UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Text Representation through Multimodal Variational Autoencoder for One-Class Learning

Marcos Paulo Silva Gôlo

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Marcos Paulo Silva Gôlo

Text Representation through Multimodal Variational Autoencoder for One-Class Learning

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Ricardo Marcondes Marcacini

USP – São Carlos
March 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
       Gôlo, Marcos Paulo Silva
G617t     Text Representation through Multimodal
       Variational Autoencoder for One-Class Learning /
       Marcos Paulo Silva Gôlo; orientador Ricardo
       Marcondes  Marcacini. -- São Carlos, 2022.
          122 p.


          Dissertação (Mestrado - Programa de Pós-Graduação
       em Ciências de Computação e Matemática
       Computacional) -- Instituto de Ciências Matemáticas
       e de Computação, Universidade de São Paulo, 2022.


          1. Multimodal Variational Autoencoder. 2. One-
       Class Learning. 3. One-Class Text Learning. 4. Text
       Mining. I. Marcacini, Ricardo Marcondes , orient.
       II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Marcos Paulo Silva Gôlo

Variational Autoencoder Multimodal para Representação de Textos na Classificação baseada em Uma Única Classe

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Ricardo Marcondes Marcacini

USP – São Carlos
Março de 2022

Dedico este trabalho a todas as pessoas que quiseram meu bem durante toda minha vida.

ACKNOWLEDGEMENTS

Aos meus pais Marco Antônio de Paula Gôlo e Josedilma Silva Gôlo pela educação que me proporcionaram e pelo apoio em todas as etapas da minha vida, além do incentivo ao contínuo estudo;

Aos meus irmãos Patrícia Silva Gôlo e Paulo Henrique Silva Gôlo por toda ajuda, ensinamentos e puxões de orelhas que me tornaram quem sou hoje;

À minha namorada Bárbara por todo apoio e incentivo a vir para São Carlos fazer mestrado, além de sempre entender meu sumiço por estar desenvolvendo esse projeto de mestrado;

Ao meu orientador Dr. Ricardo Marcondes Marcacini por me orientar e me ajudar, nunca tendo faltado com o meu desenvolvimento acadêmico e sempre achando uma brecha em seus dias corridos para atender às minhas dúvidas e ideias. Quero agradecer também todos os momentos que teve paciência para me ensinar depois de muitas vezes já ter explicado determinado assunto;

Ao meu orientador Dr. Rafael Geraldeli Rossi, que mesmo não sendo oficial, me ajudou em várias situações até o momento, colaborando fortemente para esta pesquisa. Além disso, agradeço por ter me suportado durante dois projetos de iniciação científica da graduação e até nos almoços que compartilhamos na Universidade Federal do Mato Grosso do Sul. Um agradecimento especial para todos os conselhos que me deu para que eu pudesse entrar no mestrado, além de toda base para me tornar um pesquisador;

Aos amigos Luiz Henrique, Luiz Quirino, Lucas Padilha, José Vitor, Luiz Satélis, Lucas Fidelis, Ronaldo Fiorilo, Helder Caravalho, Tiago Jesuino, Rafael Teodoro, Fernando Fernandes e Caique aos quais fiz na graduação e sempre me ajudaram no mestrado mesmo de longe;

Aos meus amigos do Laboratório de Inteligência Computacional Adailton Araújo e Ângelo Mendes, do Laboratório GBDI do ICMC, Clementino, Maxwell e Lucas Santiago, e ao meu amigo do ICMC Breno Viana, que convivem comigo e me ajudam a todo momento, além de me proporcionar os melhores *happy hours* depois de dias exaustivos de trabalho;

Ao CNPq (Processo n° 133438/2020-1) pelo apoio financeiro, que está me possibilitando ter dedicação exclusiva ao desenvolvimento deste projeto de mestrado.

“O trabalho duro ganha do talento quando o talento não trabalha duro.”

(Kevin Durant)

RESUMO

GÔLO, M. P. S. ***Variational Autoencoder* Multimodal para Representação de Textos na Classificação baseada em Uma Única Classe**. 2022. 119 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

A classificação automática de textos tem se tornado cada vez mais relevante tanto para fins acadêmicos quanto empresariais. Tradicionalmente, essa classificação é realizada pelo aprendizado de máquina multi-classe, o qual necessita de rotulação prévia dos textos de todas as classes do problema. Essa abordagem pode falhar quando não se tem um conhecimento bem definido das classes do problema, além de exigir um grande esforço na rotulação de exemplos de treinamento para cada classe. Uma abordagem conhecida como *One-Class Learning* (OCL) pode sanar essas limitações, uma vez que seu treinamento é realizado somente com exemplos rotulados de uma classe de interesse, diminuindo assim o esforço de rotulação do usuário e tornando a classificação mais apropriada para aplicações envolvendo domínio aberto. O OCL é mais desafiador devido à falta de contra-exemplos para o treinamento do modelo. Portanto, OCL requer representações textuais mais robustas. Por outro lado, a maioria dos estudos usa representações unimodais, mesmo que diferentes domínios contenham outros tipos de informações que podem ser interpretados como modalidades distintas para dados textuais. Nesse sentido, foi proposto o *Multimodal Variational Autoencoder* (MVAE). O MVAE é um método multimodal que aprende uma nova representação a partir da fusão das modalidades distintas, capturando de forma mais adequada às características da classe de interesse. O MVAE foi explorado com as modalidades de representações semânticas e sintáticas, informações de densidade, linguísticas e espaciais. Além disso, o MVAE é baseado em um *Variational Autoencoder* que é considerado um dos estados-da-arte para aprendizado de representações. Por fim, as principais contribuições desta dissertação são: (i) um método multimodal para representar textos no cenário de OCL; (ii) detecção de notícias falsas por meio de representações geradas pelo MVAE; (iii) aplicação do MVAE para representar revisões de app no filtro de revisões de app relevantes; e (iv) sensoriamento de eventos representados pelo MVAE.

Palavras-chave: Classificação de textos, Aprendizado de máquina baseado em uma única classe, Variational autoencoders multimodais.

ABSTRACT

GÔLO, M. P. S. **Text Representation through Multimodal Variational Autoencoder for One-Class Learning**. 2022. 119 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Automatic text classification has become increasingly relevant for several applications, both for academic and business purposes. Traditionally, multi-class learning methods perform text classification, which requires prior labeling of textual datasets for all classes. These methods fail when there is no well-defined information about the texts' classes and require a great effort to label the training set. One-Class Learning (OCL) can mitigate these limitations since the model training is performed only with labeled examples of an interest class, reducing the user's labeling effort and turning the classification more appropriate for open-domain applications. However, OCL is more challenging due to the lack of counterexamples for model training. Thus, OCL requires more robust text representations. On the other hand, most studies use unimodal representations, even though different domains contain other types of information that can be interpreted as distinct modalities for textual data. In this sense, the Multimodal Variational Autoencoder (MVAE) was proposed. MVAE is a multimodal method that learns a new representation from the fusion of different modalities, capturing the characteristics of the interest class in a more adequate way. MVAE explores semantic and syntactic representations, density, linguistic and spatial information as modalities. Furthermore, MVAE is based on a Variational Autoencoder, considered one of the state-of-the-art for learning representations. Finally, the main contributions of this dissertation are: (i) a multimodal method to represent texts in the OCL scenario; (ii) detection of fake news through representations generated by MVAE; (iii) applying MVAE to represent app reviews in the filtering of relevant app reviews; and (iv) sensing events represented by the MVAE.

Keywords: Text classification, One class learning, Multimodal variational autoencoder.

LIST OF FIGURES

Figure 1 – Illustration of a basic structure of an autoencoder with $\boldsymbol{d}_i$ input and $\boldsymbol{r}_i$ output. $\boldsymbol{z}_i$ represents the bottleneck. 37

Figure 2 – Illustration of a basic structure of an variational autoencoder with $\boldsymbol{d}_i$ input and $\boldsymbol{r}_i$ output. $\boldsymbol{z}_i$ represents the bottleneck, $\boldsymbol{\mu}_i$ represent the mean learned and the $\boldsymbol{\sigma}_i$ represent the standard deviation learned. 39

Figure 3 – Illustration of the reparametrization trick used by VAE to apply the backpropagation and learn the weights of the neurons. 40

Figure 4 – Multimodality scenario for textual data. named entities, topics, n-grams and sentiment are modalities of textual documents. Multimodal representation represents the combination of modalities of each textual document. 41

Figure 5 – Illustration of a decision function for OCL. White points are used to train. Green points are new examples of the interest class and red points are new examples of the non-interest class. Points within the yellow area will be assigned to the interest class. Points outside the yellow area will not be assigned to the interest class. 44

Figure 6 – Illustration of the fictitious examples generated close to the origin of the plane so that the maximum separation hyperplane is generated in the OCSVM proposed by (SCHÖLKOPF *et al.*, 2001). $+1$ means the interest class and -1 the opposite. 45

Figure 7 – Illustration of a Density Information Representation. 54

Figure 8 – Illustration of a Variational Autoencoder. 55

Figure 9 – Illustration of the proposed representation method MVAE-FakeNews For Fake News with the inputs $\boldsymbol{\lambda}_i$ and $\boldsymbol{\delta}_i$, and outputs $\boldsymbol{\lambda}'_i$ and $\boldsymbol{\delta}'_i$. 57

Figure 10 – Critical difference diagram of Friedman's statistical test with Nemenyi post-test. 62

Figure 11 – Two-dimensional projection (t-SNE) of each fake news representation method. The colors indicate fake news and real news. Representation methods that show less overlap between classes are more promising for detecting fake news. 63

Figure 12 – Pipeline with six steps representing our proposal to represent app reviews and detect relevant app reviews. 70

Figure 13 – Different cluster settings comparing the behavior of a relevant app review belonging to one of the relevant reviews topics. 72

Figure 14 – Illustration of an Autoencoder. 73

Figure 15 – Illustration of a Variational Autoencoder. 74

Figure 16 – Illustration of our MAE to represent the app reviews. 75
Figure 17 – Illustration of our MVAE to represent the app reviews. 76
Figure 18 – Critical difference diagrams with the average rankings of the Friedman test with Nemenyi's post-test considering F_1-Score and AUC-ROC on all train scenarios and datasets. 83
Figure 19 – The Pipeline of multimodal representation Learning to detect events of interest through one-class learning. 91
Figure 20 – Critical difference diagrams with the average rankings of the Friedman test with Nemenyi's post-test. We present the diagrams considering each Table of result, i.e., four scenario of labeled events (1%, 2%, 3% and 33%) and three metrics (F_1-Score, AUC-ROC and Accuracy). 102

LIST OF TABLES

Table 1 – Representation generated by the vector space model for *m* documents and *n* attributes. 32

Table 2 – BoW representation with TF term weight of the first pair of texts. BoW generated different representations for texts with the same meaning. 34

Table 3 – BoW representation with TF term weight of the second pair of texts. BoW generated equal representations for texts with the opposite meaning. 34

Table 4 – Higher values of F_1-Scores of each representation technique considering the training percentage scenarios. The amount of fake news equivalent to a percentage appears next to the %. 61

Table 5 – Comparison between our proposal and related works considering the learning type, the use of multimodality, the use of irrelevant reviews, and the text representation. 69

Table 6 – Number of relevant and irrelevant reviews of the datasets used in the experimental evaluation. 77

Table 7 – Highest **F_1-Scores** from SVDD for each representation technique on the **ARE** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 80

Table 8 – Highest **AUC-ROC** from SVDD for each representation technique on the **ARE** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 80

Table 9 – Highest **F_1-Scores** from SVDD for each representation technique on the **TEN** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 81

Table 10 – Highest **AUC-ROC** from SVDD for each representation technique on the **TEN** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 81

Table 11 – Highest $\boldsymbol{F_1}$**-Scores** from SVDD for each representation technique on the **TIT** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 82

Table 12 – Highest **AUC-ROC** from SVDD for each representation technique on the **TIT** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered. 82

Table 13 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step. 97

Table 14 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step. 97

Table 15 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step. 98

Table 16 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step. 98

Table 17 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step. 98

Table 18 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step. 99

Table 19 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step. 99

Table 20 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step. 99

Table 21 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step. 100

Table 22 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step. 100

Table 23 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step. 100

Table 24 – Results in ten event datasets considering the **accuracy** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step. 101

Table 25 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM. Values of TripleVAE are considering 60 events to train (1%) and the others representations methods consider 2000 (33%) events in the training step. 101

Table 26 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM. Values of TripleVAE are considering 60 events to train (1%) and the others representations methods consider 2000 (33%) events in the training step. 101

LIST OF ABBREVIATIONS AND ACRONYMS

AEs	Autoencoders
AlBERT	A Lite BERT
AUC-ROC	Area Under Curve Receiver Operating Characteristic
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
CDLMs	Context-Dependent Language Models
CFLMs	Context-Free Language Models
CNN	Convolutional Neural Network
Concat	Concatenation
D. Tree	Dependency Tree
DBERTML	DistilBERT Multilingual
DistilBERT	Distilled version of BERT
DNN	Deep Neural Network
ELMo	Embeddings from Language Models
FCN	Fact Checking News
FN	False Negatives
FNN	FakeNewsNet
FP	False Positive
IDC	International Data Corporation
KL	Kullback-Leibler
Lat-Long	Latitude and Longitude
LIWC	Linguistic Inquiry and Word Count
LSTM	Long-Short Term Memory
MAE	Multimodal Autoencoder
MCL	Multi-Class Learning
MVAEs	Multimodal Variational Autoencoders
OCL	One-Class Learning
OCSVM	One-Class Support Vector Machine
PUL	Positive and Unlabeled Learning
RARTR	Relevant App Reviews Topic Representation
RoBERTa	Robustly optimized BERT pretraining approach

SVDD	Support Vector Data Description
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negatives
TP	True Positives
VAE	Variational Autoencoder

CONTENTS

1 INTRODUCTION 25

1.1 Contextualization 25

1.2 Research Challenges 27

1.3 Research Goals 27

1.4 Main Contributions and Results 28

1.5 Dissertation Organization 29

2 THEORETICAL FOUNDATION 31

2.1 Text Representation 31

2.1.1 Bag-of-Words 32

2.1.2 Neural Language Models 34

2.2 Autoencoders 36

2.3 Variational Autoencoders 38

2.4 Multimodal Learning 40

2.5 Machine Learning 42

2.5.1 Multi-Class Learning 42

2.5.2 One-Class Learning 43

2.5.3 One Class Support Vector Machine 45

2.6 Concluding Remarks 47

3 LEARNING TEXTUAL REPRESENTATIONS FROM MULTIPLE MODALITIES TO DETECT FAKE NEWS THROUGH ONE-CLASS LEARNING 49

3.1 Introduction 49

3.2 Related Work 51

3.3 Multimodal Variational Autoencoder for Fake News 53

3.3.1 Density Information Representation 53

3.3.2 Variational Autoencoder 54

3.3.3 One-Class Learning 56

3.4 Experimental Evaluation 58

3.4.1 Fake News Text Collections 58

3.4.2 Experimental Settings 59

3.4.3 Results and Discussion 61

3.5 Concluding Remarks 63

4 DETECTING RELEVANT APP REVIEWS FOR SOFTWARE EVOLUTION AND MAINTENANCE THROUGH MULTIMODAL ONE-CLASS LEARNING 65
4.1 Introduction 65
4.2 Related Work 67
4.3 Multimodal Representations to Detect Relevant App Reviews . . . 69
4.3.1 Text Embedding Representation 71
4.3.2 RARTR: Relevant App Reviews Topic Representation 71
4.3.3 Multimodal Representation Learning 72
4.3.4 Relevant App Reviews Detection through One-Class Learning . . . 75
4.4 Experimental Evaluation 77
4.4.1 App Review Collections 77
4.4.2 Experimental Settings 77
4.4.3 Results and Discussion 79
4.5 Threats to work validity 84
4.6 Concluding Remarks 84

5 TRIPLE-VAE: A TRIPLE VARIATIONAL AUTOENCODER TO REPRESENT EVENTS IN ONE-CLASS EVENT DETECTION 87
5.1 Introduction 87
5.2 Related Work 89
5.3 Proposal: One-Class Multimodal Event Detection 90
5.3.1 Event Geolocation 91
5.3.2 Text Embeddings 91
5.3.3 Density Information 92
5.3.4 Multimodal Representation Learning 92
5.3.5 One-Class Learning to Detect Events of Interest 94
5.4 Experimental Evaluation 95
5.4.1 Event Collections and Experimental Settings 95
5.4.2 Results and Discussion 97
5.5 Concluding Remarks 102

6 CONCLUSION AND FUTURE WORK 105
6.1 Contributions and Scientific Innovations 105
6.2 Publications 107
6.3 Limitations and Future Work 108

BIBLIOGRAPHY 109

CHAPTER 1

INTRODUCTION

1.1 Contextualization

The International Data Corporation (IDC)[1] estimated that the volume of data generated by the world will increase from 45 zettabytes in 2019 to 175 zettabytes in 2025 (JANEV *et al.*, 2020). Much of this data is in text format since texts are one of the simplest and most traditional ways to exchange information (HASSANI *et al.*, 2020). For instance, articles, reviews, publications on social networks, medical, scientific and business reports, news, and e-mails contain textual content. Furthermore, texts are present in various areas such as finance, macroeconomics, media economics, industrial organization, and marketing (GENTZKOW; KELLY; TADDY, 2019).

In many real-world applications, manually retrieving valuable information from large textual datasets is infeasible. Therefore, computational methods for knowledge extraction from textual datasets are emerging and becoming essential in society. One of the ways to perform the automatic extraction of information from the texts is through automatic text classification (AGGARWAL, 2018a; WEISS; INDURKHYA; ZHANG, 2015).

Automatic text classification assigns a previously defined label in unlabeled textual documents (AGGARWAL, 2018a). Supervised Multi-Class Learning (MCL) is a popular strategy for automatic text classification (TAN; STEINBACH; KUMAR, 2013). In MCL, the user must know all classes of the problem and label documents for all those classes in the training step of the classification model. Therefore, the user must label documents of classes even if he/she is not interested in them (KRAWCZYK; WOŹNIAK; HERRERA, 2015). This characteristic implies two limitations. First, when the user does not label examples for all domain classes, and second when a new domain class comes up. Thus, MLC may not be viable for scenarios in which there is not a well-defined knowledge of all classes of the problem. Also, it is important

[1] <https://www.idc.com/>

to consider that supervised multi-class learning requires a greater effort to label a significant amount of examples for each possible class.

Approaches that mitigate the limitations presented by the MLC have gained emphasis in the last years (ALAM *et al.*, 2020). One approach is the One-Class Learning (OCL) (ALAM *et al.*, 2020; KHAN; MADDEN, 2009; TAX, 2001). OCL uses only labeled examples from one class (user's interest class) to perform learning, i.e., the learning is in the absence of counterexamples (PERERA; PATEL, 2019; RUFF *et al.*, 2018; KHAN; MADDEN, 2014b; TAX, 2001; MANEVITZ; YOUSEF, 2001). Thus, OCL reduces the labeling effort and is more appropriate for open-domain applications (in which we do not know all the classes in advance) or applications in which the user is interested in one class of the problem. After training with only examples of interest, OCL will be able to identify whether a new example belongs to the interest class or not. OCL can be useful for news filters, recommendation systems, information retrieval systems, web sensing, and fake news detection (ALAM *et al.*, 2020; MARCACINI *et al.*, 2017; PAN *et al.*, 2008; GÔLO *et al.*, 2021; GÔLO; ROSSI; MARCACINI, 2021).

The learning process is more challenging for the OCL due to the lack of counterexample for model training. Thus, OCL requires more robustness representations. Generally, most existing methods use the traditional Bag-of-Words (BoW) technique for text representations (MANEVITZ; YOUSEF, 2001; ZHUANG; DAI, 2006; MANEVITZ; YOUSEF, 2007; JUNIOR; ROSSI, 2017; GÔLO; MARCACINI; ROSSI, 2019). This technique represents textual documents through a set of words and their frequencies. However, BoW representation has limitations, such as high dimensionality and sparsity and inefficiency in the presence of synonyms and ambiguity (AGGARWAL, 2018a). Other studies explore dimensionality reduction techniques which are able to represent texts in a non-sparse and low-dimensional way, however, do not effectively improve the automatic text classification (KUMAR; RAVI, 2017b; KUMAR; RAVI, 2017a; GÔLO; MARCACINI; ROSSI, 2019). More recently, several studies have investigated the use of language models based on neural networks, such as Word2Vec (MIKOLOV *et al.*, 2013) and Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN *et al.*, 2019), that generates more semantic representations through word embeddings (low-dimensional vectors) Ruff *et al.* (2019), Cichosz (2020), Mayaluru (2020).

We highlight that those representation models generate a representation focused mainly on the words or sentences of textual documents. However, several domains can contain other types of information that can be useful to perform learning. Different representations can be interpreted as distinct textual data modalities, such as keywords, named entities, topics, sentiment, temporal, geographic, and semantic information (ZHOU *et al.*, 2020; GUO; WANG; WANG, 2019; BLIKSTEIN, 2013). Multimodal representation learning methods explore these different types of information to learn a more robust representation since the different modalities can be supplementary or complementary (LI; YANG; ZHANG, 2018; GUO; WANG; WANG, 2019).

1.2 Research Challenges

Despite the benefits of representations generated through multimodal learning, using multimodal methods to represent texts in the OCL scenario is a gap in the literature. Two research challenges related to multimodal learning for OCL are summarized below.

1. **Multimodal text representations for OCL**: OCL is challenging as we only have the labeled examples of the interest class. Thus, a textual representation suitable for OCL must model an m-dimensional space in which documents of the interest class are closer to each other, while they are far from examples that do not belong to the interest class. Although combining multiple modalities allows for more robust representations according to the classification task (LI; YANG; ZHANG, 2018; GUO; WANG; WANG, 2019), the vast majority of automatic text classification through OCL literature explores only unimodal methods for text representation (JUNIOR; ROSSI, 2017; KUMAR; RAVI, 2017b; KUMAR; RAVI, 2017a; GÔLO; MARCACINI; ROSSI, 2019; RUFF *et al.*, 2019; CICHOSZ, 2020; MAYALURU, 2020).

2. **Unsatisfactory classification performances using a few labeled examples of the interest class:** even though OCL decreases the user's labeling effort because he/she only labels examples of one class, the fewer the number of labeled examples, the lesser the user's effort. The reduction of training examples can harm the classification performance in various application domains. A research challenge involves investigating appropriate representations to reduce dependence on large training sets of the interest class while preserving OCL performance (JASKIE; SPANIAS, 2019; SOUZA *et al.*, 2021b).

1.3 Research Goals

This dissertation presents multimodal representation learning for OCL in textual data. In particular, we explored Multimodal Variational Autoencoders (MVAEs). MVAEs are representation learning methods that learn a representation from multiple modalities through a neural network. MVAE makes use of a Variational Autoencoder (VAE). VAEs are generative models and are considered one of the state-of-the-art for text representation learning (XU; DURRETT, 2018; XU; TAN, 2019; WANG *et al.*, 2019; LI *et al.*, 2020; CHE; YANG; WANG, 2020; FELHI; ROUX; SEDDAH, 2021).

Our proposal aims to meet the challenges discussed above, i.e., the proposed MVAE neural architectures must obtain a more suitable m-dimensional space for OCL tasks, as well as deal with scenarios considering a few labeled documents. Moreover, another research goal is to analyze the proposal in real-world applications involving textual data, such as detecting fake news, filtering relevant reviews, and web sensing. Thus, we can verify the generalizability of our MVAE architectures considering different domains and tasks.

1.4 Main Contributions and Results

The main contributions of this dissertation are:

- **Multimodal Variational Autoencoders for OCL**: we have introduced an MVAE architecture which generates more suitable textual representations for OCL tasks. In particular, our MVAE explores as modalities: (i) pre-trained embeddings from the BERT multilingual model to incorporate more semantic knowledge; (ii) topic information from the high-density regions of the original vector space of the textual data; (iii) features with the linguistic structure of the texts, such as pronouns, prepositions, conjunctions, negations and emotions; and (iv) geolocation data (latitude and longitude). In addition, our MVAE also proved to be robust for scenarios with few labeled examples in three different domains, further reducing the labeling effort.

- **Detecting fake news through MVAE representations** (GÔLO *et al.*, 2021): fake news can rapidly spread through internet users. Fake news texts constantly evolve in writing and falsehood, requiring more robust methods to represent the news. On the other hand, a textual-based unimodal representation is generally used, such as bag-of-words or representations based on linguistic categories. Thus, we use our proposed MVAE to represent the news in detecting fake news through OCL. The MVAE learns a new representation from the combination of two promising modalities for fake news detection: text embeddings and topic/density information since fake news has high-density regions representing well-defined topics such as politics, society, celebrities, science, religion, economics, and pandemic. Results show that the MVAE with 3% of labeled fake news outperforms other representation methods with 10% of labeled reviews in most scenarios. MVAE proved to be promising to represent the texts in the OCL scenario to detect fake news.

- **Textual representations for mobile app reviews to support software evolution** (GÔLO *et al.*, 2022; ARAUJO *et al.*, 2020): mobile app reviews are rich information for software evolution and maintenance. To handle the smaller number of labeled relevant reviews without harming classification performance, we use our MVAE to improve feature extraction and reviews representation. Our MVAE learns representations which explore both textual data, i.e., text embeddings, and visual information based on the high-density regions of the app reviews since these have different topics or subtopics, such as bugs, features, user experience, security, and performance. Our method achieved competitive results even using only 3% of labeled reviews compared to models that used the entire training set in most scenarios.

- **Web sensing through MVAE representations** (GÔLO; ROSSI; MARCACINI, 2021): events are phenomena that occur at a specific time and place. Event detection is a multimodal task since these events have the textual, topic, geographical, and temporal components. Most multimodal research in the literature uses the concatenation of the components to represent the events. On the other hand, we explore our MVAE to represent the events in Web sensing. MVAE learns a unified representation from textual (text embeddings), spatial (geolocation), and density (topics from the event of interest) modalities. MVAE proved to be promising to represent the events in the one-class event detection scenario.

- **Source code and datasets**: All the source codes of the proposed MVAEs are available to the community: Fake News[2], Relevant App Reviews[3], and Events[4]. Moreover, previous studies on automatic text classification through OCL perform an experimental evaluation using multi-class text collections. These collections may not reflect the real scenario of automatic text classification through OCL. Therefore, in this dissertation, 183 textual datasets for Web Sensing tasks using OCL were collected and made available (GÔLO; ROSSI; MARCACINI, 2021).

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows:

Chapter 2 presents the Theoretical Foundations of textual representation and automatic text classification processes. Thus, this chapter presents the representation process, the concepts of BoW, neural language models, neural networks such as autoencoders and variational autoencoders, and multimodal learning. In addition, this chapter presents the concepts of multi-class and one-class learning regarding the classification process.

Chapter 3 is a version of the paper *"Learning Textual Representations from Multiple Modalities to Detect Fake News Through One-Class Learning"*. This paper is about the application of the proposed MVAE in the detection of fake news and was published in the 2021 Brazilian Symposium on Multimedia and the Web (GÔLO *et al.*, 2021). In addition, an extension of this paper was submitted to the Decision Support Systems journal.

Chapter 4 is an extended version of the paper *"Detecting Relevant App Reviews for Software Evolution and Maintenance through Multimodal One-Class Learning"*. This paper is about the application of the MVAE in detecting relevant app reviews and was submitted to Information and Software Technology journal.

Chapter 5 is an extended version of the paper *"Triple-VAE: A Triple Variational Autoencoder to Represent Events in One-Class Event Detection"*. This paper is about the application of MVAE in detecting events of interest and has been accepted for publication at the 2021 National Meeting on Artificial and Computational Intelligence (GÔLO; ROSSI; MARCACINI, 2021).

[2] <https://github.com/GoloMarcos/MVAE-FakeNews_Webmedia2021>.

[3] <https://github.com/GoloMarcos/MVAE-RelevantReviews>.

[4] <https://github.com/GoloMarcos/TripleVAE-Events>.

It is important to emphasize that this work won the best paper on the conference's main track award.

Chapter 6 presents the final remarks of this dissertation, highlights the main contributions, innovations, and limitations, and points out the future work.

CHAPTER

2

THEORETICAL FOUNDATION

This chapter presents the theoretical foundations of text representation, autoencoders, multimodal learning, classification through machine learning. Section 2.1 presents the vector space model considering the bag-of-words technique and representations based on neural language models. We also discuss concepts about autoencoders in Section 2.2 and variational autoencoders in Section 2.3. In addition, Section 2.4 presents concepts about multimodal learning. Finally, Section 2.5 presents Multi-Class, One-Class learning, and the details of the OCSVM algorithm are presented in Section 2.5.2.

It is important to emphasize that the information presented in this chapter is also presented in the articles contained in the following chapters. However, the theoretical foundations are more didactic and detailed in this chapter. Therefore, if the reader knows the theoretical foundations of text representation and classification through machine learning, he/she can go straight to the following chapters.

2.1 Text Representation

Machine learning algorithms need structured data to perform learning. However, textual data is naturally unstructured. Therefore, pre-processing and textual representation techniques must be used for the automatic classification of texts to be applied. One of the most traditional ways to represent texts in machine learning is through the vector space model (AGGARWAL, 2018a; SHALEV-SHWARTZ; BEN-DAVID, 2014). In this model, a vector represents each textual document, and each vector dimension represents a characteristic.

Table 1 presents an example of the representation generated from the vector space model with m documents and n attributes. $\mathscr{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_m\}$ denotes the set of text documents, and $\mathscr{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n\}$ denotes the set of attributes. Thus, a textual collection with m documents and n attributes would result in a matrix with m rows and n columns. The cells w_{d_i,t_j} contain values that associate the $\boldsymbol{d}_i$ document with the $\boldsymbol{t}_j$ attribute. In multi-class classification, there is a column to represent the class of the respective document. However, in the One-Class Learning, this column is not required.

Table 1 – Representation generated by the vector space model for m documents and n attributes.

	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$	$\mathbf{t}_4$	$\mathbf{t}_5$	$\mathbf{t}_6$	$\mathbf{t}_7$	$\cdots$	$\mathbf{t}_n$
$\mathbf{d}_1$	w_{d_1,t_1}	w_{d_1,t_2}	w_{d_1,t_3}	w_{d_1,t_4}	w_{d_1,t_5}	w_{d_1,t_6}	w_{d_1,t_7}	$\cdots$	w_{d_1,t_n}
$\mathbf{d}_2$	w_{d_2,t_1}	w_{d_2,t_2}	w_{d_2,t_3}	w_{d_2,t_4}	w_{d_2,t_5}	w_{d_2,t_6}	w_{d_2,t_7}	$\cdots$	w_{d_2,t_n}
$\mathbf{d}_3$	w_{d_3,t_1}	w_{d_3,t_2}	w_{d_3,t_3}	w_{d_3,t_4}	w_{d_3,t_5}	w_{d_3,t_6}	w_{d_3,t_7}	$\cdots$	w_{d_3,t_n}
$\mathbf{d}_4$	w_{d_4,t_1}	w_{d_4,t_2}	w_{d_4,t_3}	w_{d_4,t_4}	w_{d_4,t_5}	w_{d_4,t_6}	w_{d_4,t_7}	$\cdots$	w_{d_4,t_n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{d}_m$	w_{d_m,t_1}	w_{d_m,t_2}	w_{d_m,t_3}	w_{d_m,t_4}	w_{d_m,t_5}	w_{d_m,t_6}	w_{d_m,t_7}	$\cdots$	w_{d_m,t_n}

The vector space model is the basis for different text representation techniques. One of them is Bag-of-Words, which is presented below.

2.1.1 Bag-of-Words

The bag-of-words (BoW) technique considers each attribute a simple term, i.e., a word from the text document (AGGARWAL, 2018a). Thus, in BoW, the values of cells w_{d_i,t_j} are called term weights. The term weights often used are (ROSSI, 2016):

- **Term Frequency (TF)**: defines the frequency of occurrence of a term in a document;
- **Term Frequency - Inverse Document Frequency (TF-IDF)**: weights the TF with the frequency of documents with that term (Equation 2.1).
- **Binary**: indicates whether or not the term occurs in the document.

$$w_{d_i,t_j} = TF\text{-}IDF_{d_i,t_j} = TF_{d_i,t_j} \cdot log\left(\frac{|D|}{df_{t_j}}\right) \tag{2.1}$$

in which, TF_{d_i,t_j} corresponds to the frequency of the term t_j in the d_i document, $|D|$ to the number of documents, and df_{t_j} to the number of documents that contain the term t_j.

Usually, studies apply some techniques before structuring the texts considering the BoW to improve the representation quality and, consequently, improve the results obtained by machine learning algorithms (AGGARWAL, 2018a; ROSSI, 2016). The techniques aim to reduce the number of terms and increase the quality of representation. Generally, the studies apply the following techniques (MANEVITZ; YOUSEF, 2007; JUNIOR; ROSSI, 2017; GÔLO; MARCACINI; ROSSI, 2019):

- **Standardize the texts:** standardize the same uppercase and lowercase words so that they become the same word, for instance, "Metric" and "metric";

- **Remove irrelevant characters:** remove irrelevant characters such as punctuation or non-alphanumeric characters. Also, depending on the classification problem, other types of characters can be removed, for instance, numeric characters if they are considered irrelevant;

- **Remove irrelevant words:** remove stopwords that do not help to discriminate the document category (e.g., prepositions, articles, and conjunctions);

- **Simplify words:** One way to perform this technique is from stemming (CONRADO, 2014). Stemming turns each word into its stem.

Even though the BoW is a simple and fast technique to represent texts and these techniques mentioned above improve the representation generated by BoW, it still has limitations. One of them is the high dimensionality since the vocabulary of a real-world textual collection can be huge. Another limitation is the high sparsity since documents contain a small portion of the words from the vocabulary. These limitations can degrade the performance of machine learning algorithms, although BoW can get satisfactory results in some text classification applications (ROSSI, 2016).

Another limitation present in the representations generated by the BoW technique is the ineffectiveness of synonyms and ambiguity since each different word is represented by a different feature in BoW, and words with the same spelling are represented by the same attribute. For instance, to represent the following sets of textual documents with the BoW:

1. “These results have the same average” and “The products get equal means”;

2. “The Company X bought the company Y” and “The Company Y bought the company from X”.

Tables 2 and 3 show the representations of texts generated by the BoW considering the pairs of sentences presented above with the data cleaning process and the term weight TF. In the first pair of texts, the texts have the same meaning. However, no occurrences of the same terms generate different representations. In the second pair of texts, the texts have opposite meanings, however equal representations.

Given the limitations of BoW, studies propose other textual representation models. Among them, the neural language models are considered one of the states-of-the-art for textual representation. In the next section, we present the details of neural language models.

