithmic processing. To achieve the latter, words are embedded as numeric vectors. The field of NLP has proposed various ways to construct such vectors. We consider two methods for text representation: frequency-based BOW embeddings and prediction-based neural embeddings (word2vec). An advantage of the BOW approach, which has been used in prior work on financial statement fraud (see Table10), is its simplicity. However, BOW represents a set of words without grammar and disrupts word order. Unlike BOW, the application of DL is still relatively new to the area of regtech (management of regulatory processes within the financial industry through technology). Therefore, the following subsections clarify neural word embeddings and address the DL components of the proposed HAN model.

4.6.1 Neural Embeddings

Within the BOW model, every word represents a feature. The amount of features denotes the dimension of the document vector (Manning et al., 2009). Since the amount of unique words within a document typically only represents a small proportion of the overall amount of unique words within the whole corpus, BOW document vectors are very sparse. A more advanced model for creating lower dimensional, dense embeddings of words is word2vec. As opposed to BOW, word2vec embeddings enable words that have similar meanings to be given similar vector representations and capture the syntactic and semantic similarities.

Word2vec (Mikolov, Sutskever, et al., 2013) is an example of a NN model that is capable of learning word representations from a large corpus. Every word within the corpus is mapped to a vector of 50 to 300 dimensions. Mikolov, Sutskever, et al. (2013) demonstrated that such vectors offer advanced capabilities to measure the semantic and syntactic similarities between words. Word2vec can employ two approaches, namely the continuous bag-of-words (CBOW) and Skip-gram. Both models employ a shallow neural network with one hidden layer. In CBOW, the model predicts a target word from a window of adjacent context words that precede and follow the target word within the sentence. The Skip-gram model, on the other hand, employs the target word for predicting the surrounding window of context words. The structure of the model weights nearby context words more heavily than more distant context words. The generated word embeddings are a suitable input for text mining algorithms based on DL, as will be observed in the next part. They constitute the first layer of the model and allow further processing of text input within the DL architecture.

The initial word2vec algorithm is followed by GloVe (Pennington et al., 2014b), FastText (Bojanowski et al., 2016), and GPT-2 (Radford et al., 2019), as well as the appearance of publicly available sets of pre-trained embeddings that are acquired by applying the above-mentioned algorithms on large text corpora. Pre-trained word embeddings accelerate training DL models and were successfully used in numerous NLP tasks (Dai & Le, 2015; Howard & Ruder, 2018; Peters et al., 2018; Radford et al., 2018; Tang et al., 2015). We apply several types of pre-trained embeddings for HAN model and a neural network with a bidirectional Gated Recurrent Unit (GRU) layer that serves as a benchmark from the field of DL. As a result of a performance-based selection, the HAN model is built with word2vec embeddings with 300 neurons, trained on the Google News corpus, with a vocabulary size of 3 million words. The DL benchmark is used with the GPT-2 pre-trained embeddings from the WebText, offered by Radford et al. (2019), as they arguable constitute the current state-of-the-art language model. The DL benchmark model is thus referred to as GPT-2 and is used together with the attention mechanism, discussed further.

4.6.2 Deep learning

After representing unstructured textual data in a numerical format, it can be used for predictive modeling. Conventional methods for classifying text involve the representation of sparse lexical features, like TF-IDF, and subsequently utilize a linear model or kernel techniques upon this representation (Joachims, 1997).

An NN can be considered a non-linear generalization of the linear classification model (Hastie, Tibshirani, Friedman, et al., 2009). NN comprised of multiple intermediate layers, called hidden layers, are referred to as deep NN (DNN), or DL networks. The weight matrices between the layers serve as intermediate parameters used by the NN to calculate a function of the inputs through the propagation of the computed values. During the training process, the NN learns to predict the output labels by changing the weights connecting the neurons with regard to how well the predicted output for a particular input matched the true output label in the training data. The process of adjusting the weights among neurons based on errors observed in prediction, to modify the calculated function to generate increased predictive accuracy, is referred to as back-propagation, while the structure of densely connected layers would be referred to as ANN (Aggarwal, 2018).

Recently, DL has incorporated new techniques, including Convolutional Neural Networks (CNN)

(Kalchbrenner et al., 2014) and Recurrent Neural Networks (RNN) (Hochreiter & Schmidhuber, 1997a) for learning textual representations (Yin et al., 2017a). The RNN architecture allows retaining the input sequence, which made it widely used for natural language understanding, language generation, and video processing (Kalchbrenner & Blunsom, 2013; Mikolov et al., 2010). An LSTM is a special type of RNN, comprised of various gates determining whether the information is kept, forgotten or updated and enabling long-term dependencies to be learned by the model (Hochreiter & Schmidhuber, 1997a). An LSTM retains or modifies previous information on a selective basis and stores important information in a separate cell c_t, which acts as a memory (Tixier, 2018). The LSTM comprises four gates called the input gate i_t, forget gate f_t, output gate o_t and input modulation gate $\hat{c}_t$. These allow the network to recall or disregard information about previous elements in an input sequence. The interaction among the gates is noted in equations below, where $\odot$ represents element-wise multiplication.

$$\begin{aligned} i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i & o_t &= \sigma(U_o x_t + W_o h_{t-1} + b) \\ f_t &= \sigma(U_f x_t + W_f h_{t-1} + b_f) & c_t &= f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\ \hat{c}_t &= tanh(U_c x_t + W_c h_{t-1} + b_c) & h_t &= tanh(c_t) \odot o_t \end{aligned}$$

By considering the present input vector x_t, as well as the previous hidden state h_{t-1}, the forget gate layer f_t determines how much of the preceding cell state c_{t-1} it should forget, while, based on the identical input, the input gate layer i_t determines the amount of new information $\hat{c}_t$ that should be learned. The combination of outputs from these filters enables updating the cell state c_t. Consequently, overwriting of important information by the new inputs does not occur, it can persist for extended periods. Lastly, the hidden state h_t is computed based on the updated memory and the output gate layer o_t. In the final stage, the output vector is calculated as a function of the newly generated hidden state $\hat{y}_t = \sigma_o(W_o h_t + b_o)$, which is analogous to the basic RNN.

4.6.3 Hierarchical Attention Network

RNN retain the sequential structure of language. More advanced DL approaches also address hierarchical patterns of language such as the hierarchy between words, sentences, and documents. Some methods have covered the hierarchical construction of documents (Rao et al., 2018; C. Zhou et al., 2015). The specific contexts of words and sentences, whereby the meaning of a word or sentence could change depending on the document, is a comparatively new concept for the process of text classification, and the HAN was developed to address this issue (Yang et al., 2016a). When computing the document encoding, HAN firstly detects the words that have importance within a sentence, and subsequently, those sentences that have importance within a document while considering the context (see Figure 25). The model recognizes the fact that an occurrence of a word may be significant when found in a particular sentence, whereas another occurrence of that word may not be important in another sentence (context).

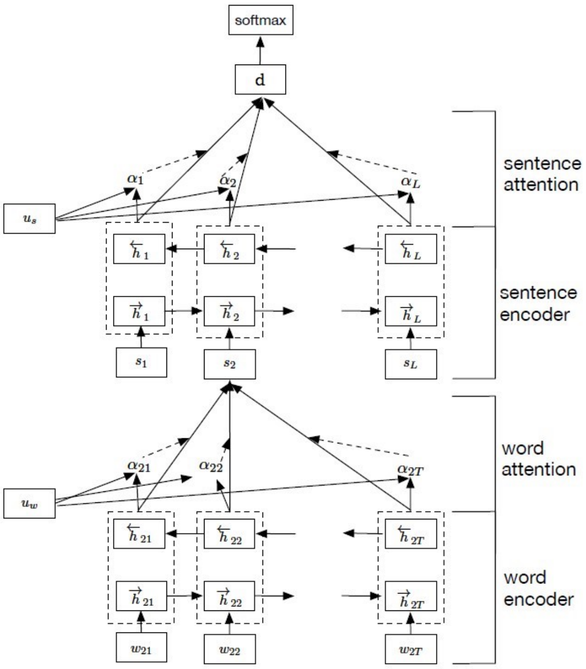


Figure 25: HAN Architecture. Image based on Yang et al. (2016a)

The HAN builds a document representation via the initial construction of sentence vectors based on words followed by the aggregation of these sentence vectors into a document representation through the application of the attention mechanism. The model consists of an encoder that generates relevant contexts and an attention mechanism, which calculates importance weights. The same algorithms are consecutively implemented at the word level and then at the sentence level.

Word Level The input is transformed into structured tokens w_{it} that denote word i in sentence $t \in [1, T]$. Tokens are further passed through a pre-trained embedding matrix W_e that allocates multidimensional vectors $x_{it} = W_e w_{it}$ to every token. As a result, words are denoted in numerical format by x_{it} as a projection of the word in a continuous vector space.

Word Encoder The vectorized tokens represent the inputs for the following layer. While Yang et al. (2016a) employed GRU for encoding, we use LSTM as it showed better performance on the large text sequences at hand (Chung et al., 2014). In the context of the current model, a bidirectional LSTM is implemented to obtain the annotations of words. The model consists of two uni-directional LSTMs, whose parameters are different apart from the word embedding matrix. Processing of the sentences in the initial forward LSTM occurs in a left to the right manner, whereas in the backward LSTM, sentences are processed from right to left. The pair of sentence embeddings are concatenated at every time step t to acquire the internal representation of the bi-directional LSTM h_{it}.

Word Attention The annotations h_{it} construct the input for the attention mechanism that learns enhanced annotations denoted by u_{it}. Additionally, the *tanh* function adjusts the input values so that they fall in the range of -1 to 1 and maps zero to near-zero. The newly generated annotations are then multiplied again with a trainable context vector u_w and subsequently normalized to an importance weight per word α_{it} via a softmax function. As part of the training procedure, the word context vector u_w is initialized randomly and concurrently learned. The total of these importance weights concatenated with the already computed context annotations is defined as the sentence vector s_i:

$$u_{it} = tanh(W_w h_{it} + b_w) \tag{20}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{21}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{22}$$

Sentence Level and Sentence Encoder Subsequently, the entire network is run at the sentence level using the same fundamental process used for the word level. An embedding layer is not required as sentence vectors s_i have previously been acquired from the word level as input. Summarization of sentence contexts is performed using a bi-directional LSTM, which analyzes the document in both forward and backward directions:

$$\overrightarrow{h}_i = \overrightarrow{LSTM}(s_i), i \in [1, L] \tag{23}$$

$$\overleftarrow{h}_i = \overleftarrow{LSTM}(s_i), i \in [T, 1] \tag{24}$$

$$h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i] \tag{25}$$

Sentence Attention For rewarding sentences that are indicators of the correct document classification, the attention mechanism is applied once again along with a sentence-level context vector u_s, which is utilized to measure the sentence importance. Both trainable weights and biases are initialized randomly and concurrently learned during the training procedure, thus yielding:

$$u_i = tanh(W_s h_i + b_s) \tag{26}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{27}$$

$$d = \sum_i \alpha_i h_i \tag{28}$$

where d denotes the document vector summarising all the information contained within each of the document's sentences. Finally, the document vector d is a high-level representation of the overall document and can be utilized as features for document classification to generate output vector $\hat{y}$:

$$\hat{y} = softmax(W_c d + b_c) \tag{29}$$

where $\hat{y}$ denotes a K dimensional vector and the components y_k model the probability that document d is a member of class k in the set $1, ..., K$.

The application of the HAN follows the application of Kränkel and Lee (2019). Training of the DL model is performed on the training data set using both textual and quantitative features. Hence, the textual data acquired in the previous section is concatenated with the financial ratios. The model is employed to predict fraud probabilities of annual statements in the corresponding validation and test partitions, that were constructed with random sampling with stratification. Figure 26 shows the architecture of the HAN based fraud detection model and the output dimensions of each layer.

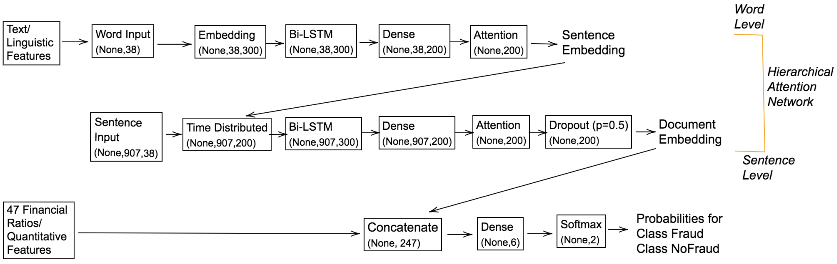


Figure 26: Architecture of the HAN based Fraud Detection Model

The LSTM layer consists of 150 neurons, a HAN dense dimension of 200, and a last dense layer dimension of 6. In this case, a combination of forward and backward LSTMs gives 300 dimensions for word and sentence annotation. The last layer of the HAN involves the application of dropout regularization to prevent over-fitting. In a final step, the resulting document-representation of dimension 200 is concatenated with 47 financial ratios and inputted to a dense layer before running through a softmax function that outputs the fraud probabilities. For training, a batch size of 32 and 17 epochs was used after hyperparameter tuning on the train validation set.

4.6.4 Evaluation metrics

The detection of financial statement fraud is considered a binary classification problem with four potential classification outcomes: True positive (TP) denotes the correct classification of a fraudulent company, false negative (FN) denotes the incorrect classification of a fraudulent company as a non-fraudulent company, true negative (TN) denotes the correct classification of a non-fraudulent company and false positive (FP) denotes the incorrect classification of a non-fraudulent company as a fraudulent company.

To estimate the predictive performance, many previous studies considered a combination of measures such as accuracy, sensitivity (also called TP rate or recall), specificity (also called TN rate), precision, and F1-score (West & Bhattacharya, 2016). In this study, model performance is evaluated by the AUC, sensitivity, specificity, F1-score, F2-score, and accuracy.

The accuracy is defined as the percentage of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{P + N} \tag{30}$$

The sensitivity measures the number of correctly classified fraudulent instances as a percentage of all fraudulent instances:

$$\text{Sensitivity} = \frac{TP}{P} = 1 - \text{FN rate} \tag{31}$$

The specificity measures the number of correctly classified non-fraudulent instances as a percentage of all non-fraudulent instances:

$$\text{Specificity} = \frac{TN}{N} = 1 - \text{FP rate} \tag{32}$$

The F-score is a combination of precision $= \frac{TP}{TP+FP}$ (correct classification of fraudulent instances as a percentage of all instances classified as fraudulent) and sensitivity (indicates how many fraudulent instances the classifier misses) and measures how precise and how robust the models classify fraudulent cases:

$$F_\beta\text{-score} = (1 + \beta^2) \times \frac{\text{precision} \times \text{sensitivity}}{(\beta^2 \times \text{precision}) + \text{sensitivity}} \tag{33}$$

Prior research emphasized that a higher sensitivity is preferred to higher specificity in financial statement fraud detection. Nevertheless, the majority of models have exhibited considerably higher performance in detecting truthful transactions in comparison to those that are fraudulent (Ravisankar et al., 2011; West & Bhattacharya, 2016). An explanation for this preference is that FN and FP rates result in considerably different misclassification costs (MC). Hajek and Henriques (2017) estimated the cost of failing to detect fraudulent statements to be twice as high as the cost of incorrectly classifying fraudulent statements. Hence, effective models should concentrate on high sensitivity and classify correctly as many positive samples as possible, rather than maximizing the number of correct classifications. Therefore, this study employs the F2-score in addition to the F1-score (harmonic mean of precision and sensitivity), as it weights sensitivity higher than precision and is, therefore, more suitable for fraud detection. The AUC denotes the area under the Receiver Operating Curve (ROC) and is preferred to accuracy in financial statement fraud detection because of the impact of fraud/non-fraud imbalance in the sample (L. Purda & Skillicorn, 2015; Throckmorton et al., 2015). This study employs the AUC as a measure of separability to compare the predictive performance of the models and determine their suitability. The higher the AUC, the better the model can distinguish between fraudulent and non-fraudulent cases.

The cutoff threshold for the probability of fraud has to be defined to quantify the F1- and F2-scores. We select the threshold that maximizes the difference between sensitivity and FP rate and use it to evaluate the classification results. For the HAN model, the optimal threshold is set at 0.03, implying that a statement is classified as fraudulent if its fraud probability is higher than 3%.

4.7 Classification results

We answer RQ 1 and 2 by means of empirical analysis and compare a set of classification models in terms of their fraud detection performance. The models generate fraud classification based on financial indicators, linguistic features of reports, the reports' text, and combinations of these groups of features. Table 11 reports corresponding results from the out-of-sample test set. The baseline accuracy of classifying all cases of the test set as non-fraudulent (majority class) is 82.81%.