Table 2 – BoW representation with TF term weight of the first pair of texts. BoW generated different representations for texts with the same meaning.

	result	**ha**	**same**	**average**	**product**	**get**	**equal**	**mean**
$\boldsymbol{d}_1$	1	1	1	1	0	0	0	0
$\boldsymbol{d}_2$	0	0	0	0	1	1	1	1

Table 3 – BoW representation with TF term weight of the second pair of texts. BoW generated equal representations for texts with the opposite meaning.

	compan	**X**	**Y**	**bought**
$\boldsymbol{d}_1$	2	1	1	1
$\boldsymbol{d}_2$	2	1	1	1

2.1.2 Neural Language Models

A recent method for textual data representation is the language models[1], which explore the idea of distributed representations. In this case, numeric vectors (usually with a lower dimension than BoW) are generated to represent words, sentences, or entire documents (AMORIM; HENNIG, 2015; MIKOLOV *et al.*, 2013). In addition, the vectors generated by language models can capture syntactic and semantic relations, thus providing better representation for several natural language processing tasks (OTTER; MEDINA; KALITA, 2020).

Mikolov *et al.* (2013) proposes one of the pioneering language models known as word2vec. The authors propose neural language models such as Continuous Bag-of-Words and Skip-Gram. These models are Context-Free Language Models (CFLMs). The CFLMs generate static embeddings for each word in the text, i.e., each word has a vector independent of the surrounding words. This fact can generate some limitations. For instance, the word “mean”, which has different meanings, such as “average” or “meaning”. However, word2vec will only generate one vector in the embedding space for this word. Thus, when processing new texts, the same vector will be used in the embedding space, regardless of the context in which the word appears in the new texts. One of the ways to solve this limitation is through the Context-Dependent Language Models (CDLMs) (DEVLIN *et al.*, 2019).

CDLMs generate a feature vector for the words considering the sentence in which they occur (DEVLIN *et al.*, 2019), i.e., the CDLMs dynamically generate the vectors in the embedding space. Thus, two sentences with the word “mean” in the two contexts presented above will have vectors of different characteristics. Furthermore, different natural language processing tasks such as named entity recognition and text classification get better results using CDLM than context-free language models (OTTER; MEDINA; KALITA, 2020).

[1] Language model is a broader concept. However, in this work, we use the term language model to represent language models based on neural networks, such as Word2Vec and BERT.

CDLMs can be unidirectional or bidirectional. Unidirectional models analyze the text in a single direction, such as the direction of reading. Bidirectional models analyze text both from left to right and from right to left. Bidirectional language models provide better performance in tasks such as next sentence prediction and imputing missing words. Consequently, they perform better in natural language processing tasks, such as detecting named entities and text classification (DEVLIN *et al.*, 2019; LIU; YIN; DU, 2019; SANH *et al.*, 2019).

One of the first bidirectional CDLM was the Embeddings from Language Models (ELMo) (PETERS *et al.*, 2018). ELMo uses recurrent neural networks of the type Long-Short Term Memory (LSTM) and, to be bidirectional, it uses two LSTM networks: (i) one to analyze the text from left to right; and (ii) one to analyze text from right to left. In addition to using LSTM, CDLM can use transformers. The transformers have the advantage of being parallelizable, whereas LSTM are difficult to parallelize. By using two LSTMs, ELMo becomes computationally more expensive than a single bidirectional model (DEVLIN *et al.*, 2019). In this sense, Devlin *et al.* (2019) proposed a CDLM based on bidirectional transformers, turning the model computationally less expensive. This advantage of reducing the computational cost made the training of models in a large textual corpus possible. Examples of transformer-based bidirectional models are: Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN *et al.*, 2019), Robustly optimized BERT pretraining approach (RoBERTa) (LIU *et al.*, 2019), Distilled version of BERT (DistilBERT) (SANH *et al.*, 2019), DistilBERT Multilingual (REIMERS; GUREVYCH, 2020), A Lite BERT (AlBERT) (LAN *et al.*, 2019). These models are considered state-of-the-art for different natural language processing tasks (OTTER; MEDINA; KALITA, 2020).

The BERT model was trained in a very large textual corpus, and the model represents sentences based on their context. In the training step, the BERT executes two tasks, and the first is complete sentences with words masked. Formally, given the word sequence of a textual document $\mathbf{d}_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,v}\}$, the BERT model generates a corrupted version of $\mathbf{d}_i$, $\hat{\mathbf{d}}_i$, in which words are selected randomly (e.g., 15%) and replaced by the `[MASK]` symbol. Also, let $\bar{\mathbf{d}}_i$ be the masked tokes of $\mathbf{d}_i$. Then, the BERT's training consists of reconstructing the masked words $\bar{\mathbf{d}}_i$ from $\hat{\mathbf{d}}_i$ Yang *et al.* (2019):

$$\max_{\Theta} \log p_{\Theta}(\bar{\mathbf{d}}_i|\hat{\mathbf{d}}_i) \approx \sum_{x_j \in \mathbf{d}_i} c_{x_j} \log \frac{\exp(H_\theta(\hat{\mathbf{d}}_\mathbf{i})_{x_j}^T \boldsymbol{e}(x_j))}{\exp(\sum_{x'} H_\theta(\hat{\mathbf{d}}_\mathbf{i})_{x_j}^T \boldsymbol{e}(x'))}, \tag{2.2}$$

in which $c_{x_j} = 1$ indicates that x_j is masked, $\boldsymbol{e}(x_j)$ indicates the embedding of the word x_j, x' is $\mathbf{d}_i$ without x_j, $H_\theta(\mathbf{d}_i) = \{\mathbf{h}_\theta(\mathbf{d}_i)_1, \mathbf{h}_\theta(\mathbf{d}_i)_2, \ldots, \mathbf{h}_\theta(\mathbf{d}_i)_v\}$ is a sequence of hidden vectors mapped by a Transformer.

The second task is to predict if a sentence $\boldsymbol{s}_i$ is the next of another sentence $\boldsymbol{s}_j$ based on the representation generated by BERT. In training, half the time, $\boldsymbol{s}_i$ is the next sentence of $\boldsymbol{s}_j$, and the other half $\boldsymbol{s}_i$ is not, i.e., it is a random sentence. In the 50% of training where $\boldsymbol{s}_i$ follows

$\boldsymbol{s}_j$, the sentence $\boldsymbol{s}_i$ is labeled *IsNext*, and the random sentences are labeled *NotNext*. BERT uses sentence embeddings as input to the next sentence classification task. Thus, BERT separates sentences and tries to predict whether one follows the other (DEVLIN *et al.*, 2019). This task helps the model learn relationships between sentences that may not be learned with the first task alone. Furthermore, it is noteworthy that this task directly influences natural language inference and the objectives of learning representations (DEVLIN *et al.*, 2019). The article Devlin *et al.* (2019) compares pre-trained neural language models with this task and without this task. This comparison shows that models without the next sentence prediction tasks obtain worse results than models that use it (DEVLIN *et al.*, 2019).

We represent the phrases used in Section 2.1.1 to highlight the limitations of the BoW in comparison with the BERT model. We calculate the similarities between the phrases[2]. A cosine similarity measure was used (TAN; STEINBACH; KUMAR, 2013).

For the phrases represented in Table 2, the result was a cosine similarity of 0.65. On the other hand, the two sentences represented in Table 3 obtained a cosine similarity of 0.98. The similarity of the two examples was high (1 is the maximum). The similarity obtained via embeddings is better than BoW that obtained a 0 (minimum) similarity for the pair in Table 2. In the second pair, even though the similarity between sentences with different contexts was high, it is better than BoW that obtained the similarity of 1 (maximum). Furthermore, considering the sentences of Table 3 represented by the BERT model, this high similarity (0.98) makes sense since the sentences deal with the same action (buy) and the same objects of purchase (company).

In addition to representing texts using more robust models such as BERT, other neural networks architecture can be used to learn representation, including those that allow combining data from multiple modalities. We highlight the use of autoencoders in this scenario (WANG *et al.*, 2019), which will be explained in the following section.

2.2 Autoencoders

Autoencoders (AEs) are a specific type of neural network in which the input has the same dimension as the output (LIU *et al.*, 2017). An autoencoder is a neural network with 3 components: encoder, bottleneck, and decoder (Figure 1). These three components form the neural network's hidden layers. The encoder compresses the input and generates the bottleneck, while the decoder reconstructs the input from that bottleneck. Typically, the hidden layers of the autoencoder consist of fewer neurons than the input. Thus, reconstructions are only possible if the weights of the hidden layer neurons capture the most representative characteristics of the input

[2] Pre-trained neural language model code to generate the representation of the phrases and to compute the cosine similarity: <https://colab.research.google.com/drive/1mambBdV8G94FA1LO0GvxpE3maKovGFlK?usp=sharing>.

data (KIEU *et al.*, 2019). Then, AEs are appropriate for learning representations (MANEVITZ; YOUSEF, 2007).

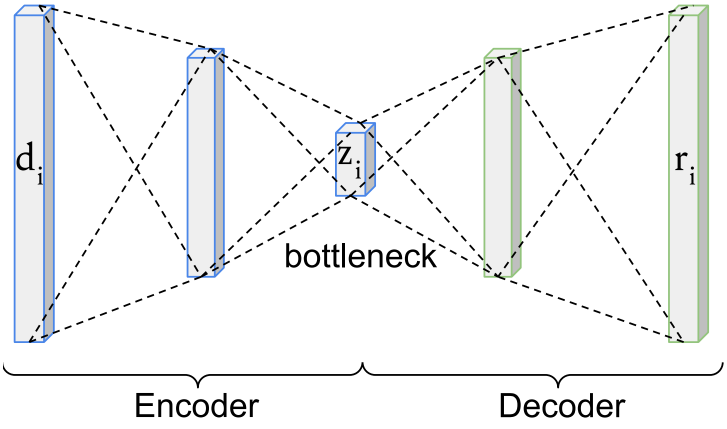
Autoencoders act as a bottleneck to carry out the neural network extract significant features from a given dataset (LIU *et al.*, 2017). Thus, there is a dimensionality reduction (GOODFELLOW; BENGIO; COURVILLE, 2016), which reduces the computational cost of the algorithms. According to Leyli-Abadi, Labiod and Nadif (2017), the representation learned by autoencoders usually presents superior quality for various tasks compared to other traditional approaches, such as Principal Component Analysis, Isometric Feature Mapping, Locally Linear Embedding, or Stochastic Neighbor Embedding.

Even though the autoencoder is composed of three components, according to Zhai *et al.* (2018), it is divided into two stages (Figure 1):

- **Encoding:** part of the neural network that compresses the input into a latent space representation (bottleneck). It can be represented by a function $f(\boldsymbol{d}_i)$ that maps the input $\boldsymbol{d}_i$ of the autoencoder to a low-dimensional space $\boldsymbol{z}_i$ (bottleneck);

- **Decoding:** part tries to reconstruct the input from the latent space representation $\boldsymbol{z}_i$. It can be represented by a function $g(\boldsymbol{z}_i)$ that reconstructs $\boldsymbol{d}_i$ by mapping $\boldsymbol{z}_i$ to the original space, generating the output $\boldsymbol{r}_i$.

Figure 1 – Illustration of a basic structure of an autoencoder with $\boldsymbol{d}_i$ input and $\boldsymbol{r}_i$ output. $\boldsymbol{z}_i$ represents the bottleneck.



Source: own authorship.

Given a training set with m documents $\mathscr{D} = \{\boldsymbol{d}_i | \boldsymbol{d}_i \in \mathbb{R}^n\}$, in which $1 \leq i \leq m$, n is the number of dimensions in the original space, and assuming that the functions of encoding and decoding are implemented by a neural network, the steps of an autoencoder can be formally defined by:

$$autoencoder = \begin{cases} \boldsymbol{z}_i = f(\Phi; \boldsymbol{d}_i) \\ \boldsymbol{r}_i = g(\Theta; \boldsymbol{z}_i) \end{cases}, \tag{2.3}$$

in which Φ are the weights and bias of neurons in the encoding neural network, and Θ are the weights and bias of neurons in the decoding neural network. Autoencoders learn these parameters from a backpropagation procedure, in which the neural network parameters are learned to minimize the loss function, for instance, the mean squared error between $\boldsymbol{d}_i$ and $\boldsymbol{r}_i$ defined in Equation 2.4 (ZHAI *et al.*, 2018).

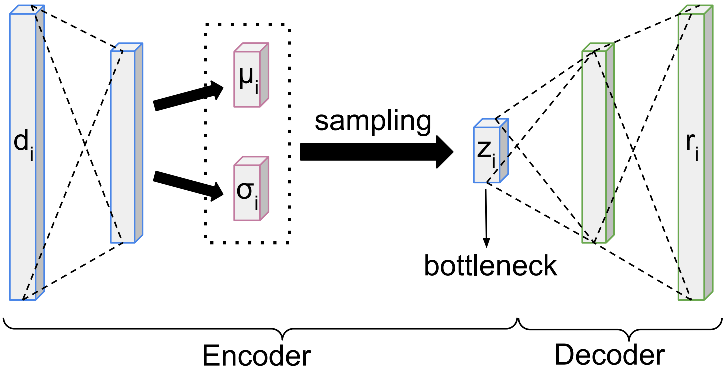
$$J(\Phi; \Theta) = \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{d}_i - \boldsymbol{r}_i\|^2 \tag{2.4}$$

With the advance of neural networks, other types of layers have been used in autoencoders, such as convolution, pooling, and recurrent layers. The literature shows that those three types of layers are commonly used for textual data (WANG *et al.*, 2019; XU; DURRETT, 2018; ZHANG *et al.*, 2017; LEYLI-ABADI; LABIOD; NADIF, 2017; BOWMAN *et al.*, 2016; LI; LUONG; JURAFSKY, 2015). We present the Variational Autoencoders (the focus of this work) in the next Section.

2.3 Variational Autoencoders

The main characteristic of the Variational Autoencoders (VAEs) is to bias learning through an informed prior distribution model (BOWMAN *et al.*, 2016; XU; DURRETT, 2018). Instead of sending the bottleneck straight to the decoder, the VAE encoder generates the bottleneck through sampling based on the previous informed probability distribution model. Therefore, VAE applies sampling considering the previously informed probability distribution model parameters (CHE; YANG; WANG, 2020; FELHI; ROUX; SEDDAH, 2021). For instance, if the model informed is Gaussian with mean 0 and standard deviation 1, the VAE will learn two representations (mean and standard deviation). The bottleneck will result from sampling the mean and standard deviation representations following the previously informed model distribution. The sampling step causes the VAE to generate a bottleneck that, when decoded, generates new data (generative model) similar to the training data since the mean and standard deviation representations were learned through these data (BOWMAN *et al.*, 2016). Still, the VAEs have another function to infer whether a given data belongs to the distribution model or not, i.e., the encoder of VAE works as a recognition model. Figure 2 shows an illustration of a VAE.

Figure 2 – Illustration of a basic structure of an variational autoencoder with $\boldsymbol{d}_i$ input and $\boldsymbol{r}_i$ output. $\boldsymbol{z}_i$ represents the bottleneck, $\boldsymbol{\mu}_i$ represent the mean learned and the $\boldsymbol{\sigma}_i$ represent the standard deviation learned.



Source: own authorship.

Formally, the VAE assumes a variable $\boldsymbol{z}_i$ that generates the data $\boldsymbol{d}_i$, by using the probability function:

$$p(\boldsymbol{z}_i|\boldsymbol{d}_i) = \frac{p(\boldsymbol{d}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)}{p(\boldsymbol{d}_i)}, \tag{2.5}$$

in which

$$p(\boldsymbol{d}_i) = \int p(\boldsymbol{d}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)dz \tag{2.6}$$

Integrals are computationally intractable. Thus, VAE uses variational inference, an approximation technique, to solve the limitation. First, VAE must approximate $p(\boldsymbol{z}_i|\boldsymbol{d}_i)$ to another distribution $q(\boldsymbol{z}_i|\boldsymbol{d}_i)$ that will be treatable. Then, the $q(\boldsymbol{z}_i|\boldsymbol{d}_i)$ parameters can be set to be similar to $p(\boldsymbol{z}_i|\boldsymbol{d}_i)$. To measure the difference between these distributions, VAE uses the divergence of Kullback-Leibler (KL). Finally, to optimize the marginal probability ($p(\boldsymbol{d}_i)$), we can use the log of the marginal probability, and $q(\boldsymbol{z}_i|\boldsymbol{d}_i)$ is approximated by $p(\boldsymbol{z}_i|\boldsymbol{d}_i)$ by KL. According to Xu and Durrett (2018), the log of the marginal probability can be described by:

$$\log p_{\Theta}(\boldsymbol{d}_i) = KL(q_{\Phi}(\boldsymbol{z}_i|\boldsymbol{d}_i)||p_{\Theta}(\boldsymbol{z}_i|\boldsymbol{d}_i)) + \mathscr{L}(\Theta,\Phi;\boldsymbol{d}_i), \tag{2.7}$$

in which

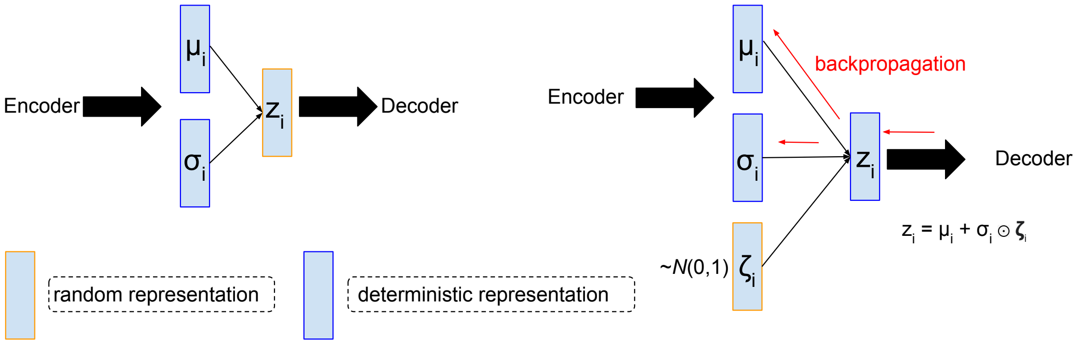
$$\mathscr{L}(\Theta,\Phi;\boldsymbol{d}_i) = \mathbb{E}_{q_{\Phi}(\boldsymbol{z}_i|\boldsymbol{d}_i)} \log p_{\Theta}(\boldsymbol{d}_i|\boldsymbol{z}_i) - KL(q_{\Phi}(\boldsymbol{z}_i|\boldsymbol{d}_i)||p_{\Theta}(\boldsymbol{z}_i)). \tag{2.8}$$

The first term of Equation 2.8 consists of the neural network reconstruction error. In the second term, we want to minimize the difference between the learned distribution $q_{\Phi}(\boldsymbol{z}_i|\boldsymbol{\lambda}_i)$ and $p_{\Theta}(\boldsymbol{z}_i)$ (prior knowledge). We replace the term $p_{\Theta}(\boldsymbol{z}_i)$ by $\mathscr{N}(\boldsymbol{z}_i;0,1)$, i.e., by a multivariate

Gaussian distribution with average 0 and standard deviation 1. Thus, we want to maximize Equation 2.8.

VAEs use backpropagation to learn neurons' weights when they are implemented by a neural network, such as the autoencoders. However, given the sampling process performed to generate the bottleneck, z_i becomes a random representation which makes the backpropagation process difficult since the weights of the encoder layers must be updated depending on the z_i weights. Therefore, a reparametrization trick must be applied to carry out the backpropagation. The procedure randomly samples a new representation ζ of a Gaussian unit and shift and scale ζ by the parameters learned in the distribution. For instance, if the parameters learned from the distribution model are the mean (μ) and the standard deviation (σ), the procedure is to shift ζ by μ and scale ζ by σ. Figure 3 illustrates the reparametrization trick.

Figure 3 – Illustration of the reparametrization trick used by VAE to apply the backpropagation and learn the weights of the neurons.



Source: own authorship.

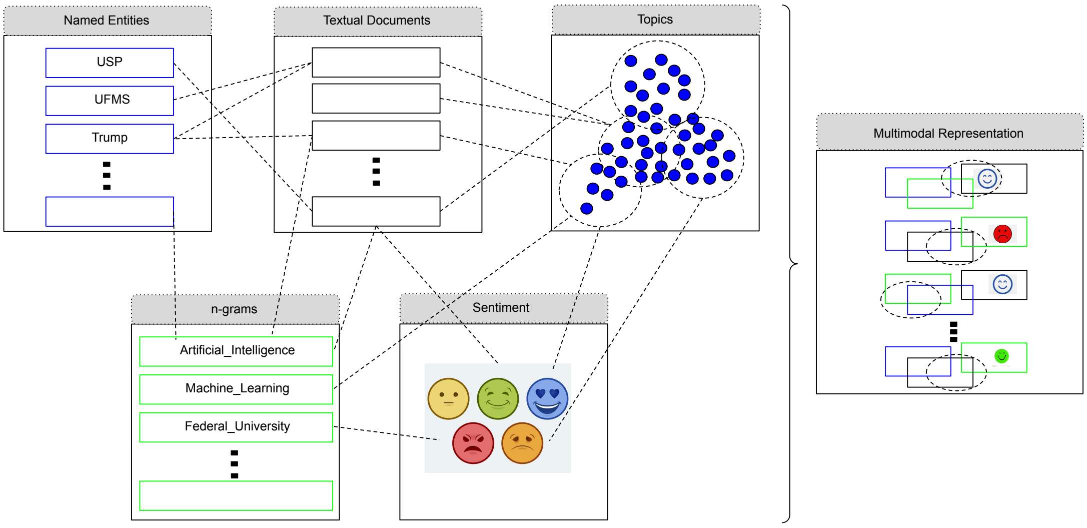
We highlight that traditionally VAE is applied to learning representation from unimodal data, i.e., data from a single type (XU; DURRETT, 2018; XU; TAN, 2019; CHE; YANG; WANG, 2020; FELHI; ROUX; SEDDAH, 2021). However, VAE can also be used as a multimodal representation learning (GUO; WANG; WANG, 2019), as presented in the next section.

2.4 Multimodal Learning

Even though most One-Class Learning studies for textual data consider only unimodal representations (MANEVITZ; YOUSEF, 2001; ZHUANG; DAI, 2006; MANEVITZ; YOUSEF, 2007; JUNIOR; ROSSI, 2017; KUMAR; RAVI, 2017b; KUMAR; RAVI, 2017a; GÔLO; MARCACINI; ROSSI, 2019; RUFF *et al.*, 2019; CICHOSZ, 2020; MAYALURU, 2020), this is not the only information to represent the texts since textual data is naturally heterogeneous and can be represented by other different types of information (BLIKSTEIN, 2013).

The information present in textual data can be named entities, X-grams, text sentiment, topics, and other representations (BLIKSTEIN, 2013). In addition, there is a scenario in which texts have external information that also characterizes textual documents, such as the latitude and longitude of events. Multimodal learning methods explore these different types of information to learn a more robust model than models generated from unimodal representations (GUO; WANG; WANG, 2019; LI; YANG; ZHANG, 2018). We present an illustration of multimodal learning in the textual data scenario in Figure 4.

Figure 4 – Multimodality scenario for textual data. named entities, topics, *n*-grams and sentiment are modalities of textual documents. Multimodal representation represents the combination of modalities of each textual document.



Source: own authorship.

The heterogeneity of texts is one of the main motivations for using multimodal representation learning (PENG; QI, 2019). Furthermore, the different modalities can be supplementary or complementary. Thus, a multimodal representation contains more information than unimodal representations (GUO; WANG; WANG, 2019), improving the machine learning process (LI; YANG; ZHANG, 2018).

To combine the modalities, we need to fuse them. In multimodal learning, the most common types of fusion are early and late fusion (KATSAGGELOS; BAHAADINI; MOLINA, 2015). The early fusion can be represented by methods that combine modalities before the machine learning process. For instance, early fusion can be done using simple operators like concatenation, addition, subtraction, multiplication, and averaging. Formally, considering the set of modalities $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3 \cdots, \boldsymbol{\psi}_a\}$, ω as the operator, and a as the number of modalities, $\boldsymbol{\Omega}_i$ represents the representation fused by early fusion, according to Equation 2.9 (LIU *et al.*, 2018).

$$\boldsymbol{\Omega}_i = \omega(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3 \cdots, \boldsymbol{\psi}_a) \tag{2.9}$$

An advantage of early fusion is using only one representation of the data in the learning process. On the other hand, early fusion has some challenges, such as dealing with different dimensions, scales, and levels of importance of each modality. However, to deal with these challenges, it is also possible to use more complex strategies than the simple operator, such as based on multimodal representation learning from neural networks (GAO *et al.*, 2020).

In late fusion, a machine learning algorithm is used in each modality. Then, the outputs of these algorithms are integrated to obtain a Final Decision, i.e., if the document will belong to the interest class or not. Formally, considering an Δ method of late fusion to obtain the final decision, and that an algorithm Γ_b is used in each modality b, in which $\{b = 1, \cdots, a\}$), according to Liu *et al.* (2018), Equation 2.10 presents the late fusion:

$$\text{Final Decision} = \Delta(\Gamma_1(\boldsymbol{\psi}_1), \Gamma_2(\boldsymbol{\psi}_2), \Gamma_3(\boldsymbol{\psi}_3) \cdots, \Gamma_a(\boldsymbol{\psi}_a)). \tag{2.10}$$

A disadvantage of this fusion is the challenge of combining the outputs of different models and defining levels of importance for each model in the final decision. Therefore, this work focuses on early fusion using multimodal representation learning through neural networks. In this way, we can explore the advantages of early fusion without the challenge of fuse representations with different dimensions since we use neural networks to fuse and learn a representation. After learning a robust representation for the texts through multimodal representation learning, the next step is to apply the One-Class Learning explained in the next Section 2.5.2.

2.5 Machine Learning

In the context of automatic text classification, most algorithms learn a surface capable of separating documents that belong to distinct classes, i.e., learn a decision surface. This surface is how the algorithm will assign a document to one of the pre-established classes, i.e., a classification model (ROSSI, 2016; TAN; STEINBACH; KUMAR, 2013; SEBASTIANI, 2002). This model is capable of classifying unlabeled texts, being able to perform the automatic text classification. It is noteworthy that the algorithm uses labeled documents to learn the decision surface during the training stage. One of the most used types of machine learning in literature and real applications is Multi-Class Learning (WEISS; INDURKHYA; ZHANG, 2015; WITTEN; FRANK; HALL, 2011; SEBASTIANI, 2002) which is explained in the next Section.

2.5.1 Multi-Class Learning

In Multi-Class Learning, the algorithm can learn two or more classes from labeled textual documents. In this way, classifiers assign one of the learned classes to new documents (WEISS; INDURKHYA; ZHANG, 2015; WITTEN; FRANK; HALL, 2011; SEBASTIANI, 2002).

Formally, we define a Multi-Class Learning text classifier as a function $f : \mathscr{D} \rightarrow \mathscr{C}$ that maps a textual document $\boldsymbol{d}_i \in \mathscr{D}$ to a set of k classes $\mathscr{C} = \{c_1, ..., c_k\}$, in which $c \in \mathscr{C}$ indicates classes of a given domain. Moreover, the text collection $\mathscr{D} \in \mathbb{R}^n$ is represented in an n-

dimensional feature space. Thus, the text classification through supervised Multi-Class Learning aims to learn a function f^* from a training set $\mathscr{H} = \{\, (\boldsymbol{d}_i, c_i) \mid \boldsymbol{d}_i \in \mathbb{R}^n,\ c_i \in \{c_1, ..., c_k\}\}$, in which $\boldsymbol{d}_i$ indicates the i-th document and c_i indicates one of the classes, in which f^* approximates the true (and unknown) mapping function f.

Despite being a widely used technique, Multi-Class Learning has some limitations that can impair the automatic classification's practical application or performance. Two limitations regard the labeling of examples: (i) when the user does not label examples for all domain classes; and (ii) when a new domain class comes up. Then, the classification model may wrongly label a new example of the class that came up or that the user does not know, as it will assign one of the learned classes to the new example. The second limitation concerns the labeling of a significant number of examples of each class to perform learning.

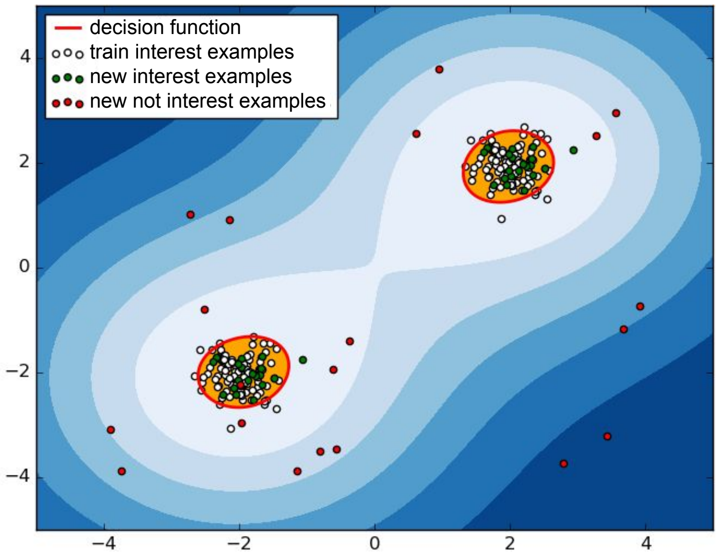
Another problem in Multi-Class Learning is class imbalance. This problem happens when there are many documents from one or a few classes and a few from the other classes. The classifier can bias the classification to the majority(ies) class(es) in this scenario. Furthermore, there are situations in which the user is only interested in one class. In this case, when using multi-class learning, the user will have to provide examples of all the other classes even he/she has no interest in them. One of the types of learning that can mitigate the Multi-class learning problems is One-Class Learning (ALAM *et al.*, 2020; RUFF *et al.*, 2018; KEMMLER *et al.*, 2013; KHAN; MADDEN, 2009; TAX, 2001), which will be explained in the next section.

2.5.2 One-Class Learning

In One-Class Learning (OCL), the user defines an interest class, and the OCL algorithm learns a classification model considering only documents of the interest class. Thus, the algorithm classifies a new document belonging to the interest class or not. Therefore, OCL is more challenging than Multi-Class Learning since there are no counterexamples or documents from different classes are provided to learn a decision surface. Recent studies indicate that OCL is a competitive classification strategy, with the advantage of reducing the user labeling effort (ALAM *et al.*, 2020; RUFF *et al.*, 2018; FERNÁNDEZ *et al.*, 2018; THEISSLER, 2017; HEMPSTALK; FRANK, 2008; ZHUANG; DAI, 2006).

Generally, OCL algorithms generate a value indicating how prone the document belongs to the interest class, and this value is compared against a threshold. The threshold can be generated by a decision function of the interest class or manually set (KEMMLER *et al.*, 2013). If the value is greater than or equal to the threshold, the document will be defined as belonging to the interest class, or it will be defined as not belonging to the interest class, otherwise (Equation 2.11). An example of a decision function is shown in Figure 5. All white points are used in the train. Red and green points will be classified, and the red line represents the decision function. Thus, all points within the yellow area have values greater than or equal to the threshold generated by the decision function and are classified as belonging to the interest class and the other points not.

Figure 5 – Illustration of a decision function for OCL. White points are used to train. Green points are new examples of the interest class and red points are new examples of the non-interest class. Points within the yellow area will be assigned to the interest class. Points outside the yellow area will not be assigned to the interest class.



Source: Adapted from (PRONOBIS, 2021).

Formally, we define an OCL text classifier as a function $g : \mathscr{D} \rightarrow \mathscr{Y}$ that maps a textual document $\boldsymbol{d}_i \in \mathscr{D}$, with $D \in \mathbb{R}^n$, for a value $y_i \in \mathscr{Y}$, indicating how close document $\boldsymbol{d}_i$ is to belong to the interest class. Thus, the text classification through OCL aims to learn a function g^* from a training set $\mathscr{H} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\}$ which approximates the unknown mapping function g. Then, the classification through OCL is done according to Equation 2.11.

$$\text{Class of } \boldsymbol{d}_i = \begin{cases} y_i \geq \mathit{threshold} \rightarrow \text{Interest} \\ y_i < \mathit{threshold} \rightarrow \text{Not Interest} \end{cases} \tag{2.11}$$

There are several OCL algorithms, such as similarity-based, probabilistic, and statistics (GÔLO; MARCACINI; ROSSI, 2019; KHAN; MADDEN, 2009). Similarity-based algorithms consider the similarity of a new document to the documents in the training set to generate the value that indicates how prone this document is to belong to the interest class. This category includes algorithms based on nearest neighbors and clustering algorithms (TAN; STEINBACH; KUMAR, 2013). On the other hand, probabilistic algorithms aim to generate a probability distribution model for the examples of the interest class and subsequently verify the probability of a new document belonging to the interest class. Probabilistic algorithms used for OCL are Naive Bayes (AGGARWAL, 2018a), Multinomial Naive Bayes, and Gaussian Mixture Models (GÔLO; MARCACINI; ROSSI, 2019). Regarding statistical algorithms, the One-Class

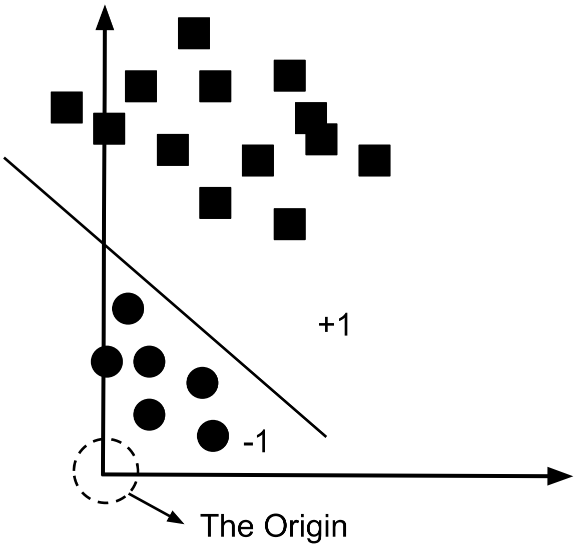
Support Vector Machine (OCSVM) algorithm (SHIN; EOM; KIM, 2005; TAX; DUIN, 2004; MANEVITZ; YOUSEF, 2001) is generally used.

This dissertation focus on learning a representation for any one-class algorithm. However, as it focuses on representations, we chose only a one-class algorithm. Therefore, we chose the OCSVM (SHIN; EOM; KIM, 2005; TAX; DUIN, 2004; MANEVITZ; YOUSEF, 2001), one of the state-of-the-art algorithms for classification through OCL (SHIN; EOM; KIM, 2005; TAX; DUIN, 2004; MANEVITZ; YOUSEF, 2001). We present the OCSVM in the next section.

2.5.3 One Class Support Vector Machine

The OCSVM is based on the Support Vector Machine (SVM) (TAN; STEINBACH; KUMAR, 2013). The SVM for Multi-Class Learning aims to generate a hyperplane of maximum separation margin between pairs of classes. The most traditional adaptation of the SVM to the OCL consists of generating fictitious examples close to the origin that correspond to counterexamples of the interest class (Figure 6) to apply a maximum separation hyperplane (SCHÖLKOPF *et al.*, 2001).

Figure 6 – Illustration of the fictitious examples generated close to the origin of the plane so that the maximum separation hyperplane is generated in the OCSVM proposed by (SCHÖLKOPF *et al.*, 2001). $+1$ means the interest class and -1 the opposite.



Adapted from (ALASHWAL; DERIS; OTHMAN, 2006).

According to Schölkopf *et al.* (2001), OCSVM will separate the data from the origin with a maximum margin. Formally, OCSVM uses Equation 2.12 to create the maximum separation hyperplane between the interest class and the origin examples.

$$\min_{z,\varepsilon,\rho} \frac{1}{2} \parallel z \parallel^2 + \frac{1}{\nu \cdot |\mathscr{D}|} \sum_{d_i \in \mathscr{D}} \varepsilon_{d_i} - \rho, \tag{2.12}$$

subject to:

$$(z \cdot \varphi(\boldsymbol{d}_i)) \geq \rho - \varepsilon_{d_i}, \varepsilon_{d_i} \geq 0, \tag{2.13}$$

in which z are the coefficients of the separation hyperplane, $\nu \in [0,1)$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors (i.e., if $\nu = 0.2$, at most 20% of the training samples are allowed to beyond the separation hyperplane, and also at least 20% of the training example will be used as support vectors), ε_{d_i} is the distance from a document $\boldsymbol{d}_i$ to the separation hyperplane, ρ is the classification error threshold, and $\varphi(\boldsymbol{d}_i)$ is a kernel function to map the documents into a linearly separable space.

After creating the hyperplane, given a document $\boldsymbol{d}_i$, the function $f(\boldsymbol{d}_i)$ indicates whether or not $\boldsymbol{d}_i$ belongs to the interest class. If $f(\boldsymbol{d}_i)$ returns $+1$, this means that the example is on the side of the hyperplane where the documents of the interest class are located, i.e., $\boldsymbol{d}_i$ will be classified as being of the interest class. On the other hand, the return -1 means that the example is on the side of the hyperplane where the fictitious examples are and will be classified as not belonging to the interest class. The function $f(\boldsymbol{d}_i)$ is given by Equation 2.14.

$$f(\boldsymbol{d}_i) = sgn(z \cdot \varphi(\boldsymbol{d}_i) - \rho), \tag{2.14}$$

in which $sgn()$ is a signal function that returns -1 when $z \cdot \varphi(\boldsymbol{d}_i) - \rho$ is negative and returns $+1$ when greater than or equal to 0.

There are other OCSVM approaches, such as Tax and Duin (2004), in which the OCSVM aims to generate the smallest hypersphere (given a radius and center) that involves the examples of the interest class. Examples allocated to the edge of the hypersphere are the support vectors. According to Tax and Duin (2004), the objective of OCSVM is to find a decision function capable of involving the textual documents of the interest class. Thus, the OCSVM wants to find the minimum volume hypersphere involving the training set documents according to Equation 2.15.

$$\boldsymbol{\mu}_{(c)} = \arg\min_{\boldsymbol{\mu} \in U} \max_{1 \leq i \leq m} ||\varphi(\boldsymbol{d}_i) - \boldsymbol{\mu}||^2, \tag{2.15}$$

in which $\boldsymbol{\mu} \in U$ is a possible center in the feature space U associated with the function kernel φ, $\varphi(\boldsymbol{d}_i)$ maps $\boldsymbol{d}_i$ into another feature space defined according to the kernel chosen, and $\boldsymbol{\mu}_{(c)}$ is the center of the hypersphere in which the highest distance between $\varphi(\boldsymbol{d}_i)$ to $\boldsymbol{\mu}_{(c)}$ is minimal.

OCSVM classifies a document as belonging to the interest class if its distance from the center is less than the radius r of the hypersphere (Considering Equation 2.11, $y_i = dist(\varphi(\boldsymbol{d}_i), \boldsymbol{\mu}_{(c)})$, and $threshold = r$). In general, we hope that the hypersphere is not too large so that the false positive rate does not increase. Therefore, we add a regularizer ν to accept a certain level of violation of the hypersphere decision function. Thus, we desire to minimize the square of the hypersphere radius and the number of violations. Formally, the minimization function is given by Equations 2.16 and 2.17, in which ε_{d_i} is the external distance between $\varphi(d_i)$ and the surface of the hypersphere, and $\nu \in (0,1]$ defines the smoothness level of the hypersphere volume.

$$\min_{\mu,\varphi,r} \quad r^2 + \frac{1}{m}\sum_{i=1}^{m}\frac{\varepsilon_{\boldsymbol{d}_i}}{\nu}, \tag{2.16}$$

$$\begin{gathered} \|\varphi(\boldsymbol{d}_i) - \boldsymbol{\mu}_{(c)}\|^2 \leq r^2 + \varepsilon_{\boldsymbol{d}_i} \\ \forall i = 1, \ldots, m. \end{gathered} \tag{2.17}$$

The OCSVM proposed in Tax and Duin (2004) has the advantage of do not need to generate fictitious examples. Moreover, there is no guarantee that the examples close to the origin truly characterize the counterexamples. Therefore, this dissertation uses the OCSVM version of Tax and Duin (2004). It is noteworthy that this method is sensitive to noise, and it is important to use strategies to learn a feature space that adequately represents the interest class.

2.6 Concluding Remarks

This chapter presented the main theoretical foundations of text representation and the limitations of more traditional representations, such as bag-of-words. Alternatives to mitigate the high dimensionality and lack of semantics in bag-of-words were also presented, such as the context-dependent language models based on neural networks.

The concepts about autoencoder and variational autoencoder, which is the basis for the representation learning techniques proposed in this dissertation, were detailed. Ways to generate representations considering multiple modalities that can be extracted from text collections were also presented. It can be highlighted that the use of multimodal representations is also the core of the proposed approaches for representation learning.

By the end, the representations learned by the proposed approaches will feed One-Class Learning algorithms. In order to explain this type of learning, we first present the traditional Multi-Class Learning and its limitation. Then, One-Class Learning was characterized, and the details of the algorithm adopted in this dissertation, OCSVM, were also presented.

The papers containing the results obtained by the proposed multimodal representation learning methods applied into One-Class Learning tasks for different application domains are presented in the next chapters.

CHAPTER

3

LEARNING TEXTUAL REPRESENTATIONS FROM MULTIPLE MODALITIES TO DETECT FAKE NEWS THROUGH ONE-CLASS LEARNING

3.1 Introduction

Fake news intentionally disseminated as real news to spread disinformation has become part of everyday life (CAMISANI-CALZOLARI, 2018; LYONS; MEROLA; REIFLER, 2020). Fake news constantly evolves in writing style and presentation, intending to deceive a target audience (CHOUDHARY; ARORA, 2021; CONROY; RUBIN; CHEN, 2015). Even for humans, determining the news veracity is a challenging task (GREIFENEDER *et al.*, 2021; SHU *et al.*, 2017; FRANK *et al.*, 2004). This happens mainly because fake news strongly appeals to our emotions and ideological postures (KUMARI *et al.*, 2021). As accepted, fake news become challenging to be corrected and tend to continue acting on the opinion of social groups (CAMISANI-CALZOLARI, 2018; GREIFENEDER *et al.*, 2021). Thus, the automatic detection of fake news is essential for society (GREIFENEDER *et al.*, 2021; SHU *et al.*, 2017)

Different news classification methods to discriminate between real and fake content have been proposed to minimize the effects of disinformation (ZHANG; GHORBANI, 2020). The most common way to deal with this problem is to characterize the fake news detection as a binary or multiclass classification problem (CHOUDHARY; ARORA, 2021), which usually requires a significant set of previously labeled news. However, covering the broad spectrum of news topics, sources, and levels of falsehood is a challenge (BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019). Furthermore, labeling a large volume of news requires a high human cost for fact-checking.

One-Class Learning (OCL) has recently been investigated as a promising approach to detect fake news in order to decrease the labeling effort (WANG; BAH; HAMMAD, 2019; KHAN; MADDEN, 2014a; FAUSTINI; COVÕES, 2019). OCL algorithms use only labeled examples of the interest class (e.g., fake news) as input to the learning algorithm, contributing to scenarios where it is difficult to label a large number of examples. Also, OCL algorithms are more robust to class imbalance, which is the case of fake news since there is significantly more real news than fake news available on news sources (ALAM *et al.*, 2020; FERNÁNDEZ *et al.*, 2018).

OCL algorithms are sensitive to the representation of textual data (BEKKER; DAVIS, 2020). Traditional document classification methods generally represent news using the Bag-of-Words (BoW) model (SILVA *et al.*, 2020). Due to the absence of semantics and assuming independence among words (AGGARWAL, 2018a), BoW has often been criticized for news classification (ZHANG; GHORBANI, 2020). Alternatively, Linguistic Inquiry and Word Count (LIWC) (SINGH; GHOSH; SONAGARA, 2021) generates representations that capture syntactic characteristics of news to the detriment of semantic features. Another alternative to represent news texts is the language model methods based on word embeddings, such as Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN *et al.*, 2019) and its derivatives. These approaches consider the syntactic and semantic relations between words to generate the news representations (OTTER; MEDINA; KALITA, 2020). In addition to the text representations, it is essential to note that fake news is more frequent in some well-defined subjects such as politics, pandemics, society, celebrities, science, religion, and economics (SILVA *et al.*, 2020). Thus, considering this topic information in the text representation can be promising to detect fake news (SHARMA; SOMAYAJI; JAPKOWICZ, 2018; KRAWCZYK; WOŹNIAK; CYGANEK, 2014).

We investigated the representation learning from multiple modalities for fake news detection in OCL scenarios. The main goal is to learn a representation model that considers the news's syntactic, semantic, and topic information. In order to do so, we proposed a multimodal representation learning method based on Variational Autoencoders (VAEs). VAEs are generative models that aim to obtain a new latent space (e.g., embeddings) that preserves the main properties of the original textual data (BOWMAN *et al.*, 2016; XU; DURRETT, 2018). Traditionally, VAEs have explored a single modality during representation learning. Thus, we propose and develop the Multimodal Variational Autoencoder for fake news detection (MVAE-FakeNews), which learns a representation from language and visual modalities. The language modality is the syntactic and semantic information from word embeddings obtained through a derivation of BERT called DistilBERT Multilingual (DBERTML) language model. We build the visual modality based on the density information that can be interpreted as the topics about fake news. Finally, we use the MVAE-FakeNews representations to train an OCL algorithm One-Class Support Vector Machine considering the fake news as interest class.

We carried out an experimental evaluation using three news collections to compare our MVAE-FakeNews with nine other textual representation models. MVAE-FakeNews got the best F_1-Scores for the interest class in most classification scenarios. Statistical analysis among all methods reveals a significant difference in the performance of MVAE-FakeNews about representations based on a single modality, such as BoW, word embeddings, topics, and syntactic characteristics (LIWC). Also, another statistical analysis between pairs of methods reveals a significant difference from the performance of the MVAE-FakeNews compared to all other methods. With only 3% of fake news labeled, MVAE-FakeNews achieves results comparable or higher to other methods when considering 10% of fake news labeled. Thus, MVAE-FakeNews proved to be a competitive alternative to the state-of-the-art and promising to practical scenarios because it requires less effort from domain experts.

The rest of this chapter is divided as follows. Section 3.2 presents related work. Section 3.3 presents the concepts and details of the proposed method and OCL. Section 3.4 presents the experimental settings used, results, and discussions. Lastly, Section 3.5 presents the conclusions and future work directions.

The proposed method and experimental results presented in this chapter were published in a paper titled "*Learning Textual Representations from Multiple Modalities to Detect Fake News Through One-Class Learning*" (GÔLO *et al.*, 2021) in the Brazilian Symposium on Multimedia and the Web (Webmedia 2021)[1].

3.2 Related Work

Approaches proposed in the literature for the news representation analyze lexical, syntactic, and semantic aspects of the publication's content to detect falsehood (CHOUDHARY; ARORA, 2021). Among the syntactic characteristics, there are the counting of grammatical classes (subjects, verbs, adjectives), presence and frequency of patterns, or specific expressions of the language and legibility analysis of the text (SILVA *et al.*, 2020). Lexical approaches analyze words used in the textual content that indicate humor, abbreviations, or temporal expressions that are informative to determine the veracity of the information (RASHKIN *et al.*, 2017). Semantic characteristics are related to analyzing the news's language patterns, structure, and meanings (ZHANG; GHORBANI, 2020).

Literature studies that carry out fake news detection, in general, represent news in a structured way using traditional models such as BoW and *n*-grams. However, such models are not powerful enough to extract semantic information from the news (ZHANG; GHORBANI, 2020). Moreover, other methods such as LIWC explore emotional, cognitive, and structural components present in the news written verbal language. Rashkin *et al.* (2017) analyzes news with the LIWC tool. The study suggests that people who write fake news use first and second-person pronouns

[1] <https://dl.acm.org/doi/abs/10.1145/3470482.3479634>.

and exaggerated and vague words in misleading news. In addition, reliable news writers tend to be impartial, using assertive words and numeric data more often.

Some studies in literature also propose the use of textual and contextual characteristics of news to enrich traditional representation models. Silva *et al.* (2020) proposed representation models, combining BoW and linguistic features for supervised scenarios involving more than two-thirds of previously labeled news. Choudhary and Arora (2021) proposes a deep learning model based on linguistic characteristics to detect fake news. Syntactic, grammatical, legibility, and sentiment expressed in the text are extracted and combined, being used as main parameters to train a sequential neural network model in supervised classification scenarios.

Singh, Ghosh and Sonagara (2021) and Khattar *et al.* (2019) propose multimodal representation models that combine text information and images associated with the news. Singh, Ghosh and Sonagara (2021) divided the characteristics into four categories: content, organization, emotiveness, and manipulation. They extract 124 textual features for each news, 81 of them with LIWC, along with 43 image characteristics. The authors use 90% of news previously labeled. Khattar *et al.* (2019) proposes a variational autoencoder considering multiple modalities for binary classification problems. The goal is to learn a unified representation model, coding the text and image modalities of the news. Khattar *et al.* (2019) proposed this method for supervised scenarios.

Faustini and Covões (2019) proposes the DCDistanceOCC, an OCL algorithm to identify fake content to minimize real news labeling efforts. For each news, linguistic characteristics are extracted, such as the proportion of capital letters, number of words per sentence, and and message sentiment. The authors create a vector from the sum of the characteristic vectors of all objects (class vector). The algorithm calculates the distance between the object and the class vector and classifies the object as being of interest or not according to a threshold. The authors executed the algorithm using cross-validation, with 90% fake news to train the model. The approach reaches 67% of $F_1 - Score$ for the Fake.Br dataset (SILVA *et al.*, 2020).

In the literature, we observe that most methods explore a single textual modality (SINGH; GHOSH; SONAGARA, 2021; HORNE; ADALI, 2017) or still need to label fake news and a wide spectrum of real news to build a classification model (BONDIELLI; MARCELLONI, 2019; MEEL; VISHWAKARMA, 2019). Studies that use multimodality, for example, require images associated with the news, which are not always present in all sources. Existing initiatives involving OCL generally use bag-of-words representations or representations based on linguistic categories. Thus, a lack of studies that used other concepts to represent news, such as semantics, was not noticed. Also, the representations are usually unimodal, and those that consider multimodality consider types of data that cannot be present in any news, e.g., images. Thus, we proposed a multimodal representation learning approach to consider different modalities presented in any new collection. The details of the proposed approach are presented in the next section.

3.3 Multimodal Variational Autoencoder for Fake News

Textual data, such as fake news, are naturally multimodal (PENG; QI, 2019) and can be represented in different ways. Moreover, the information embedded in the textual data can come from several methods, such as named entities, *n*-grams, text sentiment, topics, and linguistic categories (BLIKSTEIN, 2013; PENNEBAKER *et al.*, 2015).

Multimodal learning aims to learn models based on multiple representations, which can be complementary or supplementary, generally being more robust than unimodal models (GUO; WANG; WANG, 2019). Multimodal learning contributes to machine learning algorithms' performance and allows us to combine the data in different forms (LI; YANG; ZHANG, 2018).

Our proposal uses embeddings from a DBERTML language model and the density information extracted from these embeddings as modalities. We choose DBERTML embeddings because they allow the capture of syntactic and semantic characteristics, unlike other traditional models like BoW. Moreover, considering the news scenario with high-density regions representing well-defined topics such as politics, society, celebrities, science, religion, economics, and pandemic, we chose the density information as a visual modality since it is able to capture this spatial distribution of topics. While studies in literature frequently use BERT embeddings on text representation (KALIYAR; GOSWAMI; NARANG, 2021), the density information extracted from these embeddings is still underexplored. In the next sections, we present details about the extraction of the density modality, our multimodal variational autoencoder, and its use in one-class learning tasks to detect fake news.

3.3.1 *Density Information Representation*

One way to obtain the density information of fake news documents is to apply a clustering algorithm and use a statistical technique that calculates the cluster quality structure (SHARMA; SOMAYAJI; JAPKOWICZ, 2018; KRAWCZYK; WOŹNIAK; CYGANEK, 2014).

Consider a clustering with k clusters, i.e., $\mathscr{D} = C_1 \cup C_2, \cup \cdots \cup C_k$, in which C_j is a cluster of documents, and $2 \leq k < m$. Then, we apply the silhouette coefficient (ROUSSEEUW, 1987) in order to measure if a news belongs to a single topic or contains mixed topics. The silhouette for a news $\boldsymbol{d}_i$ represented by the embeddings of DBERTML $\boldsymbol{\lambda}_i$ assigned to a cluster C_j is given by:

$$s(\boldsymbol{\lambda}_i, k) = \frac{\beta(\boldsymbol{\lambda}_i) - \alpha(\boldsymbol{\lambda}_i)}{\max(\alpha(\boldsymbol{\lambda}_i), \beta(\boldsymbol{\lambda}_i))}, \tag{3.1}$$

in which $\alpha(\boldsymbol{\lambda}_i)$ is the average distance of $\boldsymbol{\lambda}_i$ to all documents of cluster C_j, and $\beta(\boldsymbol{\lambda}_i)$ defines the average distance of the document $\boldsymbol{\lambda}_i$ to all documents of the closest cluster C_o, $o \neq j$. $\alpha(\boldsymbol{\lambda}_i)$ and $\beta(\boldsymbol{\lambda}_i)$ are given by Equations 3.2 and 3.3 respectively.

$$\alpha(\boldsymbol{\lambda}_i) = \frac{1}{|C_j|-1} \sum_{\boldsymbol{\lambda}_l \in C_j, \boldsymbol{\lambda}_i \neq \boldsymbol{\lambda}_l} dist(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_l) \tag{3.2}$$

$$\beta(\boldsymbol{\lambda}_i) = \min_{o \neq j} \frac{1}{C_o} \sum_{\boldsymbol{\lambda}_n \in C_o} dist(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_n) \tag{3.3}$$

The silhouette values range from -1 to +1. A high value indicates that a news is well matched to its cluster and weakly matched to neighboring clusters, i.e., a news tends to belong to a single topic. We represent fake news density information by concatenating silhouette coefficient values considering each document in different clustering settings. For instance, given u different clustering settings, i.e., the clustering settings have different values of k, we perform the density modality by:

$$\boldsymbol{\delta}_i = \{s(\boldsymbol{\lambda}_i, k_1), s(\boldsymbol{\lambda}_i, k_2), \ldots, s(\boldsymbol{\lambda}_i, k_u)\} \tag{3.4}$$

in which $s(\boldsymbol{\lambda}_i, k_j)$ is the silhouette of $\boldsymbol{\lambda}_i$ in cluster setting with k_j clusters. Figure 7 presents an illustration of how we generate the density information representation.

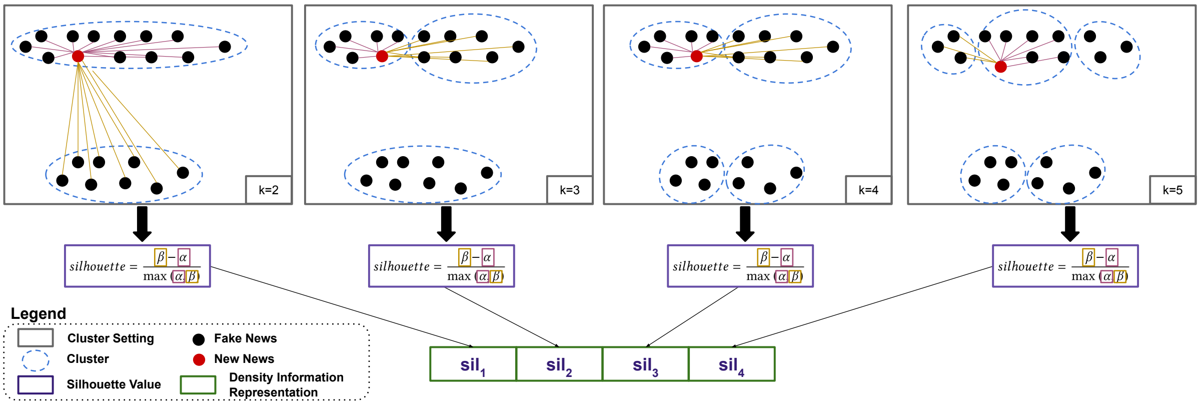

k=2
k=3
k=4
k=5
silhouette = (β−α) / max (α,β)
silhouette = (β−α) / max (α,β)
silhouette = (β−α) / max (α,β)
silhouette = (β−α) / max (α,β)
Legend
Cluster Setting
Cluster
Silhouette Value
Fake News
New News
Density Information Representation
sil1
sil2
sil3
sil4

Figure 7 – Illustration of a Density Information Representation.

3.3.2 Variational Autoencoder

Variational Autoencoders are a specific type of autoencoder (AE) (LIU *et al.*, 2017; WANG; BAH; HAMMAD, 2019). AE are neural networks that we can use to learn representations. In this paper, we refer to an AE as a dense AE, i.e., a fully connected feed-forward neural network with o layers, $o \geq 2$, in which the i-th and $(o-i)$-th layers have the same number of neurons (LIU *et al.*, 2017). We can divide the AE into the encoding and decoding steps, in which the first half of the layers encode, and the other half of the layers decode the input (ZHAI *et al.*, 2018). First, the encoding compresses the input $\boldsymbol{\lambda}_i$ into a representation of latent space z_i, and the function $f(\cdot)$ represents this step. Next, decoding attempts to reconstruct the input using z_i in the $\boldsymbol{\lambda}_i'$ output, and the function $g(\cdot)$ represents this step.

Considering that a neural network implements the functions $f(\cdot)$ and $g(\cdot)$, and that $\boldsymbol{\lambda}_i$ is a representation of the document $\boldsymbol{d}_i$. An autoencoder can be defined by:

$$autoencoder = \begin{cases} z_i = f(\Phi; \boldsymbol{\lambda}_i) \\ \boldsymbol{\lambda}_i' = g(\Theta; z_i) \end{cases} \tag{3.5}$$

in which Φ are the weights and bias of neurons in the encoding neural network, Θ are the weights and bias of neurons in the decoding neural network. The AE learns these parameters to minimize the error of reconstruction of the neural network, for instance, the mean squared error between $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_i'$.

There are variations of AEs, which aim to force the learned representations to assume useful properties for the application scenario (WANG *et al.*, 2019). One variant is the Variational Autoencoders (VAEs). The learning in a VAE is regularised in order to obtain parameters of a distribution (e.g., normal). Thus, VAEs are able to generate new data similar to the train set (BOWMAN *et al.*, 2016; XU; DURRETT, 2018). Thus, after the training step, when we send the data to the VAE, it has new data as an output. Figure 8 presents a VAE.

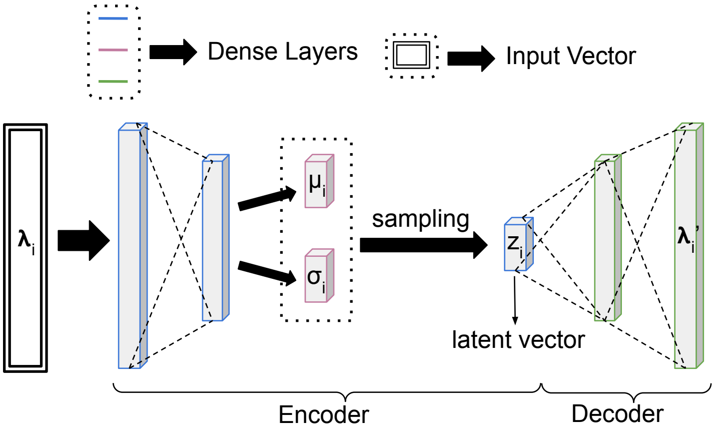


Figure 8 – Illustration of a Variational Autoencoder.

Formally, the VAE assumes a variable z_i that generates the data $\boldsymbol{\lambda}_i$, by using the probability function:

$$p(z_i|\boldsymbol{\lambda}_i) = \frac{p(\boldsymbol{\lambda}_i|z_i)p(z_i)}{p(\boldsymbol{\lambda}_i)}, \tag{3.6}$$

in which

$$p(\boldsymbol{\lambda}_i) = \int p(\boldsymbol{\lambda}_i|z_i)p(z_i)dz \tag{3.7}$$

VAE approximates $p(z_i|\boldsymbol{\lambda}_i)$ to another treatable distribution $q(z_i|\boldsymbol{\lambda}_i)$ using the Kullback-Leibler (KL) divergence to measure the divergence between distributions. To optimize the marginal likelihood ($p(\boldsymbol{\lambda}_i)$), you can use the log of the marginal likelihood (XU; DURRETT, 2018):

$$\log p_\Theta(\boldsymbol{\lambda}_i) = KL(q_\Phi(z_i|\boldsymbol{\lambda}_i)||p_\Theta(z_i|\boldsymbol{\lambda}_i)) + \mathscr{L}(\Theta,\Phi;\boldsymbol{\lambda}_i), \tag{3.8}$$

in which

$$\mathscr{L}(\Theta,\Phi;\boldsymbol{\lambda}_i) = \mathbb{E}_{q_\Phi(z_i|\boldsymbol{\lambda}_i)} \log p_\Theta(\boldsymbol{\lambda}_i|z_i) - KL(q_\Phi(z_i|\boldsymbol{\lambda}_i)||p_\Theta(z_i)). \tag{3.9}$$

The first term of Equation 2.8 consists of the neural network reconstruction error. In the second term, we want to minimize the difference between the learned distribution $q_\Phi(z_i|\boldsymbol{\lambda}_i)$ and $p_\Theta(z_i)$ (prior knowledge). We replace the term $p_\Theta(z_i)$ by $\mathscr{N}(z_i;0,1)$, i.e., by a multivariate Gaussian distribution with average 0 and standard deviation 1. Thus, we want to maximize Equation 2.8.

The MVAE-FakeNews extends VAEs to two modalities. Figure 9 presents an illustration of the Multimodal VAE neural architecture. MVAE-FakeNews is based on multimodal learning and a Variational Autoencoder and is analogous to VAE. However, it will also calculate the error of reconstructing the density information. Therefore, MVAE-FakeNews aims to maximize Equation 3.10. The first term calculates two reconstruction errors: embedding ($\boldsymbol{\lambda}_i$) and density information ($\boldsymbol{\delta}_i$). The second term wants to approximate the learned distribution $q_\Phi(z_i|\boldsymbol{\lambda}_i,\boldsymbol{\delta}_i)$ from $p_\Theta(z_i)$, as in VAE.

$$\mathscr{L}(\Theta,\Phi,\boldsymbol{\lambda}_i,\boldsymbol{\delta}_i) = \mathbb{E}_{q_\Phi(z_i|\boldsymbol{\lambda}_i,\boldsymbol{\delta}_i)} \log p_\Theta(\boldsymbol{\lambda}_i,\boldsymbol{\delta}_i|z_i) - KL(q_\Phi(z_i|\boldsymbol{\lambda}_i,\boldsymbol{\delta}_i)||p_\Theta(z_i)), \tag{3.10}$$

After MVAE-FakeNews represents the fake news, we can apply the One-Class Learning algorithms focusing on fake news. The details of One-Class Learning are presented in the next section.

3.3.3 One-Class Learning

In the One-Class Learning (OCL), the training of the algorithms is only with examples of the interest class (fake news represented), i.e., in the absence of counterexamples, aiming at less effort from the user concerning the data labeling (ALAM *et al.*, 2020; RUFF *et al.*, 2018; KEMMLER *et al.*, 2013; KHAN; MADDEN, 2009; TAX, 2001).

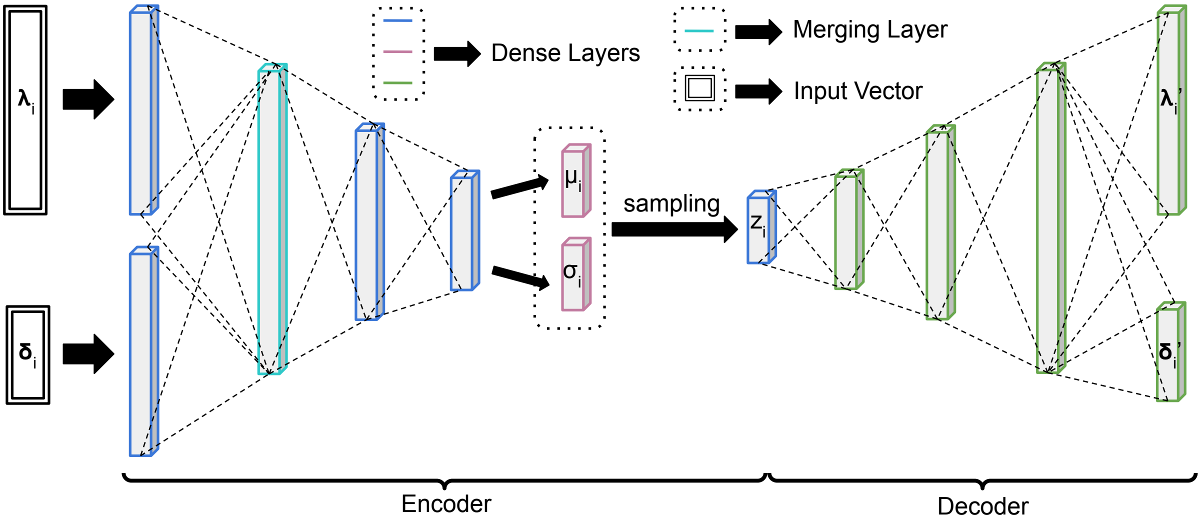


Figure 9 – Illustration of the proposed representation method MVAE-FakeNews For Fake News with the inputs $\boldsymbol{\lambda}_i$ and $\boldsymbol{\delta}_i$, and outputs $\boldsymbol{\lambda}'_i$ and $\boldsymbol{\delta}'_i$.

We formally define the OCL for fake news as a function $f : \mathscr{D} \rightarrow \mathscr{Y}$ that maps a news $\boldsymbol{d}_i \in \mathscr{D}$ to an $y_i \in \mathscr{Y}$ value, indicating how close the news $\boldsymbol{d}_i$ belongs to the interest class, i.e., fake news. The detection of fake news using OCL aims to learn a function f^* from a training set containing only fake news that comes close to the unknown mapping function f. The algorithm generates the y_i value and compares it to a threshold that determines whether the news is fake or real (Equation 3.11).

$$\text{Class of } \boldsymbol{d}_i = \begin{cases} y_i \leq threshold \rightarrow Fake \\ y_i > threshold \rightarrow Real \end{cases} \tag{3.11}$$

Among the OCL algorithms in the literature (GÔLO; MARCACINI; ROSSI, 2019), we chose the One-Class Support Vector Machines (OCSVM) (TAX; DUIN, 2004) since it is successful when the examples are represented appropriately and are considered one of the state-of-the-art in the area of OCL (ALAM *et al.*, 2020). The OCSVM (TAX; DUIN, 2004) has the objective function of finding the hypersphere of minimum volume that involves the data of the interest class (fake news). Formally, Equation 3.12 defines the center of the hypersphere (TAX; DUIN, 2004),

$$\boldsymbol{\mu}_{(c)} = \arg\min_{\boldsymbol{\mu} \in U} \max_{1 \leq i \leq m} \|\varphi(\boldsymbol{d}_i) - \boldsymbol{\mu}\|^2, \tag{3.12}$$

in which $\boldsymbol{\mu} \in U$, is a possible center in the feature space U associated with the function kernel φ, $\varphi(\boldsymbol{d}_i)$ maps $\boldsymbol{d}_i$ into another feature space defined according to the kernel chosen, and $\boldsymbol{\mu}_{(c)}$ is the center of the hypersphere in which the highest distance between $\varphi(\boldsymbol{d}_i)$ to $\boldsymbol{\mu}_{(c)}$ is minimal.

We will consider a news as fake if its distance from the center is less than the radius r of the hypersphere, i.e., $y_i = dist(\varphi(\boldsymbol{d}_i), \boldsymbol{\mu}_{(c)})$ and $threshold = r$. In general, we hope that the hypersphere is not too large so that the false positive rate does not increase. Therefore, we add a regularizer ν to accept a certain level of violation of the hypersphere decision function. Thus, we desire to minimize the square of the hypersphere radius and the number of violations. The minimization function is formalized by Equations 3.13, in which $\varepsilon_{\boldsymbol{d}_i}$ is the external distance between $\varphi(\boldsymbol{d}_i)$ and the surface of the hypersphere, and $\nu \in (0,1]$ defines the smoothness level of the hypersphere volume.

$$\min_{\boldsymbol{\mu},\varphi,r} \quad r^2 + \frac{1}{m}\sum_{i=1}^{m}\frac{\varepsilon_{\boldsymbol{d}_i}}{\nu} \tag{3.13}$$

subject to:

$$\|\varphi(\boldsymbol{d}_i) - \boldsymbol{\mu}_{(c)}\|^2 \leq r^2 + \varepsilon_{\boldsymbol{d}_i} \;\; \forall i = 1,...,m. \tag{3.14}$$

3.4 Experimental Evaluation

In the experimental evaluation, we propose to compare the MVAE-FakeNews with nine other unimodal and multimodal representation methods. We used the OCSVM algorithm to compare the fake news representation methods. Our goal is to demonstrate that the representations generated by MVAE-FakeNews outperform others commonly used in the literature for news classification. The next sections present the news collections used in the experimental evaluation, experimental settings, results, and discussion. For reproducibility reasons, all source codes and fake news collections used in the experimental evaluation are available[2].