4.7.1 Modeling of financial data

Modeling using financial data (FIN) has been the most popular approach (Table 10). The approach serves this study as a benchmark, to which we compare modeling on linguistic features (LING) and the combination of both (FIN + LING). The last two columns of Table 11 show the results of the comparison. In terms of AUC and accuracy, the tree-based models RF (Breiman, 2001) and XGB appear to excel at predicting fraud on FIN, indicating a non-linear dependency between financial indicators and the fraud status of a report. This result is in line with C. Liu et al. (2015) who showed that RF performed especially well in case of high-dimensional financial fraud data because of a higher variance reduction resulting from combining Bagging with randomly chosen subsets of input features (Hajek & Henriques, 2017). Hajek and Henriques (2017) also reported an accuracy of 88.1% on FIN data and concluded that the ensemble of tree-based algorithms including JRip, CART, C4.5 and LMT exhibit superior performance over SVM, LR and ANN due to a relatively low dimensionality achieved during feature selection. The predictive performance aligns with the results of Y. J. Kim et al. (2016), offering the LR and SVM models as the most accurate. Lin et al. (2015) and Ravisankar et al. (2011) showcased that the DNN models (ANN with more than one hidden layer) outperform LR and SVM, offering an accuracy higher by around 4.5%. The SVM is a widely recognized model and was applied both for fraud detection (Perols, 2011) and in other fields (Dumais et al., 1998). However, the results show that inherent configuration complexities make SVM a secondary choice for practitioners. ANN show less impressive predictive performance but proved to be the most efficient in terms of sensitivity. However, for model evaluation, a balanced indicator like F1- and F2-scores would provide a better perspective. These metrics suggest XGB to outperform other models. XGB represents an advancement in the in the field of ML, its high performance is noteworthy since it was not considered in prior work on fraud detection. Given

[8] F

				Finance data (FIN)				
	AUC	**Sensitivity**	**Specificity**	**F1-score**	**F2-score**	**Accuracy**		
LR	0.7620	0.6833	0.7543	0.4767	0.7480	0.8252		
RF	**0.8609**	0.7666	0.7889	0.5508	0.7892	**0.8653**		
SVM	0.7561	0.6166	0.7820	0.4625	0.7595	0.8280		
XGB	0.8470	0.6660	**0.8719**	**0.5839**	**0.8391**	0.8481		
ANN	0.7564	**0.7833**	0.6574	0.4563	0.6835	0.6790		
				Linguistics data (LING)			Comparison to FIN	
	AUC	**Sensitivity**	**Specificity**	**F1-score**	**F2-score**	**Accuracy**	**Delta AUC**	**Delta F1**
LR	0.6719	0.7000	0.6193	0.3962	0.6398	0.8280	-0.0901	-0.0805
RF	**0.7713**	**0.7500**	0.7197	**0.4839**	0.7302	**0.8424**	-0.0896	-0.0669
SVM	0.7406	0.7000	0.6747	0.4285	0.6857	0.8280	-0.0155	-0.0340
XGB	0.7219	0.3666	**0.9446**	0.4489	**0.8385**	0.8338	-0.1251	-0.1350
ANN	0.6782	0.6333	0.6747	0.3958	0.6758	0.6676	-0.0782	-0.0605
				Finance data + Linguistics data (FIN + LING)			Comparison to FIN	
	AUC	**Sensitivity**	**Specificity**	**F1-score**	**F2-score**	**Accuracy**	**Delta AUC**	**Delta F1**
LR	0.7682	0.7666	0.6782	0.4623	0.6984	0.8280	0.0062	-0.0144
RF	0.8606	0.7666	**0.7543**	0.5197	0.7610	0.8567	-0.0003	-0.0311
SVM	0.7973	0.7166	0.7439	0.4858	0.7448	0.8280	0.0567	0.0573
XGB	**0.8651**	0.8166	**0.7543**	**0.5444**	**0.7687**	**0.8653**	0.0181	-0.0395
ANN	0.7733	**0.8333**	0.6228	0.4566	0.6614	0.6590	0.0169	0.0003
				Text data, TF-IDF (TXT)			Comparison to LING	
	AUC	**Sensitivity**	**Specificity**	**F1-score**	**F2-score**	**Accuracy**	**Delta AUC**	**Delta F1**
LR	0.8371	0.7333	0.8269	0.5714	0.8145	0.8281	0.1652	0.1752
RF	0.8740	0.7166	0.9377	0.7107	**0.8998**	0.8681	0.1027	0.2268
SVM	0.8836	**0.8382**	0.7544	0.5876	0.7731	0.8796	0.1275	0.1251
XGB	0.8785	0.7660	0.8581	0.6258	0.8451	0.8853	0.1566	0.1769
ANN	0.8829	0.7121	**0.9434**	**0.7286**	0.8993	**0.8990**	0.2047	0.3328
HAN	**0.9108**	0.8000	0.8896	0.5744	0.7982	0.8457		
GPT-2+Attn	0.7729	0.7619	0.6697	0.4423	0.6905	0.6484		
				Finance data + Text data, TF-IDF (FIN + TXT)			Comparison to FIN + LING	
	AUC	**Sensitivity**	**Specificity**	**F1-score**	**F2-score**	**Accuracy**	**Delta AUC**	**Delta F1**
LR	0.8598	0.7833	0.7854	0.5562	0.7890	0.8424	0.0916	-0.0795
RF	0.8797	0.6660	0.9550	0.7079	0.9043	0.8739	0.0191	-0.1571
SVM	0.8902	0.7833	0.8961	0.6861	0.8784	0.8280	0.0929	-0.2576
XGB	0.8983	0.7000	**0.9653**	**0.7500**	**0.9187**	**0.9083**	0.0332	-0.1661
ANN	0.8911	0.7460	0.9405	0.7401	0.9055	0.9054	0.1178	-0.2838
HAN	**0.9264**	**0.9000**	0.8206	0.6506	0.8361	0.8457		
GPT-2+Attn	0.7776	0.7678	0.6791	0.4455	0.6991	0.6934		

Table 11: Comparative performance of selected binary classifiers on the different types of test data. The baseline accuracy is 0.8281

the much higher cost of missing actual fraud cases compared to false alarms, we argue that the F2-score is the most suitable threshold-based indicator of model performance. Therefore, we emphasize the F2-score together with the AUC, which allows the tuning of the threshold.

4.7.2 Modeling of linguistic data

The modeling on linguistic data (LING) was the first step towards including text in fraud detection. The earlier experiments by Cecchini et al. (2010), Humpherys et al. (2011) and Goel et al. (2010) employed SVM and achieved accuracy of 82%, 65.8%, and 89.5% respectively. The latter additionally included the BOW method that we will discuss further. Our modeling falls in line with the previous work and exhibits SVM as the second strongest predictor, yielding an AUC of 74% and accuracy of 82%. RF remained the most reliable predictor with the highest AUC, accuracy, and F2-score. Modeling done solely on LING will allow us to assess the degree to which both sources of data contribute to accurate classification. In line with Hajek and Henriques (2017), all models exhibit higher performance on FIN data than on LING data solely, leading to the conclusion that financial covariates have more predictive power than linguistic variables. However, the performance differences are not substantial and suggest a strong relationship between linguistic features and fraudulent behavior, which agrees with previous studies.

Following the ideas of Hajek and Henriques (2017), we combine FIN and LING data to evaluate if the classifier can make use of both data sources. Our results differ in terms of the leading models, with RF and XGB offering the highest AUCs of 86%. XGB is showing a definite improvement, performing well on FIN data, falling back a little in the LING set up but making better use of the combined input. Once again we observe the superior performance of XGB in terms of F2-score with 76.87% followed closely by 76.10% of RF and 74.48% of SVM, which once again advocates for the usefulness of advanced ML methods for practical tasks. Interestingly, for the rest of classifiers, the accuracy dropped a little in comparison to FIN, but the AUCs improved (with a minor exception for RF). This serves as an indication that LING and FIN data combined may provide conflicting signals to the classifier, however, the data mix is a definite improvement as it provides a stronger signal to the classifier, enhancing the predictive performance.

4.7.3 Modeling of text data

Researchers have been taking a step forward from aggregated linguistic features in an attempt to derive more predictive power from the vast amounts of text contained in annual reports. We offer the advanced methods of NLP, previously unexplored for fraud detection, and compare them to the performance of more traditional models. Goel et al. (2010), Glancy and Yadav (2011) and L. Purda and Skillicorn (2015) applied the BOW model to perform modeling on text data, while Goel and Uzuner (2016) made use of part-of-speech tagging. They utilized SVM and hierarchical clustering as classifiers and achieved accuracies of 89.5%, 83.4%, 89%, and 81.8%, respectively.

Table 11 offers an overview of the modeling results, starting with purely textual input (TXT) and continuing with text enhanced by financial data (FIN+TXT). Two new DL methods are included in TXT modeling, namely HAN and GPT-2. While traditional benchmarks take the TF-IDF transformations of word input, the DL models make use of pre-trained embeddings, discussed in the Methodology section. We can observe that modeling on TXT provides improvement across all models in comparison to LING, with the largest AUC delta of 0.2 in the case of ANN. This increase can be first and foremost attributed to the richer input of the actual MD&A content. ANN demonstrates the highest accuracy, 89%, and the best performing F1- and F2-scores, which constitutes a strong signal that the neural network architecture is a favorable candidate for the task, regardless of the BOW input. Given the complexity of textual processing, ANN proves its capacity to pick up on complex relationships between the target and explanatory variables. The improvement is also visible for the F2-score of 89.93% that closely follows the RF's 89.98%. It is interesting to compare the BOW-based ANN with GPT-2 and HAN, as all three represent a NN architecture. GPT-2 performs better on TXT than any other model on LING. Though it fails to show superior accuracy, its sensitivity metric is one of the highest, leading to the conclusion that with some threshold adjustment, it could provide better predictive performance than other models like LR or tree-based models. This example underlines the potential gains of implementing the new DL methods that allow superior insights into unstructured data. Unlike BOW-based benchmarks, embeddings-based HAN and GPT-2 retained the structure and context of the input. HAN showed superior results in terms of AUC 91.08% but fell short in terms of accuracy. However, its sensitivity is exceeding those of all other benchmarks except for SVM, making it a promising model for fraud detection. HAN represents a further advancement of the NLP with DL approaches; its performance can be explained by the intrinsic capacity to extract significant contextual similarities within documents and that pertinent cues that allow truthful text to be distinguished from deceitful ones are dependent on the context rather than the content (L. Zhou et al., 2004). All in all, the results suggest that textual data, in general, can offer much more insight than LING across all classifiers. However, the NN-based and tree-based architectures seem to benefit the most in terms of AUC.

We conclude the analysis of results by looking at the feature combination FIN+TXT, which is at the core of our study. The input setup is done in two ways: a combination of word vectors with financial indicators into one data set and a 2-step modeling approach. The latter comprises building a TXT model and using its probability prediction as an input to another DL model that will concoct it with FIN and output the final binary prediction. The first approach is applied in the case of benchmark models, including ANN, while the second one is implemented for the DL models, namely, HAN and GPT-2.

L. Purda and Skillicorn (2015) conducted a comparison of TXT with FIN data proposed by Dechow et al. (2011) separately and determined that they are complementary since both methods are capable of identifying specific types of fraud that the other cannot detect and they have a relatively low correlation. In our case, all benchmarks exhibit improved performance in comparison to the FIN + LING setup, especially LR and SVM. However, the same unanimity is observed in decreased F1-score metric, with ANN dropping by 0.28. We observe the superiority of predictive powers of full-textual input over the linguistic metrics. If we compare the additional value of FIN for the performance, we can see only a minor increase in almost all metrics, once again underlying the complexity and potential misalignment of FIN and TXT data. However, it is essential to note that unlike F1-score, F2-score increases across the ML benchmarks, which brings us to the initial assumption behind the preference toward the F2-score as key to model evaluation for practical use. We conclude that with the increased complexity of input, one should opt for advanced ML techniques for the extraction of extra insight.

The best performance is again yielded by HAN with AUC 92.64%, followed by XGB and ANN with AUCs of 89%. It is also offering the highest sensitivity of 90% across all datasets and models, making it the recommended solution for the anomaly-detection-type tasks, like fraud detection. Going back to the triad comparison between ANN, HAN, and GPT-2, we can see that the latter does not show much improvement with added FIN data across all metrics. This signals the potentially poor choice of pre-trained embeddings, highlighting the importance of this decision in the design of a DL classifier and reminding that state-of-the-art solutions do not guarantee the superior application results. ANN does not catch up with HAN AUC-wise. However, it showcases the higher F2-score of 90.55%, surpassed only by XGB, which proved to be a promising alternative to the DL methods. The results of modeling on HAN showed its capacity to incorporate and extract additional values from the diversified input, which contributes to the existing field of research and opens new opportunities to the further exploration of data enrichment for fraud detection.

The results of HAN address the RQ 1 and 2, allowing us to conclude that the proposed DL architecture offers a substantial improvement for fraud detection facilitation. Additionally, its properties allow us to offer a look into the "black box" of the DL models and provide the rationale behind the classification decision. This interpretability capacity might be particularly important for practitioners, given the need to substantiate the audit judgment, and will be further explored in the next Section.

4.8 Interpretation and decision support application

SEC developed software specifically focused on the MD&A section (L. Purda & Skillicorn, 2015) to examine the use of language for indications of fraud. The importance of the MD&A section can be observed in reforms introduced by the Sarbanes-Oxley Act (SOX) in 2002, which demanded that the relevant section should present and offer full disclosure on critical accounting estimates and policies (Rezaee, 2005). The length of MD&A sections increased after SOX became effective; nevertheless, Li (2010b) concluded that no changes were made to the information contained within MD&A sections or the style of language adopted. Taking further the fraud detection efforts, we developed a method to facilitate the audit of the MD&A section. We employ state-of-the-art textual analysis to shed light on managers' cognitive processes, which could be revealed by the language used in the MD&A section. L. Zhou et al. (2004) demonstrated that it is plausible to detect lies based on textual cues. Nonetheless, the pertinent cues that allow truthful texts to be distinguished from deceitful ones are dependent on the context. One way to support auditors would be the "red-flag" indication in the body of the

MD&A section. Hajek and Henriques (2017) explored the use of "green-flag" and "red-flag" values of financial indicators and concluded that the identification of non-fraudulent firms is less complex and can be accompanied by interpretable "green-flag" values, however because the detection of fraudulent firms requires more complex non interpretable ML models, no "red-flag" values could be derived. We will take it further and provide the suggestion for the use of textual elements as "red-flags" for auditors. This can be done on the word level or the sentence-level and is to our best knowledge, new to the field. The HAN model allows a holistic analysis of the text structure and the underlying semantics. In contrast to BOW that ignores specific contextual meanings of words, the HAN model considers the grammar, structure, and context of words within a sentence and of sentences within a document, which is essential for the identification of fraudulent behaviour. The attention mechanisms of the HAN at both word and sentence levels retain the logical dependencies of the content and learn to differentiate the important words and sentences. These valuable insights into the internal document structure together with strong predictive performance, make HAN notably advantageous in comparison to BOW-based traditional benchmarks.

Based on the assumption that fraudulent actors are capable of manipulating their writings so that they have convincing similarities to those that are non-fraudulent, only concentrating on words that focus on the content of the text while disregarding the context could be overly simplistic for differentiating truthful from misleading statements. We assume that due to their inherently higher complexity, sentence-level indicators are less prone to manipulation and thus can provide robust insight for auditing.

4.8.1 Word-level

We provide a comparative analysis of words considered to be "red-flags" by the more traditional RF model and those offered by HAN. The RF model proved to be a potent and consistent classifier throughout the comparative analysis. We apply the *lime* methodology of M. T. Ribeiro et al. (2016) to gain insight into the role of different words in the model's classification decision. *lime* stands for Local Interpretable Model-Agnostic Explanations and is based on explaining the model functioning in the locality of the chosen observation. M. T. Ribeiro et al. (2016) explains every input separately; the example of its application to one of the fraud texts can be found in Figure 27:

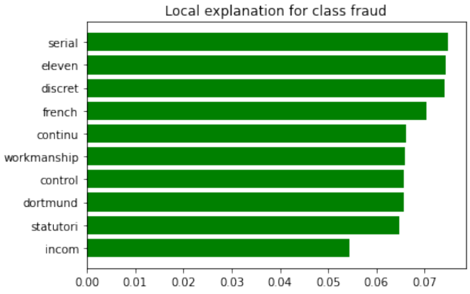
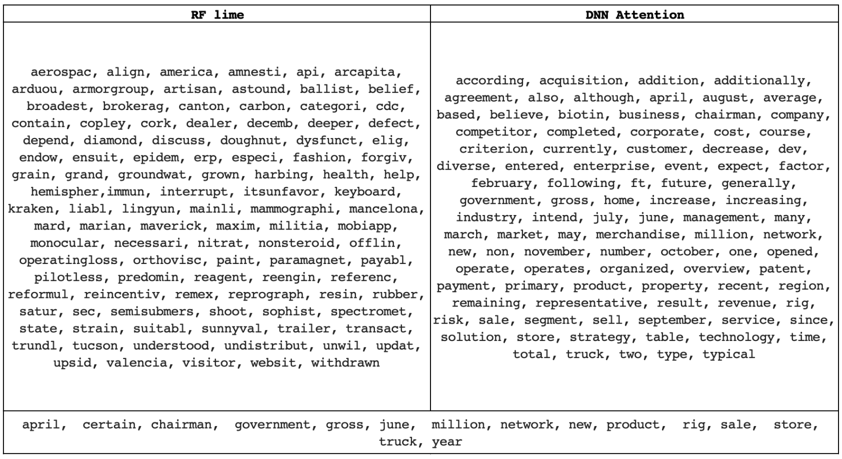


Figure 27: Words with top weights indicating fraud from a sample MD&A

We supply all fraud cases through the *lime* package and extract the top ten words, that have the strongest effect on the model in terms of fraud indication. We further aggregate these words and gain a "red-flag" vocabulary. Additionally we perform the same analysis with the DNN model and extract the weights assigned by the HAN attention layer. The results are summarized in Figure 28:



RF lime	DNN Attention
aerospac, align, america, amnesti, api, arcapita, arduou, armorgroup, artisan, astound, ballist, belief, broadest, brokerag, canton, carbon, categori, cdc, contain, copley, cork, dealer, decemb, deeper, defect, depend, diamond, discuss, doughnut, dysfunct, elig, endow, ensuit, epidem, erp, especi, fashion, forgiv, grain, grand, groundwat, grown, harbing, health, help, hemispher,immun, interrupt, itsunfavor, keyboard, kraken, liabl, lingyun, mainli, mammographi, mancelona, mard, marian, maverick, maxim, militia, mobiapp, monocular, necessari, nitrat, nonsteroid, offlin, operatingloss, orthovisc, paint, paramagnet, payabl, pilotless, predomin, reagent, reengin, referenc, reformul, reincentiv, remex, reprograph, resin, rubber, satur, sec, semisubmers, shoot, sophist, spectromet, state, strain, suitabl, sunnyval, trailer, transact, trundl, tucson, understood, undistribut, unwil, updat, upsid, valencia, visitor, websit, withdrawn	according, acquisition, addition, additionally, agreement, also, although, april, august, average, based, believe, biotin, business, chairman, company, competitor, completed, corporate, cost, course, criterion, currently, customer, decrease, dev, diverse, entered, enterprise, event, expect, factor, february, following, ft, future, generally, government, gross, home, increase, increasing, industry, intend, july, june, management, many, march, market, may, merchandise, million, network, new, non, november, number, october, one, opened, operate, operates, organized, overview, patent, payment, primary, product, property, recent, region, remaining, representative, result, revenue, rig, risk, sale, segment, sell, september, service, since, solution, store, strategy, table, technology, time, total, truck, two, type, typical
april, certain, chairman, government, gross, june, million, network, new, product, rig, sale, store, truck, year	

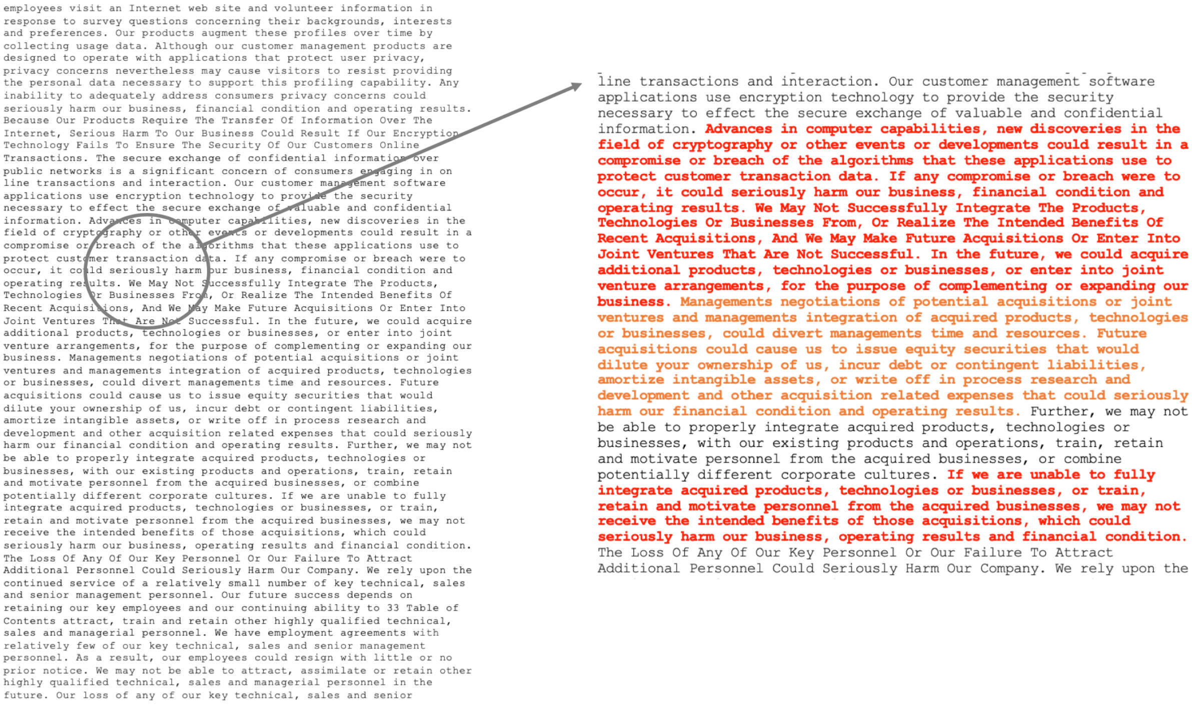
Figure 28: "Red-flag" words identified by Random Forest and HAN, the bottom section contains the words matching both sets

Fifteen words are found to be important for an indication of fraudulent activity by both algorithms, including "government", "certain", "gross", potentially indicating adverse involvement of the state institutions. It would seem that RF derives judgment from the industry: "aerospace", medical terms, "pilotless", "armourgroup". HAN picks up on financial and legal terms like "cost", "acquisition", "property". Both classifiers also include time- and calendar-related words like names of the month. It is not obvious how much the context affects this selection. Additionally, derivation of a word-based rule might potentially lead to a quick adaptation of the reporting entities for audit circumvention. Ambiguous interpretation and manipulation risks motivate the creation of the sentence-level decision support system.