3.4.1 Fake News Text Collections

This paper used three news collections, one in English and two in Portuguese. We got the first collection from FakeNewsNet (FNN) repository[3] (SHU *et al.*, 2020), which contains news from famous people fact-checked by the GossipCop[4] website. The dataset had a total of $16,095$ real news and $4,937$ fake news. After an initial analysis, we found that the news presented a significant imbalance in the number of tokens (words present in the news after removing unnecessary characters and stopwords) caused by collection errors. News ranging from 200 to 600 tokens were selected, remaining $5,298$ real news and $1,705$ fake news. FNN is the collection with the greatest unbalances in the distribution of classes. The remaining collections presented a balanced number of real and fake news.

The second collection, Fake.Br[5], is the first reference corpus in Portuguese for fake news detection. The news was manually collected and labeled. The corpus consists of $7,200$ news

[2] https://github.com/GoloMarcos/MVAE-FakeNews_Webmedia2021.git
[3] https://github.com/KaiDMML/FakeNewsNet
[4] https://www.gossipcop.com/
[5] https://github.com/roneysco/Fake.br-Corpus

items, distributed in 6 categories: politics (58%), TV and celebrities (21.4%), society and daily life (17.7%), science and technology (1.5%), economy (0.7%) and religion (0.7%). This corpus contains $3,600$ fake news and $3,600$ real news (SILVA *et al.*, 2020).

The third news collection[6], also in Portuguese, resulted from the collection of Fact Checking News (FCN) fact-checking news, in the following portals: *AosFatos*[7], *Agência Lupa*[8], *Fato ou Fake*[9], *UOL Confere*[10] and G1-Politica[11]. The collection contains $2,168$ news items, in which $1,124$ are real and $1,044$ are fake.

3.4.2 Experimental Settings

We used ten representation methods for the news, in which eight are unimodal, and two are multimodal. Three unimodal came from the traditional BoW technique (AGGARWAL, 2018a). The BoW represents each textual document in a vector, in which each dimension represents a term, that is, a word from the textual collection. The cell values of the vectors may vary according to the chosen weight scheme. We inferred BoW considering three different weight schemes:

- **Term Frequency (TF)**: defines the frequency of occurrence of a term in a document;
- **Term Frequency - Inverse Document Frequency (TFIDF)**: weights the TF with the frequency of documents with that term (Equation 3.15).
- **Binary**: indicates whether or not the term occurs in the document.

$$TF\text{-}IDF_{d_i,t_j} = TF_{d_i,t_j} \cdot log\left(\frac{|D|}{df_{t_j}}\right) \tag{3.15}$$

in which, TF_{d_i,t_j} corresponds to the frequency of the term t_j in the d_i document, $|D|$ to the number of documents, and df_{t_j} to the number of documents that contain the term t_j.

One of the unimodal methods was representations generated by the Linguistic Inquiry and Word Count (LIWC) tool (PENNEBAKER *et al.*, 2015). LIWC is a textual analysis tool, which assigns words from a text to different linguistic categories, providing a wide array of characteristics as an output. The categories involve linguistic dimensions, such as counting personal pronouns, prepositions, conjunctions, verbs, nouns, adjectives, interrogative and quantitative words, and psychological processes such as affective words, positive and negative emotions, anxiety, and sadness.

[6] https://github.com/GoloMarcos/FKTC
[7] https://aosfatos.org/noticias/
[8] https://piaui.folha.uol.com.br/lupa/
[9] https://g1.globo.com/fato-ou-fake/
[10] https://noticias.uol.com.br/confere/
[11] https://g1.globo.com/politica/

Another unimodal method that learns efficient representations is the DBERTML model (REIMERS; GUREVYCH, 2020), which is a neural language model based on Transformers (see Section 2.1.2 for more details). Context-dependent language models based on word embeddings like DBERTML explore the idea of distributed representations, in which numeric vectors represent sentences (DEVLIN *et al.*, 2019). DBERTML base these numeric vectors on dynamic word embeddings constructed based on the context of the sentence. In this way, DBERTML can apply correlation techniques, compare the vectors, and extract semantic and syntactic characteristics. (OTTER; MEDINA; KALITA, 2020).

Other methods used are density information, an AE with only dense layers, and a VAE. Such methods have as input the embedding from DBERTML. As for the two multimodal methods proposed in this work, we used the Multimodal Variational Autoencoders (MVAEs). The first is the MVAE-LIWC, which includes DBERTML embeddings and the attributes collected by LIWC. The second is the MVAE-FakeNews which includes the embeddings of the DBERTML and density information. The parameters of the representation techniques and OCSVM were:

- **BoW**: stopwords removal, stemming and cut-off = 1;
- **LIWC and DBERTML**: without parameters.
- **MVAE-FakeNews and Density Information**: clustering algorithm = k-Means and cluster quantity sets = $\{\{2,4,6,8,10\},\{3,5,7,9\}$ and $\{2,3,4,5,6,7,8,9,10\}\}$;
- **MVAEs, VAE and AE**: epoch = $\{5,10,50\}$, dimension of the dense layers = $\{(512,384,256,384,512),(512,256,512)$ and $(512,384,128,384,512)\}$, learning rate = $0,001$, optimization algorithm = $\{\text{Adam}\}$, activation function = $\{\text{linear}\}$ and number of iterations per epoch = $\frac{m}{batch\_size}$, in which m is the number of training examples and batch_size = 32;
- **MVAEs**: fusion operators = {addition, subtraction, concatenation, mean and multiplication};
- **OCSVM**: kernel = $\{RBF\}$, $\nu = \{0.001, 0.01\}$ and $0.05 * \nu, \nu \in [1..19]$, e $\gamma = \frac{1}{(n \cdot a)}$, in which n is the dimension of the input data and a is the variance of the representations.

In order to evaluate the approach, this work proposes an adaptation of the procedure k-Fold Cross-Validation (TAN; STEINBACH; KUMAR, 2013), considering the OCL classification scenario. This adaptation consists of applying the procedure only to the interest class (fake news). First, we divided the fake news set into 10 folds. To simulate a more realistic scenario of fake news, we use 1 fold for training and 9 folds for testing in this work. Then, we included the real news in the test set. In addition, we extracted only a percentage of training document from the training fold. The percentages were 30%, 50%, 70% and 100%, equivalent to 3%, 5%, 7% and 10% of all fake news documents in the collection. Thus, we alternate the folds so that they are in the training set and the test set. We use the F_1-Score as an evaluation measure, which is a harmonic average between precision (P) and recall (R).

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad (3.16) \qquad P = \frac{TP}{TP + FP} \quad (3.17) \qquad R = \frac{TP}{TP + FN} \quad (3.18)$$

In which True Positives (TP) is the number of fake news that the algorithm has correctly classified; False Positive (FP) is the number of real news that have been classified as fake news; and False Negatives (FN) is the number of fake news classified as real news.

3.4.3 Results and Discussion

Table 4 presents the results obtained for the three textual collections considering each percentage of fake news used in training. The results compare the ten representation models of fake news. The proposed MVAE-FakeNews obtained the highest F_1-Score in ten of the twelve evaluated scenarios. In the remaining two scenarios, the density information got the highest F_1-Score. BoW with the TFIDF term weight obtained the lowest F_1-Score in all scenarios.

Table 4 – Higher values of F_1-Scores of each representation technique considering the training percentage scenarios. The amount of fake news equivalent to a percentage appears next to the %.

	%	TFIDF	TF	Binary	DistilBERT	Density	LIWC	AE	VAE	MVAE-LIWC	MVAE-FakeNews
FakeBr	**3% (108)**	0.029±0.01	0.600±0.01	0.617±0.00	0.574±0.01	0.621±0.03	0.597±0.01	0.622±0.01	0.632±0.01	0.637±0.00	**0.642±0.00**
	5% (180)	0.102±0.01	0.603±0.01	0.619±0.00	0.602±0.01	0.633±0.02	0.607±0.01	0.635±0.00	0.637±0.00	0.636±0.01	**0.644±0.00**
	7% (252)	0.182±0.01	0.607±0.00	0.620±0.00	0.618±0.01	**0.647±0.02**	0.614±0.00	0.638±0.00	0.637±0.00	0.637±0.00	0.645±0.00
	10% (360)	0.263±0.01	0.610±0.00	0.621±0.00	0.628±0.01	**0.650±0.01**	0.620±0.00	0.640±0.00	0.638±0.00	0.639±0.00	0.646±0.00
FCN	**3% (31)**	0.001±0.00	0,556±0.02	0,591±0.02	0.426±0.06	0.487±0.07	0.557±0.03	0.375±0.07	0.741±0.04	0.726±0.03	**0.805±0.02**
	5% (52)	0.010±0.01	0,582±0.01	0,605±0.01	0.568±0.05	0.575±0.08	0.587±0.02	0.631±0.04	0.796±0.03	0.736±0.02	**0.813±0.03**
	7% (73)	0.037±0.01	0,591±0.01	0,610±0.01	0.640±0.03	0.584±0.06	0.600±0.02	0.697±0.01	0.801±0.01	0.749±0.03	**0.811±0.02**
	10% (104)	0.110±0.02	0,596±0.01	0,614±0.00	0.706±0.02	0.625±0.03	0.613±0.01	0.722±0.03	0.804±0.02	0.753±0.03	**0.808±0.02**
FNN	**3% (51)**	0.001±0.01	0,327±0.01	0,355±0.01	0.321±0.01	0.325±0.02	0.379±0.01	0.337±0.01	0.365±0.01	0.386±0.01	**0.395±0.03**
	5% (85)	0.026±0.01	0,344±0.01	0,360±0.01	0.345±0.01	0.345±0.04	0.382±0.00	0.353±0.01	0.367±0.01	0.388±0.01	**0.403±0.03**
	7% (120)	0.081±0.01	0,349±0.00	0,361±0.00	0.353±0.01	0.353±0.01	0.384±0.00	0.358±0.00	0.367±0.00	0.390±0.00	**0.397±0.01**
	10% (170)	0.169±0.01	0,353±0.00	0,362±0.00	0.363±0.01	0.357±0.02	0.386±0.00	0.362±0.00	0.367±0.00	**0.393±0.00**	**0.393±0.01**

We performed Friedman's statistical test with Nemenyi's post-test and Wilcoxon statistical test (TRAWINSKI *et al.*, 2012) to compare the representation methods considering all training percentage scenarios and datasets, i.e., we use each row of the table as a sample for the test. Figure 10 presents the result of the Friedman test with Nemenyi's post-test through the critical difference diagram[12]. The diagram presents the average rankings of the methods. Methods connected by a line do not present statistically significant differences between them. The Wilcoxon test is a non-parametric hypothesis test used when it is desired to compare pairs of methods to assess whether there is a statistically significant difference between them (TRAWINSKI *et al.*, 2012). The MVAE-FakeNews has a statistical difference compared to each of the methods with a confidence level of 95%.

[12] The test parameters were the default parameters of the KEEL tool (<https://sci2s.ugr.es/keel/index.php>).

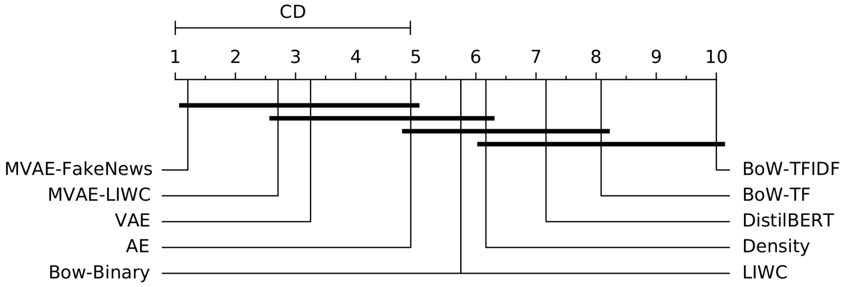


Figure 10 – Critical difference diagram of Friedman's statistical test with Nemenyi post-test.

MVAE-FakeNews obtain the best results. Multimodal methods were the ones that got the highest F_1-Scores since their average rankings are the best. It is worth mentioning that the multimodal methods did not obtain a statistically significant difference from the VAE and the AE[13]. This shows that representations from embeddings proved to generate better classification results. In addition to obtaining the best average ranking, MVAE-FakeNews got statistical differences from unimodal representations, such as all representations generated by BoW, density information, LIWC, and DBERTML.

For FakeNewsNet and Fact Checked News collections, the MVAE-FakeNews, when trained with only 3% of labeled fake news, got better F_1-Scores than the other nine methods, when these consider 10% of labeled fake news. MVAE-FakeNews outperforms all other methods, obtaining better results even with few labeled fake news. It is worth mentioning that, without considering the representation of density information, that got the best results considering 7% and 10% in the Fake.Br collection, we can observe the same behavior.

Even without presenting good average rankings, we can see that the density information obtain a higher F_1-Score in two scenarios for the Fake.Br collection. This indicates that news from different subjects can be located in different high-density regions. Based on the results, only density information is not enough to represent fake news. However, it is an important modality. The same behavior occurs with embeddings that capture syntactic and semantic information. However, when a model is learned from embeddings that capture syntactic, semantic, and Density information, as MVAE-FakeNews, all the information in the representation contributes to generating a more discriminative representation.

Figure 11 presents two-dimensional projections of the best representation model obtained by each method considering the Fact Checked News collection. We generated the representations using the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm (MAATEN; HINTON, 2008) with the default parameters from the sklearn library. Although there was a lot of information loss, we can infer that representations generated by BoW-TF, BoW-Binary, and LIWC were the representations with the greatest overlap among the classes of fake and real news. The representations of density information and BoW-TFIDF were able to represent the data from

[13] We highlight that the lack of statistically significant difference can occur due to the high number of methods, which implies a high critical difference.

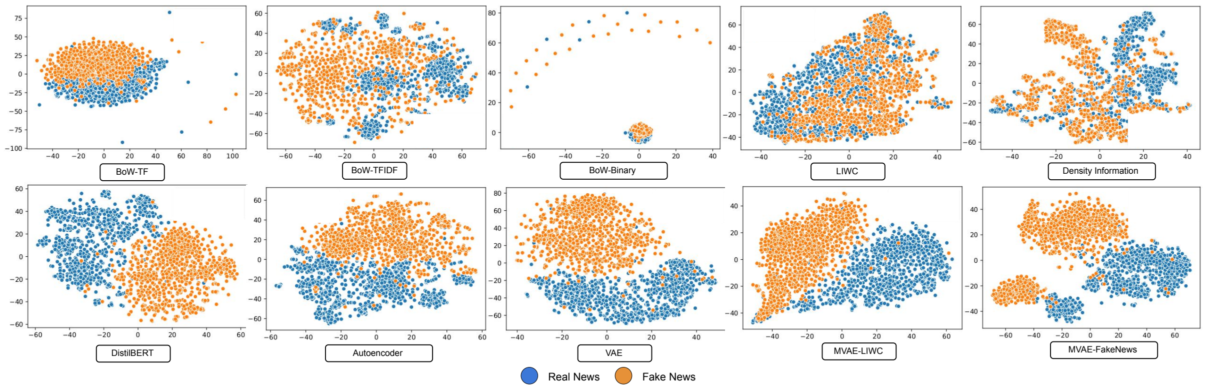


Figure 11 – Two-dimensional projection (t-SNE) of each fake news representation method. The colors indicate fake news and real news. Representation methods that show less overlap between classes are more promising for detecting fake news.

different classes into different regions. The representations of DBERTML and the methods that use this representation as input, such as AE, VAE, MVAE-LIWC, and MVAE-FakeNews, generated more discriminative representations, i.e., with less class overlap, which can provide better classification models.

The fake news Fact Checked News collection is composed of the news that authors collected from August 2018 to May 2019. At this time, Brazil witnessed a political competition between the *Partido dos Trabalhadores* political party, of the former president of Brazil, Luiz Inácio Lula da Silva, and the current president of Brazil, Jair Messias Bolsonaro. A large portion of the news in this dataset addresses information about presidential elections, and another portion does not. MVAE-FakeNews was the only method that performed the separation of real and fake news in two different hubs. Our intuition is that MVAE-FakeNews divided the news in the latent space by subject, which may have helped the classification process.

3.5 Concluding Remarks

Fake news is increasingly present in society, impacting sectors like politics and the economy. This impact requires methods to detect fake news. Generally, literature studies detect fake news with machine learning algorithms. One-Class Learning is an appropriate method to detect fake news since it learns only with examples from the interest class. Also, it is more suitable for unbalanced scenarios and scenarios that are difficult to label, in addition to the advantage of having less labeling effort. However, naturally unstructured texts must be structured considering a representation model to apply machine learning algorithms. Different characteristics can be used to generate more discriminative representations of fake news, such as syntactic and semantic relations between words, topics, and sentiment, in order to improve classification performance.

This work proposed the Multimodal Variational Autoencoder (MVAE) that has as modalities the embeddings generated by the model DBERTML and density information. MVAE-

FakeNews obtained better results than nine other text representation techniques and got better F_1-Scores in ten out of twelve scenarios. The proposed method also presented a statistically significant difference concerning other techniques. When we use the MVAE-FakeNews with the One-Class Learning, its results with only 3% fake news labeled do not significantly differ from its results with a higher number of labeled data. Also, the results obtained by MVAE-FakeNews with 3% of fake news labeled are comparable or higher than other methods when considering 10% of fake news labeled and generated more discriminative representations. Thus, MVAE-FakeNews proved to be promising to represent news using One-Class Learning to detect fake news.

As future work, we will explore more modalities for MVAE-FakeNews, mainly in features that consider features of location, time, as well as names of people and organizations extracted from the news. We also intend to use semi-supervised OCL algorithms (Positive and Unlabeled Learning (BEKKER; DAVIS, 2020)) with representations generated by the MVAE-FakeNews.

CHAPTER

4

DETECTING RELEVANT APP REVIEWS FOR SOFTWARE EVOLUTION AND MAINTENANCE THROUGH MULTIMODAL ONE-CLASS LEARNING

4.1 Introduction

The popularization of smartphones and social media has transformed the relationship between developers of mobile applications (apps) and users (Maalej *et al.*, 2016). As a result, popular apps receive hundreds of thousands or even millions of reviews written in natural language (Maalej *et al.*, 2016). These reviews present user feedback on various aspects, such as functionality, usability, performance, bug report, and provide app developers a new and valuable channel to extract knowledge for software evolution and maintenance (CHEN *et al.*, 2014; GUZMAN; EL-HALIBY; BRUEGGE, 2015). However, manual analysis of this large volume of information becomes an impractical task (STANIK; HAERING; MAALEJ, 2019). In this context, several studies have proposed machine learning-based methods for automatically detecting relevant app reviews to support the monitoring of users' opinions about the app (ARAUJO *et al.*, 2020; CHEN *et al.*, 2014; STANIK; HAERING; MAALEJ, 2019).

Panichella et al. (PANICHELLA *et al.*, 2015) introduced a pioneering study on review classification to help developers in accomplishing software maintenance and evolution. The study showed that natural language processing and machine learning are useful to detect reviews describing problem discovery and feature requests, distinguishing them from other irrelevant reviews. Further studies showed the impact of relevant reviews on the software development lifecycle, such as release planning (VILLARROEL *et al.*, 2016), recommending software changes (SORBO *et al.*, 2016), improving automated testing tools (GRANO *et al.*, 2018), identifying security issues (TAO; GUO; HUANG, 2020), and requirements engineering tasks (ARAÚJO;

MARCACINI, 2021; PANICHELLA; RUIZ, 2020). Although promising results have been reported, most previous studies require many labeled reviews to train a classifier. In practical terms, it requires a great human effort and a costly process for labeling reviews. This fact can even make review-based software evolution unfeasible for scenarios with a high rate of software changes since labeled reviews would quickly become outdated. Thus, we raise the question: how do domain experts detect relevant reviews to support software evolution and maintenance with less labeling effort?

This paper presents an approach for detecting relevant reviews through multimodal one-class learning (OCL). In OCL, we only need to label reviews of the interest class (e.g., relevant app reviews) for classifier training. Moreover, OCL can be promising in relation to traditional multi-class learning considering unbalanced datasets, such as the app reviews scenario (FERNÁNDEZ *et al.*, 2018). We also present methods to improve feature extraction and reviews representation to handle the smaller amount of labeled reviews during model training without harming classification performance. In special, we propose multimodal representations which explore both textual data and visual information based on the density of the reviews dataset. Density information can be interpreted as a summary of the main topics extracted from the reviews.

In short, our multimodal one-class methods train a review classifier from two modalities: (i) textual embeddings generated by the Bidirectional Encoder from Transformers (BERT), which outperforms other text representation methods in different natural language processing tasks (ARAUJO *et al.*, 2020; DEVLIN *et al.*, 2019); and (ii) high-density regions of the app reviews representing different topics or subtopics, such as bugs, feature, user experience, security, and performance. We propose a deep neural network based on a multimodal autoencoder to merge the two modalities into a single latent representation, called embedding space. Finally, we use the Support Vector Data Description (SVDD) classifier to train a model to detect relevant reviews from the learned embedding space.

We carried out an experimental evaluation with tree app reviews collections to compare our multimodal OCL methods to the other eight text representation methods used in the literature. Our proposal performed best in most classification scenarios for detecting relevant app reviews. Moreover, one of our multimodal methods statistically outperforms other methods in different scenarios. We observed that our method achieved competitive results even using only 3% and 5% of labeled reviews compared to models that used the entire training set (90% of labeled reviews). Thus, our multimodal methods proved to be a competitive alternative to the state-of-the-art and promising to practical scenarios, requiring less effort from domain experts.

The methods and results presented in this chapter are part of a preliminary study called "*From Bag-of-Words to Pre-trained Neural Language Models: Improving Automatic Classification of App Reviews for Requirements Engineering*" (ARAUJO *et al.*, 2020), with a focus on evaluating textual representation for app reviews. In addition, this preliminary study was extended to incorporate the multimodal representation learning task in the One-Class Learning scenario (GÔLO *et al.*, 2022).

The remainder of this chapter is organized as follows. First, we present the related work in Section 4.2, the concepts and details of the proposed method and OCL in Section 4.3, and the experimental settings, results, and discussions in Section 4.4. Finally, we present the conclusions and future work directions in Section 4.6.

4.2 Related Work

App developers need to understand user feedback to make strategic decisions about software evolution (Maalej *et al.*, 2016). App reviews contain useful information to improve, evolve or maintain software (PANICHELLA; RUIZ, 2020). The volume and unstructured nature of textual reviews, combined with the high volume of irrelevant reviews, makes identifying relevant reviews essential in this process (GUZMAN; EL-HALIBY; BRUEGGE, 2015).

Tao, Guo and Huang (2020), Kifetew *et al.* (2021), Zhang, Wang and Xie (2019) use rule-based learning approaches, in which a set of app reviews are previously analyzed in search of linguistic patterns (Dependency Tree (D. Tree)), common to the passages they describe the information of interest, such as a software requirement. However, the literature points out limitations for the rule-based approaches (AGGARWAL, 2018a):

- **Efficiency**: a textual expression can be represented by a large number of corresponding rules or by rules with high complexity;
- **Accuracy**: a large volume of rules can generate a lot of conflicts, or a small number of rules may not be representative enough to extract the desired information;
- **Generalization**: rules extracted by analyzing data from a specific domain generally do not meet the context of other domains;
- **Dependency**: requires the use of external dictionaries to extract context information. For example, they identify whether a token belongs to a specific type, such as titles, places, organizations, etc.

In addition to the rule-based strategy, the supervised multi-class classification strategy is widely used in the literature to identify relevant reviews (ARAUJO *et al.*, 2020). Al Kilani, Tailakh and Hanani (2019), Messaoud *et al.* (2019) use the multi-class strategy without the irrelevant review category. This reduces annotation volume but significantly increases the classifier's chances of incorrectly assigning one of the classes to an irrelevant review. For instance, if an irrelevant app review comes up to be classified, it will be assigned to one of the learned classes, which can prejudice the automatic app review classification. Al Kilani, Tailakh and Hanani

(2019), Araujo *et al.* (2020), Maalej *et al.* (2016), Messaoud *et al.* (2019), Stanik, Haering and Maalej (2019), Wang *et al.* (2018), Rungta *et al.* (2020), Panichella *et al.* (2016) use multi-class learning to detect relevant app reviews by labeling irrelevant reviews. However, multi-class learning has different gaps, such as the great user effort to label all the classes, including the irrelevant app reviews. Furthermore, the scope of irrelevant reviews is too broad (CHEN *et al.*, 2014), culminating in the challenge to label irrelevant reviews. Binary classification methods, such as Guzman, El-Haliby and Bruegge (2015), Maalej *et al.* (2016), Wu *et al.* (2021), Tao, Guo and Huang (2020), decrease the labeling effort, but users should still label irrelevant app reviews. Furthermore, such a strategy often leads to unbalanced datasets, harming the classifier's performance (FERNÁNDEZ *et al.*, 2018).

In addition to the classification model for reviews, existing methods also investigate different approaches to obtain a structured representation. Al Kilani, Tailakh and Hanani (2019), Araujo *et al.* (2020), Guzman, El-Haliby and Bruegge (2015), Messaoud *et al.* (2019), Maalej *et al.* (2016), Wang *et al.* (2018), Zhang, Wang and Xie (2019), Kifetew *et al.* (2021), Tao, Guo and Huang (2020), Wu *et al.* (2021) use unimodal methods to structure the app reviews. Al Kilani, Tailakh and Hanani (2019), Guzman, El-Haliby and Bruegge (2015), Messaoud *et al.* (2019), Maalej *et al.* (2016), Panichella *et al.* (2016) use the Bag-of-Words (BoW), which generates sparse representations with high dimensionality, and without any semantic information. Stanik, Haering and Maalej (2019), Rungta *et al.* (2020), Wang *et al.* (2018) use context-free language models (CFLM), which does not have the limitations of BoW. However, CFLM represents the words regardless of their context. Finally, Araujo *et al.* (2020), Wu *et al.* (2021) use the context-dependent language model (CDLM) that outperforms CFLM or BoW in different natural language tasks.

We did not find studies in the literature that explore multimodality and OCL to detect relevant reviews. However, we found studies that explore multimodality and multi-class learning. Stanik, Haering and Maalej (2019), Panichella *et al.* (2016) explore multimodality concatenating textual modalities such as the BoW representation, the text sentiment, keywords, etc. On the other hand, Rungta *et al.* (2020) proposed a multimodal approach that learns a representation from information available in application packages (APKs) to categorize apps. The main characteristics of this approach are:

- use of multi-class learning;
- learn a representation using a neural network;
- use the concatenation operator in the modality fusion;
- the use of CFLM and BoW to represent textual modalities.

Table 5 shows a comparison between our proposal and related work considering the learning type, multimodality, the use of irrelevant reviews, and the text representation type. Previous studies use multi-class or binary classification and generally explore unimodal representations. However, some studies explore multimodality. Our proposal differs from these studies in:

- we use one-class learning to reduce the effort in labeling app reviews;
- we use a Context-Dependent Language Model (CDLM) for text embeddings;
- we propose a visual topic-based modality extracted from relevant app reviews.

Table 5 – Comparison between our proposal and related works considering the learning type, the use of multimodality, the use of irrelevant reviews, and the text representation.

	Learning type			**Multi**	**Irrelevant**	Text Representation			
	Multi	Binary	**One**	**modal**	**Reviews**	D. Tree	BoW	CFLM	**CDLM**
(Al Kilani; Tailakh; Hanani, 2019)	✓						✓		
(ARAUJO *et al.*, 2020)	✓				✓		✓		✓
(GUZMAN; EL-HALIBY; BRUEGGE, 2015)		✓			✓		✓		
(Maalej *et al.*, 2016)	✓	✓			✓		✓		
(MESSAOUD *et al.*, 2019)	✓						✓		
(RUNGTA *et al.*, 2020)	✓			✓	✓		✓	✓	
(STANIK; HAERING; MAALEJ, 2019)	✓			✓	✓		✓	✓	
(WANG *et al.*, 2018)	✓				✓		✓	✓	
(PANICHELLA *et al.*, 2016)	✓			✓	✓		✓		
(WU *et al.*, 2021)		✓			✓				✓
(TAO; GUO; HUANG, 2020)		✓			✓	✓			
(KIFETEW *et al.*, 2021)					✓	✓			
(ZHANG; WANG; XIE, 2019)					✓	✓			
Our			✓	✓	✓				✓

4.3 Multimodal Representations to Detect Relevant App Reviews

Textual data, such as relevant app reviews, are naturally multimodal and can be represented in different ways. Multimodal representation learning aims to learn a new unified representation based on multiple representations, which can be complementary or supplementary (GUO; WANG; WANG, 2019; LI; YANG; ZHANG, 2018).

Our proposal consists of a pipeline with six steps to detect relevant app reviews through one-class learning via multimodal representations. Figure 12 shows the pipeline:

- **First Step:** consists of creating a dataset with only relevant reviews;
- **Second Step:** we represent the relevant app reviews texts through a CDLM. We choose a CDLM because they allow the capture of syntactic and semantic characteristics, unlike other traditional text representation models (DEVLIN *et al.*, 2019; OTTER; MEDINA; KALITA, 2020). We present this step background on Section 4.3.1;

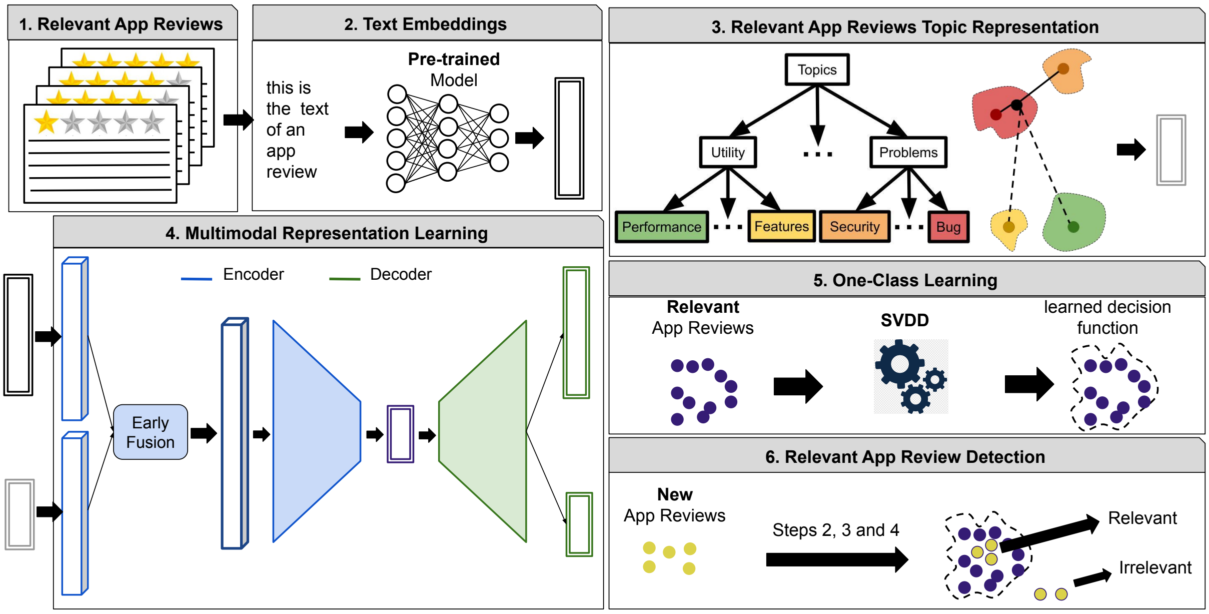
- **Third Step:** we create a second modality based on density information called Relevant App Reviews Topic Representation (RARTR). In this modality, high-density regions represent well-defined topics such as bugs, features, user experience, security, and performance (Al Kilani; Tailakh; Hanani, 2019; ARAUJO *et al.*, 2020; GUZMAN; EL-HALIBY; BRUEGGE, 2015; Maalej *et al.*, 2016; STANIK; HAERING; MAALEJ, 2019). Furthermore, while previous studies frequently use modalities such as BERT and BERT variants embeddings (ARAUJO *et al.*, 2020), word embeddings (STANIK; HAERING; MAALEJ, 2019), bag-of-words (Al Kilani; Tailakh; Hanani, 2019), or information available in application packages (APKs) (RUNGTA *et al.*, 2020), the use of topic information as a modality for app reviews is still unexplored. We present this step background on Section 4.3.2;

- **Fourth Step:** an autoencoder learns a multimodal representation through the two modalities (Steps 2 and 3). Thus, the text embeddings are language information. The RARTR is visual information with the spatial distribution of the relevant app reviews. We present this step background on Section 4.3.3;

- **Fifth Step:** we use the OCL to learn a decision function to identify relevant app reviews (interest class). We present this step background on Section 4.3.4;

- **Sixth Step:** the classifier obtained in the previous step is used in new app review datasets to identify relevant information for developers (filtering irrelevant reviews). We present this step background on Section 4.3.4;

Figure 12 – Pipeline with six steps representing our proposal to represent app reviews and detect relevant app reviews.



4.3.1 Text Embedding Representation

To represent the texts from app reviews, we use the text embeddings generated by the multilingual version of the Bidirectional Encoder from Transformers (BERT) model (DEVLIN *et al.*, 2019). The BERT model was trained in a very large textual corpus that represents sentences based on their context. In the training step, the BERT executes two tasks, mask language model and next sentence prediction. Given the word sequence of a textual document $\mathbf{d}_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,v}\}$, the BERT model generates a corrupted version of $\mathbf{d}_i$, $\hat{\mathbf{d}}_i$, in which words are selected randomly (e.g., 15%) and replaced by a `[MASK]` symbol. Also, let the $\bar{\mathbf{d}}_i$ be the masked tokens of $\mathbf{d}_i$. Then, the BERT's training consists of reconstructing the masked words $\bar{\mathbf{d}}_i$ from $\hat{\mathbf{d}}_i$ (YANG *et al.*, 2019):

$$\max_{\Theta} \log p_{\Theta}(\bar{\mathbf{d}}_i|\hat{\mathbf{d}}_i) \approx \sum_{w_i \in \mathbf{d}_i} c_{w_t} \log \frac{\exp(H_\theta(\hat{\mathbf{d}}_\mathbf{i})^T_{w_t} \boldsymbol{e}(w_t))}{\exp(\sum_{w'} H_\theta(\hat{\mathbf{d}}_\mathbf{i})^T_{w_t} \boldsymbol{e}(w'))}, \tag{4.1}$$

in which $c_{w_t} = 1$ indicates that w_t is masked, $\boldsymbol{e}(w_t)$ indicates the embedding of the word w_t, w' is $\mathbf{d}_i$ without w_t, $H_\theta(\mathbf{d}_i) = \{\mathbf{h}_\theta(\mathbf{d}_i)_1, \mathbf{h}_\theta(\mathbf{d}_i)_2, \ldots, \mathbf{h}_\theta(\mathbf{d}_i)_v\}$ is a sequence of hidden vectors mapped by a Transformer.

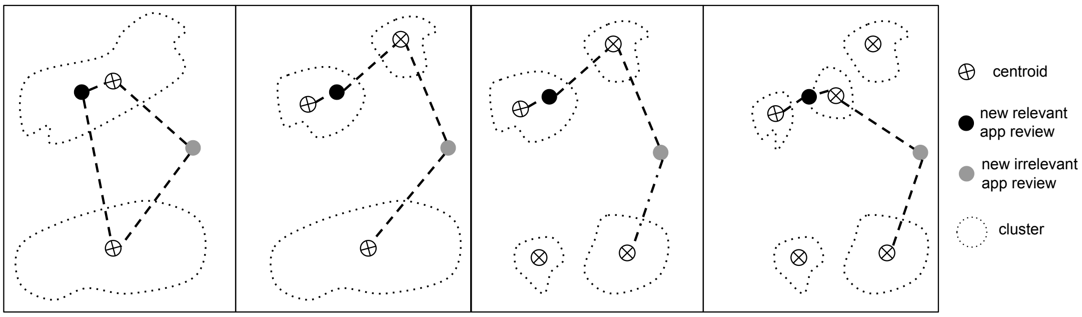
The second task is to predict if a sentence is the next of another sentence based on the representation generated by BERT. This second task helps BERT to learn relationships between sentences that may not be learned with the first task alone. We use a Multilingual Distilled version of BERT (DBERTML) pre-trained model to represent texts. Given a new document $\mathbf{d}_i$, the DBERTML uses a number of words to create an embedding $\boldsymbol{\lambda}_i$ with 512 real values representing document $\mathbf{d}_i$. Given the number of words that the model will consider, e.g., 128, if the document has more than 128 words, DBERTML uses only the 128 first words, 128 words from the middle, or the final 128 words. Therefore, we use the 128 first words from the reviews.

4.3.2 RARTR: Relevant App Reviews Topic Representation

As a second modality, we present a representation based on topic information, which assumes that different high-density regions group the documents of the interest class (KRAWCZYK; WOŹNIAK; CYGANEK, 2014; SHARMA; SOMAYAJI; JAPKOWICZ, 2018). Figure 13 shows an example of different cluster settings comparing the behavior of a relevant app review belonging to one of the relevant topics (one of the high-density regions) and an irrelevant review. The relevant review is always closer to a topic (cluster centroid) than the irrelevant review, i.e., it is better allocated regardless of the cluster setting. One way to obtain the topic information of relevant app reviews documents is to apply a clustering algorithm and use a statistical technique to calculate the merit of the obtained cluster structure.

Consider a clustering with k clusters, i.e., $\mathscr{D} = C_1 \cup C_2, \cup \cdots \cup C_k$, in which $\mathscr{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_m\}$ is the collection of app reviews documents, C_j is a cluster of documents,

Figure 13 – Different cluster settings comparing the behavior of a relevant app review belonging to one of the relevant reviews topics.



and $2 \leq k < m$. Then, we apply the silhouette coefficient (ROUSSEEUW, 1987) in order to measure if a document belongs to a single topic or contains mixed topics. The silhouette for a document $\boldsymbol{d}_i$ assigned to a cluster C_j is defined in Equation:

$$s(\boldsymbol{d}_i, k) = \frac{\beta(\boldsymbol{d}_i) - \alpha(\boldsymbol{d}_i)}{\max(\alpha(\boldsymbol{d}_i), \beta(\boldsymbol{d}_i))}, \tag{4.2}$$

in which $\alpha(\boldsymbol{d}_i)$ is the average distance of $\boldsymbol{d}_i$ to all documents of cluster C_j, and $\beta(\boldsymbol{d}_i)$ defines the average distance of a document $\boldsymbol{d}_i$ to all documents of the closest cluster C_o, $o \neq j$. $\alpha(\boldsymbol{d}_i)$ and $\beta(\boldsymbol{d}_i)$ are given by Equations 4.3 and 4.4 respectively.

$$\alpha(\boldsymbol{d}_i) = \frac{1}{|C_j| - 1} \sum_{\boldsymbol{d}_l \in C_j, \boldsymbol{d}_i \neq \boldsymbol{d}_l} dist(\boldsymbol{d}_i, \boldsymbol{d}_l) \tag{4.3}$$

$$\beta(\boldsymbol{d}_i) = \min_{o \neq j} \frac{1}{C_o} \sum_{\boldsymbol{d}_n \in C_o} dist(\boldsymbol{d}_i, \boldsymbol{d}_n) \tag{4.4}$$

The silhouette values range from -1 to +1. A high value indicates that a review is well allocated to its cluster and is far from the neighboring clusters, i.e., a review tends to belong to a single topic. We generate the representation of relevant app reviews topic information by concatenating silhouette coefficient values considering each document in different clustering settings. For instance, given u different clustering settings (i.e., the clustering settings have different values of k), we generate the RARTR modality by:

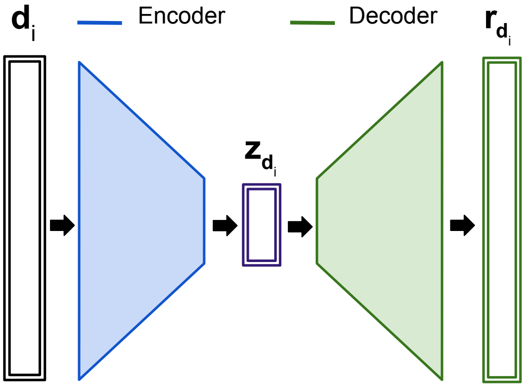
$$\boldsymbol{\delta}_i = \{s(\boldsymbol{d}_i, k_1), s(\boldsymbol{d}_i, k_2), \ldots, s(\boldsymbol{d}_i, k_u)\} \tag{4.5}$$

in which $s(\boldsymbol{d}_i, k_j)$ is the silhouette of $\boldsymbol{d}_i$ in cluster setting with k_j clusters.

4.3.3 Multimodal Representation Learning

Our method uses Autoencoders (AE) to learn a joint representation for the two proposed modalities. An AE has two steps: encoding and decoding (AGGARWAL, 2018b). The first half of the layers corresponds to the encoding, in which the input $\boldsymbol{d}_i$ is compressed to a latent space $\boldsymbol{z}_{\boldsymbol{d}_i}$. The last half of the layers corresponds to the decoding step, which attempts to reconstruct the input encoded in $\boldsymbol{z}_{\boldsymbol{d}_i}$ in the output $\boldsymbol{r}_{\boldsymbol{d}_i}$. $\boldsymbol{z}_{\boldsymbol{d}_i}$ and $\boldsymbol{r}_{d_i}$ are computed according to Equation 4.6. We present an illustration of an AE in Figure 14.

Figure 14 – Illustration of an Autoencoder.



$$Autoencoder = \begin{cases} \mathbf{z}_{\boldsymbol{d}_i} = f(\Phi; \mathbf{d}_i) \\ \mathbf{r}_{\boldsymbol{d}_i} = g(\Theta; \mathbf{z}_{\boldsymbol{d}_i}) \end{cases} \tag{4.6}$$

in which Φ is respectively the weights and biases of neurons in the encoding neural network, Θ is the weights and biases of neurons in the decoding neural network. The AE learns these parameters to minimize the reconstruction error of the neural network, e.g., the mean squared error (Equation 4.7).

$$J(\Phi;\Theta) = \frac{1}{|\mathscr{D}|} \sum_{\boldsymbol{d}_i \in \mathscr{D}} ||\boldsymbol{d}_i - \boldsymbol{r}_{\boldsymbol{d}_i}||^2 \tag{4.7}$$

There are variations of the AE, which aim to force the learned representations to assume useful properties for the application scenario (AGGARWAL, 2018b). One variant is the Variational Autoencoder (VAE). A VAE imposes a specific probabilistic structure on the hidden units. One of the simplest constraints is that the activation in the hidden units should be drawn from the standard Gaussian distribution (i.e., zero mean and unit variance) (AGGARWAL, 2018b).

Another characteristic is that we can feed the decoder with samples from the standard normal distribution in order to generate samples of the training data, i.e., the VAE performs sampling in the encoding step to generate new data (XU; DURRETT, 2018).

Formally, the VAE assumes a variable $\boldsymbol{z}_{\boldsymbol{d}_i}$ that generates the data $\boldsymbol{d}_i$, by using the probability function:

$$p(\boldsymbol{z}_{\boldsymbol{d}_i}|\boldsymbol{d}_i) = \frac{p(\boldsymbol{d}_i|\boldsymbol{z}_{\boldsymbol{d}_i})p(\boldsymbol{z}_{\boldsymbol{d}_i})}{p(\boldsymbol{d}_i)}, \tag{4.8}$$

in which

$$p(\boldsymbol{d}_i) = \int p(\boldsymbol{d}_i|z_{\boldsymbol{d}_i})p(z_{\boldsymbol{d}_i})dz_{\boldsymbol{d}_i}. \tag{4.9}$$

VAE approximates $p(z_{\boldsymbol{d}_i}|\boldsymbol{d}_i)$ to another treatable distribution $q(z_{\boldsymbol{d}_i}|\boldsymbol{d}_i)$ using the Kullback-Leibler (KL) divergence that is responsible for measuring the divergence between distributions. To optimize the marginal likelihood ($p(\boldsymbol{d}_i)$), you can use the log of the marginal likelihood (XU; DURRETT, 2018):

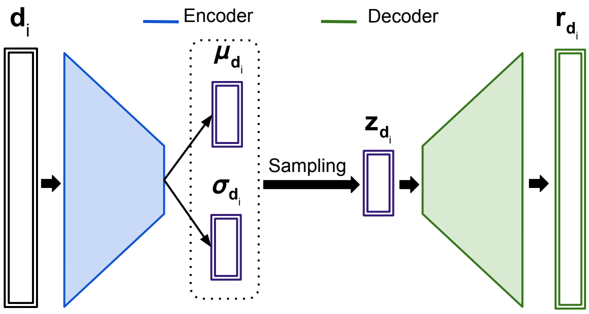
$$\log p_{\Theta}(\mathbf{d}_i) = KL(q_{\Phi}(\mathbf{z}_{\boldsymbol{d}_i}|\mathbf{d}_i)||p_{\Theta}(\mathbf{z}_{\boldsymbol{d}_i}|\mathbf{d}_i)) + \mathscr{L}(\Theta,\Phi;\mathbf{d}_i), \tag{4.10}$$

in which

$$\mathscr{L}(\Theta,\Phi,\mathbf{d}_i) = \mathbb{E}_{q_{\Phi}(\mathbf{z}_{\boldsymbol{d}_i}|\mathbf{d}_i)} \log p_{\Theta}(\mathbf{d}_i|\mathbf{z}_{\boldsymbol{d}_i}) - KL(q_{\Phi}(\mathbf{z}_{\boldsymbol{d}_i}|\mathbf{d}_i)||p_{\Theta}(\mathbf{z}_{\boldsymbol{d}_i})). \tag{4.11}$$

When a neural network implements a VAE, it learns the encoder's Φ parameters and the decoder's Θ parameters through the weights of the neurons of the neural network layers. The first term of Equation 4.11 is related to the neural network reconstruction error. In the second term, we want to minimize the difference between the learned distribution $q_{\Phi}(\mathbf{z}_{\boldsymbol{d}_i}|\mathbf{d}_i)$ and $p_{\Theta}(\mathbf{z}_{\boldsymbol{d}_i})$ (prior knowledge). It is worth mentioning that literature studies replace the term $p_{\Theta}(\mathbf{z}_{\boldsymbol{d}_i})$ by $\mathscr{N}(\mathbf{z}_{\boldsymbol{d}_i};0,1)$, that is, by a multivariate Gaussian distribution with average 0 and standard deviation 1. Thus, we want to maximize Equation 4.11. We present an illustration of a VAE in Figure 15.

Figure 15 – Illustration of a Variational Autoencoder.



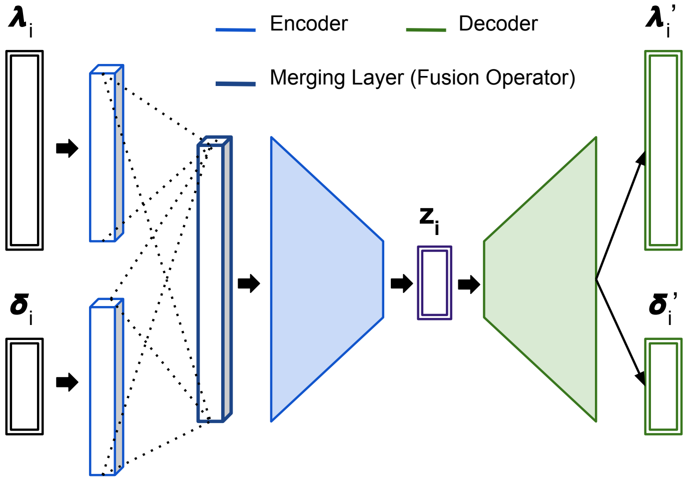
We propose a Multimodal Autoencoder (MAE) and a Multimodal Variatinal autoencoder (MVAE). MAE and MVAE extend AE and VAE to multiple modalities, respectively. Therefore, the proposals have two inputs, two outputs, and the fusion of the modalities in its encoder steps. We use early fusion to combine the modalities through two dense layers with the same number of neurons. Thus, we can use different early fusion operators, such as average, addition, or subtract, to combine the modalities.

AE and VAE calculate the error of reconstructing only one of the modalities. MAE and MVAE will calculate the error of reconstructing from both $\boldsymbol{\lambda}_i$ (embedding from DistilBERT) and $\boldsymbol{\delta}_i$ (RARTR). Thus, we define MAE as:

$$MAE = \begin{cases} \mathbf{z}_i = f(\Phi; \boldsymbol{\lambda}_i, \boldsymbol{\delta}_i) \\ \boldsymbol{\lambda}'_i, \boldsymbol{\delta}'_i = g(\Theta; \mathbf{z}_i) \end{cases} \tag{4.12}$$

MAE learns Φ and Θ to minimize the mean squared error between $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}'_i$, and $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}'_i$. We present an illustration of our MAE in Figure 16.

Figure 16 – Illustration of our MAE to represent the app reviews.



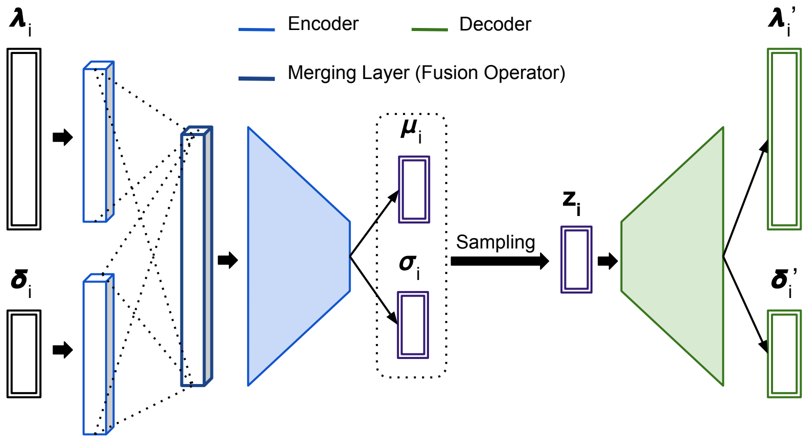
Our MVAE aims to maximize Equation 4.13. The first term calculates two reconstruction errors (embedding and RARTR). The second term wants to approximate the learned distribution $q_\Phi(\mathbf{z}_i|\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i)$ from $p_\Theta(\mathbf{z}_i)$. We present an illustration of our MVAE in Figure 17.

$$\mathscr{L}(\Theta, \Phi, \boldsymbol{\lambda}_i, \boldsymbol{\delta}_i) = \mathbb{E}_{q_\Phi(\mathbf{z}_i|\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i)} \log p_\Theta(\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i|\mathbf{z}_i) - KL(q_\Phi(\mathbf{z}_i|\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i)||p_\Theta(\mathbf{z}_i)), \tag{4.13}$$

4.3.4 Relevant App Reviews Detection through One-Class Learning

Given a set of app review documents, we formally define the OCL as a function $f : \mathscr{D} \rightarrow \mathscr{Y}$ that maps an app review $\boldsymbol{d}_i \in \mathscr{D}$ to a $y_i \in \mathscr{Y}$ value, indicating how close the review $\boldsymbol{d}$ belongs to the interest class, i.e., to be a relevant app review. The detection of relevant app reviews using OCL aims to learn a function f^* from a training set containing only relevant app reviews close to the unknown mapping function f. After learning the function f^*, the classifier is able to generate

Figure 17 – Illustration of our MVAE to represent the app reviews.



a y_i value to a new review $\boldsymbol{d}_i$ and compares it to a threshold in order to define whether the app review is relevant or irrelevant (Equation 4.14).

$$\text{Class of } \boldsymbol{d}_i = \begin{cases} y_i \leq threshold \rightarrow \text{Relevant} \\ y_i > threshold \rightarrow \text{Irrelevant} \end{cases} \tag{4.14}$$

Among the OCL algorithms (GÔLO; MARCACINI; ROSSI, 2019), we chose the Support Vector Data Description (SVDD) (TAX; DUIN, 2004) since it is successful when the examples are represented appropriately (ALAM *et al.*, 2020). The SVDD objective function aims to obtain the hypersphere of minimum volume that involves the data of the relevant app reviews. Formally, Equation 4.15 defines the center of the hypersphere (TAX; DUIN, 2004),

$$\mu_{(c)} = \underset{\boldsymbol{\mu} \in U}{\arg\min} \max_{1 \leq i \leq m} ||\varphi(\boldsymbol{d}_i) - \boldsymbol{\mu}||^2, \tag{4.15}$$

in which m is the number of relevant app reviews, $\boldsymbol{\mu} \in U$, is a possible center in the feature space U associated with the function kernel φ, $\varphi(\boldsymbol{d}_i)$ maps $\boldsymbol{d}_i$ into another feature space defined according to the kernel chosen, and $\boldsymbol{\mu}_{(c)}$ is the center of the hypersphere in which the highest distance between $\varphi(\boldsymbol{d}_i)$ to $\boldsymbol{\mu}_{(c)}$ is minimal.

We will consider an app review as relevant if its distance from the center is less than the radius r of the hypersphere, i.e., $y_i = dist(\varphi(\boldsymbol{d}_i), \mu_{(c)})$ and $threshold = r$. In general, we hope that the hypersphere is not too large so that the false positive rate does not increase. Therefore, we add a regularizer ν to accept a certain level of violation of the hypersphere decision function. Thus, we desire to minimize the square of the hypersphere radius and the number of violations. The formalization of the minimization function is given by Equations 4.16 and 4.17, in which ε_{d_i} is the external distance between $\varphi(d_i)$ and the surface of the hypersphere, and $\nu \in (0,1]$ defines the smoothness level of the hypersphere volume.

$$\min_{\mu,\varphi,r} \quad r^2 + \frac{1}{m}\sum_{i=1}^{m}\frac{\varepsilon_{\boldsymbol{d}_i}}{\nu}, \tag{4.16}$$

$$\|\varphi(\boldsymbol{d}_i) - \mu_{(c)}\|^2 \le r^2 + \varepsilon_{\boldsymbol{d}_i} \quad \forall i = 1,...,m. \tag{4.17}$$

4.4 Experimental Evaluation

In the experimental evaluation, we propose to compare two proposed multimodal representation methods with eight other representation methods from the literature. We want to demonstrate that the representations generated by our two multimodal methods outperform others usually used in the literature for relevant app review classification. The following sections present the app reviews collections that we use in the experimental evaluation, experimental settings, results, and discussion. All source codes are available for reproducibility purposes[1].

4.4.1 App Review Collections

We used the datasets created by Stanik, Haering and Maalej (2019) in the experimental evaluation. These datasets contain software product reviews collected from Twitter and the App Store. According to the authors, the annotation process followed a previously elaborated and validated guide. At least two people took the data labeling (three in case of disagreement). Table 6 summarizes the labeled data. The Relevant review class represents the combination of the ”Problem report” and ”Inquiry” classes from the original datasets (STANIK; HAERING; MAALEJ, 2019). There are 6,406 app reviews and 26,166 tweets. Of these, 16,770 with texts in English and 15,802 in Italian. The reviews are into two classes, “Relevant review” with 12,883 and “Irrelevant review” with 19,689.

Table 6 – Number of relevant and irrelevant reviews of the datasets used in the experimental evaluation.

	App Reviews	Tweets	
	English (ARE)	English (TEN)	Italian (TIT)
Relevant review	2,537	4,338	6.008
Irrelevant review	3,869	6,026	9,794
Total	6,406	10,364	15,802

4.4.2 Experimental Settings

We used ten representation methods for the relevant app reviews, of which seven are unimodal, and three are multimodal. Three unimodal representations came from the traditional bag-of-words (BoW) technique (AGGARWAL, 2018a). We generated BoW representations

[1] <https://github.com/GoloMarcos/MVAE-RelevantReviews.git>.

considering three different term weight schemes (AGGARWAL, 2018a): binary that indicates only whether or not the term occurs in the document; Term Frequency (tf) that defines the frequency of occurrence of a term in a document; and Term Frequency - Inverse Document Frequency (tf-idf) that weighs tf by the inverse of the number of documents in the collection in which the term occurs, according to the following equation: $tf - idf = tf_{d,t} \cdot log\left(\frac{|D|}{df_t}\right)$, in which $tf_{d,t}$ is the frequency of the term t in the document d, $|D|$ is to the number of documents, and df_t is the number of documents with the term t.

We also used the representation proposed in (STANIK; HAERING; MAALEJ, 2019) (the same study where we collected the datasets). We refer to this representation by Maalej. Maalej representation concatenates the modalities: the review sentiment, a keyword vector, a Part-of-speech (POS) tags vector, a verb tense vector, a tfidf vector, and a vector generated by the FastText Model. Also, in order to show the gains provided by the two approaches of multimodal representation learning proposed in this paper, MAE and MVAE, we considered unimodal representations generated by DBERTML model (4.3.1), the relevant app reviews topic representation (RARTR) (4.3.2), an AE with only dense layers, and a VAE. Both AE and VAE use as input the embedding generated through DBERTML. The parameters of the representation techniques and SVDD were:

- **BoW**: stopwords removal and stemming;
- **DBERTML and Maalej**: parameter-free;
- **MVAE, MAE and RARTR**: we used the k-Means with the sets of $k = \{3,6,7,8\}$, $\{\{2,3,4,5,6,7,8,9,10\}$ and $\{2,3,8,9\}\}$;
- **MVAE, MAE, VAE and AE**: learning rate = $0,001$, optimization algorithm = $\{$Adam$\}$, linear activation function, dimensions of the dense layers = $\{(512,384,128,384,512)$, $(512,256,512)$ and $(512,256,128,256,512)\}$, maximum number of epochs = $\{5,10,50\}$ and batch_size = 32;
- **MVAE and MAE**: fusion operators = $\{$addition, subtraction, concatenation, average and multiplication$\}$;
- **SVDD**: kernels = $\{RBF, Linear, Sigmoid, Polynomial\}$, degree =$\{2,3,4\}$, $\nu = \{0.001$, 0.01 and $0.05 * \nu, \nu \in [1..18]\}$, e $\gamma = \frac{1}{(n \cdot a)}$, in which n is the dimension of the input data and a is the variance of the representations.

We use two adaptations of the procedure 10-Fold Cross-Validation (TAN; STEINBACH; KUMAR, 2013) considering the OCL classification scenario. The adaptations consist of applying the procedure only to the interest class (relevant app reviews). We divided the relevant app reviews set into 10 folds. Also, in order to simulate a more realistic scenario of relevant app reviews labeling, we considered different percentages of documents from the training folds. Thus, the adaptations consist in:

- **More Labeling**: We use 9 folds to train and the remaining 1 fold to test. Also, we use the following percentage of training documents: 25%, 50%, 75%, and 100%, equivalent to 22.5%, 45%, 67.5%, and 90% of all relevant review documents in the collection;
- **Less Labeling**: We use 1 fold to train and 9 folds to test iteratively. Also, we use the following percentage of training documents: 30%, 50%, 70%, and 100%, equivalent to 3%, 5%, 7%, and 10% of all relevant review documents in the collection.

We use the F_1-Score and Area Under Curve Receiver Operating Characteristic (AUC-ROC) as evaluation measures. F_1-Score (Equation 4.22) is a harmonic average between precision (Equation 4.18) and recall (Equation 4.19). AUC-ROC computes the area under curve ROC. A ROC curve presents the relation between the TPR (equivalent to Recall) and false positive rate (FPR) (Equation 4.20) at different threshold settings. To generate the ROC curve, we use the real classes from the app reviews and the score generated by SVDD for each app review (FAWCETT, 2006). After obtaining the ROC curve, we calculate the AUC-ROC from Equation 4.21.

$$Precision = \frac{TP}{TP+FP}, \quad (4.18) \qquad Recall = \frac{TP}{TP+FN}, \quad (4.19) \qquad FPR = \frac{FP}{FP+TN}, \quad (4.20)$$

$$AUC\text{-}ROC = \int_{\infty}^{-\infty} TPR(t)FPR'(t)\,\mathrm{d}t \qquad (4.21) \qquad F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \quad (4.22)$$

in which True Positives (TP) is the number of relevant app reviews that the algorithm has correctly classified; True Negatives (TN) is the number of irrelevant app reviews that the algorithm has correctly classified; False Positives (FP) is the number of irrelevant app reviews that have been classified as relevant; False Negatives (FN) is the number of relevant app reviews classified as irrelevant; and t is a classification threshold.

4.4.3 Results and Discussion

In Tables 7, 8, 9, 10, 11, and 12 we present the highest values of F_1-Score and AUC-ROC obtained by ten app review representation techniques considering each percentage of training documents on three app review collections. We obtain the results through the SVDD algorithm. Bold values indicate that the method obtained the highest value in a percentage of training documents (column).

In relation to the ARE dataset (Tables 7 and 8), MVAE obtained better results since it obtained the highest F_1 and AUC-ROC scores than the other methods. Considering MAE, the

MVAE was better in the scenario with less labeling. Moreover, MVAE with only $76(3\%)$ relevant reviews obtained the highest F_1 and AUC-ROC scores than Maalej and all unimodal methods with $2,283(90\%)$ relevant reviews. Also, with only $127(5\%)$ relevant reviews, MVAE obtained competitive results related to MAE with $1,142(45\%)$ relevant reviews considering the F_1-Score. Furthermore, MVAE, with only $76(3\%)$ relevant reviews, obtained the highest AUC-ROC in relation to MAE with $2,283(90\%)$ relevant reviews.

Table 7 – Highest $\boldsymbol{F_1}$**-Scores** from SVDD for each representation technique on the **ARE** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#76	**#127**	**#178**	**#254**	**#570**	**#1,142**	**#1,712**	**#2,283**
Tfidf	0.55±0.02	0.57±0.02	0.58±0.01	0.60±0.01	0.60±0.03	0.63±0.01	0.64±0.02	0.65±0.02
Tf	0.54±0.00	0.54±0.00	0.54±0.00	0.54±0.00	0.57±0.00	0.57±0.00	0.57±0.00	0.57±0.00
Binary	0.54±0.00	0.54±0.00	0.54±0.00	0.54±0.00	0.57±0.00	0.57±0.00	0.57±0.00	0.57±0.00
Maalej	0.66±0.02	0.67±0.01	0.67±0.01	0.67±0.01	0.63±0.01	0.64±0.01	0.64±0.01	0.64±0.00
DBERTML	0.67±0.02	0.68±0.02	0.68±0.01	0.67±0.01	0.68±0.01	0.68±0.01	0.68±0.00	0.68±0.00
RARTR	0.66±0.03	0.66±0.02	0.65±0.02	0.64±0.02	0.65±0.02	0.64±0.01	0.64±0.01	0.64±0.02
AE	0.64±0.02	0.64±0.01	0.64±0.02	0.65±0.02	0.66±0.01	0.65±0.01	0.65±0.01	0.65±0.01
VAE	0.68±0.01	0.69±0.02	0.69±0.01	0.67±0.01	0.67±0.01	0.66±0.02	0.66±0.02	0.66±0.01
MAE	0.70±0.01	0.71±0.01	0.72±0.02	0.72±0.01	0.73±0.02	**0.75±0.02**	**0.77±0.01**	**0.78±0.01**
MVAE	**0.72±0.03**	**0.75±0.02**	**0.74±0.03**	**0.74±0.01**	**0.74±0.01**	0.73±0.01	0.73±0.02	0.72±0.02

Table 8 – Highest **AUC-ROC** from SVDD for each representation technique on the **ARE** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#76	**#127**	**#178**	**#254**	**#570**	**#1,142**	**#1,712**	**#2,283**
Tfidf	0.69±0.04	0.70±0.02	0.70±0.01	0.71±0.02	0.71±0.01	0.73±0.02	0.73±0.02	0.73±0.02
Tf	0.28±0.04	0.28±0.05	0.31±0.06	0.30±0.04	0.30±0.03	0.31±0.02	0.32±0.02	0.32±0.02
Binary	0.21±0.03	0.23±0.02	0.22±0.02	0.21±0.01	0.23±0.00	0.24±0.01	0.24±0.01	0.24±0.01
Maalej	0.79±0.01	0.80±0.00	0.80±0.00	0.80±0.00	0.76±0.01	0.77±0.01	0.77±0.01	0.77±0.01
DBERTML	0.80±0.01	0.80±0.02	0.80±0.02	0.80±0.01	0.78±0.01	0.78±0.01	0.78±0.01	0.78±0.01
RARTR	0.78±0.04	0.76±0.03	0.76±0.02	0.72±0.03	0.72±0.04	0.70±0.03	0.69±0.02	0.67±0.03
AE	0.77±0.02	0.77±0.02	0.77±0.01	0.77±0.01	0.76±0.02	0.76±0.01	0.75±0.02	0.74±0.02
VAE	0.82±0.01	0.83±0.01	0.82±0.01	0.81±0.01	0.77±0.02	0.76±0.01	0.77±0.02	0.75±0.02
MAE	0.87±0.01	0.87±0.02	0.87±0.01	0.86±0.01	0.84±0.02	0.85±0.01	**0.87±0.01**	**0.87±0.00**
MVAE	**0.88±0.01**	**0.89±0.02**	**0.88±0.01**	**0.87±0.02**	**0.86±0.01**	**0.86±0.01**	0.85±0.01	0.84±0.01

In relation to the TEN dataset (Tables 9 and 10),MAE obtained better results since it obtained the highest F_1 and AUC-ROC scores than the other methods considering more labeling. With less labeling, Maalej obtained the highest AUC-ROCs, and RARTR obtained better F_1-Scores. Maalej, with $304(7\%)$ relevant reviews, obtained the highest AUC-ROC. MAE, with $976(22.5\%)$ relevant reviews, obtained the highest F_1-Score. On the other hand, RARTR, with 130(3%) relevant reviews, obtained a highest AUC-ROC than all other methods with 3,904(90%) relevant reviews.

Table 9 – Highest **F_1-Scores** from SVDD for each representation technique on the **TEN** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#130	**#216**	**#304**	**#433**	**#976**	**#1,952**	**#2,928**	**#3,904**
Tfidf	0.51±0.01	0.52±0.01	0.53±0.01	0.54±0.01	0.55±0.01	0.56±0.02	0.57±0.02	0.57±0.01
Tf	0.56±0.00	0.56±0.00	0.56±0.00	0.56±0.00	0.59±0.00	0.59±0.00	0.59±0.00	0.59±0.00
Binary	0.56±0.00	0.56±0.00	0.56±0.00	0.56±0.00	0.59±0.00	0.59±0.00	0.59±0.00	0.59±0.00
Maalej	0.60±0.01	0.60±0.00	0.61±0.00	0.61±0.00	0.62±0.01	0.62±0.01	0.62±0.01	0.62±0.01
DBERTML	0.57±0.01	0.56±0.01	0.57±0.00	0.57±0.00	0.59±0.00	0.59±0.00	0.59±0.00	0.59±0.00
RARTR	**0.63±0.01**	**0.62±0.01**	**0.63±0.02**	**0.62±0.01**	0.63±0.02	0.63±0.00	0.63±0.00	0.62±0.00
AE	0.57±0.01	0.57±0.00	0.57±0.01	0.57±0.01	0.59±0.00	0.59±0.00	0.59±0.00	0.59±0.00
VAE	0.57±0.01	0.57±0.01	0.57±0.01	0.57±0.00	0.59±0.00	0.60±0.01	0.59±0.00	0.59±0.00
MAE	0.62±0.01	**0.62±0.01**	0.62±0.02	0.62±0.00	**0.64±0.01**	**0.64±0.01**	**0.64±0.01**	**0.64±0.01**
MVAE	0.60±0.01	0.60±0.01	0.61±0.01	0.61±0.01	0.62±0.01	0.62±0.00	0.62±0.01	0.62±0.00

Table 10 – Highest **AUC-ROC** from SVDD for each representation technique on the **TEN** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#130	**#216**	**#304**	**#433**	**#976**	**#1,952**	**#2,928**	**#3,904**
Tfidf	0.57±0.01	0.58±0.01	0.58±0.01	0.58±0.01	0.60±0.02	0.61±0.02	0.61±0.02	0.61±0.02
Tf	0.33±0.01	0.33±0.01	0.33±0.01	0.33±0.01	0.34±0.01	0.35±0.01	0.35±0.01	0.35±0.01
Binary	0.33±0.01	0.33±0.01	0.33±0.01	0.32±0.01	0.35±0.01	0.35±0.01	0.35±0.01	0.35±0.01
Maalej	0.68±0.01	**0.68±0.00**	**0.69±0.00**	**0.69±0.00**	0.65±0.01	**0.66±0.01**	0.66±0.01	0.66±0.01
DBERTML	0.61±0.01	0.60±0.01	0.61±0.01	0.61±0.00	0.60±0.01	0.60±0.01	0.60±0.01	0.60±0.01
RARTR	**0.68±0.03**	0.66±0.01	0.68±0.04	0.66±0.03	0.64±0.03	0.62±0.02	0.63±0.03	0.62±0.02
AE	0.63±0.01	0.62±0.01	0.62±0.01	0.62±0.01	0.58±0.02	0.58±0.00	0.58±0.00	0.57±0.00
VAE	0.62±0.02	0.62±0.01	0.63±0.01	0.62±0.01	0.61±0.02	0.61±0.02	0.61±0.01	0.61±0.01
MAE	0.67±0.02	0.66±0.01	0.68±0.02	0.66±0.02	**0.66±0.03**	0.65±0.01	**0.67±0.02**	**0.67±0.02**
MVAE	0.65±0.01	0.65±0.01	0.66±0.01	0.66±0.01	0.64±0.02	0.63±0.01	0.63±0.02	0.62±0.02

In relation to the TIT dataset (Tables 11 and 12), RARTR obtained better results considering less labeling. However, with more labeling, MAE obtained the highest AUC-ROCs and F_1-Scores in relation to all other methods and all other percentages of relevant reviews labeling.