4.8.2 Sentence-level

The added contextual information extracted by the HAN shows improved performance on the test set in comparison to linguistic features and other DNN models. It can be partially explained by the hierarchical structure of language, that entails the unequal roles of words in the overall structure. Following RQ 3, we want to benefit from the structural and context insight retained in sentence-level analysis, provided uniquely by the HAN model.

We extract the sentence-level attention weights for 200 fraudulent reports gained as a result of prediction by HAN and filter the top ten most important sentences per report. The mean weight of a sentence that can be considered a "red-flag" is 0.05, with a maximum at 0.61. We devise a rule, dictating that sentences with weights higher than 0.067 (top 25% quantile) will be referred to as "extra important", sentences between 0.04 and 0.67 (top half) are "important" and those between 0.022 and 0.04 are "noteworthy". These three groups of words get respective coloring and are highlighted in the considered MD&A, as depicted in Figure 29.



employees visit an Internet web site and volunteer information in response to survey questions concerning their backgrounds, interests and preferences. Our products augment these profiles over time by collecting usage data. Although our customer management products are designed to operate with applications that protect user privacy, privacy concerns nevertheless may cause visitors to resist providing the personal data necessary to support this profiling capability. Any inability to adequately address consumers privacy concerns could seriously harm our business, financial condition and operating results. Because Our Products Require The Transfer Of Information Over The Internet, Serious Harm To Our Business Could Result If Our Encryption Technology Fails To Ensure The Security Of Our Customers Online Transactions. The secure exchange of confidential information over public networks is a significant concern of consumers engaging in on line transactions and interaction. Our customer management software applications use encryption technology to provide the security necessary to effect the secure exchange of valuable and confidential information. Advances in computer capabilities, new discoveries in the field of cryptography or other events or developments could result in a compromise or breach of the algorithms that these applications use to protect customer transaction data. If any compromise or breach were to occur, it could seriously harm our business, financial condition and operating results. We May Not Successfully Integrate The Products, Technologies Or Businesses From, Or Realize The Intended Benefits Of Recent Acquisitions, And We May Make Future Acquisitions Or Enter Into Joint Ventures That Are Not Successful. In the future, we could acquire additional products, technologies or businesses, or enter into joint venture arrangements, for the purpose of complementing or expanding our business. Managements negotiations of potential acquisitions or joint ventures and managements integration of acquired products, technologies or businesses, could divert managements time and resources. Future acquisitions could cause us to issue equity securities that would dilute your ownership of us, incur debt or contingent liabilities, amortize intangible assets, or write off in process research and development and other acquisition related expenses that could seriously harm our financial condition and operating results. Further, we may not be able to properly integrate acquired products, technologies or businesses, with our existing products and operations, train, retain and motivate personnel from the acquired businesses, or combine potentially different corporate cultures. If we are unable to fully integrate acquired products, technologies or businesses, or train, retain and motivate personnel from the acquired businesses, we may not receive the intended benefits of those acquisitions, which could seriously harm our business, operating results and financial condition. The Loss Of Any Of Our Key Personnel Or Our Failure To Attract Additional Personnel Could Seriously Harm Our Company. We rely upon the continued service of a relatively small number of key technical, sales and senior management personnel. Our future success depends on retaining our key employees and our continuing ability to 33 Table of Contents attract, train and retain other highly qualified technical, sales and managerial personnel. We have employment agreements with relatively few of our key technical, sales and senior management personnel. As a result, our employees could resign with little or no prior notice. We may not be able to attract, assimilate or retain other highly qualified technical, sales and managerial personnel in the future. Our loss of any of our key technical, sales and senior

line transactions and interaction. Our customer management software applications use encryption technology to provide the security necessary to effect the secure exchange of valuable and confidential information. **Advances in computer capabilities, new discoveries in the field of cryptography or other events or developments could result in a compromise or breach of the algorithms that these applications use to protect customer transaction data. If any compromise or breach were to occur, it could seriously harm our business, financial condition and operating results. We May Not Successfully Integrate The Products, Technologies Or Businesses From, Or Realize The Intended Benefits Of Recent Acquisitions, And We May Make Future Acquisitions Or Enter Into Joint Ventures That Are Not Successful. In the future, we could acquire additional products, technologies or businesses, or enter into joint venture arrangements, for the purpose of complementing or expanding our business. Managements negotiations of potential acquisitions or joint ventures and managements integration of acquired products, technologies or businesses, could divert managements time and resources. Future acquisitions could cause us to issue equity securities that would dilute your ownership of us, incur debt or contingent liabilities, amortize intangible assets, or write off in process research and development and other acquisition related expenses that could seriously harm our financial condition and operating results.** Further, we may not be able to properly integrate acquired products, technologies or businesses, with our existing products and operations, train, retain and motivate personnel from the acquired businesses, or combine potentially different corporate cultures. **If we are unable to fully integrate acquired products, technologies or businesses, or train, retain and motivate personnel from the acquired businesses, we may not receive the intended benefits of those acquisitions, which could seriously harm our business, operating results and financial condition.** The Loss Of Any Of Our Key Personnel Or Our Failure To Attract Additional Personnel Could Seriously Harm Our Company. We rely upon the

Figure 29: A page from MD&A (on the left) and its extract with "red-flag" phrases for the attention of the auditor (on the right). Sentences that contributed the most to the decision towards "fraud" are labeled by HAN as extra important and important. Additional examples are provided in Online Appendix

We propose to use the probability prediction of the HAN model and assign sentence weights as a two-step decision support system for auditors. Given its strong predictive performance, HAN can provide an initial signal about the risks of fraud. Given the selected sensitivity threshold, auditors may select to evaluate a potentially fraudulent report with extra caution and use the highlighted sentences as additional guidance. Given the lengthiness of an average MD&A and limited physical concentration capacities associated with the manual audit, this sort of visual guidance can offer higher accuracy of fraud detection.

4.9 Discussion

As reported in the literature review, Hajek and Henriques (2017), Throckmorton et al. (2015) have tackled the task of combined mining financial and linguistic data for financial statement fraud prediction, and no study was found on the combination of financial and textual data. Given the managerial efforts to conceal bad news by using particular wording (Humpherys et al., 2011) and by generating less understandable reports (Li, 2008; T. Loughran & Mcdonald, 2014), it is pivotal to adopt more advanced text processing techniques.

In line with the findings of Perols (2011) and Y. J. Kim et al. (2016), SVM showed good performance across most experimental setups. This can be explained by the fact that both models can deal with a huge number of features and with correlated predictors. Due to its ability to deal with high dimensional and sparse features, SVM has achieved the best performances in previous studies (Goel & Gangolly, 2012; L. Purda & Skillicorn, 2015) that incorporated the BOW approach. RF came up as the leader in predictive performance, managing to extract knowledge from both financial and BOW-based textual sources. DL models proved capable of distinguishing fraudulent cases. However, only the HAN architecture showcased exceptional capacity to extract signals from the FIN + TXT setting, which is in the center of the current research. The HAN detects a high number of fraudulent cases compared to remaining models, strengthening the statement by L. Zhou et al. (2004) that the detection of deception based on text necessitates contextual information.

The results of the AUC measures indicate that the linguistic variables extracted with HAN and TF-IDF add significant value to fraud detection models in combination with financial ratios. The heterogeneity in performance shifts among different data types for models, showing that different models pick up on different signals, and a combination of these models might be more appropriate to support the decision-making processes of stakeholders in the determination of fraud than the choice of a single model. The use of additional performance metrics like F2-score addressed the practical applicability of the classification models, given the imbalance of error costs. The superior predictive capacity should be considered in combination with the model's sensitivity in order to account for the implications of non-detecting the fraudulent case.

We have explored the interpretation capacities of RF and HAN models on the word and sentence levels. Both models agreed on a specific "red-flag" vocabulary; however, mostly, they picked up on different terms. Also, out of context, these words might be misleading. The indication of "red-flags" words is becoming increasingly unreliable with the adaptive response of the alleged offending parties. The offered sentence-level markup showed a more robust approach to the provision of decision support for the auditors.

4.10 Conclusion

The detection of financial fraud is a challenging endeavor. The continually adapting and complex nature of fraudulent activities necessitates the application of the latest technologies to confront fraud. This research investigated the potential of a state-of-the-art DL model to add to the development of advanced financial fraud detection methods. Minimal research has been conducted on the subject of methods that combine the analysis of financial and linguistic information, and no studies were discovered on the application of text representation based on DL to detect financial statement fraud. In addition to quantitative data, we investigated the potential of the accompanying text data in annual reports, and have emphasized the increasing significance of textual analysis for the detection of signals of fraud within financial documentation. The proposed HAN method concentrates on the content as well as the context of textual information. Unlike the BOW method, which disregards word order and additional grammatical information, DL is capable of capturing semantic associations and discerning the meanings of different word and phrase combinations.

The results have shown that the DL model achieved considerable improvement in AUC compared to the benchmark models. The findings indicate that the DL model is well suited to identify the fraudulent cases correctly, whereas most ML models fail to detect fraudulent cases while performing better at correctly identifying the truthful statements. The detection of fraudulent firms is of great importance due to the significantly higher MC associated with fraud. Thus, specifically in the highly unbalanced case of fraud detection, it is advisable to use multiple models designed to capture different aspects.

Based on these findings, we conclude that the textual information of the MD&A section extracted through HAN has the potential to enhance the predictive accuracy of financial statement fraud models, particularly in the generation of warning signals for the fraudulent behavior that can serve to support the decision making-process of stakeholders. The distorted word order handicaps the ability of the BOW-based ML benchmarks to offer a concise indication of the "red-flags". We offered the decision support solution to the auditors that allows a sentence-level indication of text fragments that trigger the classifier to treat the submitted case as fraudulent. The user can select the degree of impact of indicated sentences and improve the timing and accuracy of the audit process.

4.11 Appendix

4.11.1 Financial Variables (FIN)

Variable (named as in dataset)	Compustat Index	Description
Balance Sheet Variables:		
Acc.Payables	Data 70	Accounts Payable
AccRec.DbtsT	Data 2	Accounts Receivable
AmrtOfIntang	Data 65	Amortisation of Intangibles
Assets.Total	Data 6	Total Assets
AverageTA		Average Total Assets = ($Total\ Assets_t +\ Total\ Assets_{t-1}$)/2
CashSTInvest	Data1	Cash and Short-term Investments
COGS.costsgd	Data 41	Cost of Goods Sold
Comm.OrdinEQ	Data 60	Common/Ordinary Equity
CurrAssetTot	Data 4	Current Assets Total
CurrLiabsTot	Data 5	Current Liabilities Total
DeferrTaxInc	Data 50	Income Taxes - Deferred
DeprAmortTot	Data 14	Depreciation Amortization Total
Fin.Actv.NCF	Data 313	Financing Activities/Net Cash Flow
Income.IBEXi	Data 18	Income Before Extraordinary Items
InventoryPrx	Data 3	Inventories Total
LT.Debt.Totl	Data 9	Long-term Debt Total
Oper.Act.NCF	Data 308	Operating Activities/Net Cash Flow
PPE.TotalNet	Data 8	Property Plant and Equipment (PPE)
Sales.TurnNt	Data 12	Sales/Turnover Net
SG.A.Expense	Data 189	Selling, General and Administrative Expense (SGA Expense)

Financial Ratios:

EBIT.Marginn	Earnings before Interest and Taxes Margin
EBITDA.Marginn	Earnings before Interest, Tax, Depreciation and Amortisation Margin
GrProfitMarg	Gross Profit Margin
NtProfitMargin	Net Profit Margin
CashFlMargin	Cash Flow Margin
ROA.finratio	Return on Assets
ROE.RnCommEQ	Return on Equity

Beneish score (Beneish, 1999):

PROBM	Probability of manipulation to overstate earnings = $-4.84 + .920 * DSR + .528 * GMI + .404 * AQI + .892 * SGI + .155 * DEPI - .172 * SGAI + 4.679 * Accruals - .327 * LEVI$
ACCRUALS	Total Accruals to Total Assets Index =$(Income\ Before\ Extraordinary\ Items -\ Operating\ Activities/Net\ Cash\ Flow)/Total\ Assets$
AQI	Asset Quality Index= $[(Total\ Assets_t - Current\ Assets_t - PPE_t)/Total\ Assets_t]/[(Total\ Assets_{t-1} - Current\ Assets_{t-1} - PPE_{t-1})/Total\ Assets_{t-1}]$
DEPI	Depreciation Index = $[Depreciation_{t-1}/(Depreciation_{t-1} +\ PPE_{t-1})]/[Depreciation_t/(Depreciation_t + PPE_t)]$
DSR	Days Sales in Receivables Index = $(Receivables_t/Sales_t)/(Receivables_{t-1}/Sales_{t-1})$
GMI	Gross Margin Index =$[(Sales_{t-1} - Cost\ of\ Goods\ Sold_{t-1})/Sales_{t-1}]/[(Sales_t - Cost\ of\ Goods\ Sold_t)/Sales_t]$
LEVI	Leverage Index = $[(Longterm\ Debt_t + Current\ Liabilities_t)/Total\ Assets_t]/[(Longterm\ Debt_{t-1} + Current\ Liabilities_{t-1})/Total\ Assets_{t-1}]$
SGAI	Sales General and Administrative Expenses Index = $[(SGA\ Expense_t)/Sales_t]/[(SGA\ Expense_{t-1})/Sales_{t-1}]$

SGI	Sales Growth Index = $Sales_t/Sales_{t-1}$
Accruals quality related variables (Dechow et al., 2011):	
dWC_Accruals	Change in working capital accruals = $\Delta Current\ Assets - \Delta Current\ Liabilities - \Delta Cash\ and\ Shortterm\ Investments$
dInventories	Change in Inventories = $\Delta Inventories/Average\ Total\ Assets$
dReceivables	Change in Receivables = $\Delta Receivables/Average\ Total\ Assets$
Performance variables (Dechow et al., 2011):	
dCash_Margin	Change in Cash Margin = $(Cost\ of\ Goods\ Sold - \Delta Inventories + \Delta Accounts\ Payables)/(Sales\ Turnover\ Net - \Delta Accounts\ Receivable)$
dCash_Sales	Change in Cash Sales = $Sales\ Turnover\ Net - \Delta Accounts\ Receivable$
dDef_Tax_Expense	Change in Deferred Tax Expense = $Deferred\ tax\ expense_t/Total\ Assets_{t-1}$
dEarnings	Change in Earnings =$\Delta(Income.IBEXi/Average\ Total\ Assets)$
Market-related incentives (Dechow et al., 2011):	
Leverage	Leverage = $Longterm\ Debt/Total\ Assets$
CFF	Level of finance raised = $(Financing\ Activities/Net\ Cash\ Flow)/Average\ Total\ Assets$

4.11.2 Linguistic Variables (LING):

Word Category Proportions based on L&M word lists by T. I. M. Loughran and Mcdonald (2011):	
mda_word_count	Total Word Count
mda_positive_word_proportion	Positive Word Count/Total Word Count
mda_negative_word_proportion	Negative Word Count/Total Word Count
mda_uncertainty_word_proportion'	Uncertainty Word Count/Total Word Count
mda_constraining_word_proportion	Constraining Word Count/Total Word Count
mda_tone	(Positive Word Count – Negative Word Count)/ (Posistive Word Count+Negative Word Count))
Readability ratios by Humpherys et al. (2011), Li (2008):	
mda_average_sentence-_length	Total Word Count/Total Sentence Count
mda_complex_word_proportion	Complex Word (more than 2 syllables) Count/Total Word Count
mda_fog_index	0.4 * (Average Sentence Length + Complex Word Proportion)

4.11.3 Data Preprocessing

For BOW TF-IDF - benchmark models

- words and sentences are tokenised
- numbers and punctuation marks are removed
- all characters are transformed to lower case
- stop words are removed using a standard vocabulary of english
- company names that are specific to each MD&A section are filtered out as they are not characteristic for either the fraudulent nor non fraudulent annual reports

- all words are stemmed
- the resulting corpus consists of **21 213** unique terms.

For DL - GPT-2 and HAN models

In order to obtain the input for the DL models, a vectorised representation of the data is created. Many of the DL models make use of word embeddings learned with word2vec.

- all numbers are filtered out
- documents are split into sentences then each sentence into words
- all punctuation is removed
- all words are converted to lower case
- words are converted into a list of unique indexed tokens, that forms the dictionary of the corpus
- in contrast to the pre-processing steps needed for the BOW approach, stemming and stopwords removal are not necessary since the pre-trained word2vec captures these.
- rare words occurring less than twice in the corpus are further filtered out, which account for 28.86% of the text, yielding a dictionary of **18 302** unique terms

In the case of variable length input sequences for the DL model, it is necessary to ensure the equal sequence length for every observation. This procedure is defined as padding and truncation and must be implemented at both document and sentence levels. The same fixed length of sentences per document N_w and words per sentence N_s is fixed in the manner described below:

- longer sentences are truncated to N_w words, such that words are eliminated, whereas shorter sentences are padded with zeros to compensate for the missing words
- longer documents are truncated to N_s sentences and the shorter documents are padded with zero vectors to compensate for the missing sentences

In order to find appropriate values for N_w and N_s a comparison of the median number, the 90% and the 95% quantiles of the number of words per sentence and of sentences per document for the fraudulent and non fraudulent pre-processed MD&A sections is conducted to make sure that there are no major differences between the two classes and that no valuable information about the fraudulent text is being lost.

As can be seen in Table 12, no major differences between the fraudulent and non fraudulent cases in terms of sequence length can be observed. The resulting data of dimension $1163 \times 907 \times 38$, whereby each word is being represented by its word index, is now ready to be inputted into the DL model.

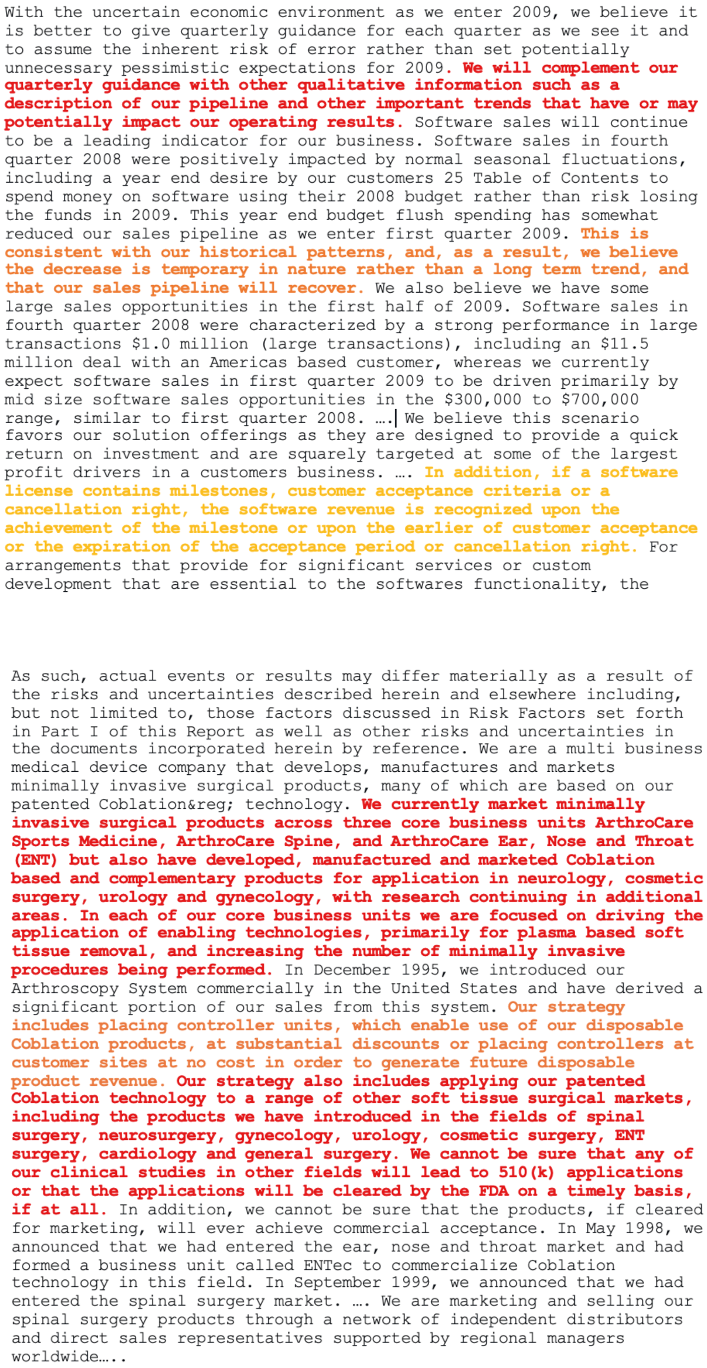
	fraud	no fraud	fraud &no fraud
mean num. words/sentence	22	23	23
median num. words/sentence	20	20	20
90% quantile num. words/sentence	37	38	38
95% quantile num. words/sentence	45	47	47
mean num. sentences/document	549	599	591
median num. sentences/document	380	360	364
90% quantile num. sentences/document	953	901	908
95% quantile num. sentences/document	1849	1366	1443

Table 12: Sequence length distribution of fraudulent and non fraudulent MD&As

4.11.4 Hyperparamteres of benchamrk models

Algortithm	**Parameters**
LR	L2 regularization and the "lbfgs" solver
SVM	regularization parameter C =1.0, "rbf" kernel, gamma=0.0024
RF	number of estimators=100, maximum depth=None, minimum samples split=2, maximum features=20
XGB	learning rate=0.1, number of estimators=100, maximum depth=3
ANN	120 neurons, batch size=20, 100 epochs

4.11.5 Examples of sentence-level "red-flag" marking



With the uncertain economic environment as we enter 2009, we believe it is better to give quarterly guidance for each quarter as we see it and to assume the inherent risk of error rather than set potentially unnecessary pessimistic expectations for 2009. **We will complement our quarterly guidance with other qualitative information such as a description of our pipeline and other important trends that have or may potentially impact our operating results.** Software sales will continue to be a leading indicator for our business. Software sales in fourth quarter 2008 were positively impacted by normal seasonal fluctuations, including a year end desire by our customers 25 Table of Contents to spend money on software using their 2008 budget rather than risk losing the funds in 2009. This year end budget flush spending has somewhat reduced our sales pipeline as we enter first quarter 2009. **This is consistent with our historical patterns, and, as a result, we believe the decrease is temporary in nature rather than a long term trend, and that our sales pipeline will recover.** We also believe we have some large sales opportunities in the first half of 2009. Software sales in fourth quarter 2008 were characterized by a strong performance in large transactions $1.0 million (large transactions), including an $11.5 million deal with an Americas based customer, whereas we currently expect software sales in first quarter 2009 to be driven primarily by mid size software sales opportunities in the $300,000 to $700,000 range, similar to first quarter 2008. …. We believe this scenario favors our solution offerings as they are designed to provide a quick return on investment and are squarely targeted at some of the largest profit drivers in a customers business. …. **In addition, if a software license contains milestones, customer acceptance criteria or a cancellation right, the software revenue is recognized upon the achievement of the milestone or upon the earlier of customer acceptance or the expiration of the acceptance period or cancellation right.** For arrangements that provide for significant services or custom development that are essential to the softwares functionality, the

As such, actual events or results may differ materially as a result of the risks and uncertainties described herein and elsewhere including, but not limited to, those factors discussed in Risk Factors set forth in Part I of this Report as well as other risks and uncertainties in the documents incorporated herein by reference. We are a multi business medical device company that develops, manufactures and markets minimally invasive surgical products, many of which are based on our patented Coblation&reg; technology. **We currently market minimally invasive surgical products across three core business units ArthroCare Sports Medicine, ArthroCare Spine, and ArthroCare Ear, Nose and Throat (ENT) but also have developed, manufactured and marketed Coblation based and complementary products for application in neurology, cosmetic surgery, urology and gynecology, with research continuing in additional areas. In each of our core business units we are focused on driving the application of enabling technologies, primarily for plasma based soft tissue removal, and increasing the number of minimally invasive procedures being performed.** In December 1995, we introduced our Arthroscopy System commercially in the United States and have derived a significant portion of our sales from this system. **Our strategy includes placing controller units, which enable use of our disposable Coblation products, at substantial discounts or placing controllers at customer sites at no cost in order to generate future disposable product revenue. Our strategy also includes applying our patented Coblation technology to a range of other soft tissue surgical markets, including the products we have introduced in the fields of spinal surgery, neurosurgery, gynecology, urology, cosmetic surgery, ENT surgery, cardiology and general surgery. We cannot be sure that any of our clinical studies in other fields will lead to 510(k) applications or that the applications will be cleared by the FDA on a timely basis, if at all.** In addition, we cannot be sure that the products, if cleared for marketing, will ever achieve commercial acceptance. In May 1998, we announced that we had entered the ear, nose and throat market and had formed a business unit called ENTec to commercialize Coblation technology in this field. In September 1999, we announced that we had entered the spinal surgery market. …. We are marketing and selling our spinal surgery products through a network of independent distributors and direct sales representatives supported by regional managers worldwide…..

Figure 30: Sentences that contributed the most to the decision towards "fraud" label by HAN. Extra important, important and noteworthy sentences are highlighted and should be revised by the auditor.

5 Quantification of Economic Uncertainty: a deep learning application

A.Kim, N.Gillmann, S.Lessmann

5.1 Abstract

Research on the measurement of uncertainty has a long tradition. Recently, the creation of the economic policy uncertainty index sparked a new wave of research on this topic. The index is based on major American newspapers with the use of manual labeling and counting of specific keywords. Several attempts of automating this procedure have been undertaken since, using Support Vector Machine and LDA analysis. The current paper takes these efforts one step further and offers an algorithm based on natural language processing and deep learning techniques for the quantification of economic policy uncertainty. The new approach allows an accurate distillation of the latent "uncertainty" underlying newspaper articles, enables an automated construction of a new index for the measurement of economic policy uncertainty, and improves on existing methods. The potential use of our new index extends to the areas of political uncertainty management, business cycle analysis, financial forecasting, and potentially, derivative pricing.

5.2 Introduction

Leading economic experts agree that the slowdown of world economic growth during the period 2018-2019 can be mainly attributed to high uncertainty about political decisions (Tripier, 2019). Events like the "trade war" between the US and China, Brexit, US sanctions against Iran, and the demonstrations in Hong Kong fall into this period. One of the main channels through which the high level of uncertainty affects world economic growth is a falling investment rate of private companies around the world (Bobasu et al., 2020).

Economic researchers are still debating about the exact effects of uncertainty shocks on economic activity. Economic theory puts forward precautionary savings as the most prominent explanation (Kimball, 1990). This theory states that when uncertainty increases, actors will put their activities on hold until there is more clarity (Leduc & Liu, 2016). Contrarily, the risk premium theory mentioned in Christiano et al. (2014) argues that the effect of increased uncertainty can even be positive in specific scenarios. Bloom (2009) suggested a "wait-and-see" effect. However, at the moment, no dominating theory can be established.

One of the significant reasons why research cannot agree on the effects of uncertainty shocks is that there is disagreement on how to measure uncertainty. Recent literature proposes several proxies: volatility of the stock market (Bloom, 2009), dispersion in forecasts of professional forecasters (Glas, 2019; Y. Liu & Sheng, 2019; Sill, 2014), disagreement in the expectations of survey participants (Bachmann et al., 2013a; Claveria, 2019), as well as some data-driven approaches (Jurado et al., 2015). The first three proxies share the same shortcoming: they measure perceptions of individual uncertainty instead of the general underlying state of uncertainty.

This personal perception tends to differ from the aggregated uncertainty in the economy, especially during periods of high volatility, when the formulation of expectations about the future is nontrivial. The latter proxy is trying to overcome this problem by aggregating individual information. Its potential shortcomings lie in the large amounts of economic data required, which may cause slower response to the change in underlying uncertainty.

Most recently, economists adopted text data as a source to obtain additional information about the economy. Alexopoulos and Cohen (2015) and Baker et al. (2016) presented the first papers that used text to quantify economic uncertainty. The latter one explored the information potential of newspaper articles by constructing an economic policy uncertainty (EPU) index. The index is based on the share of articles classified as uncertain in the pool of general articles per newspaper per month. This method became widely accepted regardless of some underlying limitations (discussed further), stemming from the absence of previous research. The authors had to come up with a set of rules to identify an EPU article. Manual labeling led to the creation of a dictionary-based method (further referred to as "BBD") that allowed further automated labeling. In 2018 a new World Uncertainty Index (WUI) was introduced by Ahir et al. (2018) - an index also relying on the explanatory power of textual data from Quarterly Economist Intelligence Unit Country reports. WUI is built using a keyword methodology similar to BBD. Advanced text mining methods have been applied in the economic literature since 2017. The unsupervised learning approach is represented by LDA, an algorithm that facilitates identifying latent topics in a document without pre-labeling the data. After identifying the topics in a set of newspaper articles, the researcher can choose those she considers relevant and construct an index from them. Examples are Azqueta-Gavaldón (2017), Larsen (2017), and Thorsrud (2018). Unfortunately, the topics resulting from LDA are not named and do not necessarily match the particular research interest. This limits the usefulness of LDA for the quantification of economic uncertainty. Supervised learning provides a way of identifying relevant keywords inside the text corpus without manual definition and arbitrarily constructed topics. For example, Tobback et al. (2018) used Support Vector Machine (SVM) and applied it to the corpus of six Belgian newspapers over the time from 2000 until 2013. They restricted their initial sample of newspaper articles to those talking about uncertainty in Belgium or the EU. The SVM-based classification model was used to predict the binary label (containing or not containing economic policy uncertainty) of every article and reconstruct the index using the BBD methodology. The resulting time-series had superior predictive power over some of the Belgian macro indicators like bond yield and spread, the credit default swap spread, and consumer confidence as opposed to an index based on the original BBD method.

Some other examples of the recent use of text data for economic questions are (Audrino et al., 2020) who use different newspapers and social media as data sources to assess the impact of attention and sentiment variables on stock market volatility, and (Ardia et al., 2019) who analyze the value of sentiment variables for economic growth forecasting using numerous textual data sources.

Our goal is to offer a new method to quantify economic policy uncertainty that would demonstrate an accurate, robust, and adaptable performance, with an additional insight stemming from its interpretability. In pursuing this goal, the paper contributes to the area of economic policy uncertainty quantification. It introduces a state-of-the-art deep learning model for textual analysis with improved predictive power (as opposed to previously used algorithms) and adaptability features in the setting of the changing newspaper rhetoric. The latter is substantiated by the predictive power of the reconstructed index for economic indicators associated with uncertainty, like stock market returns, employment, and industrial production. Thus, the reconstructed index may be used by economic and financial institutions for the evaluation and forecasting of economic behavior, business cycles, as well as the assessment of effects of monetary policy and political decisions. Additionally, we offer insights on the role different words play in the classification of EPU through time, which constitutes additional value for economic and social analysis.

In line with the announced goals, we have formulated three research questions (RQ) that define the empirical design:

- RQ 1: can a deep learning classifier learn to distinguish the latent concept of uncertainty without using any keywords but the textual semantics of a newspaper article instead?

We have considered the recently developed natural language processing (NLP) models that make use of deep learning (DL) and transfer learning (Radford et al., 2019). The proposed binary classifier distinguishes between articles containing or not containing EPU. We train the model on an article corpus labeled according to the BBD methodology, given its wide adoption by practitioners (Ghirelli et al., 2019; Soric & Lolic, 2017; Zalla, 2017) and absence of non-manual labeling alternatives. We compare the performance of the proposed approach with some well-known algorithms like SVM and Random Forest, as well as test its robustness with 10-fold cross-validation with stratification (given a major target label imbalance).

- RQ 2: how does the uncertainty rhetoric change in time?

To evaluate the temporal dynamics of the newspaper vocabulary, underlying the concept of uncertainty in analyzed corpora, we open the black box of a neural network and analyze which words of the input article were considered the most important by the classifier. We perform this task for 1000 EPU articles from every year, select the top ten words per article, assign them a rank from 10 to 1 and then sum up the ranks of the entire vocabulary. We further select the ten highest ranking words that will represent the "uncertainty drivers" for the analyzed year.

- RQ 3: can the selected DL methodology show better adaptability to the changing rhetoric than the BBD index?

To explore the adaptive capacities of the model, we have transformed the values predicted during cross-validation into an index using the original BBD methodology. We obtain two indices: one reconstructed from predicted values and one reconstructed from "true" EPU values based on our newspaper data. We offer a comparative analysis of the original and predicted time-series indices. Firstly, we explore their co-movement with alternative uncertainty proxies to identify comparable proxies, and then assess their predictive power over a set of key macroeconomic variables.

5.3 New deep learning-based EPU index

Our benchmark throughout the paper will be the methodology of BBD. In this section, we describe the original EPU index methodology, then discuss our dataset and explain the reconstruction procedure of a new DL-based EPU index.

5.3.1 Original BBD methodology

Baker et al. (2016) label every article in their newspaper sample as "0" if it contains no EPU (also referred to as simply economic (E) article) and "1" if it contains EPU. The labeling decision is based on the presence of three sets of keywords: "economy or economic" + a term from a group referring to policy (Congress, deficit, Federal Reserve, legislation, regulation, White House) + "uncertainty or uncertain". If an article contains at least one keyword from all three groups, it is labeled as containing EPU (target="1"). The share of EPU articles when applying the BBD methodology to our corpus can be seen in Figure 31:

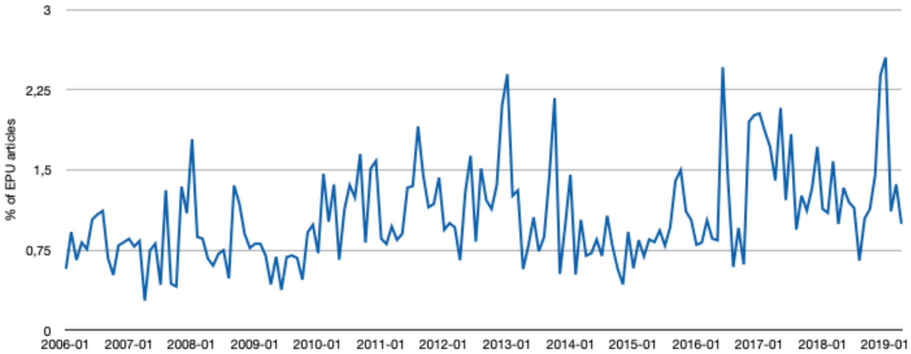


Figure 31: Share of EPU articles (label="1") in the dataset per month (%).

The set of keywords that BBD use for labeling was derived from an extensive manual audit of a large corpus of articles from ten leading US newspapers. Since the use of the "economic" and "uncertainty" keywords is undisputed, the BBD authors decided to only include articles containing the terms for "economic" and "uncertainty" in their audit study. The audit study then helped to identify which policy terms are necessary to identify relevant EPU articles. In detail, groups of researchers were reading random samples of newspaper articles from the collection and labeling them as either related to EPU, not related to EPU or hard to tell. BBD refined the set of keywords until both human labeling and search requests yielded the same EPU index.

The output of the labeling is a binary classification of newspaper articles into those containing economic policy uncertainty (target=1) and those with general economic news (target=0).

This classification can be transformed into a monthly index by calculating the share of EPU articles per newspaper per month and standardizing the share for each newspaper individually. Additionally the average over all newspapers for each month is standardized to a mean of 100.

The method proposed by BBD has a number of limitations that were partially addressed in the years following the original publication. The set of fixed keywords, while easy to implement, inevitably leads to an oversimplification. Moreover, the constant set of filter words means potential failure of the devised rule to pick up on changes in the vocabulary, which may occur over time.

5.3.2 Data and Index reconstruction

Data availability forced us to make several filtering decisions for the set of articles that we could use for modeling. Thus, when interpreting the resulting indices, one should keep in mind that our starting sample might differ slightly from that of Baker et al. (2016). The use of all available articles did not appear possible due to the limitations of our data source Lexis Nexis Uni. Thus, we had to narrow down the search. We followed the example of Tobback et al. (2018), assuming that most relevant articles must contain the "economy" or "economic" keywords. Baker et al. (2016) performed this alternative filtering for their audit study to an even greater extent and collected 12 009 full-text articles. The most recent contribution in the literature by Tobback et al. (2018)) offers an analysis of 210 000 full-text articles. Our analysis is done on 315 543 articles, from 01 Jan 2006 to 30 Apr 2019, offering the biggest text corpus so far. The start date before the Global Financial Crisis (GFC) is selected in order to capture both periods with normal and high levels of EPU. We aspired to include newspapers that guarantee coverage across the whole of the USA, however these are not the exact BBD newspapers. Our articles come from The Washington Post, Pittsburgh Post-Gazette (Pennsylvania), The Atlanta Journal-Constitution, St. Louis Post-Dispatch (Missouri), The Philadelphia Inquirer (Pennsylvania), USA Today, Star Tribune (Minneapolis, MN), The Orange County Register (California), Tampa Bay Times (Florida, previously known as St. Petersburg Times) and The New York Post. The distribution of articles across newspapers is shown in Table 13:

Table 13: Number of economic articles per newspaper, 01 Jan 2006 - 30 April 2019.

Newspaper	Number of articles
The Washington Post	81 734
Pittsburgh Post-Gazette (Pennsylvania)	41 225
Tampa Bay Times	36 436
USA Today	26 267
The Atlanta Journal-Constitution	26 038
St. Louis Post-Dispatch (Missouri)	25 400
The Philadelphia Inquirer (Pennsylvania)	22 502
Star Tribune (Minneapolis, MN)	21 422
The Orange County Register (California)	19 983
The New York Post	14 536

The uneven distribution of articles is partially explained by the size of newspapers' editorial offices and a large amount of reprints and reposts of existing articles (NYP in particular) that were dropped from the sample. The articles are used for modeling without regard to the source, while the index reconstruction method accounts for the distribution skews.