Considering the results from Tables 7, 8, 9, 10, 11, and 12, our proposal obtained a higher F_1-Score and AUC-ROC in thirty-one of the forty-eight evaluated scenarios (64.6%). RARTR obtained a higher F_1-Score and AUC-ROC in thirteen of the forty-eight evaluated scenarios (27.1%). Finally, Maalej obtained a higher F_1-Score and AUC-ROC in four of the forty-eight evaluated scenarios (8.3%). In general, BoW got the lowest F_1-Score and AUC-ROC in most scenarios. AE, VAE, and DBERTML outperform the BoW representations. However, they do not obtain higher F_1-Scores and AUC-ROC considering the RARTR, Maalej, or our

Table 11 – Highest **F_1-Scores** from SVDD for each representation technique on the **TIT** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#180	**#300**	**#420**	**#608**	**#1,351**	**#2,703**	**#4,055**	**#5,407**
Tfidf	0.53±0.01	0.54±0.01	0.55±0.01	0.56±0.01	0.55±0.02	0.58±0.01	0.58±0.02	0.57±0.01
Tf	0.52±0.00	0.52±0.00	0.52±0.00	0.52±0.00	0.55±0.00	0.55±0.00	0.55±0.00	0.55±0.00
Binary	0.52±0.00	0.52±0.00	0.52±0.00	0.52±0.00	0.55±0.00	0.55±0.00	0.55±0.00	0.55±0.00
Maalej	0.58±0.01	0.58±0.01	0.58±0.01	0.58±0.01	0.60±0.00	0.60±0.00	0.60±0.00	0.60±0.00
DBERTML	0.54±0.01	0.54±0.01	0.54±0.01	0.54±0.00	0.56±0.00	0.56±0.00	0.56±0.00	0.56±0.00
RARTR	**0.61±0.03**	**0.64±0.02**	**0.63±0.02**	**0.64±0.02**	**0.65±0.02**	0.66±0.02	0.67±0.02	0.67±0.02
AE	0.55±0.01	0.55±0.01	0.55±0.01	0.55±0.01	0.56±0.00	0.56±0.00	0.56±0.00	0.56±0.00
VAE	0.55±0.01	0.56±0.01	0.55±0.01	0.55±0.02	0.57±0.01	0.57±0.01	0.57±0.00	0.56±0.01
MAE	0.59±0.02	0.61±0.03	0.61±0.01	0.62±0.01	0.65±0.01	**0.67±0.02**	**0.68±0.02**	**0.68±0.01**
MVAE	0.57±0.02	0.58±0.01	0.58±0.01	0.58±0.01	0.60±0.01	0.61±0.02	0.61±0.02	0.61±0.01

Table 12 – Highest **AUC-ROC** from SVDD for each representation technique on the **TIT** dataset. We present the values considering eight different train relevant app review percentages. We also present the number of documents equivalent to the percentage considered.

	3%	**5%**	**7%**	**10%**	**22.5%**	**45%**	**67.5%**	**90%**
	#180	**#300**	**#420**	**#608**	**#1,351**	**#2,703**	**#4,055**	**#5,407**
Tfidf	0.65±0.01	0.65±0.01	0.65±0.01	0.65±0.01	0.65±0.01	0.66±0.02	0.66±0.02	0.68±0.02
Tf	0.33±0.02	0.34±0.02	0.33±0.02	0.34±0.03	0.30±0.01	0.30±0.01	0.30±0.01	0.31±0.01
Binary	0.31±0.03	0.31±0.01	0.30±0.01	0.30±0.03	0.27±0.01	0.28±0.01	0.28±0.01	0.28±0.01
Maalej	0.70±0.00	0.70±0.00	0.70±0.00	0.70±0.00	0.70±0.01	0.70±0.01	0.70±0.01	0.70±0.01
DBERTML	0.63±0.02	0.63±0.02	0.63±0.02	0.63±0.01	0.64±0.01	0.64±0.01	0.64±0.01	0.64±0.01
RARTR	**0.73±0.04**	**0.77±0.03**	**0.75±0.02**	**0.76±0.03**	0.75±0.02	0.75±0.02	0.76±0.02	0.78±0.02
AE	0.64±0.02	0.64±0.01	0.64±0.01	0.64±0.03	0.63±0.02	0.62±0.01	0.62±0.02	0.62±0.01
VAE	0.66±0.02	0.66±0.02	0.64±0.02	0.66±0.02	0.66±0.03	0.66±0.02	0.65±0.02	0.66±0.01
MAE	0.68±0.03	0.72±0.03	0.71±0.02	0.73±0.02	**0.76±0.03**	**0.78±0.03**	**0.79±0.02**	**0.80±0.02**
MVAE	0.68±0.04	0.69±0.03	0.69±0.03	0.70±0.02	0.69±0.01	0.70±0.01	0.70±0.04	0.69±0.02

proposal. Therefore, our proposals obtained the best results. It is worth mentioning the other best results were obtained by a unimodal modality proposed by this work (RARTR) and by another multimodal method (Maalej). Thus, in general, the multimodality was better than unimodality and the unimodal modality proposed was better than other unimodal techniques.

We performed Friedman's statistical test with Nemenyi's post-test (TRAWINSKI *et al.*, 2012) to compare the representation techniques in all training percentage scenarios for each dataset and metric[2]. In Figure 18, we present the results of the Friedman test with Nemenyi's post-test through the critical difference diagram. The diagram presents the methods' average rankings, and the methods connected by a line do not present statistically significant differences between them.

[2] The test parameters were the default parameters of the KEEL tool (<https://sci2s.ugr.es/keel/index.php>).

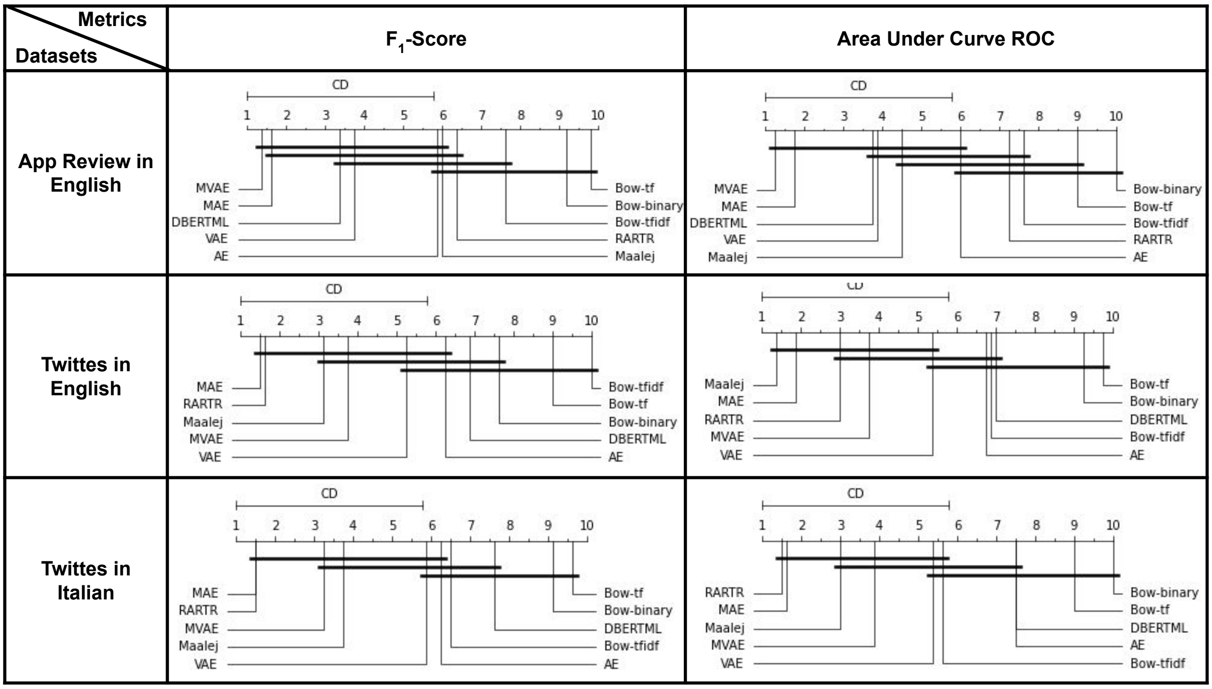


Figure 18 – Critical difference diagrams with the average rankings of the Friedman test with Nemenyi's post-test considering F_1-Score and AUC-ROC on all train scenarios and datasets.

In general, Bag-of-words models have the worst average rankings. MVAE, MAE, RARTR, and Maalej have the best average rankings. VAE and AE have the best average rankings than BoW models and worst average rankings than MVAE, MAE, RARTR, and Maalej. In the ARE dataset considering F_1-Score and AUC-ROC, MVAE has the best average ranking and presents difference statically significant considering RARTR and BoW models.

In the TEN dataset considering the F_1-Score, MAE has the best average ranking, has difference statically significant considering DBERTML and BoW models. On the other hand, considering the AUC-ROC, Maalej has the best average ranking with a statically significant difference considering AE, DBERTML, and BoW models and do not have a statically significant difference considering the other methods.

In the TIT dataset considering the F_1-Score, MAE and RARTR have the best average rankings, have difference statically significant considering DBERTML and BoW models, and do not have difference statically significant considering the other methods. Considering the AUC-ROC, RARTR has the best average ranking with difference statically significant considering AE, DBERTML, BoW-TF and BoW-Binary.

The experimental results show that DBERTML and RARTR are promising modalities that together in our proposal obtained the best average rankings. Furthermore, the best performance of the multimodal methods in different languages, app review sources, metrics, and train reviews percentages shows that learning a representation from more than one modality generates more robust representations than unimodal representations.

4.5 Threats to work validity

Threats to the internal validity of our study relate to experimenter bias. In the relevant review detection, we use data labeled by third parties. However, the work used some annotation protocols, which reduced the risk of labeling bias. Furthermore, we built our relevant class by joining two relevant classes (bug and feature). We adopted this strategy so that the work was as generic as possible, i.e., to be used in any scenario of relevant reviews, and not only with a relevant review type, for instance, detection of bug reviews or detection of feature reviews (LEDEL; HERBOLD, 2021).

Threats to external validity relate to the generalizability of our findings. We based our conclusions on the experimental evaluation of a subset of app reviews in this work. On the other hand, we use reviews of different types of apps from various domains such as google play, apple store, and Twitter to mitigate this limitation.

Finally, threats to the construct validity of our study refer to the suitability of our evaluation measures and experimental protocol. In one adaptation of the 10-fold cross-validation, we chose to keep the test set equal to the test set of (STANIK; HAERING; MAALEJ, 2019) in order to compare the results and show that it is possible to obtain satisfactory results with even less labeling of reviews. On the other hand, we use one more adaptation of the 10-fold cross-validation with less labeling and all the irrelevant reviews to mitigate this limitation. We use only an OCL algorithm (SVDD) to detect relevant reviews. However, this work focuses on representing app reviews to the relevant review detection scenario through OCL since OCL is more suitable for this scenario. Furthermore, we present the results based on the F_1 measure and AUC-ROC. F_1 is the harmonic average of two other metrics (precision and recall), and AUC-ROC presents a relation between two other metrics (fpr and recall) in different thresholds.

4.6 Concluding Remarks

Relevant reviews can help developers with software evolution and maintenance. However, the detection of relevant reviews manually is impracticable since there are a lot of reviews. Recently, there has been a significant increase in machine learning methods to detect relevant reviews. Multi-Class learning is often explored in this context, even with well-known limitations, such as the great effort to label the reviews. Another known limitation is in the review representation models, where text-based unimodal representations are usually explored. However, multimodal representations can represent app reviews to consider different aspects of relevant reviews.

This paper proposes one-class learning to detect relevant reviews through multimodal representations. We use as modalities the embedding from a context-depend language model and a representation that captures density information from these embeddings. In most scenarios, our two multimodal methods obtained higher F_1-Scores and AUC-ROC than four state-of-the-art methods and four other text representation techniques, even with less labeled reviews. This favors the scenario that most closely matches review-based software evolution and decreases the user's effort. Thus our methods proved to be promising to represent relevant review in the one-class relevant review classification.

Directions for future work involve extending our method to a multi-domain scenario. In this case, we plan to develop a pre-trained one-class model from cross-domain learning, which explores reviews from multiple apps so that detection of relevant reviews can be generalized to new apps without further training. Thus, review-based software evolution and maintenance tasks can be a practical alternative to help developers in their own apps or analyze competing apps' strengths and weaknesses.

CHAPTER

5

TRIPLE-VAE: A TRIPLE VARIATIONAL AUTOENCODER TO REPRESENT EVENTS IN ONE-CLASS EVENT DETECTION

5.1 Introduction

Nowadays, social networks and news portals share and publish different events that affect our daily lives (CHEN; LI, 2020). Social protests, pandemic effects, natural disasters, political and economic actions are examples of events that occur in a specific time and place (DENG; RANGWALA; NING, 2020). Event analysis is the field that investigates how to organize and extract knowledge from large event databases (RADINSKY; HORVITZ, 2013; ZHAO, 2021). Such knowledge is useful for exploratory analysis tasks, building decision-making indicators, and improving machine learning models by providing new (extra) features on the world's external factors. A crucial step in event analysis is filtering which events are interesting for a given application, as thousands of events are published daily. Event classification methods usually carried out this step considering the textual information of an event, as well as its geographic information and other metadata (SETTY; HOSE, 2018; ZHAO, 2021).

Recent event classification methods have some limitations (ZHOU *et al.*, 2020; CHEN; LI, 2020; ZHAO, 2021). The first limitation is to propose event classification considering a multi-class scenario, a decision that makes the practical use of the model unfeasible. In this case, the event dataset's volume, diversity, and frequent update rate surpass the human capacity to label and maintain a training set. Some studies model the event classification as a binary problem (ZHOU *et al.*, 2020), in which the positive class identifies events of interest and the negative class defines non-relevant events. However, both classes require significant labeling of training examples. In this sense, the one-class learning paradigm is a promising alternative as it requires labeling only of events of interest (ALAM *et al.*, 2020).

A second limitation is the event representation model (ZHOU *et al.*, 2020). Events are composed of textual information, geographic location, names of people, organizations, and other metadata. Traditional methods usually concatenate these different features into a single representation that is used to train a model. More recent methods explore representation learning, such as deep autoencoders, to extract a new latent space (embeddings) from the concatenated representation of features (BLANDFORT *et al.*, 2019). Although both strategies obtain competitive results, few studies evaluate the performance of these representation strategies for event analysis. We argue that different information from events represents different data modalities misused through concatenation strategies. Thus, our focus is to explore event representation as a multimodal representation learning task.

This paper presents an approach to learning multimodal representation for one-class event classification. Our approach is called Triple-VAE and explores three main event modalities: textual information, geographic location, and topic metadata. First, we propose a multimodal variational autoencoder capable of learning a single representation from triple modalities. Unlike concatenation-based methods, our approach merges modalities more naturally, automatically learning the importance of modalities in the final representation. Second, we also argue that our approach is more appropriate for one-class classification since it learns a representation space that approximates events of interest in high-density regions — which significantly improves the event classification step. In short, our proposed approach has the following contributions:

- We naturally incorporate latitude and longitude data into the embedding space, along with textual and topic information. However, previous studies use geographic location only as extra features in the concatenated representation.
- We leverage pre-trained neural language models to represent events. In particular, we use the DistilBERT Multilingual model, which is trained in a large textual corpus and has some general-purpose knowledge. In practice, this is a strategy to carry out transfer learning from the pre-trained model for our multimodal representation learning.
- We explore event topic information as a visual modality, in which each topic represents high-density information in a given dimensional space. In fact, density information facilitates visual exploration of the spatial distribution of events, thus providing useful information about related events.

We carried out an experimental evaluation involving ten real-world event datasets. We compared our proposed approach with the other seven multimodal and three unimodal strategies. Our proposal outperforms existing methods considering F_1-Score and accuracy in the most explored scenarios, especially the more realistic scenario considering less labeling.

The proposed methods and experimental results presented in this chapter were published in a paper titled "*Triple-VAE: A Triple Variational Autoencoder to Represent Events in One-Class*

Event Detection" (GÔLO; ROSSI; MARCACINI, 2021), which was selected as the best paper in XVIII National Meeting on Artificial and Computational Intelligence (ENIAC 2021).

5.2 Related Work

The proposal presented in Zeppelzauer and Schopfhauser (2016) uses texts and images as modalities to perform event detection. Both modalities are unstructured and therefore need preprocessing. The authors preprocessed the text using the Bag-of-Words (BoW) and the dimensionality reduction technique Latent Dirichlet Allocation. The study uses the bag-of-visual-words for representation in the image modality. After representing the text and image, the authors explored both modalities through early and late fusions. Early fusion is made through the concatenation of the representations, while late fusion is made through additive and hierarchical late fusion. The approach uses a binary classifier to ignore non-relevant events, and then multi-class learning is applied to built classification models. The experimental evaluation shows that using two modalities improves the event classification task. It is worth pointing out that early fusion outperforms late fusion.

In Kang and Kang (2017), the authors use multi-class learning to predict crime events defined by visual (neighborhood appearance), spatial and temporal information. First, the study uses a Convolutional Neural Network (CNN) to represent the images, and the spatial and temporal data are already structured. Then, a Deep Neural Network (DNN) is used to predict if the event is a crime. In the DNN, the authors use the early fusion with the concatenation operator and a softmax activation function in the output layer. Results show that the DNN outperforms the Kernel Density Estimation and Support Vector Machines (SVM) algorithms with a simple concatenation of modalities.

Blandfort *et al.* (2019) explores the detection of gang violence events through Twitter. The authors use text and image modalities to represent the events. The study uses Linguistic features, word embeddings, and a CNN to learn text representations. Furthermore, the authors use the Faster R-CNN and global image features generated by the deep convolutional model Inception-v3 to represent the images. The study uses the early fusion considering the concatenation operator, and the late fusion is performed considering an ensemble of algorithms trained on each modality. The authors use the multi-class SVM learning algorithm. Results show that multimodal representations outperform unimodal ones. Early fusion presents high results than late fusion.

Zhou *et al.* (2020) is a survey of multimodal event detection. The authors compare studies that make event detection through multi-class or binary learning, clustering, and other tasks considering unimodal and multimodal representations. The study shows that multimodality can be promising with representation learning through artificial neural networks. Furthermore, the work concludes that concatenate the modalities can result in a representation with a high dimension which can negatively affect event detection. The survey also points that future work should involve information enrichment, i.e., news modalities to enrich the representation learned and the use of different state-of-the-art neural network architectures in representation learning.

We observed that existing multimodal studies for event detection (i) use multi-class learning, which generates more user's effort on labeling, and if a new class arises, the classifier will make wrong predictions since it was not trained on that event class; (ii) use binary learning, which generates less user's effort on labeling, and the chance of the user not labeling events of one of the classes is smaller in comparison with the multi-class learning. However, labeling uninteresting events requires knowing a wide range of classes so that the user cannot label enough examples; (iii) do not explore other early fusion operators such as addition, subtraction, multiplication, and average; (iv) use the event image as a modality and, consequently, its models only work on events with images (note that many events do not have associated images); and (v) shows that early fusion outperforms late fusion.

Given all these facts and gaps mentioned in this section and the future work presented in the survey Zhou *et al.* (2020), we propose an event detection approach considering one-class learning (OCL) over a multimodal representation. OCL avoids multi-class and binary learning limitations since the user labels only one class and classifies a new example as belonging to the interest class or not. Also, we propose the generative model variational autoencoder (VAE) to learn a representation from a set of three modalities: the event text, geolocation, and density information. We considered a VAE because it is a powerful method to learn representations since it is one of the state-of-the-art in representation learning. Furthermore, we propose to use early fusion with different fusion operators. We present the details of the proposed approach in the next section.

5.3 Proposal: One-Class Multimodal Event Detection

According to Zhou *et al.* (2020), an event is: *"A story related to some news topic comprising of patterns that occurred at some specific time and space"*. Based on this definition, we define an event ($\boldsymbol{e}$) as its text representation ($\boldsymbol{\gamma}$), its density information ($\boldsymbol{\lambda}$), its geolocation ($\boldsymbol{\iota}$), and its date (τ). Therefore, we formally define an event $\boldsymbol{e}_i$ by the quadruple:

$$\boldsymbol{e_i} = \{\boldsymbol{\gamma}_i, \boldsymbol{\lambda}_i, \boldsymbol{\iota}_i, \tau_i\} \tag{5.1}$$

In this multimodal scenario, we propose a pipeline to detect events through OCL via multimodal representation (Figure 19). The pipeline has six steps. The first step consists of collecting events with description, geolocation, and date. In the second step, we represent the event's text through a neural language model. In the third step, we use a modality based on density information generated from the text's representation. In the fourth step, a variational autoencoder learns a multimodal representation from the three modalities ($\boldsymbol{\gamma}, \boldsymbol{\lambda}$ and $\boldsymbol{\iota}$), considering events that occurred prior to date τ. We use events that occur after date τ to evaluate the event classification model. The fifth step consists in using OCL to learn a decision function. Finally, in the sixth step, we make the detection of the events of interest.

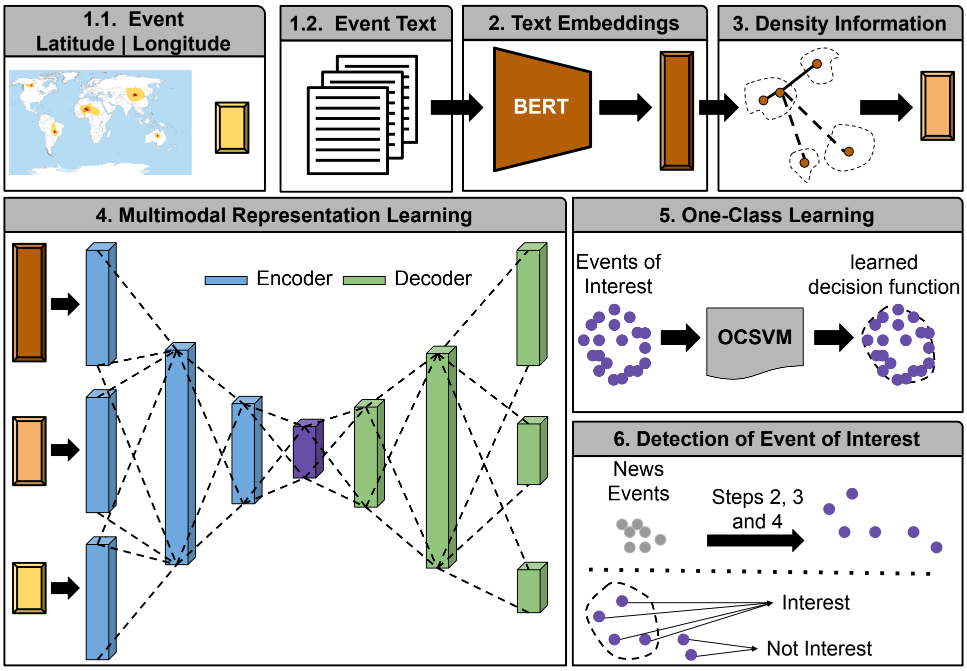


Figure 19 – The Pipeline of multimodal representation Learning to detect events of interest through one-class learning.

5.3.1 Event Geolocation

We obtain the event geolocation through the Latitude and Longitude coordinates. Latitude and Longitude refer to a place's position or geographic coordinates on Earth. Latitude ranges from -90 to 90, in which -90 represents the south pole, 90 represents the north pole, and 0 represents the Earth's equator. Longitude ranges from -180 to 180, in which values $\in [-180, 0)$ represent places in the west, values $\in (0, 180]$ represent places in the east, and 0 represents the Greenwich meridian. Therefore, modality $\boldsymbol{\iota}$ is a vector with two dimensions in which the first is the Latitude and the second is the Longitude.

5.3.2 Text Embeddings

One of the states-of-the-art to represent text is the context-dependent neural language model Bidirectional Encoder from Transformers (BERT) (DEVLIN *et al.*, 2019). It is noteworthy that this model obtains better results in natural language processing tasks than other models, such as based on word embeddings models or traditional models (e.g., BoW) (OTTER; MEDINA; KALITA, 2020). Therefore, we use the Distilled version of BERT in its multilingual version (DBERTML) (REIMERS; GUREVYCH, 2020) to represent the event's text. First, we use the sentence-transformers library[1] to use the model DBERTML. Then, we make the preprocessing,

[1] https://www.sbert.net/

providing the text to the pre-trained DBERTML model, and it returns an embedding $\boldsymbol{\gamma}_i$ with 512 real values. Details of the model DBERTML and its training parameters to obtain the embeddings are available in (REIMERS; GUREVYCH, 2020).

5.3.3 Density Information

We explore a modality based on density information. This modality is based on an OCL assumption, which assumes that high-density regions contain examples of the interest class (KRAWCZYK; WOŹNIAK; CYGANEK, 2014; SHARMA; SOMAYAJI; JAPKOWICZ, 2018). To obtain the events density information, we apply a clustering algorithm and use a statistical technique that calculates the consistency within data clusters.

We use the silhouette coefficient (ROUSSEEUW, 1987) to generate the density information ($\boldsymbol{\lambda}$). In this modality, the density representation $\boldsymbol{\lambda}_i$ of an event $\boldsymbol{e}_i$ is given by the concatenation of silhouette values computed considering a different number of clusters. Thus, given u different clustering settings, $\boldsymbol{\lambda}_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,u}\}$, in which $s_{i,j}$ is the silhouette of $\boldsymbol{\gamma}_i$ in j-th setting, and $s(\boldsymbol{\gamma}_i, k)$ is given by:

$$s(\boldsymbol{\gamma}_i, k) = \frac{\beta(\boldsymbol{\gamma}_i) - \alpha(\boldsymbol{\gamma}_i)}{\max(\alpha(\boldsymbol{\gamma}_i), \beta(\boldsymbol{\gamma}_i))} \tag{5.2}$$

in which k is the number of clusters, $2 \leq k < m$ and m is the number of events, $\alpha(\boldsymbol{\gamma}_i)$ is the average distance of $\boldsymbol{\gamma}_i$ to the centroid of its cluster, and $\beta(\boldsymbol{\gamma}_i)$ defines the average distance of $\boldsymbol{\gamma}_i$ to all $\boldsymbol{\gamma}$ of the closest cluster.

The event scenario has high-density regions representing well-defined topics of the events (BIDE; DHAGE, 2021). Furthermore, density information can be explored as a visual modality to analyze the spatial distribution of events. However, we highlight that the use of density information as a modality for events is still unexplored in literature.

5.3.4 Multimodal Representation Learning

After we have $\boldsymbol{\iota}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$ represented, we use a variational autoencoder with multimodal representation learning to learn a joint representation. An autoencoder (AE) is a neural network that learns data representations using two steps: encoding and decoding (AGGARWAL, 2018a). First, the encoder ($f()$) compresses $\boldsymbol{e}_i$ to a latent space $\boldsymbol{z}_{e_i}$. Then, the decoder ($g()$) reconstructs $\boldsymbol{e}_i$ from $\boldsymbol{z}_{e_i}$ in the output $\boldsymbol{r}_{e_i}$. Thus, the training of AE consists of making $\boldsymbol{r}_{e_i} \approx \boldsymbol{e}_i$. The encoder and decoder, and the AE optimization function are respectively given by:

$$autoencoder = \begin{cases} \mathbf{z}_{e_i} = f(\Phi; \mathbf{e}_i) \\ \mathbf{r}_{e_i} = g(\Theta; \mathbf{z}_{e_i}) \end{cases} \tag{5.3}$$

$$J(\Phi;\Theta) = \frac{1}{m}\sum_{e_i} \|\mathbf{e}_i - \mathbf{r}_{e_i}\|^2 \tag{5.4}$$

in which Φ is the weights and biases of neurons in the encoder, Θ is the weights and biases of neurons in the decoder. Thus, the AE is adequate in scenarios with examples belonging to one class because it trains in an unsupervised way.

There are variations of the AE that impose constraints on the hidden units (AGGARWAL, 2018a). For instance, the Variational Autoencoder (VAE) constraint that the activation in the hidden units should be drawn from the standard Gaussian with zero mean and unit variance (XU; DURRETT, 2018). This constraint also allows generating samples of the training data just feeding the decoder with samples generated from a normal distribution. Formally, the VAE assumes that a variable $\mathbf{z}_{e_i}$ generates the data $\mathbf{e}_i$ (Equation 5.5).

$$p(\mathbf{z}_{e_i}|\mathbf{e}_i) = \frac{p(\mathbf{e}_i|\mathbf{z}_{e_i})p(\mathbf{z}_{e_i})}{p(\mathbf{e}_i)} \quad (5.5) \qquad p(\mathbf{e}_i) = \int p(\mathbf{e}_i|\mathbf{z}_{e_i})p(\mathbf{z}_{e_i})d\mathbf{z}_{e_i} \quad (5.6)$$

VAE approximates $p(\mathbf{z}_{e_i}|\mathbf{e}_i)$ to another treatable distribution $q(\mathbf{z}_{e_i}|\mathbf{e}_i)$ using the Kullback-Leibler (KL) divergence, which is responsible for measuring the divergence between two distributions. To optimize the marginal likelihood ($p(\mathbf{e}_i)$), you can use the log of the marginal likelihood (XU; DURRETT, 2018):

$$\log p_\Theta(\mathbf{e}_i) = KL(q_\Phi(\mathbf{z}_{e_i}|\mathbf{e}_i)||p_\Theta(\mathbf{z}_{e_i}|\mathbf{e}_i)) + \mathscr{L}(\Theta,\Phi;\mathbf{e}_i) \tag{5.7}$$

in which

$$\mathscr{L}(\Theta,\Phi,\mathbf{e}_i) = \mathbb{E}_{q_\Phi(\mathbf{z}_{e_i}|\mathbf{e}_i)} \log p_\Theta(\mathbf{e}_i|\mathbf{z}_{e_i}) - KL(q_\Phi(\mathbf{z}_{e_i}|\mathbf{e}_i)||p_\Theta(\mathbf{z}_{e_i})) \tag{5.8}$$

We implement a VAE using a neural network. Thus, it learns the encoder's Φ parameters and the decoder's Θ parameters through the weights of the neurons of the neural network layers. The first term of Equation 5.8 is related to the neural network reconstruction error. In the second term, we want to minimize the difference between the learned distribution $q_\Phi(\mathbf{z}_{e_i}|\mathbf{e}_i)$ and $p_\Theta(\mathbf{z}_{e_i})$ (prior knowledge). It is worth mentioning that literature studies replace the term $p_\Theta(\mathbf{z}_{e_i})$ by a multivariate Gaussian distribution $\mathscr{N}(\mathbf{z}_{e_i};0,1)$.

In this paper, we propose a Triple-VAE: a VAE that learns from three modalities. Therefore, the Triple-VAE has three inputs and three outputs. To learn a representation from three modalities, Triple-VAE combines them through early fusion. We opt to use the early fusion because of the advantage of using only one representation for the events in the classification step. Furthermore, to deal with the challenge of combine modalities with different dimensions, we use three dense layers with the same number of neurons that receive the inputs of Triple-VAE (Figure 19). The proposed architecture allows us to combine the modalities with different literature fusion operators.

Our proposed Triple-VAE aims to maximize Equation 5.9. Given an event $\boldsymbol{e}_i$, the first term calculates the reconstruction errors of $\boldsymbol{\gamma}_{e_i}, \boldsymbol{\lambda}_{e_i}, \boldsymbol{\iota}_{e_i}$. The second term wants to approximate the learned distribution $q_{\Phi}(\boldsymbol{z}_{e_i}|\boldsymbol{\gamma}_{e_i},\boldsymbol{\lambda}_{e_i},\boldsymbol{\iota}_{e_i})$ from $p_{\Theta}(\boldsymbol{z}_{e_i})$.

$$\begin{aligned}\mathscr{L}(\Theta,\Phi,\boldsymbol{\gamma}_{e_i},\boldsymbol{\lambda}_{e_i},\boldsymbol{\iota}_{e_i}) = \mathbb{E}_{q_{\Phi}(\boldsymbol{z}_{e_i}||\boldsymbol{\gamma}_{e_i},\boldsymbol{\lambda}_{e_i},\boldsymbol{\iota}_{e_i})} &\log p_{\Theta}(|\boldsymbol{\gamma}_{e_i},\boldsymbol{\lambda}_{e_i},\boldsymbol{\iota}_{e_i}|\boldsymbol{z}_{e_i}) \\ -KL(q_{\Phi}(\boldsymbol{z}_{e_i}|\boldsymbol{\gamma}_{e_i},\boldsymbol{\lambda}_{e_i},\boldsymbol{\iota}_{e_i})&||p_{\Theta}(\boldsymbol{z}_{e_i}))\end{aligned} \tag{5.9}$$

5.3.5 One-Class Learning to Detect Events of Interest

After we have a multimodal representation for events, we are able to classify them. We use one-class learning (OCL) (TAX, 2001) to detect events of interest. In the OCL, the algorithms train using only examples of the interest class. Thus, we do not suffer from new event categories or not knowing a non-relevant event category. Moreover, even if the user is interested in a single event category, the OCL is most appropriate since OCL does not label examples of classes that are not the interest class (ALAM *et al.*, 2020). Another advantage of OCL over multi-class or binary learning is (i) the user has less effort in data labeling; and (ii) it is more appropriate in unbalanced scenarios (FERNÁNDEZ *et al.*, 2018).

Let the domain of events be $\mathscr{E}$, and the domain of labels be $\mathscr{Y}$, in which $y_i = \{+1,-1\}$ for $y_i \in \mathscr{Y}$ and +1 represents the label of the interest class, while -1 represents the label of the non-interest class. Then, given a set of m training events $\{(\mathbf{e}_j, y_j)\}_{j=1}^{m}$, in which $y_j = +1$, the goal of OCL is to learn a function $f : \mathscr{E} \rightarrow \mathscr{Y}$ given only labeled events from the interest class. After learning the function f, the classifier is able to predict y_i for a new event $\mathbf{e}_i$ comparing $f(\mathbf{e}_i)$ with a threshold as presented in Equation 5.10.

$$y_i = \begin{cases} +1 \text{ (Interest)} & \text{if } f(\mathbf{e}_i) \leq \mathit{threshold} \\ -1 \text{ (Non Interest)} & \text{otherwise} \end{cases} \tag{5.10}$$

Among the OCL algorithms (GÔLO; MARCACINI; ROSSI, 2019), we chose the One-Class Support Vector Machine (OCSVM) (TAX; DUIN, 2004) since it is considered state-of-the-art in OCL (ALAM *et al.*, 2020). The training of OCSVM consists of finding and hypersphere of minimum volume that involves the training events. The center of the hypersphere is defined in Equation 5.11 (TAX; DUIN, 2004):

$$\mu_{(c)} = \underset{\mu \in U}{\arg\min} \max_{1 \leq i \leq m} ||\varphi(\boldsymbol{e}_i) - \mu||^2 \tag{5.11}$$

in which m is the number of events of interest, U is the feature space associated with the function kernel φ, $\mu_{(c)}$ is the center of the hypersphere. Since the goal is to obtain the hypersphere with minimum volume, we minimize the radius (r), i.e., r^2. Slack variables ($\varepsilon_i \geq 0$) can also allow a trade-off between hypersphere volume and coverage of the training events. Then, the constraint that almost all training events are within the sphere is given by Equation 5.12. OCSVM aims to minimize Equation 5.13 subject to Equation 5.12.

$$\|\varphi(\boldsymbol{e}_i) - \mu_{(c)}\|^2 \leq r^2 + \varepsilon_{\boldsymbol{e}_i}, \quad \forall i = 1, \ldots, m, \ \varepsilon_{\boldsymbol{e}_i} \geq 0 \qquad (5.12)$$

$$\min_{\mu,\varphi,r} \quad r^2 + \frac{1}{m} \sum_{i=1}^{m} \frac{\varepsilon_{\boldsymbol{e}_i}}{\nu} \qquad (5.13)$$

In which $\nu \in (0, 1]$ is a parameter to control the trade-off between the radius and the errors so that the hypersphere is not too large and the false positive rate increases. We will consider an event as belonging to the interest class if its distance from the center is less than the radius r of the hypersphere, i.e., $f(\boldsymbol{e}_i) = dist(\varphi(\boldsymbol{e}_i), \mu_{(c)})$ and $threshold = r$.