5.3.3 Index reconstruction

Addressing the stated RQ resulted in the creation of a new EPU index. The data points are obtained through predicting the label of newspaper articles as EPU or non-EPU with a DL algorithm. The initial labeling of the train and test sets was performed according to the original BBD methodology, as well as the transformation of the binary results into an index time-series. The proposed novel method for classification is established within the NLP framework and comprises learning the latent representation of EPU from textual input. Superior predictive performance is achieved using a deep neural network architecture with advanced components like pre-trained embeddings, bidirectional GRU layers, and attention layer (Bengio et al., 2003).

5.4 Methodology

DL applications in economics are yet sparse. Thus, we will revisit the principles of DL and NLP, as well as provide a detailed configuration of the selected classification model. Following RQ 1, we are offering a classification model that is capable of identifying uncertainty in newspaper articles without a fixed set of keywords. This model is based on DL and NLP techniques, namely a recurrent neural network that uses GPT-2 pre-trained embeddings (Radford et al., 2019) and an attention mechanism (Vaswani et al., 2017). This section begins with the description of the data pre-processing steps, followed by the introduction of the embeddings concept and the GPT-2 language model. We further provide clarifications on the DL architecture and elaborate on the attention mechanism. The latter will be instrumental for RQ 2, when we address the change of the "uncertainty drivers" over time.

5.4.1 Data pre-processing

In order for an NLP model to process text, the words are converted into a numeric representation. We analyzed the average lengths of the article body and the headline. Table 14 shows that EPU articles tend to be longer. This particularity is accounted for during the text pre-processing.

Table 14: Average number of words in E- and EPU-labeled articles and corresponding headlines before preprocessing.

	Average length of the article body	Average length of the headline
All	823	9.31
E articles	817	9.2
EPU articles	1 087	10.2

The corpus vocabulary has to be carefully considered in order to facilitate the task of knowledge extraction. This entails homogenizing and cleaning the provided textual data from noise. The headlines were integrated into the text. The pre-processing steps included three main stages. The first stage comprised of vocabulary filtering: opening up and converting the contraptions ("can't" into "cannot") and removal of the usual stopwords (excluding negations ("not")). All words that occur less than ten times were also dropped (bringing the vocabulary size from 468 997 to 114 763), which allowed accounting for misspelling as well. Importantly, train and validation sets were stripped of the keywords "policy or political"+"uncertainty or uncertain" to ensure that the classifier does not learn only based on their presence or absence. During the second stage, numbers and irrelevant components like internet links and punctuation were removed. During the third stage, text got transformed to lower case. As a result of pre-processing, the average length of the article shrank to 407 words. The cleaned article text is broken into a list of words (tokens).

5.4.2 Natural Language Processing: language models

NLP focuses on the methods that allow machines to analyze and evaluate human language. The task of text representation in a numeric format lies at the basis of NLP. However, modeling a system as complex and intricate as a human language proved to be a very complex task, even with the appearance of large digital corpora of text in the 90s. Teaching computers to understand the written text involved the necessity of approximating the irregular structure of the human expression and modeling language rules, leading to the introduction of Language Models (LM). Nowadays, LM are used in machine translation, text classification, speech recognition, handwriting recognition, information retrieval, and many other (Bahdanau et al., 2014; Graves et al., 2013; Hirschberg & Manning, 2015).

Two main classes are statistical LM and neural LM. The first class uses traditional statistical techniques like N-grams and linguistic rules to learn the probability distribution of words in a studied text (one of the early examples is Bahl et al. (1989)). Widely used solutions included one-hot encoded bag-of-words (BoW) vector representations and the TF-IDF representations (also known as frequency embeddings, Salton et al. (1975). The latter represents the matrix of document vectors, containing term occurrence frequencies (TF) or their transformation by weighting with the inverse document frequency (IDF). The key idea of TF-IDF representation lies in the assignment of larger weights to words with higher discriminatory ability. This principle entails that frequent occurrence of a term in the document does not lead to high importance; rather, the word must be unique for that document at the corpus level. This ranking is widely used in NLP, in particular, for sorting data into categories, as well as keywords extraction.

The second class became a new powerful tool for NLP with the adoption of neural networks to model language (Bengio et al., 2003). This area saw tremendous developments in recent years and became industrial state-of-the-art, used in Google translate, virtual assistants like Apple's Siri and Amazon's Alexa. We have applied the solutions developed by OpenAI, who released a new language model called GPT-2 in 2019. GPT-2 is a transformer-based generative language model that was trained on 40 GB of curated text from the internet (Radford et al., 2019).

5.4.3 Embeddings

In order to preserve the semantic meaning and linguistic characteristics of a word, we can transform it into a vector representation, called word embedding. Although known before (frequency-based embeddings and vector-space model), the concept of word embeddings re-emerged in 2013 with the introduction of prediction-based "neural" embeddings. The work of Mikolov, Chen, et al. (2013) started a new chapter in the development of the field, allowing to represent words as numeric vectors without the sparsity of one-hot encoded matrices and retention of the semantic meaning as opposed to TF-IDF representations. The proposed word2vec is an advanced model for word embedding, composed of a neural network model that is capable of learning word representations during training on a large text corpus. Mikolov, Chen, et al. (2013) offer two types of training task for the procurement of embeddings: CBOW and Skip-gram. The former forces the model to predict a target word from a window of adjacent context words, while the latter entails prediction of a context window from the provided target word. The resulting word vectors ("inflation"=[0.5, -0.0123, ... 2.1]) are located within the multi-dimensional vector space in such a way that words sharing common contexts within the corpus are positioned next to each other.

The initial word2vec algorithm was followed by GloVe (Pennington et al., 2014a), FastText (Bojanowski et al., 2016), and GPT-2 (Radford et al., 2019), as well as the appearance of publicly available sets of pre-trained embeddings that were acquired by applying the above-mentioned algorithms on large text corpora.

The GPT-2 pre-trained embeddings used for the proposed model, were trained on 250 thousand documents from the WebText, as stated by Radford et al. (2019). A machine learning method, where a model developed for a specific task, is reused and becomes a starting point for a model on a different task, got known as transfer learning. As defined by Goodfellow et al. (2016), "transfer learning and domain adaptation refer to the situation where what has been learned in one setting . . . is exploited to improve generalization in another setting". Usage of pre-trained embeddings proved to be useful for achieving a superior performance in most NLP tasks (Dai & Le, 2015; Howard & Ruder, 2018; Peters et al., 2018; Radford et al., 2018). Given the limited size of our newspaper corpus, we use word embeddings that were trained on a much larger sample as part of our model for EPU classification. To that end, we replace the words of an article with its pre-trained embedding feature vector. This approach maintains the word order in an article. Given the sequential nature of the data, the architecture of a classifier plays a critical role in obtaining a prediction accuracy.

5.4.4 Deep Learning: recurrent and bidirectional neural networks

DL is a subset of Machine Learning primarily based on the hierarchical approach, where each step converts information from the previous step into more complex representations of the data (Goodfellow et al., 2016). We refer to deep learning when the used Artificial Neural Network (ANN) uses multiple layers (L. Deng & Yu, 2014). DL methodology aims at learning multiple levels of representations from data, with higher levels reflecting more abstract concepts, thus capturing the complex relations between the data set features (A. Kim et al., 2020). This ability made DL a popular solution for a wide range of modeling tasks. The adoption of DL methods in scientific areas like economics, however, was limited by the necessary computational capacities and interpretability issues. Neural networks notoriously represent a 'black box' - a shortcoming originating of its inherent internal complexity (Gilpin et al., 2018).

Regardless of these shortcomings, the development of DL offered a versatile toolbox for the processing of sequential data i.e., time series and text. New DL architectures like convolutional neural networks (CNN, Kalchbrenner et al. (2014)), recurrent neural networks (RNN, Hochreiter and Schmidhuber (1997a)), Hierarchical Attention Networks (HAN, Yang et al. (2016b)) were successfully employed for learning textual representations (Krauss et al., 2017). In particular, Athiwaratkun and Stokes (2017), Yin et al. (2017b), Zhang et al. (2018) showcase the ability of RNN variation like Gated Recurrent Unit (GRU) to show improved performance on NLP tasks. As opposed to RNN that may fail to capture the long-term information due to the gradient vanishing problem, the GRU is equipped with a set of "gates" that allow GRU to dynamically remember and forget the information flow, which is crucial for longer text inputs (Cho et al., 2014).

Addressing the non-linear nature of text understanding, Schuster and Paliwal (1997) suggested a further reinforcement of the RNN with a bidirectional component. For the case of uncertainty classification, analyzing the preceding, as well as the following observations, is equally important for the extraction of the semantic concepts (F. Liu et al., 2020). Their ability to grasp long-term dependencies motivated the choice of a bidirectional GRU layer as a significant component of the suggested classifier. The full DL architecture is represented in Figure 32, where x_1 to x_T represent the textual input transformed into *tokens*. As mentioned before, the average length of an article is 407 words, which was established as a fixed input length (T=407). IN order to feed in the article into the DL model, text strings must be numeric. We transform every word into a token ("inflation"-> "34") and create a lookup vocabulary that allows to map the tokens back to words. We further truncate longer articles and *pad* shorter articles. Padding means adding fixed values (in our case "0", which doesn"t have any semantic meaning to it) in the beginning of an article until it reaches a length of 407 tokens. The output is represented by the single neuron with sigmoid activation, given the binary classification task. The model outputs probabilities for the supplied array of *tokens* representing an article to be containing EPU (target=1). The layer that follows the input is a dense matrix of embeddings. As discussed above, we use a set of pre-trained GPT-2 embeddings. Every word in the dictionary (114 763 words) is assigned an embedding vector of 768 neurons (defined by the authors of the GPT-2 language model). Thus, the embedding matrix has dimensions 768 x 114 763 and functions as a look-up table. Input integers are used as the index to access this table. We have e_t vectors as output, each representing an input word. These vectors are supplied to the bidirectional GRU layer that will process them word by word. This layer's output will be a hidden state h^t vector that will go into the dropout layer (also depicted in Figure 32) and further passed to the attention layer.

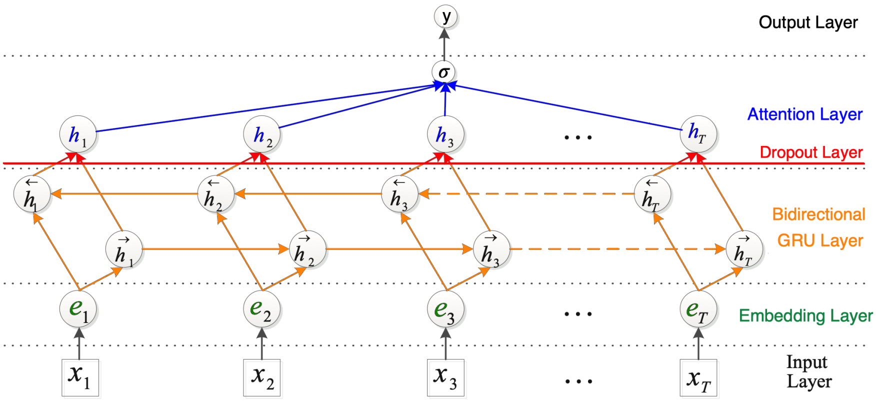


Figure 32: Architecture of the proposed DL model for EPU classification (based on illustration provided by P. Zhou et al. (2016))

Equations 34-37 showcase the internal functionality of a GRU layer. As opposed to LSTM, GRU does not have a component called *cell state* and uses the *hidden state* to transfer information (Cho et al., 2014). It also has only two gates: a *reset gate* and *update gate*. The reset gate is used to decide how much past information to forget, while the update gate is used to decide which information will be discarded or added. Equation 34 defines the reset gate, Equation 35 - the update gate, and Equations 36 and 37 describe the transformations to obtain the hidden state. The single-layer GRU computes the hidden state h^t for word x^t with W and U representing weight matrices and b bias vectors of corresponding elements of the GRU cell, $\odot$ denotes the element-wise multiplication of two vectors:

$$r^t = \sigma(W_r x^t + U_r h^{t-1} + b_r) \tag{34}$$

$$z^t = \sigma(W_z x^t + U_z h^{t-1} + b_z) \tag{35}$$

$$\tilde{h^t} = tanh(W_h x^t + U_h(r^t \odot h^{t-1}) + b_h) \tag{36}$$

$$h^t = z^t \odot h^{t-1} + (1 - z^t) \odot \tilde{h^t} \tag{37}$$

As we are using a bidirectional GRU, the network will contain two sub-networks for the left and right sequence context, which develop forward and backward, respectively. The output of the t word is thus represented by the element-wise sum that combines the forward and backward pass outputs:

$$h^t = \overrightarrow{h_t} \odot \overleftarrow{h_t} \tag{38}$$

The hidden states from the bidirectional GRU layer will be further passed on to the dropout and attention layers. The output of the attention layer s is supplied into the output layer with a sigmoid activation, that produces the probability of the article a, containing T words, to be containing EPU:

$$y_a = \sigma(W_o s + b_o) \tag{39}$$

The binary cross-entropy loss is used for end-to-end training:

$$\mathcal{L}(y_a, \hat{y_a}) = -\frac{1}{T}\sum_{i=1}^{T}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \; = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log(p_{ij}) \tag{40}$$

5.4.5 Regularization measures: dropout layer

Like all complex systems, neural networks are vulnerable to overfitting (G. E. Hinton et al., 2012). To make sure that the model learns to generalize from the training set without picking up the noise, our GPT-2 DNN includes a dropout layer after the bidirectional GRU layer. The concept of dropout comprises removal at random of hidden layer neurons and their corresponding connection weights during training. The probability of a hidden neuron being dropped out follows a Bernoulli distribution with a given dropout rate, in our case, a 50% chance.

5.4.6 Attention layer

Another important component of the proposed model from Figure 32 is the attention layer (Vaswani et al., 2017). As pointed out by P. Zhou et al. (2016), attention has been successfully adopted for several NLP-related tasks, like reading comprehension, abstractive summarization, textual entailment, and learning task-independent sentence representations. An attention function is mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are words vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

We are using a weighted average attention mechanism as applied by Chorowski et al. (2014) and P. Zhou et al. (2016), which produces a weight vector and merges word-level features from each time step into a sentence-level feature vector, by multiplying the weight vector. The calculation is depicted in Equations 41 and 42: let H be a matrix consisting of output vectors $[h_1, h_2, ..., h_T]$ from the bidirectional GRU layer and w - a trained parameter vector after Dropout was applied. T remains a sentence length of 407 words. The weighted sum of these output vectors forms the representation s of the sentence:

$$\alpha = softmax(w^T H) \tag{41}$$

$$s = H\alpha^T \tag{42}$$

An additional value of the attention layer stems from its interpretation features that will be explored further.

5.5 Results

In this section, we first present the best performing model for the prediction of EPU articles. All data-processing and modeling computations are performed in `python` with the use of packages like `numpy`, `pandas`, `scikit-learn`, `nltk`, `gensim`, for DL implementation the high-level neural network library `keras` is used as well as the `transformers` package by HuggingFace.

5.5.1 Classification analysis

According to the experimental design, we have developed a DL-NLP-model that allows the accurate classification of articles according to the previously discussed labeling. The test set represents 30% of the data and contains approximately 2% of EPU cases, matching the label balance of the train set. For evaluation, we have selected the AUC (Area under the Curve) and the F1-score. The former reflects how much a model is capable of distinguishing between classes regardless of the threshold. and is robust toward class imbalance. The latter allows evaluating the accuracy of the predictor by considering both precision (number of correct positive results divided by the number of all positive results) and recall (or sensitivity, correct positive results divided by the number of all relevant samples) of the test set. The F1-score represents a harmonic mean and measures how precise and how robust the models classify EPU cases:

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{43}$$

We considered a variety of different neural network architectures for the model training process. The neural networks with bidirectional and attention layers provided the best performance in the selected metrics. Apart from GPT-2, we have tried two other widely-used pre-trained embeddings: Google News Embeddings and GloVe. The results were inferior to GPT-2 and will not be discussed further.

Table 15: Evaluation of classifier models on the randomized out-of-sample test set.

Models	AUC	F1-score
LR	0.9116	0.1550
SVM	0.8966	0.2083
RF	0.9063	0.0356
XGB	0.9054	0.0171
GPT-2 DNN	**0.9606**	**0.6500**

Training of the benchmarks was performed with `scikit-learn` package and the following hyperparameters: LR - L2 regularization and the "lbfgs" solver; SVM - regularization parameter C =1.0, "rbf" kernel, gamma=0.0024; RF - number of estimators=100, maximum depth=None, minimum samples split=2, maximum features=20; XGB - learning rate=0.1, number of estimators=100, maximum depth=3.

Table 15 shows the results of the GPT-2 DNN model and selected benchmarks: TF-IDF vector-based logistic regression (LR), SVM, Random Forest (RF) and XGBoost (XGB). GPT-2 DNN outperforms other models with the highest AUC of 0.96, but its improvement for the F1-score is even more substantial, reaching 0.65 as compared to other models. Tree-based models seem to be particularly weak with the precision and recall. LR and SVM with non-linear kernel capture the case of interest more accurately. Given that the classifiers were trained on the dataset without the original keywords, and considering the strong performance of GPT-2 DNN, Table 15 allows us to conclude that RQ 1 was answered positively: we have successfully constructed a DL model that can capture the concept of EPU using text mining. However, to examine the robustness of the proposed solution, we performed 10-fold stratified cross-validation. Table 16 shows that GPT-2 DNN keeps up the excelling performance with an AUC standard deviation of 0.014 and an F1-score standard deviation of 0.04. However, the heterogeneity of input is visible through the folds, regardless of the randomized splitting.

Table 16: Results of the 10-fold cross-validation (with stratification of samples) of the GPT-2 DNN.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	**Mean**
AUC	0,9507	0.9725	0.9553	0,9587	0,9316	0,9760	0,9606	0,9557	0,9722	0,9829	**0,9616**
F1-score	0,7281	0,7068	0,6500	0,6294	0,5787	0,6613	0,6571	0,6567	0,6584	0,6698	**0,6590**

To further examine the potential presence of narrative shifts (topics, used vocabulary) in EPU articles over time, we have looked into the decision-making mechanism of GPT-2 DNN, in particular, its attention layer. The next section illustrates the analysis of the changing semantics in the newspaper articles over time.

5.5.2 Evolution of uncertainty rhetoric

RQ 2 concerned the potential shortcomings of a static keyword approach. Our goal was to analyze if there is a change in the words that entail EPU. We have sub-sampled all the EPU articles by year (on average, 235 articles per year), dropped the three groups of EPU keywords, and used them as a test set for the trained classifier. We extracted the weights assigned by the attention layer to the word inputs after the model was trained i.e., in the inference phase. The top ten words with the highest values were selected and assigned points from ten to 1. Points accumulated during the year constituted a ranking of every word by its "uncertainty impact". The top ten words for every year are showcased in Table 17. One can observe an evident change of the newspaper agenda and the introduction of new "uncertainty drivers" through time. The years 2008-2009 are focused on economic "crisis" and "recession", followed by concerns on "fiscal" policies and changes in "legislation". The pre-election years see the rise of national agenda with "american" and "america" leading and "trump" first appearing in 2015 and firmly dominating the ranks from 2016 to 2019. In 2016 "brexit" enters the ranks, followed by "tariffs" and "immigration" in 2018. Change of "uncertainty drivers" in time indicates a strong interpretation capacity of the DNN classifier and demonstrates its ability to adapt to the new topics with time. Further, the presented evidence raises concern if a set of fixed keywords is enough to capture uncertainty during different periods like the financial crisis in 2009, the trade-war in 2018, or the COVID-19 pandemic in 2020.

Table 17: Top 10 words associated with uncertainty with corresponding rank, as evaluated by the attention layer of the proposed classifier.

2006		**2007**		**2008**		**2009**		**2010**	
subject	120	federal	134	presidential	200	presidents	88	presidential	110
federal	112	subject	84	economic	72	presidential	85	subject	90
government	69	newspaper	72	subject	69	subject	79	federal	78
economics	65	economics	61	crisis	54	stimulus	68	republican	69
newspaper	62	republican	48	republican	54	recession	68	newspaper	68
economic	45	budgets	45	federal	53	economic	62	recession	57
republican	41	presidential	44	budgets	45	crisis	60	economic	55
presidents	37	economic	40	newspaper	44	bailouts	56	presidents	54
english	32	government	38	presidents	42	newspaper	52	legislation	52
legislation	32	presidents	36	bailouts	42	federal	52	unemployment	50
2011		**2012**		**2013**		**2014**		**2015**	
presidential	118	presidential	196	presidential	93	american	73	rates	98
subject	84	subject	101	subject	84	america	48	american	71
newspaper	74	cliff	79	federal	63	republican	38	trump	48
debt	72	republican	74	newspaper	60	americans	38	rate	42
republican	70	presidents	52	ceiling	58	legislation	36	americans	42
recession	57	recession	50	republican	54	economic	33	america	40
federal	55	economics	46	recession	50	rates	32	federal	32
presidents	48	fiscal	46	cliff	49	federal	31	democrats	25
ceiling	46	economic	43	debt	48	newspaper	25	april	24
economics	46	federal	41	economics	43	subject	24	republican	22
2016		**2017**		**2018**		**2019**			
trump	343	trump	679	trump	774	trump	389		
brexit	153	rates	45	brexit	46	brexit	149		
americans	57	americans	44	tariffs	43	american	48		
american	49	american	37	american	38	america	44		
rates	49	america	26	april	22	tariffs	36		
america	41	legislation	23	california	19	rates	36		
rate	31	republican	21	rates	15	americans	23		
april	23	ms	14	americans	15	rate	19		
democrats	22	english	14	republican	14	republican	18		
republican	15	federal	12	immigration	13	true	15		

5.5.3 Adaptability analysis

In this section, we compare our reconstructed index to the proxies established in the literature and identify meaningful benchmarks. Our goal is to assess the ability of our index to explain variation in real macroeconomic variables.

There are several approaches to building an uncertainty proxy. They can be assigned into four different categories: proxies based on the number of search requests or newspaper articles during a certain period (Baker et al., 2016), proxies based on variation in a large group of macro variables (Jurado et al., 2015), proxies based on disagreement in expectations among survey participants (Ozturk & Sheng, 2018), and proxies based on the volatility of economic variables (Bloom, 2009).

BBD represents the first category. The survey-based uncertainty index relies on data from the consensus survey, an aggregator that collects surveys of economic forecasters from many different sources. The forecast error of the different survey participants can be interpreted as a proxy for uncertainty in the economy. The macro-based uncertainty index consists of a large collection of macroeconomic and financial data. The volatility-based index is the VIX from the US stock exchange. Table 18 showcases the different indices and the corresponding labels:

Table 18: Uncertainty proxies.

Name	Label	Source
BBD method on our data	BBD	own data
GPT-2 DNN	GPT-2	own construction
Total Uncertainty	S	Ozturk and Sheng (2018)
Real Uncertainty (h=1)	M	Jurado et al. (2015)
Stock market volatility	V	Bloom (2009)

Co-movement between uncertainty proxies

We start our economic analysis by looking at descriptive statistics of the uncertainty proxies in Table 19. Naturally, they need to be standardized for visual comparisons due to the varying value ranges. The kurtosis of all proxies except GPT-2 exceeds the value of the normal distribution, meaning that four out of five proxies show considerably high peaks. Stock market volatility and macro-based uncertainty include the highest peaks. Additionally, all proxies except GPT-2 are right-skewed, providing further evidence of relatively high values included in most proxies. GPT-2 is the closest to a normal distribution.

Table 19: Summary Statistics from Jan 2006 until Sept 2017.

Variable	Mean	Std. Dev	Min	Max	Skew	Kurt
BBD	94.95	47.51	17.56	248.77	1.29	4.57
GPT-2	92.56	35.40	29.35	175.38	0.14	1.98
S	0.43	0.31	0.11	1.23	1.30	3.59
M	0.64	0.05	0.58	0.83	2.08	7.13
V	19.44	9.18	10.26	62.64	2.33	9.66

We further look at correlations between the different proxies to identify potential groups. Figure 33 showcases the correlations among the different proxies, clustered by proximity:

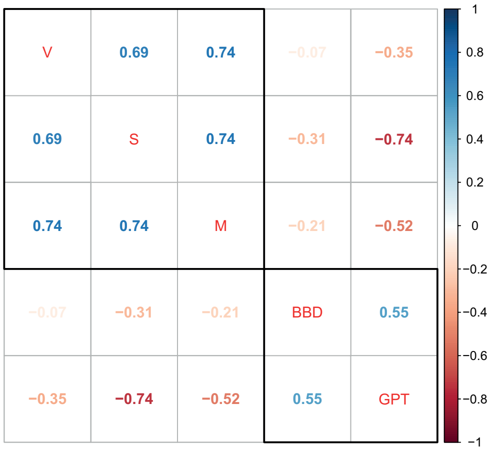


Figure 33: Pearson Correlation between the different uncertainty proxies from Jan 2006 until Sept 2017.

We can see two distinct clusters formed by the newspaper-based proxies and all other proxies. The similarity among the other group of proxies seems to be higher than the similarity between the two economic policy uncertainty proxies. The EPU indices seem to be negatively correlated with the other group of uncertainty proxies. A negative correlation is counter-intuitive because it implies that if uncertainty measured by one group increases, the other group will decrease. To better understand this finding, we plot the different time series.

Figure 34 shows the five different proxies in one time series plot. We can see two major patterns: the group of indices that is not based on newspaper data has its peak around 2009 during the GFC and otherwise does not have any prominent peaks, while the newspaper-based indices have several.

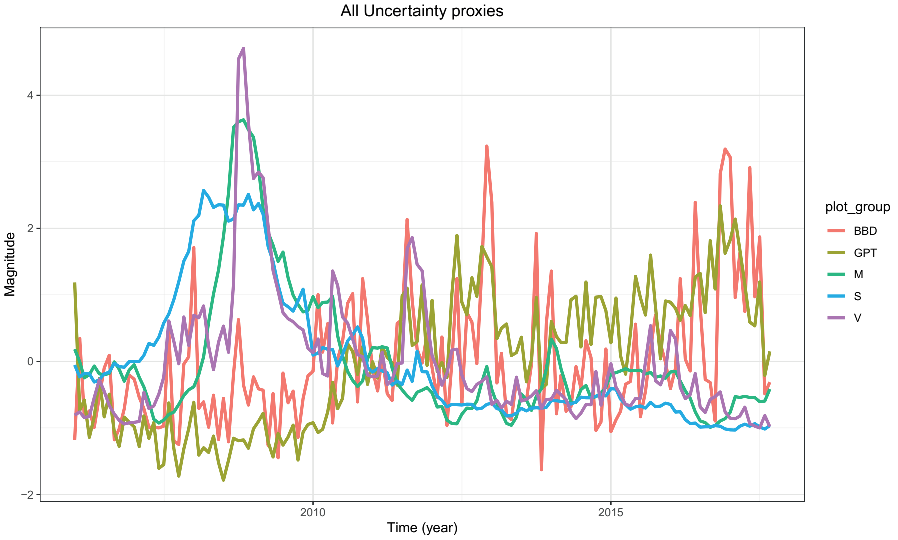


Figure 34: Time series plot of all uncertainty proxies from Jan 2006 until Sep 2017.

BBD and GPT-2 in Figure 34 have a relatively high variance. They also move up during all the times one would expect uncertainty to increase. Both indices are quite similar. The BBD-based index behaves slightly differently during the GFC from 2009 to 2012; otherwise, our GPT-2 index and the BBD index move up during all major events related to high uncertainty.

Survey uncertainty is especially visible during the GFC in 2009. Towards the end of the sample period, the uncertainty indicated by this proxy seems to fade out. For macro uncertainty, we obtain a similar picture: it reaches its maximum during the GFC and shows comparatively little movement later on. Stock market volatility exhibits more variation than a survey- or macro-uncertainty but also peaks during 2009. It remains relatively smooth with a small peak during the European sovereign debt crisis in 2014.

To sum up, the uncertainty proxies based on macro-data, surveys, and volatility show similar behavior, potentially stemming from relying on all individual information in the economy that is available to individual agents before an uncertainty shock hits. BBD and GPT-2 show very different behavior from the other group of proxies. They have the highest variation among all uncertainty proxies and also the largest number of peaks. Instead of relying on an individual information, they are based on newspapers that already contain aggregated information.

The higher movement of the newspaper-based indices might indicate that these indices capture fast-moving uncertainty in the economy better than the other proxies, that mainly move during a small number of massive shocks.

Interaction with real economic variables

We investigate if GPT-2 is better than BBD at predicting the movements of the economy and capturing the change in newspaper vocabulary. We measure how BBD and GPT-2 correlate with different real economic variables within the second half of our sample. In general, the latter should be negatively correlated with the uncertainty proxies, so that when uncertainty increases, the affected variables decrease. Based on theoretical literature in economics (Arellano et al., 2019; Bernanke, 1983), uncertainty affects the variables in Table 20. We remove time trends from the stock market, employment, and industrial production using an HP-Filter (Ravn & Uhlig, 2002) and also take the natural logarithm of every variable except for the 10-year government bond yields, which include negative values.

Table 20: Macroeconomic and Financial variables.

Name	Label	Specification
S&P 500	SP	log hp
Employment (manufacturing)	EM	log hp
Industrial production	IP	log hp
Federal funds rate	FF	log
10-year government bond yield	BY	level

We explore the relationship between the time series with Pearson correlation and mean directional accuracy (MDA). We want to investigate how our GPT-2 uncertainty index behaves through time compared to BBD. Therefore, we decided to divide our sample into three different periods defined in Table 21. Period 2 and Period 3 are defined by the shifts in "uncertainty drivers" of EPU articles, discussed in Subsection 5.5.2, and can be defined as "Pre Trump" and "Post Trump".

Table 21: Sample periods for empirical investigations.

Time period	Sample	Abbreviation
Jan 2006 - Apr 2019	Full Sample	P1
Jan 2006 - Dec 2014	Pre Trump	P2
Jan 2015 - Apr 2019	Post Trump	P3

Since we do not know which lag yields the most substantial relationship between uncertainty and economic variables, we explore this dynamic by showing the contemporaneous correlation, the correlation with economic variables one year ahead, as well as the highest absolute correlation with economic variables in the range from t_0 up to t_{0+k}, where $k = 12$.

An uncertainty proxy that behaves according to economic theory should lead to a decrease in economic activity, entailing a negative correlation between uncertainty at t_0 and economic variables in the future at t_{0+k}.

We investigate MDA by comparing the predictive power of the uncertainty proxies for different lags. All statistics for Period 1 can be found in Table 22, for Periods 2 and 3 - in Table 23.

Table 22: Interaction with real and financial variables in Period 1.

Name	Corr(0)	Corr (12)	maxCorr	MDA
Stock market				
BBD	-0.13	-0.02	-0.13 (0)	0.50 (12)
GPT-2	-0.02	0.23	0.23 (12)	0.55 (1)
Employment				
BBD	-0.09	0.02	-0.09 (0)	0.52 (9)
GPT-2	-0.02	0.24	0.24 (12)	0.63 (10)
Industrial production				
BBD	-0.06	0.05	-0.07 (0)	0.56 (2)
GPT-2	0.05	0.23	0.23 (11)	0.60 (3)
Federal funds rate				
BBD	-0.08	-0.03	-0.08 (0)	0.55 (12)
GPT-2	-0.26	-0.03	-0.26 (0)	0.59 (12)
10-year government bond yield				
BBD	-0.01	-0.16	-0.16 (12)	0.52 (5)
GPT-2	-0.12	-0.42	-0.42 (12)	0.53 (1)

This table shows correlation between EPU and economic variables based on the following equation: $corr(Uncertainty_t, EconomicVariable_{t+k}$, where k ranges from 0 to 12, as well as MDA. The numbers in brackets indicate lag numbers.

Our index shows the desired feature of a negative correlation for four out of five variables.

However, for the stock market and employment, the negative correlation is weak. Additionally, GPT-2 shows a stronger correlation with economic variables than BBD for the federal funds rate and the government bond yield. BBD shows a stronger correlation for the other three variables.

We also investigate the timing of the relationship between uncertainty and economic activity. Column 2 displays the correlation between economic variables at twelve months in the future and current EPU, while Column 3 shows the period ahead with the strongest correlation between the two. For the first three variables, there is virtually no negative correlation between EPU and economic activity twelve months ahead. For the federal funds rate, we observe a weak negative correlation, and only the government bond yield shows a noticeable negative correlation with uncertainty twelve months before.

Regarding the strongest correlation in column three for the first three variables, the two indices show the opposite behavior. While for BBD lag zero has the strongest correlation, and this correlation is negative, for GPT-2 lag twelve has the strongest correlation and it is positive. Only for the last two variables, both indices show the strongest correlation for the same lag, and in both cases, the strongest correlation is negative.

MDA further shows which time lag yields more accuracy for the prediction of economic activity. Employment and the federal funds rate show the best performance for rather long lags. Industrial production and the government bond yield show the best performance for shorter lags. Only for the stock market, the best lag length differs.

In Table 23 we explore the performance during the sub-samples period 2 and 3.

Table 23: Interaction with real and financial variables in Period 2 and 3.

Name	Corr(0)	Corr (12)	maxCorr	MDA	Corr(0)	Corr (12)	maxCorr	MDA
Stock market P2					Stock market P3			
BBD	-0.11	-0.02	-0.12 (1)	0.56 (12)	-0.32	0.28	0.48 (10)	0.50 (2)
GPT-2	0.02	0.32	0.32 (12)	0.51 (1)	-0.29	0.26	-0.29 (0)	0.63 (1)
Employment P2					Employment P3			
BBD	-0.09	0.03	-0.09 (0)	0.61 (9)	-0.46	-0.11	-0.57 (5)	0.51 (3)
GPT-2	-0.05	0.29	0.29 (12)	0.66 (10)	-0.19	-0.20	-0.29 (9)	0.60 (10)
Industrial production P2					Industrial production P3			
BBD	0.03	0.05	0.07 (11)	0.62 (1)	-0.27	0.43	0.43 (12)	0.55 (3)
GPT-2	0.14	0.40	0.40 (11)	0.60 (3)	0.09	0.17	0.18 (11)	0.56 (9)
Federal funds rate P2					Federal funds rate P3			
BBD	-0.31	-0.35	-0.35 (12)	0.53 (12)	0.34	0.48	0.48 (9)	0.63 (12)
GPT-2	-0.60	-0.37	-0.60 (0)	0.56 (12)	0.50	0.30	0.50 (0)	0.70 (12)
10-year government bond yield P2					10-year government bond yield P3			
BBD	0.24	0.12	0.24 (0)	0.51 (4)	-0.17	-0.39	-0.39 (12)	0.64 (7)
GPT-2	0.25	-0.09	0.25 (0)	0.50 (1)	-0.48	-0.19	-0.48 (0)	0.59 (1)

This table shows correlation between EPU and economic variables based on the following equation: $corr(Uncertainty_t, EconomicVariable_{t+k}$, where k ranges from 0 to 12, as well as MDA. The numbers in brackets indicate lag numbers.

Period 2 shows the same pattern for all variables but the government bond yield. BBD still has a negative correlation with the first two variables, while GPT-2 has positive. GPT-2 shows a stronger negative correlation with the federal funds rate. However, for the government bond yield, nearly all correlations are now positive, and the lag with the most defined relationship changed from lag twelve to lag zero. The lag pattern for MDA stayed the same.

Results for Period 3 change significantly. While industrial production and federal funds rate show mostly positive correlation with EPU, the stock market, employment, and the government bond yield exhibit noticeable negative correlation with uncertainty. The change is greater for GPT-2, where the sign switched from positive to negative correlation.

Generally, it seems that the connection between uncertainty and economic activity differs substantially between the two sub-samples. While BBD shows performance more in line with economic theory during Period 2, GPT-2 catches up and sometimes outperforms BBD in Period 3. This observation reinforces the argument that we need an uncertainty proxy that can deal with the changing vocabulary in newspapers over time, yielding results that are in line with the economic theory even when there are structural breaks. Our DL-NLP algorithm is the first step in that direction.

Forecasting performance In this Section, we evaluate the potential of our newly created index for forecasting, following the practice of Claveria et al. (2007), D'Amuri and Marcucci (2017), Tarassow (2019). We are forecasting five different variables with an ARIMAX model, where either our GPT-2 index or the BBD index is added as an exogenous variable for forecasting. The forecasting is done for rolling windows of 18, 12, and 6 months during the three different periods explained in the previous subsection and performed with the *auto.arima()* function from the forecast package in R by Hyndman and Khandakar (2007).

For each period and variable, we obtain one forecast where the model is augmented by the BBD labels and one forecast where the model is augmented by the predictions of the GPT-2 DNN model. Tables 24 and 25 exhibit the RMSE for all forecasts. Additionally, we perform Diebold-Mariano tests to identify superior forecasting performance among the ARIMAX models.

For the whole sample period, the model, including the BBD index, generally seems to result in lower RMSE for six months and the government bond yield. This difference in RMSE is only statistically significant for the forecasts of the stock market. GPT-2 provides lower RMSE only when forecasting the federal funds rate with a rolling window of six months. Otherwise, the forecasts are very similar.

Since our goal is to investigate whether our index can deal better with the change in the vocabulary used by the newspapers, we carry out the same forecasting exercise for the previously defined Periods 2 and 3. The results can be found in Table 25.

Periods 1 and 2 exhibit a similar pattern. BBD provides forecasts of higher accuracy for the stock market at a rolling window of 6 months, as well as the forecasts of government bond yields at a window of 18 months. GPT-2 only yields lower RMSE for the federal funds rate at a

Table 24: ARIMAX forecasts for Period 1.

Model	S&P 500	Fed funds	Empl.	Ind. prod.	Bonds
Window of 18 months					
BBD	0.048	0.026	0.002	0.009	0.171*
GPT-2	0.047*	0.026	0.002	0.009	0.178
Window of 12 months					
BBD	0.050	0.029	0.002	0.008	0.182
GPT-2	0.050	0.030	0.002	0.009	0.201
Window of 6 months					
BBD	0 .051**	0.039	0.006	0.010	0.242
GPT-2	0.068	0.036	0.007	0.012	0.250

This table shows RMSE for rolling window ARIMAX models with BBD and GPT-2 as external regressors. Stars indicate significance levels of Diebold-Mariano Tests for higher forecast accuracy: ** = 0.05; * = 0.1.

Table 25: ARIMAX forecasts for Period 2 and 3.