5.4 Experimental Evaluation

In the experimental evaluation, we compared our proposed Triple-VAE with the other seven multimodal and three unimodal literature representation methods. We want to demonstrate that the representations generated by Triple-VAE outperform others usually explored in the literature for event detection. We use the OCSVM algorithm to compare all the events representation methods. The next subsections present the event collections, experimental settings, results, and discussion. All source codes that we use in the experimental evaluation are available online[2].

5.4.1 Event Collections and Experimental Settings

We obtain the event collections from the GDELT project, which monitors real-time events worldwide. We collect ten datasets in which each dataset represents a theme and contains 6000 events. We populate the datasets by using the google cloud big query. We use the same process of collection described in Gôlo, Rossi and Marcacini (2021), however, collecting the geolocation from the events.

We compare our Triple-VAE with three unimodal strategies in which each of one is the inputs of the Triple-VAE, i.e., DBERMTL, Latitude and Longitude (Lat-Long), and Density Information. Also, we compare our Triple-VAE with seven multimodal strategies. Four are Bimodal strategies considering the DBERTML and Lat-Long as modalities. In this context, we explore the Concatenation (Concat) of the modalities, an AE applied in the Concat representation, a VAE applied in the Concat representation, and a Bi-VAE that learns a representation in the same way as our proposal but only considering two modalities. Moreover, we evaluate trimodal strategies considering the DBERTML, Lat-Long, and Density as modalities. The parameters from the seven multimodal strategies, three unimodal strategies, our proposal (Triple-VAE), and the OCSVM algorithm are:

- **DBERTML, Lat-Long and Concat (Bi-Modal)**: parameters free;

[2] <https://github.com/GoloMarcos/TripleVAE-Events>.

- **Density and Trimodals**: we used the k-Means with the sets of $k = \{\{3,6,9,12\}, \{2,4,6,8,10\}, \{3,5,7,9,11\}, \{2,3,4,5,6,7,8,9,10,11\}\}$;
- **Triple-VAE, Bi-VAE, VAE and AE**: learning rate = 0.001, seed = 1, optimization algorithm = $\{$Adam$\}$, maximum number of epochs = $\{5,8,10,50\}$, linear activation function, dense layers dimensions= $\{(512,128,512), (512,384,128,384,512), (512,384,512)$, and $(512,384,266,128,266,384,512)\}$, and batch_size = 32;
- **Triple-VAE and Bi-VAE**: fusion operators = $\{$addition, subtraction, concatenation, average and multiplication$\}$;
- **OCSVM**: $kernels = \{rbf, linear, sigmoid, polynomial\}$, the kernel coefficients $degree$ $=\{2,3,4\}$ and $gamma = \{1/(na), 1/n\}$, in which n is the dimension of the input data and a is the variance of the representations, and $\nu = \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1 * h, h \in [1..9]\}$.

We chose some specific experimental setups to simulate a more suitable scenario closer to the real-world applications. First, we use event dates to separate training and testing. Events with older dates are from the training set. Second, we explored the use of a few labeled examples in the training set. Thus, we explored the use of 60, 120, 180, and 2000 events in training set. In the test, we use 4000 events. Also, we randomly selected 4000 events from different event datasets and added them to the test set in order to have counter-examples of the interest class during the evaluation process.

We analyze the classification performances using F_1-Score (Equation 5.14), accuracy (Equation 5.19), and Area Under Curve Receiver Operating Characteristic (AUC-ROC). F_1-Score is a harmonic average between precision (Equation 5.15) and recall (Equation 5.16). AUC-ROC (Equation 5.18) computes the area under curve ROC. A ROC curve presents the relation between the tpr (equivalent to recall) and false positive rate (fpr) (Equation 5.17) at different threshold settings.

$$f1 = \frac{2 \cdot p \cdot r}{p + r} \quad (5.14) \qquad p = \frac{tp}{tp + fp} \quad (5.15) \qquad r = \frac{tp}{tp + fn} \quad (5.16) \qquad fpr = \frac{fp}{fp + tn} \quad (5.17)$$

$$auc\text{-}roc = \int_{\infty}^{-\infty} tpr(t) fpr'(t)\, \mathrm{d}t \quad (5.18) \qquad acc = \frac{tp + tn}{tp + tn + fp + fn} \quad (5.19)$$

In the equations presented above, tp (true positives) is the number of events of interest that the algorithm has correctly classified; tn (true negatives) is the number of non-interest events that the algorithm has correctly classified; fp (false positives) is the number of non-interest events that have been classified as interest; fn (false negatives) is the number of events of interest classified as non-interest; and t is a classification threshold.

5.4.2 Results and Discussion

Tables 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, and 24 present the experimental results (F_1-Score, accuracy, and AUC-ROC) in the ten event datasets. The results consist in the F_1-Score, accuracy, and AUC-ROC, among all representations method parameters and OCSVM parameters. Bold values indicate that the method obtained the highest value considering each metric.

Considering the AUC-ROC in all scenarios (Tables 14, 17, 20, 23) and the F_1-Score and accuracy with 2000 events for training (Tables 22, 24), DBERTML obtain the highest values. In all other scenarios (Tables 13, 15, 16, 18, 19 and 21), i.e., F_1-Score and accuracy with 60, 120, and 180 events for training, the Triple-VAE obtain the highest values. Therefore, DBERTML was the best method for event detection in the scenario with more labeled events. On the other hand, considering the scenario with fewer labeled events closer to real-world applications, Triple-VAE was the best method for event detection as it achieved the best F_1-Score and accuracy values.

Table 13 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	Lat-Long	Density	Concat	AE	VAE	BiVAE	Concat	AE	VAE	TripleVAE
War	0,663	0,672	0,730	0,675	0,678	0,682	0,721	0,675	0,680	0,684	**0,780**
Tsunami	0,881	0,664	0,727	0,664	0,671	0,669	0,897	0,664	0,672	0,673	**0,918**
Covid	0,841	0,664	0,754	0,688	0,743	0,741	**0,952**	0,688	0,741	0,736	0,946
Corruption	0,728	0,692	0,681	0,694	0,654	0,640	0,939	0,694	0,641	0,652	**0,958**
Earthquake	0,866	0,668	0,676	0,668	0,680	0,687	0,896	0,668	0,681	0,684	**0,916**
Immigration	0,748	0,674	0,691	0,670	0,699	0,697	0,942	0,674	0,692	0,689	**0,950**
Racism	0,825	0,691	0,685	0,663	0,674	0,671	0,961	0,663	0,675	0,675	**0,964**
Inflation	0,809	0,670	0,664	0,670	0,685	0,667	0,930	0,671	0,672	0,679	**0,953**
Terrorism	0,768	0,669	0,709	0,671	0,667	0,670	0,908	0,672	0,668	0,668	**0,937**
Agriculture	0,757	0,666	0,689	0,667	0,666	0,662	0,970	0,667	0,665	0,665	**0,979**

Table 14 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	Lat-Long	Density	Concat	AE	VAE	BiVAE	Concat	AE	VAE	TripleVAE
War	**0,904**	0,640	0,778	0,628	0,646	0,643	0,772	0,628	0,650	0,654	0,856
Tsunami	**0,983**	0,630	0,767	0,620	0,648	0,648	0,947	0,621	0,654	0,667	0,975
Covid	**0,997**	0,683	0,864	0,699	0,792	0,803	0,991	0,700	0,800	0,794	0,996
Corruption	**0,995**	0,656	0,676	0,659	0,652	0,646	0,983	0,659	0,654	0,652	0,989
Earthquake	0,965	0,636	0,754	0,655	0,653	0,653	0,958	0,655	0,653	0,668	**0,972**
Immigration	**0,996**	0,685	0,668	0,690	0,732	0,738	0,989	0,690	0,726	0,731	0,989
Racism	**0,997**	0,669	0,701	0,701	0,708	0,710	0,995	0,701	0,711	0,712	0,996
Inflation	**0,999**	0,606	0,591	0,610	0,624	0,622	0,981	0,610	0,621	0,611	0,991
Terrorism	**0,994**	0,648	0,725	0,651	0,618	0,618	0,984	0,651	0,621	0,621	0,986
Agriculture	**0,996**	0,589	0,705	0,584	0,632	0,623	0,992	0,584	0,631	0,606	0,995

We performed an analysis in relation to the Triple-VAE using 1% labeling (60 events) compared to the other methods using 33% labeling (2000 events). This comparison is shown in Tables 25 and 26. Triple-VAE achieved a higher F_1-Score and Accuracy than all other methods in

Table 15 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 60 (1%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML | Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	0,730	0,605	0,719	0,603	0,619	0,633	0,703	0,605	0,634	0,630	**0,779**
Tsunami	0,892	0,593	0,768	0,591	0,605	0,615	0,897	0,591	0,613	0,610	**0,917**
Covid	0,849	0,642	0,764	0,702	0,735	0,727	**0,952**	0,679	0,732	0,737	0,949
Corruption	0,787	0,625	0,626	0,637	0,598	0,598	0,941	0,640	0,599	0,599	**0,958**
Earthquake	0,874	0,589	0,674	0,575	0,619	0,632	0,892	0,575	0,615	0,626	**0,915**
Immigration	0,800	0,626	0,607	0,622	0,707	0,690	0,945	0,622	0,680	0,682	**0,952**
Racism	0,851	0,654	0,644	0,649	0,664	0,663	0,961	0,648	0,665	0,665	**0,965**
Inflation	0,821	0,580	0,562	0,582	0,594	0,591	0,934	0,584	0,595	0,584	**0,954**
Terrorism	0,811	0,604	0,705	0,609	0,591	0,585	0,912	0,610	0,594	0,591	**0,938**
Agriculture	0,805	0,557	0,658	0,556	0,603	0,604	0,970	0,558	0,605	0,593	**0,979**

Table 16 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML | Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,813**	0,672	0,688	0,673	0,679	0,682	0,771	0,673	0,680	0,681	0,778
Tsunami	**0,909**	0,673	0,674	0,664	0,666	0,662	0,875	0,664	0,667	0,670	0,902
Covid	0,674	0,673	0,740	0,698	0,770	0,771	**0,935**	0,688	0,776	0,772	**0,935**
Corruption	0,904	0,691	0,735	0,690	0,657	0,652	0,930	0,689	0,664	0,665	**0,956**
Earthquake	0,858	0,668	0,664	0,668	0,682	0,684	0,870	0,668	0,688	0,688	**0,917**
Immigration	0,914	0,674	0,678	0,671	0,677	0,677	0,950	0,673	0,677	0,677	**0,956**
Racism	0,880	0,689	0,720	0,665	0,684	0,683	0,969	0,665	0,689	0,689	**0,976**
Inflation	0,862	0,674	0,662	0,674	0,668	0,667	0,941	0,673	0,669	0,669	**0,955**
Terrorism	0,843	0,673	0,667	0,676	0,671	0,674	0,937	0,672	0,675	0,672	**0,944**
Agriculture	0,863	0,665	0,680	0,665	0,673	0,670	0,952	0,665	0,670	0,670	**0,978**

Table 17 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML | Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,916**	0,635	0,712	0,631	0,664	0,662	0,815	0,631	0,666	0,665	0,812
Tsunami	**0,984**	0,638	0,707	0,625	0,636	0,641	0,943	0,625	0,664	0,641	0,967
Covid	**0,997**	0,728	0,788	0,758	0,854	0,858	0,992	0,760	0,861	0,860	0,994
Corruption	**0,996**	0,656	0,742	0,659	0,623	0,603	0,980	0,659	0,613	0,602	0,989
Earthquake	0,964	0,628	0,685	0,634	0,636	0,641	0,962	0,634	0,648	0,644	**0,974**
Immigration	**0,996**	0,679	0,617	0,685	0,696	0,697	0,992	0,685	0,697	0,696	0,993
Racism	**0,997**	0,677	0,784	0,699	0,717	0,716	0,996	0,700	0,720	0,718	**0,997**
Inflation	**0,999**	0,622	0,664	0,627	0,650	0,637	0,979	0,627	0,635	0,635	0,987
Terrorism	**0,994**	0,644	0,684	0,648	0,704	0,706	0,981	0,649	0,708	0,697	0,984
Agriculture	**0,997**	0,604	0,647	0,586	0,651	0,654	0,991	0,585	0,653	0,649	0,996

the Corruption, Earthquake, Immigration, Racism, Inflation, Terrorism, Agriculture collections (7 of 10 datasets). Furthermore, in the War and Tsunami collections, Triple-VAE achieved a higher F_1-Score and Accuracy than all other methods except DBERTML.

Table 18 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 120 (2%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,824**	0,609	0,649	0,603	0,625	0,623	0,741	0,609	0,625	0,627	0,757
Tsunami	**0,914**	0,602	0,674	0,601	0,609	0,605	0,873	0,601	0,618	0,609	0,902
Covid	0,743	0,646	0,747	0,733	0,758	0,761	0,938	0,724	0,763	0,765	**0,939**
Corruption	0,912	0,628	0,687	0,630	0,613	0,605	0,929	0,630	0,606	0,606	**0,957**
Earthquake	0,866	0,610	0,651	0,622	0,610	0,621	0,871	0,626	0,617	0,628	**0,918**
Immigration	0,921	0,624	0,583	0,624	0,626	0,625	0,951	0,624	0,635	0,648	**0,957**
Racism	0,894	0,641	0,723	0,647	0,671	0,674	0,970	0,647	0,671	0,673	**0,977**
Inflation	0,879	0,589	0,622	0,598	0,605	0,603	0,944	0,596	0,607	0,595	**0,957**
Terrorism	0,863	0,611	0,664	0,615	0,620	0,621	0,939	0,615	0,626	0,620	**0,946**
Agriculture	0,880	0,551	0,607	0,566	0,634	0,634	0,953	0,570	0,635	0,635	**0,978**

Table 19 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,841**	0,675	0,694	0,671	0,669	0,671	0,777	0,671	0,674	0,669	0,774
Tsunami	**0,915**	0,683	0,677	0,663	0,672	0,664	0,866	0,663	0,678	0,669	0,914
Covid	0,864	0,679	0,765	0,698	0,794	0,789	0,936	0,708	0,796	0,799	**0,952**
Corruption	0,924	0,686	0,725	0,691	0,661	0,658	0,921	0,690	0,660	0,659	**0,955**
Earthquake	0,865	0,668	0,694	0,667	0,679	0,686	0,872	0,668	0,685	0,686	**0,935**
Immigration	0,933	0,674	0,679	0,671	0,695	0,698	0,939	0,676	0,695	0,696	**0,945**
Racism	0,886	0,689	0,713	0,665	0,678	0,677	0,962	0,666	0,682	0,684	**0,967**
Inflation	0,897	0,671	0,682	0,671	0,666	0,691	0,936	0,673	0,667	0,667	**0,953**
Terrorism	0,881	0,673	0,664	0,676	0,671	0,671	0,926	0,676	0,671	0,670	**0,948**
Agriculture	0,907	0,666	0,738	0,665	0,691	0,694	0,959	0,665	0,692	0,687	**0,971**

Table 20 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,916**	0,626	0,715	0,632	0,668	0,667	0,828	0,632	0,669	0,668	0,831
Tsunami	**0,986**	0,657	0,698	0,609	0,646	0,637	0,939	0,613	0,653	0,643	0,971
Covid	**0,997**	0,729	0,849	0,745	0,863	0,861	0,996	0,745	0,869	0,864	0,996
Corruption	**0,996**	0,662	0,738	0,665	0,586	0,586	0,969	0,665	0,612	0,617	0,989
Earthquake	0,963	0,642	0,756	0,632	0,637	0,636	0,958	0,635	0,641	0,643	**0,978**
Immigration	**0,996**	0,678	0,643	0,683	0,700	0,700	0,990	0,683	0,702	0,701	0,990
Racism	**0,997**	0,679	0,774	0,705	0,711	0,711	0,994	0,706	0,713	0,712	0,994
Inflation	**0,999**	0,628	0,713	0,627	0,628	0,632	0,984	0,627	0,631	0,631	0,991
Terrorism	**0,994**	0,646	0,577	0,650	0,707	0,703	0,980	0,650	0,707	0,701	0,986
Agriculture	**0,996**	0,590	0,764	0,590	0,705	0,707	0,991	0,590	0,705	0,705	0,995

We performed Friedman's statistical test with Nemenyi's post-test to compare the approaches considering all metric scenarios and datasets (TRAWINSKI *et al.*, 2012)[3]. Figure 20 presents a critical difference diagram generated through the results of the Friedman test with

[3] The test parameters were the default parameters of the KEEL tool (<https://sci2s.ugr.es/keel/index.php>).

Table 21 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM applied in the eleven representations methods with 180 (3%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	Lat-Long	Density	Concat	AE	VAE	BiVAE	Concat	AE	VAE	TripleVAE
War	**0,844**	0,608	0,662	0,608	0,594	0,593	0,761	0,611	0,604	0,615	0,751
Tsunami	**0,920**	0,611	0,646	0,611	0,619	0,621	0,865	0,611	0,618	0,617	0,912
Covid	0,868	0,639	0,775	0,684	0,799	0,797	0,937	0,692	0,800	0,804	**0,952**
Corruption	0,930	0,622	0,669	0,631	0,599	0,599	0,922	0,631	0,606	0,622	**0,957**
Earthquake	0,872	0,600	0,677	0,609	0,611	0,612	0,880	0,618	0,617	0,629	**0,934**
Immigration	0,937	0,626	0,591	0,624	0,654	0,649	0,941	0,631	0,663	0,648	**0,946**
Racism	0,898	0,641	0,724	0,647	0,672	0,670	0,963	0,649	0,673	0,674	**0,967**
Inflation	0,907	0,585	0,654	0,591	0,604	0,598	0,939	0,591	0,610	0,601	**0,956**
Terrorism	0,892	0,609	0,566	0,616	0,629	0,630	0,929	0,616	0,631	0,633	**0,948**
Agriculture	0,915	0,549	0,695	0,574	0,658	0,663	0,960	0,578	0,662	0,660	**0,972**

Table 22 – Results in ten event datasets considering the F_1**-Score** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	Lat-Long	Density	Concat	AE	VAE	BiVAE	Concat	AE	VAE	TripleVAE
War	**0,855**	0,665	0,732	0,677	0,685	0,688	0,726	0,681	0,688	0,689	0,765
Tsunami	**0,933**	0,647	0,680	0,667	0,684	0,675	0,811	0,670	0,685	0,680	0,815
Covid	**0,955**	0,677	0,748	0,732	0,771	0,771	0,949	0,737	0,773	0,775	0,941
Corruption	**0,931**	0,676	0,665	0,692	0,691	0,684	0,868	0,692	0,692	0,691	0,870
Earthquake	**0,912**	0,660	0,693	0,666	0,667	0,675	0,816	0,671	0,667	0,679	0,825
Immigration	**0,928**	0,668	0,664	0,693	0,762	0,780	0,900	0,693	0,781	0,814	0,860
Racism	**0,940**	0,666	0,825	0,688	0,729	0,745	0,910	0,694	0,754	0,786	0,911
Inflation	**0,950**	0,666	0,796	0,681	0,660	0,658	0,862	0,681	0,659	0,662	0,861
Terrorism	**0,925**	0,676	0,690	0,679	0,681	0,682	0,895	0,683	0,681	0,682	0,893
Agriculture	**0,914**	0,657	0,677	0,668	0,730	0,728	0,866	0,670	0,728	0,728	0,859

Table 23 – Results in ten event datasets considering the **auc-roc** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step.

Datasets	Unimodal			Bimodal (DBERTML | Lat-Long)				Trimodal			
	DBERTML	Lat-Long	Density	Concat	AE	VAE	BiVAE	Concat	AE	VAE	TripleVAE
War	**0,927**	0,633	0,829	0,625	0,683	0,685	0,746	0,625	0,691	0,678	0,764
Tsunami	**0,985**	0,674	0,738	0,643	0,668	0,688	0,846	0,644	0,670	0,690	0,835
Covid	**0,997**	0,779	0,850	0,805	0,847	0,848	0,977	0,805	0,850	0,847	0,976
Corruption	**0,996**	0,663	0,674	0,666	0,699	0,687	0,895	0,666	0,699	0,687	0,893
Earthquake	**0,961**	0,657	0,792	0,634	0,674	0,683	0,826	0,635	0,674	0,686	0,830
Immigration	**0,996**	0,685	0,610	0,692	0,767	0,798	0,947	0,692	0,792	0,815	0,924
Racism	**0,997**	0,696	0,921	0,705	0,705	0,711	0,953	0,706	0,704	0,737	0,955
Inflation	**0,999**	0,699	0,879	0,620	0,606	0,587	0,889	0,621	0,613	0,597	0,870
Terrorism	**0,994**	0,642	0,691	0,645	0,683	0,682	0,939	0,645	0,685	0,682	0,938
Agriculture	**0,996**	0,576	0,699	0,586	0,756	0,751	0,894	0,585	0,751	0,744	0,898

Nemenyi's post-test. The diagram presents the methods' average rankings, and the methods connected by a line do not present statistically significant differences between them.

In addition to obtaining the highest results, DBERTML and our proposed Triple-VAE presented better average rankings. Lat-Long and the concatenation of two or three modalities

Table 24 – Results in ten event datasets considering the **accuracy** of the algorithm OCSVM applied in the eleven representations methods with 2000 (33%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML I Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,846**	0,608	0,705	0,609	0,623	0,623	0,686	0,609	0,627	0,630	0,742
Tsunami	**0,933**	0,618	0,684	0,672	0,650	0,616	0,820	0,674	0,649	0,624	0,813
Covid	**0,956**	0,681	0,769	0,754	0,754	0,770	0,950	0,758	0,756	0,775	0,942
Corruption	**0,931**	0,629	0,607	0,639	0,635	0,623	0,868	0,638	0,635	0,626	0,871
Earthquake	**0,907**	0,605	0,697	0,636	0,647	0,656	0,806	0,638	0,648	0,658	0,823
Immigration	**0,926**	0,615	0,598	0,650	0,713	0,755	0,900	0,652	0,742	0,795	0,857
Racism	**0,939**	0,634	0,823	0,648	0,676	0,682	0,910	0,653	0,701	0,746	0,911
Inflation	**0,950**	0,620	0,797	0,612	0,570	0,577	0,861	0,613	0,577	0,578	0,860
Terrorism	**0,922**	0,601	0,618	0,629	0,642	0,639	0,897	0,628	0,644	0,641	0,894
Agriculture	**0,914**	0,556	0,654	0,568	0,683	0,679	0,864	0,568	0,680	0,679	0,854

Table 25 – Results in ten event datasets considering the $\boldsymbol{F_1}$**-Score** of the algorithm OCSVM. Values of TripleVAE are considering 60 events to train (1%) and the others representations methods consider 2000 (33%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML I Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,855**	0,665	0,732	0,677	0,685	0,688	0,726	0,681	0,688	0,689	0,780
Tsunami	**0,933**	0,647	0,680	0,667	0,684	0,675	0,811	0,670	0,685	0,680	0,918
Covid	**0,955**	0,677	0,748	0,732	0,771	0,771	0,949	0,737	0,773	0,775	0,946
Corruption	0,931	0,676	0,665	0,692	0,691	0,684	0,868	0,692	0,692	0,691	**0,958**
Earthquake	0,912	0,660	0,693	0,666	0,667	0,675	0,816	0,671	0,667	0,679	**0,916**
Immigration	0,928	0,668	0,664	0,693	0,762	0,780	0,900	0,693	0,781	0,814	**0,950**
Racism	0,940	0,666	0,825	0,688	0,729	0,745	0,910	0,694	0,754	0,786	**0,964**
Inflation	0,950	0,666	0,796	0,681	0,660	0,658	0,862	0,681	0,659	0,662	**0,953**
Terrorism	0,925	0,676	0,690	0,679	0,681	0,682	0,895	0,683	0,681	0,682	**0,937**
Agriculture	0,914	0,657	0,677	0,668	0,730	0,728	0,866	0,670	0,728	0,728	**0,979**

Table 26 – Results in ten event datasets considering the **Accuracy** of the algorithm OCSVM. Values of TripleVAE are considering 60 events to train (1%) and the others representations methods consider 2000 (33%) events in the training step.

Datasets	**Unimodal**			**Bimodal (DBERTML I Lat-Long)**				**Trimodal**			
	DBERTML	**Lat-Long**	**Density**	**Concat**	**AE**	**VAE**	**BiVAE**	**Concat**	**AE**	**VAE**	**TripleVAE**
War	**0,846**	0,608	0,705	0,609	0,623	0,623	0,686	0,609	0,627	0,630	0,779
Tsunami	**0,933**	0,618	0,684	0,672	0,650	0,616	0,820	0,674	0,649	0,624	0,917
Covid	**0,956**	0,681	0,769	0,754	0,754	0,770	0,950	0,758	0,756	0,775	0,949
Corruption	0,931	0,629	0,607	0,639	0,635	0,623	0,868	0,638	0,635	0,626	**0,958**
Earthquake	0,907	0,605	0,697	0,636	0,647	0,656	0,806	0,638	0,648	0,658	**0,915**
Immigration	0,926	0,615	0,598	0,650	0,713	0,755	0,900	0,652	0,742	0,795	**0,952**
Racism	0,939	0,634	0,823	0,648	0,676	0,682	0,910	0,653	0,701	0,746	**0,965**
Inflation	0,950	0,620	0,797	0,612	0,570	0,577	0,861	0,613	0,577	0,578	**0,954**
Terrorism	0,922	0,601	0,618	0,629	0,642	0,639	0,897	0,628	0,644	0,641	**0,938**
Agriculture	0,914	0,556	0,654	0,568	0,683	0,679	0,864	0,568	0,680	0,679	**0,979**

obtain the worsts average rankings. In most scenarios, Triple-VAE has a statistically significant difference compared to Lat-Long and the strategies bimodal and Trimodal of Concat, AE, and VAE. BiVAE, DBERTML, and Density, in most scenarios, do not have a statistically

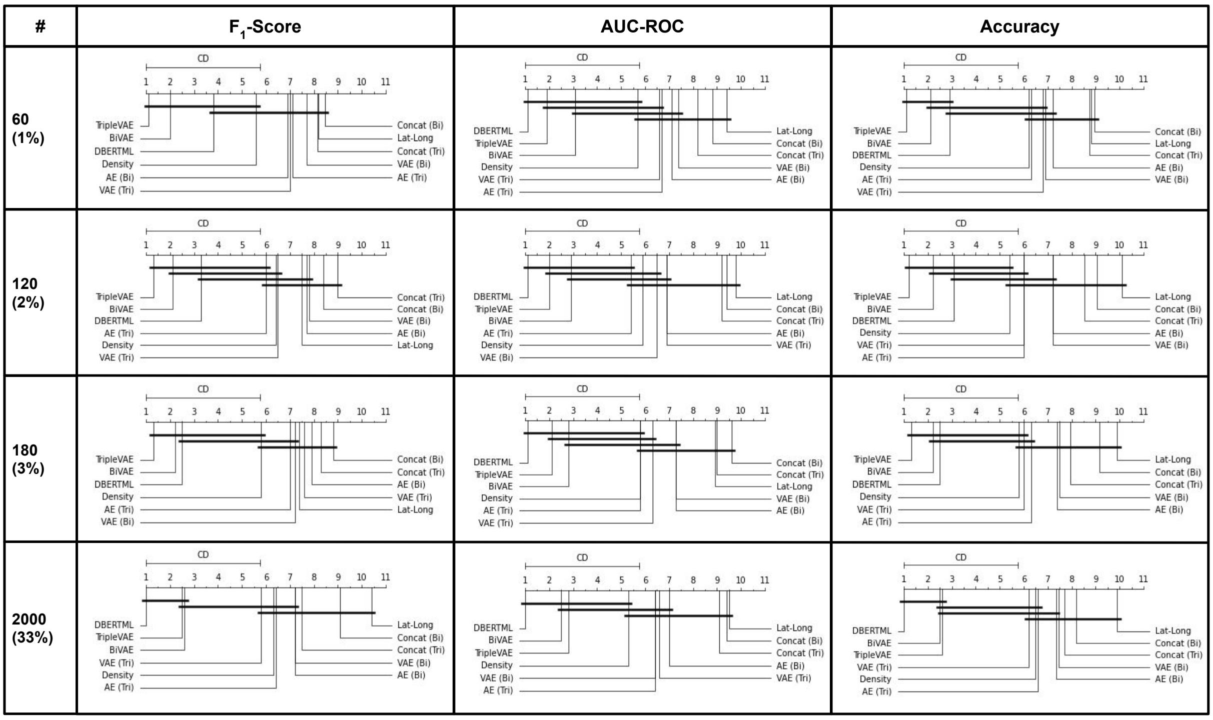


Figure 20 – Critical difference diagrams with the average rankings of the Friedman test with Nemenyi's post-test. We present the diagrams considering each Table of result, i.e., four scenario of labeled events (1%, 2%, 3% and 33%) and three metrics (F_1-Score, AUC-ROC and Accuracy).

significant difference from Triple-VAE. In addition, Triple-VAE was able to learn highly non-linear relationships, redundancies, and dependencies between modalities, structuring the events in a dimensional space more suitable for OCL methods.

5.5 Concluding Remarks

Event detection can be used to sense, analyze and comprehend important events that happen in our society. These social events have textual, geographical, and temporal components. Thus, multimodal representations have been investigated to represent the events since these components directly influence the detection of events.

This paper proposes a multimodal method (Triple-VAE) to learn a representation from three modalities (text, density, and geolocation), with different early fusion operators (concatenate, add, subtract, and multiply), and using a variational autoencoder to learn a unified representation from those modalities. We also applied the OCSVM in order to perform OCL in the generated representations. The results obtained in our experimental evaluation show that Triple-VAE, considering F_1-Score and Accuracy using only 60 (1% of the dataset) events to train, outperforms literature methods with 2000 events (33% of the dataset) to represent events on the OCL scenario. Our proposal also presented better results with statistically significant differences concerning other unimodal and multimodal techniques. It is noteworthy that the models built through OCL and considering the representations generated by Triple-VAE were able to differentiate events of interest and non-interest satisfactorily.

We intend to extend our Triple-VAE to handle incomplete modalities in future work. We note that some events may have incomplete or inaccurate information regarding geographic information and other metadata. Thus, a multimodal representation learning method must be robust to these scenarios. We also intend to use semi-supervised OCL algorithms (Positive and Unlabeled Learning (BEKKER; DAVIS, 2020)) with the representations obtained by Triple-VAE.

CHAPTER

6

CONCLUSION AND FUTURE WORK

This chapter presents the conclusions of this dissertation. First, the contributions and scientific innovations achieved during the development of this dissertation are presented, referring to the scientific challenges and research goals defined in the introduction of this dissertation. Second, we present the publications resulting from this dissertation. Finally, we present the limitations and directions for future work.

6.1 Contributions and Scientific Innovations

This dissertation presents a multimodal method developed to represent textual data considering the scenario of automatic text classification through OCL (OCL). The method (i) allows the use of a different number of modalities as input; (ii) allows the use of modalities with different dimensions; and (iii) is language and domain-independent. We apply the multimodal method proposed in three real-world application domains: (i) fake news classification; (ii) relevant app reviews detection; and (iii) web sensing from news events. It is noteworthy that we carried out an extensive empirical evaluation considering: (i) several multimodal variational autoencoder architectures; (ii) textual languages; and (iii) different sizes of training sets. We highlight the following innovations and contributions obtained with the development of this dissertation:

- **Multimodal method to represent the texts in automatic text classification through OCL:** we propose and develop a new multimodal method to represent textual data for OCL. The method is called Multimodal Variational Autoencoder (MVAE) and explores as modalities: (i) pre-trained embeddings from the DistilBERT multilingual; (ii) topic information from the high-density regions of the interest class; (iii) linguistic features of the texts; and (iv) geolocation (latitude and longitude). We explore the proposed method using two or three modalities as input. On the other hand, the proposed method can be extended to use more than three modalities. As a result, the MVAE outperforms different state-of-the-art text representations in the OCL scenario.

- **Detecting fake news through proposed multimodal representations:** we explore the MVAE to represent textual data for fake news classification (Chapter 3). We explore two MVAEs in this domain: (i) with the embeddings from the DistilBERT multilingual and topic information as modalities; and (ii) with the embeddings from the DistilBERT multilingual and linguistic features. Finally, it is worth highlighting the satisfactory performance in classifying fake news, outperforming other state-of-the-art methods in most evaluation scenarios.

- **Filtering relevant app reviews through multimodal text representations from the proposed MVAE:** we explore the MVAE to represent the app reviews for irrelevant app reviews filtering (Chapter 4). We explore our MVAE with the DistilBERT multilingual and topic information as modalities. Experimental results showed that our proposal is promising to support software engineers in analyzing app feedback from user reviews.

- **Web sensing through event analysis:** we use the MVAE to represent the news events and perform the classification of events of interest (Chapter 5). We explore our MVAE with the embeddings from the DistilBERT multilingual, topic information, and geolocation (latitude and longitude) as modalities. Our proposal proved to be useful for web sensing with a smaller set of labeled data, which is a relevant requirement in event analysis.

- **Textual collection involving news events for web sensing tasks:** In this dissertation, we collect 183 textual datasets for the OCL scenario. Each textual dataset consists of 6000 texts from the event titles of the Global Data of Events, Language, and Tone (GDELT) project. Collections can be used through a created library in an online public repository[1] (GÔLO; ROSSI; MARCACINI, 2021).