Model	S&P 500	Fed funds	Empl.	Ind. prod.	Bonds	S&P 500	Fed funds	Empl.	Ind. prod.	Bonds
Window of 18 months P2						Window of 18 months P3				
BBD	0.052	0.028	0.002	0.001	0.200*	0.856	0.240	0.009	0.141	1.559*
GPT-2	0.052	0.028	0.002	0.001	0.204	0.856	0.239	0.009	0.141	1.577
Window of 12 months P2						Window of 12 months P3				
BBD	0.056	0.032*	0.002	0.008	0.212	0.796	0.228	0.096	0.129	1.487
GPT-2	0.055	0.034	0.002	0.010	0.235	0.795	0.230	0.096	0.129	1.438*
Window of 6 months P2						Window of 6 months P3				
BBD	0.055**	0.043	0.007	0.012	0.277	0.745	0.221	0.102	0.116	1.443
GPT-2	0.079	0.038	0.009	0.014	0.285	0.735	0.211**	0.101	0.113*	1.433

This table shows RMSE for rolling window ARIMAX models with BBD and GPT-2 as external regressors. Stars indicate significance levels of Diebold-Mariano Tests for higher forecast accuracy: ** = 0.05; * = 0.1.

window of six months. For all other variables and window sizes, there is no statistical difference in RMSE between the two models.

For the Period 3, RMSE is generally much higher. BBD does not yield lower RMSE anymore. Instead, GPT-2 shows lower RMSE for all variables for a window of six months. The difference in RMSE is statistically significant for forecasts of the federal funds rate and industrial production. For the longer windows, both models show similar forecast accuracy.

To sum up, for P1 and P2, BBD generally yields forecasts with lower RMSE, even though there is rarely a statistically significant difference between the two models. In Period 3, when a change of newspaper agenda occurred, our model provides lower RMSE and more accurate forecasts for two out of five variables. This serves as evidence that a DL-NLP-based index can better deal with changing newspaper agendas over time.

5.6 Conclusion

Following the importance of risk assessment and agents' expectations in economic development, we offered a novel DL-NLP-based method for the quantification of economic policy uncertainty. The method is applied to the corpus of articles from ten major USA newspapers from 01 Jan 2006 to 30 Apr 2019. The predictive performance of the model surpassed the available benchmarks with an AUC of 0.96 and an F1-score of 0.65. The model remained robust in 10-fold cross-validation.

Our method offers high interpretability and adaptability, which was demonstrated by the analysis of the top ten words responsible for EPU over time. We exposed a definite change of agenda in the newspaper articles. The first part of the sample, from Jan 2006 until Dec 2014, did not feature the word "trump". Starting in Jan 2015 until the end of our sample in Apr 2019, the word "trump" always featured in the top ten. These shifts show the necessity to adapt to changing political and economic trends when trying to capture economic uncertainty from newspaper articles.

The necessity to take into account changing newspaper rhetoric was further illustrated by investigating the correlations between our uncertainty proxies and economic activity for the two different periods. We showed that the co-movement between EPU and economic variables switches from positive to negative from Period 2 to Period 3. With our forecasting experiment, we showed that during the later period, forecasting accuracy reduced drastically. Our uncertainty index based on DL-NLP had superior forecasting ability for two out of five variables and resulted in lower RMSE for all variables. In the earlier period, none of the two models provided higher accuracy for four out of five variables. This way, the proposed method proved its fitness to deal with the change in newspaper agenda better than the methodology of (Baker et al., 2016).

Our approach shows pathways towards capturing economic policy uncertainty over long periods while keeping track of changes in the way that news and uncertainty are reported. Two recent examples that changed newspaper reporting are the Trump presidency and the recent COVID-19 pandemic. The approach might prove especially useful for governments and institutions in countries with scarce, timely information sources on the level of uncertainty in the economy as newspaper articles are widely available over time and therefore represent a feasible alternative data source to assess economic policy uncertainty.

References

Abbasi, A., Albrecht, C., Vance, A., & Hansen, J. (2012). Metafraud: A meta-learning framework for detecting financial fraud. *MIS Quarterly: Management Information Systems*, *36*(4), 1293–1327. https://doi.org/10.2307/41703508

ACFE. (2019). *Report to the Nations 2018 Global Study on Occupational Fraud and Abuse* (tech. rep.). https://doi.org/10.1002/9781118929773.oth1

Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning. *Risks*, *6*(2), 1–20.

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning - A Textbook.* Springer International Publishing.

Ahir, H., Bloom, N., & Furceri, D. (2018). The world uncertainty index. *Available at SSRN 3275033.*

Akaike, H. (1987). Factor analysis and AIC, In *Selected papers of hirotugu akaike.* Springer.

Albrecht, W. S., Albrecht, C., & Albrecht, C. C. (2008). Current trends in fraud and its detection. *Information Security Journal*, *17*(1), 2–12. https://doi.org/10.1080/19393550801934331

Alexopoulos, M., & Cohen, J. (2015). The power of print: Uncertainty shocks, markets, and the economy. *International Review of Economics and Finance*, *40*, 8–28.

Antoniou, A., & Holmes, P. (1995). Futures trading, information and spot price volatility: Evidence for the FTSE-100 stock index futures contract using garch. *Journal of Banking & Finance*, *19*(1), 117–129.

Ardia, D., Bluteau, K., & Boudt, K. (2019). Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values. *International Journal of Forecasting*, *35*(4), 1370–1386.

Arellano, C., Bai, Y., & Kehoe, P. J. (2019). Financial frictions and fluctuations in volatility. *Journal of Political Economy*, *127*(5), 2049–2103.

Athiwaratkun, B., & Stokes, J. W. (2017). Malware classification with lstm and gru language models and a character-level cnn, In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE.

Audrino, F., Sigrist, F., & Ballinari, D. (2020). The impact of sentiment and attention measures on stock market volatility. *International Journal of Forecasting*, *36*(2), 334–357.

Azqueta-Gavaldón, A. (2017). Developing news-based Economic Policy Uncertainty index with unsupervised machine learning. *Economics Letters*, *158*, 47–50.

Bachmann, R., Elstner, S., & Sims, E. R. (2013a). Uncertainty and Economic Activity : Evidence from Business Survey Data. *American Economic Journal: Macroeconomics*, *5*(2), 217–249.

Bachmann, R., Elstner, S., & Sims, E. R. (2013b). Uncertainty and economic activity: Evidence from business survey data. *American Economic Journal: Macroeconomics*, *5*(2), 217–49.

Baek, Y., & Kim, H. Y. (2018). Modaugnet: A new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module. *Expert Systems with Applications*, *113*, 457–480.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(7), 1001–1008.

Bai, B., Yen, J., & Yang, X. (2008). False financial statements: Characteristics of China's listed companies and cart detecting approach. *International Journal of Information Technology and Decision Making*, *7*(2), 339–359. https://doi.org/10.1142/S0219622008002958

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring Economic Policy Uncertainty. *The Quarterly Journal of Economics*, *131*(4), 1593–1636.

Bakshi, G., Cao, C., & Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of Finance*, *52*(5), 2003–2049.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, *12*(7).

Bates, D. S. (1991). The crash of 1987: Was it expected? the evidence from options markets. *The Journal of Finance*, *46*(3), 1009–1044.

Beneish, M. D. (1999). The Detection of Earnings Manipulation. *Financial Analysts Journal*, *55*(5), 24–36. https://doi.org/10.2469/faj.v55.n5.2296

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1–127.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures, In *Neural networks: Tricks of the trade*. Springer.

Bengio, Y., & Delalleau, O. (2009). Justifying and generalizing contrastive divergence. *Neural Computation*, *21*(6), 1601–1621.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), 1137–1155.

Bengio, Y., Goodfellow, I., & Courville, A. (2016). *Deep learning*. MIT Press.

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., Et al. (2007). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, *19*, 153.

Benos, A. V. (1998). Aggressiveness and survival of overconfident traders. *Journal of Financial Markets*, *1*(3), 353–383.

Beque, A., Coussement, K., Gayler, R., & Lessmann, S. (2017). Approaches for credit scorecard calibration: An empirical analysis. *Knowledge-Based Systems*, *134*(15), 213–227.

Bernanke, B. S. (1983). Irreversibility, uncertainty, and cyclical investment. *The Quarterly Journal of Economics*, *98*(1), 85–106.

Biktimirov, E. N., & Wang, C. (2017). Model-based versus model-free implied volatility: Evidence from North American, European, and Asian index option markets. *The Journal of Derivatives*, *24*(3), 42–68.

Black, F., & Scholes, M. (1976). Taxes and the pricing of options. *The Journal of Finance*, *31*(2), 319–332.

Blitz, D., Huij, J., Lansdorp, S., & Verbeek, M. (2013). Short-term residual reversal. *Journal of Financial Markets*, *16*(3), 477–504.

Bloom, N. (2009). The Impact of Uncertainty Shocks. *Econometrica*, *77*(3), 623–685.

Bobasu, A., Geis, A., Quaglietti, L., & Ricci, M. (2020). Tracking global economic uncertainty: Implications for global investment and trade. *Economic Bulletin Boxes*, (1).

Bodnaruk, A., Loughran, T., & McDonald, B. (2015). Using 10-K text to gauge financial constraints. *Journal of Financial and Quantitative Analysis*, *50*(4), 623–646. https://doi.org/10.1017/S0022109015000411

Bohn, H. (2001). Social security and demographic uncertainty: The risk-sharing properties of alternative policies, In *Risk aspects of investment-based social security reform*. University of Chicago Press.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327.

Bouri, E., Gupta, R., & Roubaud, D. (2018). Herding behaviour in cryptocurrencies. *Finance Research Letters*.

Brady, C., & Ramyar, R. (2006). White paper on spread betting. *Lond. Cass Bus. Sch.*

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Britten-Jones, M., & Neuberger, A. (2000). Option prices, implied price processes, and stochastic volatility. *The Journal of Finance*, *55*(2), 839–866.

Brown, S. V., & Tucker, J. W. (2011). Large-Sample Evidence on Firms' Year-over-Year MD&A Modifications. *Journal of Accounting Research*, *49*(2), 309–346. https://doi.org/10.1111/j.1475-679X.2010.00396.x

Busch, T., Christensen, B. J., & Nielsen, M. Ø. (2011). The role of implied volatility in forecasting future realized volatility and jumps in foreign exchange, stock, and bond markets. *Journal of Econometrics*, *160*(1), 48–57.

Business Insider, 2016. (n.d.). *"bitcoin is still storming higher"* [[Online; accessed 28-Dec-2016]].

Caggiano, G., Castelnuovo, E., & Groshenny, N. (2014). Uncertainty shocks and unemployment dynamics in us recessions. *Journal of Monetary Economics*, *67*, 78–92.

Cassidy, J. (2020). In an Environment of Chronic Economic Uncertainty, the White House Is Only Making It Worse [[Online; accessed 01-June-2020]].

Cboe. (2009). The cboe volatility index-vix. *White Paper*, 1–23.

Cecchini, M., Aytug, H., Koehler, G. J., & Pathak, P. (2010). Making words work: Using financial text as a predictor of financial events. *Decision Support Systems*, *50*(1), 164–175. https://doi.org/10.1016/j.dss.2010.07.012

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, *112*, 353–371.

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of china stock market, In *2015 ieee international conference on big data (big data)*. IEEE.

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2016). A multi-kernel support tensor machine for classification with multitype multiway data and an application to cross-selling recommendations. *European Journal of Operational Research*, *255*(1), 110–120.

Chimienti, M. T., Kochanska, U., & Pinna, A. (2019). Understanding the crypto-asset phenomenon, its risks and measurement issues. *Economic Bulletin Articles*, *5*.

Chiriac, R., & Voev, V. (2011). Modelling and forecasting multivariate realized volatility. *Journal of Applied Econometrics*, *26*(6), 922–947.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chorowski, J., Bahdanau, D., Cho, K., & Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*.

Christiano, L. J., Motto, R., & Rostagno, M. (2014). Risk shocks. *American Economic Review*, *104*(1), 27–65.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.*

Claveria, O. (2019). Forecasting the unemployment rate using the degree of agreement in consumer unemployment expectations. *Journal for Labour Market Research*, *53*, 1–10.

Claveria, O., Pons, E., & Ramos, R. (2007). Business and consumer expectations and macroeconomic forecasts. *International Journal of Forecasting*, *23*(1), 47–69.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, *74*(368), 829–836. https://doi.org/10.1080/01621459.1979.10481038

CoinMarketCap. (2018). Charts [[Online; accessed 01-April-2018]].

Cointelegraph. (2019). Crypto asset manager ledgerx launches bitcoin volatility index by zmudzinski, adrian [[Online; accessed 10-Feb-2019]].

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196.

Crawford, G., & Sen, B. (1996). *Derivatives for decision makers: Strategic management issues* (Vol. 1). John Wiley & Sons.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, *183*(3), 1447–1465.

da Gama Silva, P. V. J., Klotzle, M. C., Pinto, A. C. F., & Gomes, L. L. (2019). Herding behavior and contagion in the cryptocurrency market. *Journal of Behavioral and Experimental Finance*, *22*, 41–50.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning, In *Advances in neural information processing systems.*

D'Amuri, F., & Marcucci, J. (2017). The predictive power of google searches in forecasting us unemployment. *International Journal of Forecasting*, *33*(4), 801–816.

Davis, A. K., Piger, J. M., & Sedor, L. M. (2012). Beyond the Numbers: Measuring the Information Content of Earnings Press Release Language. *Contemporary Accounting Research*, *29*(3), 845–868. https://doi.org/10.1111/j.1911-3846.2011.01130.x

Dechow, P. M., Ge, W., Larson, C. R., & Sloan, R. G. (2011). Predicting Material Accounting Misstatements. *Contemporary Accounting Research*, *28*(1), 17–82. https://doi.org/10.1111/j.1911-3846.2010.01041.x

Demeterfi, K., Derman, E., Kamal, M., & Zou, J. (1999). A guide to volatility and variance swaps. *The Journal of Derivatives*, *6*(4), 9–32.

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197–387.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 653–664.

DePaulo, B. M., Rosenthal, R., Rosenkrantz, J., & Rieder Green, C. (1982). Actual and Perceived Cues to Deception: A Closer Look at Speech. *Basic and Applied Social Psychology*, *3*(4), 291–312. https://doi.org/10.1207/s15324834basp0304{_}6

Deribit. (2019). deribit: Bitcoin futures and options exchange.

Dixon, M., Klabjan, D., & Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, *6*(3-4), 67–77.

Doran, C. F. (1999). Why forecasts fail: The limits and potential of forecasting in international relations and economics. *International Studies Review*, *1*(2), 11–41.

Dowd, K. (2000). Adjusting for risk: An improved Sharpe ratio. *International Review of Economics & Finance*, *9*(3), 209–222.

du Jardin, P. (2016). A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research*, *254*(1), 236–252.

Duffie, D., Pan, J., & Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, *68*(6), 1343–1376.

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Proceedings of the 7th International Conference on Information and Knowledge Management*, 148–155. https://doi.org/10.1145/288627.288651

Dyck, A., Morse, A., & Zingales, L. (2010). Who blows the whistle on corporate fraud? *Journal of Finance*, *65*(6), 2213–2253. https://doi.org/10.1111/j.1540-6261.2010.01614.x

Ederington, L. H., & Lee, J. H. (1996). The creation and resolution of market uncertainty: The impact of information releases on implied volatility. *Journal of Financial and Quantitative Analysis*, *31*(4), 513–539.

Elendner, H., Trimborn, S., Ong, B., & Lee, T. M. (2018). The cross-section of crypto-currencies as financial assets: Investing in crypto-currencies beyond bitcoin. *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1*, 145–173.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, *11*, 625–660.

Feldman, R., Govindaraj, S., Livnat, J., & Segal, B. (2010). Management's tone change, post earnings announcement drift and accruals. *Review of Accounting Studies*, *15*(4), 915–953. https://doi.org/10.1007/s11142-009-9111-x

Fengler, M. R., Härdle, W. K., & Villa, C. (2003). The dynamics of implied volatilities: A common principal components approach. *Review of Derivatives Research*, *6*(3), 179–202. https://doi.org/10.1023/B:REDR.0000004823.77464.2d

Fischer, T., & Krauss, C. (2018a). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Fischer, T., & Krauss, C. (2018b). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Freel, M. S. (2005). Perceived environmental uncertainty and innovation in small firms. *Small Business Economics*, *25*(1), 49–64.

French, K. R., Schwert, G. W., & Stambaugh, R. F. (1987). Expected stock returns and volatility. *Journal of Financial Economics*, *19*(1), 3–29.

Gaganis, C. (2009). Classification techniques for the identification of falsified financial statements: a comparative analysis. *Intelligent Systems in Accounting, Finance & Management*, *16*(3), 207–229. https://doi.org/10.1002/isaf.303

Gee, J., & Button, M. (2019). *The Financial Cost of Fraud 2019* (tech. rep.). Crowe.

Geng, R., Bose, I., & Chen, X. (2015). Prediction of financial distress: An empirical study of listed chinese companies using data mining. *European Journal of Operational Research*, *241*(1), 236–247.

Ghirelli, C., Perez, J. J., & Urtasun, A. (2019). A new economic policy uncertainty index for spain. *Economics Letters*, (182), 64–67.

Giles, C. L., Lawrence, S., & Tsoi, A. C. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning*, *44*(1), 161–183.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning, In *2018 ieee 5th international conference on data science and advanced analytics (dsaa)*. IEEE.

Glancy, F. H., & Yadav, S. B. (2011). A computational model for financial reporting fraud detection. *Decision Support Systems*, *50*(3), 595–601. https://doi.org/10.1016/j.dss.2010.08.010

Glas, A. (2019). Five dimensions of the uncertainty-disagreement linkage. *International Journal of Forecasting*, *36*(2), 607–627.

Goel, S., & Gangolly, J. (2012). Beyond the numbers: Mining the annual reports for hidden cues indicative of financial statement fraud. *Intelligent Systems in Accounting, Finance and Management*, *19*(2), 75–89. https://doi.org/10.1002/isaf.1326

Goel, S., Gangolly, J., Faerman, S. R., & Uzuner, O. (2010). Can linguistic predictors detect fraudulent financial filings? *Journal of Emerging Technologies in Accounting*, *7*(1), 25–46. https://doi.org/10.2308/jeta.2010.7.1.25

Goel, S., & Uzuner, O. (2016). Do Sentiments Matter in Fraud Detection? Estimating Semantic Orientation of Annual Reports. *Intelligent Systems in Accounting, Finance and Management*, *23*(3), 215–239. https://doi.org/10.1002/isaf.1392

Gollier, C. (2018). *The economics of risk and uncertainty*. Edward Elgar Publishing Limited.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning: Adaptive computation and machine learning*. MIT press.

Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks, In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE.

Gray, G. L., & Debreceny, R. S. (2014). A taxonomy to guide research on the application of data mining to fraud detection in financial statement audits. *International Journal of Accounting Information Systems*, *15*(4), 357–380. https://doi.org/10.1016/j.accinf.2014.05.006

Hajek, P., & Henriques, R. (2017). Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods. *Knowledge-Based Systems*, *128*, 139–152. https://doi.org/10.1016/j.knosys.2017.05.001

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the roc curve. *Machine Learning*, *77*(1), 103–123.

Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a garch (1, 1)? *Journal of Applied Econometrics*, *20*(7), 873–889.

Härdle, W. K., Harvey, C. R., & Reule, R. C. (2020). Understanding cryptocurrencies. *Journal of Financial Econometrics*. https://doi.org/10.2139/ssrn.3360304

Härdle, W. K., & Trimborn, S. (2015). Crix or evaluating blockchain based currencies. *Mathematisches Forschungsinstitut Oberwolfach*, (Report No. 42/2015), 17–20. https://doi.org/10.4171/OWR/2015/42

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning* (2nd). New York, Springer.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2009). *The elements of statistical learning: data mining, inference and prediction* (2nd ed.). New York, Springer. https://doi.org/10.1007/BF02985802

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, *21*(9), 1263–1284.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios [asmb.2209]. *Applied Stochastic Models in Business and Industry*, *33*(1), 3–12.

Heaton, J., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. *arXiv preprint arXiv:1602.06561*.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, *6*(2), 327–343.

Hey, J. D. (1996). Uncertainty in economics, In *A guide to modern economics*. Routledge.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hinton, G., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527–1554.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Hochreiter, S., & Schmidhuber, J. (1997a). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hochreiter, S., & Schmidhuber, J. (1997b). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hou, A. J., Wang, W., Chen, C. Y. H., & Härdle, W. K. (2020). Pricing Cryptocurrency Options*. *Journal of Financial Econometrics*, *18*(2), https://academic.oup.com/jfec/article-pdf/18/2/250/33218360/nbaa006.pdf, 250–279. https://doi.org/10.1093/jjfinec/nbaa006

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Hsinchun, C., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, *36*(4), 1165–1188.

Huang, S. Y., Tsaih, R. H., & Yu, F. (2014). Topological pattern discovery and feature extraction for fraudulent financial reporting. *Expert Systems with Applications*, *41*(9), 4360–4372. https://doi.org/10.1016/j.eswa.2014.01.012

Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research*, *196*(2), 819–825.

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research*, *207*(3), 1702–1716.

Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, *278*(1), 330–342.

Humpherys, S. L., Moffitt, K. C., Burns, M. B., Burgoon, J. K., & Felix, W. F. (2011). Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems*, *50*(3), 585–594. https://doi.org/10.1016/j.dss.2010.08.009

Hyndman, R. J., & Khandakar, Y. (2007). *Automatic time series for forecasting: The forecast package for r*. Monash University, Department of Econometrics; Business Statistics ...

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of Finance*, *45*(3), 881–898.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *the 14th International Conference on Machine Learning (ICML '97)*, 143–151.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 27. https://doi.org/10.1186/s40537-019-0192-5

Jurado, K., Ludvigson, C. S., & Ng, S. (2015). Measuring Uncertainty. *American Economic Review*, *105*(3), 1177–1216.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, *100*, 234–245.

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models, In *Emnlp 2013 - 2013 conference on empirical methods in natural language processing, proceedings of the conference*, Association for Computational Linguistics (ACL).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences, In *52nd annual meeting of the association for computational linguistics, acl 2014 - proceedings of the conference*, Association for Computational Linguistics (ACL). https://doi.org/10.3115/v1/p14-1062

Karpoff, J. M., Koester, A., Lee, D. S., & Martin, G. S. (2014). Database Challenges in Financial Misconduct Research. *Working Paper*, 1–66.

Katsiampa, P. (2017). Volatility estimation for bitcoin: A comparison of garch models. *Economics Letters*, *158*, 3–6.

Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.-C., & Johnson, J. E. (2020). Can deep learning predict risky retail investors? a case study in financial risk behavior forecasting. *European Journal of Operational Research*, *283*(1), 217–234.

Kim, H. Y., & Won, C. H. (2018a). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, *103*, 25–37.

Kim, H. Y., & Won, C. H. (2018b). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, *103*, 25–37.

Kim, Y. J., Baik, B., & Cho, S. (2016). Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning. *Expert Systems with Applications*, *62*, 32–43. https://doi.org/10.1016/j.eswa.2016.06.016

Kimball, M. S. (1990). Precautionary Savings in the Small and in the Large. *Econometrica*, *58*(1), 53–73.

Klein, T., Thu, H. P., & Walther, T. (2018). Bitcoin is not the new gold–a comparison of volatility, correlation, and portfolio performance. *International Review of Financial Analysis*, *59*, 105–116.

Knight Frank, H. (1921). Risk, uncertainty and profit.

Kohler, H.-P., & Kohler, I. (2002). Fertility decline in russia in the early and mid 1990s: The role of economic uncertainty and labour market crises. *European Journal of Population/Revue européenne de Démographie*, *18*(3), 233–262.

Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid.*

Kränkel, M., & Lee, H.-E. L. (2019). Text Classification with Hierarchical Attention Networks. https://humboldt-wi.github.io/blog/research/information_systems_1819/group5_han/

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, *104*, 38–48.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689–702.

Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in neural information processing systems 4* (pp. 950–957). Morgan Kaufman.

Larcker, D. F., & Zakolyukina, A. A. (2012). Detecting Deceptive Discussions in Conference Calls. *Journal of Accounting Research*, *50*(2), 495–540. https://doi.org/10.1111/j.1475-679X.2012.00450.x

Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, *10*, 1–40.

Larsen, V. (2017). Components of uncertainty. *Norges Bank Working Paper Series.*

Lavington, F. (1912). Uncertainty in its relation to the net rate of interest. *The Economic Journal*, *22*(87), 398–409.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Leduc, S., & Liu, Z. (2016). Uncertainty shocks are aggregate demand shocks. *Journal of Monetary Economics*, *82*, 20–35.

Lee, J., Jang, D., & Park, S. (2017). Deep learning-based corporate performance prediction model considering technical capability. *Sustainability*, *9*(6), 899–911.

Lehmann, B. N. (1990). Fads, martingales, and market efficiency. *The Quarterly Journal of Economics*, *105*(1), 1–28.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136.

Li, F. (2008). Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics*, *45*(2-3), 221–247. https://doi.org/10.1016/j.jacceco.2008.02.003

Li, F. (2010a). Textual analysis of corporate disclosures: A survey of the literature. *Journal of accounting literature*, *29*, 143.

Li, F. (2010b). The information content of forward- looking statements in corporate filings-A naïve bayesian machine learning approach. *Journal of Accounting Research*, *48*(5), 1049–1102. https://doi.org/10.1111/j.1475-679X.2010.00382.x

Lilien, G. L. (2011). Bridging the academic–practitioner divide in marketing decision models. *Journal of Marketing*, *75*(4), 196–210.

Lin, C. C., Chiu, A. A., Huang, S. Y., & Yen, D. C. (2015). Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. *Knowledge-Based Systems*, *89*, 459–470. https://doi.org/10.1016/j.knosys.2015.08.011

Liu, C., Chan, Y., Alam Kazmi, S. H., & Fu, H. (2015). Financial Fraud Detection Model: Based on Random Forest. *International Journal of Economics and Finance*, *7*(7). https://doi.org/10.5539/ijef.v7n7p178

Liu, F., Zheng, J., Zheng, L., & Chen, C. (2020). Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification. *Neurocomputing*, *371*, 39–50.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26.

Liu, Y., & Sheng, X. S. (2019). The measurement and transmission of macroeconomic uncertainty: Evidence from the us and bric countries. *International Journal of Forecasting*, *35*(3), 967–979.

Loughran, T. I. M., & Mcdonald, B. (2011). When is a Liability not a Liability ? Textual Analysis , Dictionaries , and 10-Ks Journal of Finance , forthcoming. *Journal of Finance*, *66*(1), 35–65.

Loughran, T., & Mcdonald, B. (2014). Measuring readability in financial disclosures. *Journal of Finance*. https://doi.org/10.1111/jofi.12162

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, *65*, 465–470.

Luo, R., Zhang, W., Xu, X., & Wang, J. (2018). A neural stochastic volatility model, In *Thirty-second aaai conference on artificial intelligence.*

LXVX. (2019). Ledger X [[Online; accessed 10-Feb-2019]].

Madan, D. B., Carr, P. P., & Chang, E. C. (1998). The variance gamma process and option pricing. *Review of Finance*, *2*(1), 79–105.

Manning, C. D., Ragahvan, P., & Schutze, H. (2009). An Introduction to Information Retrieval. *Information Retrieval*, (100), 1–18. https://doi.org/10.1109/LPT.2009.2020494

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.

Merton, R. C. Et al. (1973). Theory of rational option pricing. *Theory of Valuation*, 229–288.

Mihoci, A., Althof, M., Chen, C. Y.-H., & Härdle, W. K. (2019). FRM financial risk meter. *Advances in Econometrics*, *42*. ssrn.com/abstract=3429549

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. ‘, & Khudanpur, S. (2010). Recurrent Neural Network Language Modeling, In *Interspeech.* https://doi.org/10.1021/jp056727x

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space Tomas. *IJCAI International Joint Conference on Artificial Intelligence.* https://doi.org/10.1162/153244303322533223

Montufar, G. F., Pascanu, R., Cho, K., & Bengio, Y. (2014). On the number of linear regions of deep neural networks, In *Advances in neural information processing systems.*

Neuberger, A. (1994). The log contract, the new instrument to hedge volatility. *Journal of Portfolio Management*, *20*(2).

Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles. https://doi.org/10.1177/0146167203029005010

Nguyen, K. (1995). Financial statement fraud:Motives, Methods, Cases and Detection. *The Secured Lender*, *51*(2), 36. https://doi.org/10.1002/9781118527436

Odean, T. (1998). Volume, volatility, price, and profit when all traders are above average. *The journal of finance*, *53*(6), 1887–1934.

Oztekin, A., Delen, D., Turkyilmaz, A., & Zaim, S. (2013). A machine learning-based usability evaluation method for elearning systems. *Decision Support Systems*, *56*, 63–73.

Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016a). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, *253*(3), 697–710.

Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016b). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, *253*(3), 697–710.

Ozturk, E. O., & Sheng, X. S. (2018). Measuring global and country-specific uncertainty. *Journal of International Money and Finance*, *88*, 276–295.

Paleologo, G., Elisseeff, A., & Antonini, G. (2010). Subagging for credit scoring models. *European Journal of Operational Research*, *201*(2), 490–499.

Parker, S. C. (1996). A time series model of self-employment under uncertainty. *Economica*, 459–475.

Patton, A. J., & Sheppard, K. (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics*, *97*(3), 683–697.

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological Aspects of Natural Language Use: Our Words, Our Selves. *Annual Review of Psychology*, *54*(1), 547–577. https://doi.org/10.1146/annurev.psych.54.101601.145041

Pennington, J., Socher, R., & Manning, C. D. (2014a). Glove: Global vectors for word representation, In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*.

Pennington, J., Socher, R., & Manning, C. D. (2014b). GloVe: Global vectors for word representation, In *Emnlp 2014 - 2014 conference on empirical methods in natural language processing, proceedings of the conference*, Association for Computational Linguistics (ACL). https://doi.org/10.3115/v1/d14-1162

Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing*, *30*(2), 19–50. https://doi.org/10.2308/ajpt-50009

Persons, O. S. (2011). Using Financial Statement Data To Identify Factors Associated With Fraudulent Financial Reporting. *Journal of Applied Business Research (JABR)*, *11*(3), 38. https://doi.org/10.19030/jabr.v11i3.5858

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pichl, L., & Kaizoji, T. (2017). Volatility analysis of bitcoin price time series. *Quantitative Finance and Economics*, *1*(QFE-01-00474), 474.

Pourhabibi, T., Ong, K.-L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, *133*, 113303. https://doi.org/10.1016/J.DSS.2020.113303

Pryor, M. (2011). *The financial spread betting handbook 2e: A guide to making money trading spread bets*. Harriman House Limited.

Purda, L. D., & Skillicorn, D. (2012). Reading between the Lines: Detecting Fraud from the Language of Financial Reports. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1572065

Purda, L., & Skillicorn, D. (2015). Accounting Variables, Deception, and a Bag of Words: Assessing the Tools of Fraud Detection. *Contemporary Accounting Research*, *32*(3), 1193–1223. https://doi.org/10.1111/1911-3846.12089

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *Amazon AWS*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, *308*, 49–57. https://doi.org/10.1016/j.neucom.2018.04.045

Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, *50*(2), 491–500. https://doi.org/10.1016/j.dss.2010.11.006

Ravn, M. O., & Uhlig, H. (2002). On adjusting the hodrick-prescott filter for the frequency of observations. *Review of economics and statistics*, *84*(2), 371–376.

Rezaee, Z. (2005). Causes, consequences, and deterence of financial statement fraud. *Critical Perspectives on Accounting*, *16*(3), 277–298. https://doi.org/10.1016/S1045-2354(03)00072-8

Ribeiro, B., & Lopes, N. (2011). Deep belief networks for financial prediction (B.-L. Lu, L. Zhang, & J. Kwok, Eds.). In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *Proccedings of the international conference on neural information processing (iconip'2011)*, Berlin, Heidelberg, Springer Berlin Heidelberg.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier, In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, san francisco, ca, usa, august 13-17, 2016*.

Richardson, S. A., Sloan, R. G., Soliman, M. T., & Tuna, I. (2005). Accrual reliability, earnings persistence and stock prices. *Journal of Accounting and Economics*, *39*(3), 437–485. https://doi.org/10.1016/j.jacceco.2005.04.005

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Securities and Exchange Commission. (2019a). Division of Corporation Finance: Standard Industrial Classification (SIC) Code List. https://www.sec.gov/info/edgar/siccodes.htm

Securities and Exchange Commission. (2019b). Form 10-K. https://www.sec.gov/files/form10-k.pdf

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., & Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Progress in brain research*, *165*, 33–56.

Sevim, C., Oztekin, A., Bali, O., Gumus, S., & Guresen, E. (2014). Developing an early warning system to predict currency crises. *European Journal of Operational Research*, *237*(3), 1095–1104.

Shen, F., Chao, J., & Zhao, J. (2015). Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing*, *167*, 243–253.

Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: Arima vs. lstm. *arXiv preprint arXiv:1803.06386*.

Sill, K. (2014). Forecast disagreement in the survey of professional forecasters. *Business Review Q*, *2*, 15–24.

Singleton, T. W., & Singleton, A. J. (2011). *Fraud Auditing and Forensic Accounting, Fourth Edition*. https://doi.org/10.1002/9781118269183

Sirignano, J. (2016). Deep learning for limit order books. *CoRR*, *abs/1601.01987*. https://arxiv.org/abs/1601.01987

Sirignano, J. A., Sadhwani, A., & Giesecke, K. (2016). Deep learning for mortgage risk. https://people.stanford.edu/giesecke/

Sirignano, J., Sadhwani, A., & Giesecke, K. (2016). Deep learning for mortgage risk. *arXiv preprint arXiv:1607.02470.*

Siriopoulos, C., & Fassas, A. (2009). Implied volatility indices–a review.

Smith, R. C. (2013). *Uncertainty quantification: Theory, implementation, and applications* (Vol. 12). Siam.

Soric, P., & Lolic, I. (2017). Economic uncertainty and its impact on the croation economy. *Public Sector Economics*, *4*(41), 443–477.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Taleb, N. (1997). *Dynamic hedging: Managing vanilla and exotic options* (Vol. 64). John Wiley & Sons.

Tang, D., Qin, B., & Liu, T. (2015). *Document Modeling with Gated Recurrent Neural Network for Sentiment Classification* (tech. rep.). Association for Computational Linguistics. http://ir.hit.edu.cn/

Tarassow, A. (2019). Forecasting us money growth using economic uncertainty measures and regularisation techniques. *International Journal of Forecasting*, *35*(2), 443–457.

Teece, D., Peteraf, M., & Leih, S. (2016). Dynamic capabilities and organizational agility: Risk, uncertainty, and strategy in the innovation economy. *California Management Review*, *58*(4), 13–35.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, *62*(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

Thorsrud, L. A. (2018). Words are the New Numbers: A Newsy Coincident Index of the Business Cycle. *Journal of Business and Economic Statistics*, 1–17.

Throckmorton, C. S., Mayew, W. J., Venkatachalam, M., & Collins, L. M. (2015). Financial fraud detection using vocal, linguistic and financial cues. *Decision Support Systems*, *74*, 78–87. https://doi.org/10.1016/j.dss.2015.04.006

Tixier, A. J.-P. (2018). Notes on Deep Learning for NLP. https://arxiv.org/abs/1808.09772

Tobback, E., Naudts, H., Daelemans, W., Junqué de Fortuny, E., & Martens, D. (2018). Belgian economic policy uncertainty index: Improvement through text mining. *International Journal of Forecasting*, *34*(2), 355–365.

Trimborn, S., & Härdle, W. K. (2018). CRIX an Index for cryptocurrencies. *Journal of Empirical Finance*, *49*, 107–122. https://doi.org/10.1016/j.jempfin.2018.08.004.

Trimborn, S., Li, M., & Härdle, W. K. (2019). Investing with Cryptocurrencies—a Liquidity Constrained Investment Approach*. *Journal of Financial Econometrics*, *18*(2), 280–306. https://doi.org/10.1093/jjfinec/nbz016

Tripier, F. (2019). Assessing the cost of uncertainty created by brexit. *EconPol opinion*, (25).

Ulaş, A., Yıldız, O. T., & Alpaydın, E. (2012). Eigenclassifiers for combining correlated classifiers. *Information Sciences*, *187*(0), 109–120.

US Securities and Exchange Comission. (2019). Agency financial report. *US Department of State*. https://www.sec.gov/files/sec-2019-agency-financial-report.pdf#mission

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need, In *Advances in neural information processing systems*.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, *218*(1), 211–229.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders, In *Proceedings of the 25th international conference on machine learning*. ACM.

Wang, H.-z., Li, G.-q., Wang, G.-b., Peng, J.-c., Jiang, H., & Liu, Y.-t. (2017). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied energy*, *188*, 56–70.

Warburton, K. (2003). Deep learning and education for sustainability. *International Journal of Sustainability in Higher Education*, *4*(1), 44–56.

Weber, M., & Camerer, C. F. (1998). The disposition effect in securities trading: An experimental analysis. *Journal of Economic Behavior & Organization*, *33*(2), 167–184.

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Elsevier Ltd. https://doi.org/10.1016/j.cose.2015.09.005

Whaley, R. E. (1993). Derivatives on market volatility: Hedging tools long overdue. *The Journal of Derivatives*, *1*(1), 71–84.

Xiong, R., Nichols, E. P., & Shen, Y. (2016). Deep learning stock volatility with google domestic trends. *CoRR*, *arXiv:1512.04916v3*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016a). *Hierarchical Attention Networks for Document Classification* (tech. rep.). In Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016b). 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*.

Yeh, S. H., Wang, C. J., & Tsai, M. F. (2015). Deep belief networks for predicting corporate defaults, In *Proceedings of the 24th wireless and optical communication conference (wocc)*, IEEE Computer Society.

Yermack, D. (2015). Is bitcoin a real currency? An economic appraisal. *Handbook of Digital Currency*, 31–43.

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017a). Comparative Study of CNN and RNN for Natural Language Processing. http://arxiv.org/abs/1702.01923

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017b). Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Zalla, R. (2017). Economic policy uncertainty in ireland. *Atlantic Economic Journal*, *2*(45), 269–271.

Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network, In *European semantic web conference*. Springer.

Zhao, Y., Li, J., & Yu, L. (2017). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, *66*, 9–16.

Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). A C-LSTM Neural Network for Text Classification. http://arxiv.org/abs/1511.08630

Zhou, L., Burgoon, J. K., Nunamaker, J. F., & Twitchell, D. (2004). Automating Linguistics-Based Cues for detecting deception in text-based asynchronous computer-mediated communication. *Group Decision and Negotiation*, *13*(1), 81–106. https://doi.org/10.1023/B:GRUP.0000011944.62889.6f

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification, In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*.

Selbständigkeitserklärung

Ich versichere, die von mir vorgelegte Dissertation selbständig und ohne unerlaubte Hilfe und Hilfsmittel angefertigt, sowie die benutzten Quellen und Daten anderen Ursprungs als solche kenntlich gemacht zu haben.

Ich bezeuge durch meine Unterschrift, dass meine Angaben über die bei der Abfassung meiner Dissertation benutzten Hilfsmittel, über die mir zuteil gewordene Hilfe sowie über frühere Begutachtungen meiner Dissertation in jeder Hinsicht der Wahrheit entsprechen.

Berlin, 03. Aug 2020