- **Source code of the proposed MVAEs for the different applications investigated in this dissertation:** all the source codes developed in this dissertation to pre-process collections, generation of representations, and classification through the One-Class Support Vector Machines (OCSVM) are available to the community in the repositories: Fake News[2], Relevant App Reviews[3], and Events[4].

[1] <https://github.com/GoloMarcos/OCTCMG>.

[2] <https://github.com/GoloMarcos/MVAE-FakeNews_Webmedia2021>.

[3] <https://github.com/GoloMarcos/MVAE-RelevantReviews>.

[4] <https://github.com/GoloMarcos/TripleVAE-Events>.

6.2 Publications

Publications in journals and conferences disseminated the contributions obtained with the development of this dissertation. We present a list of these publications separated by publication type:

- **Journals**

  1. **Gôlo, M. P. S.**; Rossi, R. G.; Marcacini R. M. *Learning to sense from events via semantic variational autoencoder. In: PLoS One*, 2021. (Qualis A1)
  2. Araujo, A. F.; **Gôlo, M. P. S.**; Marcacini R. M. *Opinion mining for app reviews: an analysis of textual representation and predictive models. In: Automated Software Engineering.* Springer, 2021. (Qualis A2)

- **Conferences**

  1. **Gôlo, M. P. S.**; Rossi, R. G.; Marcacini R. M. *Triple-VAE: A Triple Variational Autoencoder to Represent Events in One-Class Event Detection. In: National Meeting of Artificial and Computational Intelligence*, 2021[5]. (Qualis B4)
  2. **Gôlo, M. P. S.**; Souza C. M. ; ROSSI, R. G. ; REZENDE, S. O. ; NOGUEIRA, B. M. ; MARCACINI, R. M *Learning Textual Representations from Multiple Modalities to Detect Fake News Through One-Class Learning.* In: Brazilian Symposium on Multimedia and the Web (WebMedia '21), 2021. (Qualis A4)
  3. Araujo, A. F.; **Gôlo, M. P. S.**;Viana, B. M.; Sanches, F.; Romero, Roseli; Marcacini R. M. *From Bag-of-Words to Pre-trained Neural Language Models: Improving Automatic Classification of App Reviews for Requirements Engineering. In: National Meeting of Artificial and Computational Intelligence* SBC, 2020. (Qualis B4)

- **Submitted to Journals**

  1. **Gôlo, M. P. S.**; Souza C. M. ; ROSSI, R. G. ; REZENDE, S. O. ; NOGUEIRA, B. M. ; MARCACINI, R. M *One-Class Learning for Fake News Detection Through Representations learned by Multimodal Variational Autoencoders. Submitted to: ACM Transactions on Information Systems*, 2022. (Qualis A1)
  2. **Gôlo, M. P. S.**; Araujo, A. F.; Rossi, R. G.; Marcacini R. M. *Detecting Relevant App Reviews for Software Evolution and Maintenance through Multimodal One-Class Learning. Submitted to: Information and Software Technology.* (Qualis A1)

[5] It is important to emphasize that this work won the award for best paper in the main track of the conference.

6.3 Limitations and Future Work

This dissertation explored the representations generated by the pre-trained context-dependent neural language model DistilBERT Multilingual as a modality. However, this model limits the number of words used to generate the structured representation. The DistilBERT Multilingual used in this dissertation used only the 128 first words of the texts. Therefore, a future direction is using context-dependent neural language models that consider all the words in the text. The other modality explored was Density Information which uses a clustering algorithm to generate its representations. This work used the k-Means algorithm to generate the representation of the density information used as input to the MVAE, which learned good representations results in satisfactory classification performances as presented in Chapters 3, 4, and 5. However, other promising clustering algorithms from the literature can be used, such as graph-based clustering algorithms (SANTOS; ROSSI, 2020).

Regarding the MVAE method, there was no variation in the learning rate and batch size parameters. We do not carry out this variation to maintain smaller parameters in the MVAE method and the other neural networks. Nevertheless, as shown in the experimental evaluations of Chapters 3, 4, and 5, the MVAE presented a classification performance superior to different traditional and states-of-the-art representation methods. Therefore, a future direction is to variate the learning rate and batch size for the MVAE to generate more robust models.

Due to time constraints, we decide to use only the OCSVM algorithm since the project's main focus is on textual representation learning. However, one direction for future work is using other OCL algorithms (GÔLO; MARCACINI; ROSSI, 2019; KHAN; MADDEN, 2014b; KHAN; MADDEN, 2009).

For future work, we plan to investigate different semi-supervised learning scenarios for OCL, known in the literature as Positive and Unlabeled Learning (PUL) (ELKAN; NOTO, 2008; SOUZA *et al.*, 2021b; SOUZA *et al.*, 2021a). Algorithms will learn from the examples of the interest class and the unlabeled ones. Therefore, it is possible to use the advantage of MVAE of obtaining good representations with few labeled data and the advantage of PUL of learning from unlabeled examples.

Finally, this work explored the automatic text classification through OCL using two steps: (i) text representation; and (ii) text classification. Even been a natural division, this division can generate limitations since the representations generated are non-customized and agnostic to the text classification algorithm. Future work would be to carry out the classification and representation of texts through OCL in an end-to-end framework, i.e., with a single learning process.

BIBLIOGRAPHY

AGGARWAL, C. **Machine Learning for Text**. 1st. ed. United States: Springer Publishing Company, Incorporated, 2018. ISBN 3319735306, 9783319735306. Citations on pages 25, 26, 31, 32, 44, 50, 59, 67, 77, 78, 92, and 93.

AGGARWAL, C. C. **Neural Networks and Deep Learning: A Textbook**. [S.l.]: Springer International Publishing, 2018. ISBN 9783319944630. Citations on pages 72 and 73.

Al Kilani, N.; Tailakh, R.; Hanani, A. Automatic classification of apps reviews for requirement engineering: Exploring the customers need from healthcare applications. In: **Proceedings of the 2019 International Conference on Social Networks Analysis, Management and Security (SNAMS)**. [S.l.: s.n.], 2019. p. 541–548. Citations on pages 67, 68, 69, and 70.

ALAM, S.; SONBHADRA, S. K.; AGARWAL, S.; NAGABHUSHAN, P. One-class support vector classifiers: A survey. **Knowledge-Based Systems**, Elsevier, v. 196, p. 1–19, 2020. Citations on pages 26, 43, 50, 56, 57, 76, 87, and 94.

ALASHWAL, H.; DERIS, S.; OTHMAN, R. M. One-class support vector machines for protein-protein interactions prediction. **International Journal of Biological and Medical Sciences**, v. 1, n. 2, 2006. Citation on page 45.

AMORIM, R. C. D.; HENNIG, C. Recovering the number of clusters in data sets with noise features using feature rescaling factors. **Information Sciences**, Elsevier, v. 324, p. 126–145, 2015. Citation on page 34.

ARAUJO, A.; GOLO, M.; VIANA, B.; SANCHES, F.; ROMERO, R.; MARCACINI, R. From bag-of-words to pre-trained neural language models: Improving automatic classification of app reviews for requirements engineering. In: SBC. **Proceeding of the 2020 National Meeting on Artificial and Computational Intelligence**. Rio Grande do Sul, 2020. p. 378–389. Citations on pages 28, 65, 66, 67, 68, 69, and 70.

ARAÚJO, A. F.; MARCACINI, R. M. Re-bert: automatic extraction of software requirements from app reviews using bert language model. In: **Proceedings of the 2021 Annual ACM Symposium on Applied Computing**. [S.l.: s.n.], 2021. p. 1321–1327. Citation on page 66.

BEKKER, J.; DAVIS, J. Learning from positive and unlabeled data: a survey. **Machine Learning**, v. 1, n. Apr, p. 1–45, 2020. Citations on pages 50, 64, and 103.

BIDE, P.; DHAGE, S. Similar event detection and event topic mining in social network platform. In: IEEE. **Proceedings of the 2021 International Conference for Convergence in Technology**. [S.l.], 2021. p. 1–11. Citation on page 92.

BLANDFORT, P.; PATTON, D. U.; FREY, W. R.; KARAMAN, S.; BHARGAVA, S.; LEE, F.-T.; VARIA, S.; KEDZIE, C.; GASKELL, M. B.; SCHIFANELLA, R. *et al.* Multimodal social media analysis for gang violence prevention. In: **Proceedings of the 2019 International AAAI Conference on web and social media**. [S.l.: s.n.], 2019. v. 13, p. 114–124. Citations on pages 88 and 89.

BLIKSTEIN, P. Multimodal learning analytics. In: **Proceedings of the 2013 International Conference on Learning Analytics and Knowledge**. Belgium: ACM, 2013. p. 102–106. Citations on pages 26, 40, and 53.

BONDIELLI, A.; MARCELLONI, F. A survey on fake news and rumour detection techniques. **Information Sciences**, Elsevier, v. 497, p. 38–55, 2019. Citations on pages 49 and 52.

BOWMAN, S.; VILNIS, L.; VINYALS, O.; DAI, A.; JOZEFOWICZ, R.; BENGIO, S. Generating sentences from a continuous space. In: **Proceedings of the 2016 Conference on Computational Natural Language Learning**. Germany: Association for Computational Linguistics, 2016. p. 10–21. Citations on pages 38, 50, and 55.

CAMISANI-CALZOLARI, M. **The Fake News Bible: A Guide to Fake News. How Do They Start? Who Originates Fake News? How Do They Become Viral? Answers to Questions on a New Phenomenon That is Changing Our Lives.** [S.l.]: Independently published, 2018. ISBN 1980920427. Citation on page 49.

CHE, L.; YANG, X.; WANG, L. Text feature extraction based on stacked variational autoencoder. **Microprocessors and Microsystems**, Elsevier, v. 76, p. 103063, 2020. Citations on pages 27, 38, and 40.

CHEN, N.; LIN, J.; HOI, S. C. H.; XIAO, X.; ZHANG, B. Ar-miner: Mining informative reviews for developers from mobile app marketplace. In: **Proceedings of the 2014 International Conference on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2014. (ICSE 2014), p. 767–778. ISBN 9781450327565. Citations on pages 65 and 68.

CHEN, X.; LI, Q. Event modeling and mining: a long journey toward explainable events. **The VLDB Journal**, Springer, v. 29, n. 1, p. 459–482, 2020. Citation on page 87.

CHOUDHARY, A.; ARORA, A. Linguistic feature based learning model for fake news detection and classification. **Expert Systems with Applications**, Elsevier, v. 169, p. 1–15, 2021. Citations on pages 49, 51, and 52.

CICHOSZ, P. Unsupervised modeling anomaly detection in discussion forums posts using global vectors for text representation. **Natural Language Engineering**, Cambridge University Press, v. 26, n. 5, p. 551–578, 2020. Citations on pages 26, 27, and 40.

CONRADO, M. **Extração automática de termos simples baseada em aprendizado de máquina**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2014. Citation on page 33.

CONROY, N. J.; RUBIN, V. L.; CHEN, Y. Automatic deception detection: Methods for finding fake news. In: **ASIST 2015: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community**. United States: American Society for Information Science, 2015. p. 1–4. Citation on page 49.

DENG, S.; RANGWALA, H.; NING, Y. Dynamic knowledge graph based multi-event forecasting. In: **Proceedings of the 2020 International Conference on Knowledge Discovery & Data Mining**. [S.l.: s.n.], 2020. p. 1585–1595. Citation on page 87.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Citations on pages 26, 34, 35, 36, 50, 60, 66, 69, 71, and 91.

ELKAN, C.; NOTO, K. Learning classifiers from only positive and unlabeled data. In: **Proceedings of the 2008 International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2008. p. 213–220. Citation on page 108.

FAUSTINI, P.; COVÕES, T. F. Fake news detection using one-class classification. In: **Proceedings of the 2019 Brazilian Conference on Intelligent Systems**. Brazil: IEEE, 2019. p. 592–597. Citations on pages 50 and 52.

FAWCETT, T. An introduction to roc analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006. Citation on page 79.

FELHI, G.; ROUX, J. L.; SEDDAH, D. Challenging the semi-supervised vae framework for text classification. In: **Proceedings of the 2021 Second Workshop on Insights from Negative Results in NLP**. [S.l.: s.n.], 2021. p. 136–143. Citations on pages 27, 38, and 40.

FERNÁNDEZ, A.; GARCÍA, S.; GALAR, M.; PRATI, R. C.; KRAWCZYK, B.; HERRERA, F. **Learning from imbalanced data sets**. Switzerland: Springer, 2018. Citations on pages 43, 50, 66, 68, and 94.

FRANK, M. G.; FEELEY, T. H.; PAOLANTONIO, N.; SERVOSS, T. J. Individual and small group accuracy in judging truthful and deceptive communication. **Group Decision and Negotiation**, Springer, v. 13, n. 1, p. 45–59, 2004. Citation on page 49.

GAO, J.; LI, P.; CHEN, Z.; ZHANG, J. A survey on deep learning for multimodal data fusion. **Neural Computation**, MIT Press, v. 32, n. 5, p. 829–864, 2020. Citation on page 42.

GENTZKOW, M.; KELLY, B.; TADDY, M. Text as data. **Journal of Economic Literature**, v. 57, n. 3, p. 535–74, 2019. Citation on page 25.

GÔLO, M.; CARAVANTI, M.; ROSSI, R.; REZENDE, S.; NOGUEIRA, B.; MARCACINI, R. Learning textual representations from multiple modalities to detect fake news through one-class learning. In: **Proceedings of the 2021 Brazilian Symposium on Multimedia and the Web**. [S.l.: s.n.], 2021. p. 197–204. Citations on pages 26, 28, 29, and 51.

GÔLO, M.; MARCACINI, R.; ROSSI, R. An extensive empirical evaluation of preprocessing techniques and supervised one class learning algorithms for text classification. In: SBC. **Proceeding of the 2019 National Meeting on Artificial and Computational Intelligence**. Brazil: SBC, 2019. p. 262–273. Citations on pages 26, 27, 32, 40, 44, 57, 76, 94, and 108.

GÔLO, M. P.; ROSSI, R. G.; MARCACINI, R. M. Triple-vae: A triple variational autoencoder to represent events in one-class event detection. In: SBC. **Proceeding of the 2021 National Meeting on Artificial and Computational Intelligence.** [S.l.], 2021. p. 643–654. Citations on pages 26, 28, 29, and 89.

GÔLO, M. P. S.; ARAUJO, A. F.; ROSSI, R. G.; M., M. R. Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning. **Information and Software Technology (SUBMETIDO)**, 2022. Citations on pages 28 and 67.

GÔLO, M. P. S.; ROSSI, R. G.; MARCACINI, R. M. Learning to sense from events via semantic variational autoencoder. **Plos one**, Public Library of Science San Francisco, CA USA, v. 16, n. 12, p. e0260701, 2021. Citations on pages 29, 95, and 106.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. 499-523 p. <http://www.deeplearningbook.org>. Citation on page 37.

GRANO, G.; CIURUMELEA, A.; PANICHELLA, S.; PALOMBA, F.; GALL, H. C. Exploring the integration of user feedback in automated testing of android applications. In: IEEE. **Proceedings of the 2018 International Conference on Software Analysis, Evolution and Reengineering**. [S.l.], 2018. p. 72–83. Citation on page 65.

GREIFENEDER, R.; JAFFE, M.; NEWMAN, E.; SCHWARZ, N. **The psychology of fake news: Accepting, sharing, and correcting misinformation**. London: Routledge, 2021. Citation on page 49.

GUO, W.; WANG, J.; WANG, S. Deep multimodal representation learning: A survey. **IEEE Access**, IEEE, v. 7, p. 63373–63394, 2019. Citations on pages 26, 27, 40, 41, 53, and 69.

GUZMAN, E.; EL-HALIBY, M.; BRUEGGE, B. Ensemble methods for app review classification: An approach for software evolution. In: **Proceedings of the 2015 International Conference on Automated Software Engineering**. [S.l.]: IEEE/ACM, 2015. p. 771–776. Citations on pages 65, 67, 68, 69, and 70.

HASSANI, H.; BENEKI, C.; UNGER, S.; MAZINANI, M. T.; YEGANEGI, M. R. Text mining in big data analytics. **Big Data and Cognitive Computing**, Multidisciplinary Digital Publishing Institute, v. 4, n. 1, p. 1, 2020. Citation on page 25.

HEMPSTALK, K.; FRANK, E. Discriminating against new classes: One-class versus multi-class classification. In: SPRINGER. **Proceedings of the 2008 Australasian Conference on Artificial Intelligence**. [S.l.], 2008. p. 325–336. Citation on page 43.

HORNE, B.; ADALI, S. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: **Proceedings of the 2017 International AAAI Conference on Web and Social Media**. Canada: Association for the Advancement of Artificial Intelligence, 2017. v. 11, p. 759–766. Citation on page 52.

JANEV, V.; PUJIĆ, D.; JELIĆ, M.; VIDAL, M.-E. Survey on big data applications. In: **Knowledge Graphs and Big Data Processing**. [S.l.]: Springer, Cham, 2020. p. 149–164. Citation on page 25.

JASKIE, K.; SPANIAS, A. Positive and unlabeled learning algorithms and applications: A survey. In: IEEE. **2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)**. [S.l.], 2019. p. 1–8. Citation on page 27.

JUNIOR, D. D. S.; ROSSI, R. G. Classificaç ao automática de textos utilizando aprendizado supervisionado baseado em uma unica classe. **Trabalho de conclusão de curso de Sistemas de Informação UFMS-CPTL**, 2017. Citations on pages 26, 27, 32, and 40.

KALIYAR, R. K.; GOSWAMI, A.; NARANG, P. Fakebert: Fake news detection in social media with a bert-based deep learning approach. **Multimedia Tools and Applications**, Springer, v. 80, n. 8, p. 11765–11788, 2021. Citation on page 53.

KANG, H.-W.; KANG, H.-B. Prediction of crime occurrence from multi-modal data using deep learning. **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 4, p. e0176244, 2017. Citation on page 89.

KATSAGGELOS, A. K.; BAHAADINI, S.; MOLINA, R. Audiovisual fusion: Challenges and new approaches. **IEEE**, IEEE, v. 103, n. 9, p. 1635–1653, 2015. Citation on page 41.

KEMMLER, M.; RODNER, E.; WACKER, E.-S.; DENZLER, J. One-class classification with gaussian processes. **Pattern recognition**, Elsevier, v. 46, n. 12, p. 3507–3518, 2013. Citations on pages 43 and 56.

KHAN, S.; MADDEN, M. A survey of recent trends in one class classification. In: **Proceedings of the 2009 Conference on Artificial Intelligence and Cognitive Science**. Ireland: Springer, 2009. p. 188–197. Citations on pages 26, 43, 44, 56, and 108.

____. One-class classification: taxonomy of study and review of techniques. **The Knowledge Engineering Review**, Cambridge University Press, v. 29, n. 3, p. 345–374, 2014. Citation on page 50.

KHAN, S.; MADDEN, M. G. One-class classification: taxonomy of study and review of techniques. **The Knowledge Engineering Review**, Cambridge University Press, v. 29, n. 3, p. 345–374, 2014. Citations on pages 26 and 108.

KHATTAR, D.; GOUD, J. S.; GUPTA, M.; VARMA, V. Mvae: Multimodal variational autoencoder for fake news detection. In: **Proceedings of the 2019 World Wide Web Conference**. United States: ACM, 2019. p. 2915–2921. Citation on page 52.

KIEU, T.; YANG, B.; GUO, C.; JENSEN, C. S. Outlier detection for time series with recurrent autoencoder ensembles. In: **Proceedings of the 2019 International Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. p. 2725–2732. Citation on page 37.

KIFETEW, F. M.; PERINI, A.; SUSI, A.; SIENA, A.; MUñANTE, D.; MORALES-RAMIREZ, I. Automating user-feedback driven requirements prioritization. **Information and Software Technology**, v. 138, p. 106635, 2021. ISSN 0950-5849. Citations on pages 67, 68, and 69.

KRAWCZYK, B.; WOŹNIAK, M.; CYGANEK, B. Clustering-based ensembles for one-class classification. **Information sciences**, Elsevier, v. 264, p. 182–195, 2014. Citations on pages 50, 53, 71, and 92.

KRAWCZYK, B.; WOŹNIAK, M.; HERRERA, F. On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. **Pattern Recognition**, Elsevier, v. 48, n. 12, p. 3969–3982, 2015. Citation on page 25.

KUMAR, B.; RAVI, V. One-class text document classification with OCSVM and LSI. In: **Art. Intel. & Evolutionary Computations in Eng. Systems**. [S.l.: s.n.], 2017. p. 597–606. Citations on pages 26, 27, and 40.

KUMAR, B. S.; RAVI, V. Text document classification with PCA and one-class SVM. In: **Proceedings of the 2017 International Conference on Frontiers in Intel. Computing: Theory and Applications**. [S.l.: s.n.], 2017. p. 107–115. Citations on pages 26, 27, and 40.

KUMARI, R.; ASHOK, N.; GHOSAL, T.; EKBAL, A. A multitask learning approach for fake news detection: Novelty, emotion, and sentiment lend a helping hand. In: IEEE. **2021 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2021. p. 1–8. Citation on page 49.

LAN, Z.; CHEN, M.; GOODMAN, S.; GIMPEL, K.; SHARMA, P.; SORICUT, R. Albert: A lite bert for self-supervised learning of language representations. In: **Proceedings on the 2019 International Conference on Learning Representations**. [S.l.: s.n.], 2019. Citation on page 35.

LEDEL, B.; HERBOLD, S. Broccoli: Bug localization with the help of text search engines. **arXiv.**, 2021. Citation on page 84.

LEYLI-ABADI, M.; LABIOD, L.; NADIF, M. Denoising autoencoder as an effective dimensionality reduction and clustering of text data. In: SPRINGER. **Proceedings of the 2017 Pacific-Asia Conference on Knowledge Discovery and Data Mining**. [S.l.], 2017. p. 801–813. Citations on pages 37 and 38.

LI, J.; LUONG, M.-T.; JURAFSKY, D. A hierarchical neural autoencoder for paragraphs and documents. In: **Proceedings of the 2015 Internacional Conference Natural Language Processing**. China: [s.n.], 2015. p. 1106–1115. Citation on page 38.

LI, R.; LI, X.; CHEN, G.; LIN, C. Improving variational autoencoder for text modelling with timestep-wise regularisation. In: **Proceedings of the 2020 International Conference on Computational Linguistics**. [S.l.: s.n.], 2020. p. 2381–2397. Citation on page 27.

LI, Y.; YANG, M.; ZHANG, Z. A survey of multi-view representation learning. **IEEE transactions on knowledge and data engineering**, IEEE, v. 31, n. 10, p. 1863–1883, 2018. Citations on pages 26, 27, 41, 53, and 69.

LIU, B.; YIN, G.; DU, W. Text classification with pixel embedding. **arXiv preprint arXiv:1911.04115**, 2019. Citation on page 35.

LIU, K.; LI, Y.; XU, N.; NATARAJAN, P. Learn to combine modalities in multimodal deep learning. **arXiv preprint arXiv:1805.11730**, v. 1, p. 1–15, 2018. Citations on pages 41 and 42.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. E. A survey of deep neural network architectures and their applications. **Neurocomputing**, Elsevier, v. 234, p. 11–26, 2017. Citations on pages 36, 37, and 54.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019. Citation on page 35.

LYONS, B. A.; MEROLA, V.; REIFLER, J. How bad is the fake news problem? **The Psychology of Fake News: Accepting, Sharing, and Correcting Misinformation**, Routledge, v. 1, p. 11–26, 2020. Citation on page 49.

Maalej, W.; Nayebi, M.; Johann, T.; Ruhe, G. Toward data-driven requirements engineering. **IEEE Software**, v. 33, n. 1, p. 48–54, Jan 2016. ISSN 1937-4194. Citations on pages 65, 67, 68, 69, and 70.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, v. 9, n. 11, p. 2579–2605, 2008. Citation on page 62.

MANEVITZ, L.; YOUSEF, M. One-class document classification via neural networks. **Neurocomputing**, Elsevier, v. 70, n. 7-9, p. 1466–1481, 2007. Citations on pages 26, 32, 37, and 40.

MANEVITZ, L. M.; YOUSEF, M. One-class svms for document classification. **Journal of machine Learning research**, v. 2, n. Dec, p. 139–154, 2001. Citations on pages 26, 40, and 45.

MARCACINI, R. M.; ROSSI, R. G.; NOGUEIRA, B. M.; MARTINS, L. V.; CHERMAN, E. A.; REZENDE, S. O. Websensors analytics: Learning to sense the real world using web news events. In: **Proceedings of the 2017 Simposio Brasileiro de Sistemas Multimídia e Web**. [S.l.: s.n.], 2017. p. 169–173. Citation on page 26.

MAYALURU, H. K. R. **One Class Text Classification using an Ensemble of Classifiers**. Phd Thesis (PhD Thesis) — Rheinische Friedrich-Wilhelms-Universität Bonn, 2020. Citations on pages 26, 27, and 40.

MEEL, P.; VISHWAKARMA, D. K. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. **Expert Systems with Applications**, Elsevier, v. 153, p. 1–26, 2019. Citations on pages 49 and 52.

MESSAOUD, M.; JENHANI, I.; JEMAA, N.; MKAOUER, M. W. A multi-label active learning approach for mobile app user review classification. In: **Springer**. [S.l.: s.n.], 2019. p. 805–816. Citations on pages 67, 68, and 69.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Proceedings of the 2013 Conference on Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2013. p. 3111–3119. Citations on pages 26 and 34.

OTTER, D.; MEDINA, J.; KALITA, J. A survey of the usages of deep learning for natural language processing. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 32, n. 2, p. 604–624, 2020. Citations on pages 34, 35, 50, 60, 69, and 91.

PAN, R.; ZHOU, Y.; CAO, B.; LIU, N. N.; LUKOSE, R.; SCHOLZ, M.; YANG, Q. One-class collaborative filtering. In: **Proceedings of the 2008 International Conference on Data Mining**. [s.n.], 2008. p. 502–511. ISBN 978-0-7695-3502-9. Available: <http://dx.doi.org/10.1109/ICDM.2008.16>. Citation on page 26.

PANICHELLA, S.; RUIZ, M. Requirements-collector: automating requirements specification from elicitation sessions and user feedback. In: IEEE. **Proccedings of the 2020 International Requirements Engineering Conference**. [S.l.], 2020. p. 404–407. Citations on pages 66 and 67.

PANICHELLA, S.; SORBO, A. D.; GUZMAN, E.; VISAGGIO, C. A.; CANFORA, G.; GALL, H. C. How can i improve my app? classifying user reviews for software maintenance and evolution. In: IEEE. **Proceedings of the 2015 International Conference on Software Maintenance and Evolution**. [S.l.], 2015. p. 281–290. Citation on page 65.

____. Ardoc: App reviews development oriented classifier. In: **Proceedings of the 2016 International Symposium on Foundations of Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2016. (FSE 2016), p. 1023–1027. ISBN 9781450342186. Citations on pages 68 and 69.

PENG, Y.; QI, J. Cm-gans: Cross-modal generative adversarial networks for common representation learning. **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)**, ACM New York, NY, USA, v. 15, n. 1, p. 1–24, 2019. Citations on pages 41 and 53.

PENNEBAKER, J.; BOYD, R.; JORDAN, K.; BLACKBURN, K. **The development and psychometric properties of LIWC**. [S.l.], 2015. Citations on pages 53 and 59.

PERERA, P.; PATEL, V. M. Learning deep features for one-class classification. **IEEE Transactions on Image Processing**, IEEE, v. 28, n. 11, p. 5450–5463, 2019. Citation on page 26.

PETERS, M.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2018. p. 2227–2237. Citation on page 35.

PRONOBIS, A. **One-Class Image. Availabel at: <https://stackoverflow.com/questions/37775470/machine-learning-one-class-classification-novelty-detection-anomaly-assessment>. Last Access: December 2, 2021**. 2021. Citation on page 44.

RADINSKY, K.; HORVITZ, E. Mining the web to predict future events. In: **Proceedings of the 2013 International Conference on Web Search and Data mining**. [S.l.: s.n.], 2013. p. 255–264. Citation on page 87.

RASHKIN, H.; CHOI, E.; JANG, J. Y.; VOLKOVA, S.; CHOI, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In: **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**. Denmark: Association for Computational Linguistics, 2017. p. 2931–2937. Citation on page 51.

REIMERS, N.; GUREVYCH, I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**. Edinburgh: Association for Computational Linguistics, 2020. p. 4512. Available: <https://arxiv.org/abs/2004.09813>. Citations on pages 35, 60, 91, and 92.

ROSSI, R. G. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2016. Citations on pages 32, 33, and 42.

ROUSSEEUW, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, Elsevier, v. 20, p. 53–65, 1987. Citations on pages 53, 72, and 92.

RUFF, L.; VANDERMEULEN, R.; GOERNITZ, N.; DEECKE, L.; SIDDIQUI, S. A.; BINDER, A.; MÜLLER, E.; KLOFT, M. Deep one-class classification. In: **Proceedings of the 2018 International Conference on Machine Learning**. United States: Machine Learning Research, 2018. p. 4393–4402. Citations on pages 26, 43, and 56.

RUFF, L.; ZEMLYANSKIY, Y.; VANDERMEULEN, R.; SCHNAKE, T.; KLOFT, M. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In: **Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. p. 4061–4071. Citations on pages 26, 27, and 40.

RUNGTA, M.; SHERKI, P. P.; DHALIWAL, M. P.; TIWARI, H.; VALA, V. Two-phase multimodal neural network for app categorization using apk resources. In: **Proceedings of the 2020 International Conference on Semantic Computing**. [S.l.: s.n.], 2020. p. 162–165. Citations on pages 68, 69, and 70.

SANH, V.; DEBUT, L.; CHAUMOND, J.; WOLF, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**, 2019. Citation on page 35.

SANTOS, J.; ROSSI, R. Unsupervised machine learning based on heterogeneous networks for text clustering. In: SBC. **Proceeding of the 2020 National Meeting on Artificial and Computational Intelligence**. [S.l.], 2020. p. 35–46. Citation on page 108.

SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. Estimating the support of a high-dimensional distribution. **Neural computation**, MIT Press, v. 13, n. 7, p. 1443–1471, 2001. Citations on pages 15 and 45.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 34, n. 1, p. 1–47, 2002. Citation on page 42.

SETTY, V.; HOSE, K. Event2vec: Neural embeddings for news events. In: **Proceedings of the 2018 International Conference Research & Development in Information Retrieval**. [S.l.: s.n.], 2018. p. 1013–1016. Citation on page 87.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014. Citation on page 31.

SHARMA, S.; SOMAYAJI, A.; JAPKOWICZ, N. Learning over subconcepts: Strategies for 1-class classification. **Computational Intelligence**, Wiley Online Library, v. 34, n. 2, p. 440–467, 2018. Citations on pages 50, 53, 71, and 92.

SHIN, H. J.; EOM, D.-H.; KIM, S.-S. One-class support vector machines—an application in machine fault detection and classification. **Computers & Industrial Engineering**, Elsevier, v. 48, n. 2, p. 395–408, 2005. Citation on page 45.

SHU, K.; MAHUDESWARAN, D.; WANG, S.; LEE, D.; LIU, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. **Big Data**, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New ..., v. 8, n. 3, p. 171–188, 2020. Citation on page 58.

SHU, K.; SLIVA, A.; WANG, S.; TANG, J.; LIU, H. Fake news detection on social media: A data mining perspective. **SIGKDD 2017: Special Interest Group on Knowledge Discovery in Data Explorations Newsletter**, ACM, v. 19, n. 1, p. 22–36, 2017. Citation on page 49.

SILVA, R.; SANTOS, R.; ALMEIDA, T.; PARDO, T. Towards automatically filtering fake news in portuguese. **Expert Systems with Applications**, Elsevier, v. 146, p. 1–14, 2020. Citations on pages 50, 51, 52, and 59.

SINGH, V. K.; GHOSH, I.; SONAGARA, D. Detecting fake news stories via multimodal analysis. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 72, n. 1, p. 3–17, 2021. Citations on pages 50 and 52.

SORBO, A. D.; PANICHELLA, S.; ALEXANDRU, C. V.; SHIMAGAKI, J.; VISAGGIO, C. A.; CANFORA, G.; GALL, H. C. What would users change in my app? summarizing app reviews for recommending software changes. In: **Proceedings of the 2016 International Symposium on Foundations of Software Engineering**. [S.l.: s.n.], 2016. p. 499–510. Citation on page 65.

SOUZA, M. C. d.; NOGUEIRA, B. M.; ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. A heterogeneous network-based positive and unlabeled learning approach to detect fake news. In: SPRINGER. **Proceedings of the 2021 Brazilian Conference on Intelligent Systems**. [S.l.], 2021. p. 3–18. Citation on page 108.

SOUZA, M. C. de; NOGUEIRA, B. M.; ROSSI, R. G.; MARCACINI, R. M.; SANTOS, B. N. D.; REZENDE, S. O. A network-based positive and unlabeled learning approach for fake news detection. **Machine Learning**, Springer, p. 1–44, 2021. Citations on pages 27 and 108.

STANIK, C.; HAERING, M.; MAALEJ, W. Classifying multilingual user feedback using traditional machine learning and deep learning. In: **Proceedings of the 2019 International Requirements Engineering Conf. Workshops**. [S.l.: s.n.], 2019. p. 220–226. Citations on pages 65, 68, 69, 70, 77, 78, and 84.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining: Pearson New International Edition**. United States: Pearson Education Limited, 2013. ISBN 9781292038551. Citations on pages 25, 36, 42, 44, 45, 60, and 78.

TAO, C.; GUO, H.; HUANG, Z. Identifying security issues for mobile applications based on user review summarization. **Information and Software Technology.**, Elsevier, 2020. Citations on pages 65, 67, 68, and 69.

TAX, D.; DUIN, R. Support vector data description. **Machine Learning**, Springer, v. 54, n. 1, p. 45–66, 2004. Citations on pages 45, 46, 47, 57, 76, and 94.

TAX, D. M. J. **One-class classification: Concept learning in the absence of counter-examples**. Phd Thesis (PhD Thesis) — Technische Universiteit Delft, 2001. Citations on pages 26, 43, 56, and 94.

THEISSLER, A. Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. **Knowledge-Based Systems**, Elsevier, v. 123, p. 163–173, 2017. Citation on page 43.

TRAWINSKI, B.; SMETEK, M.; TELEC, Z.; LASOTA, T. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. **Applied Mathematics and Computer Science**, v. 22, n. 4, p. 867–881, 2012. Citations on pages 61, 82, and 99.

VILLARROEL, L.; BAVOTA, G.; RUSSO, B.; OLIVETO, R.; PENTA, M. D. Release planning of mobile apps based on user reviews. In: IEEE. **Proceedings of the 2016 International Conference on Software Engineering**. [S.l.], 2016. p. 14–24. Citation on page 65.

WANG, C.; ZHANG, F.; LIANG, P.; DANEVA, M.; SINDEREN, M. van. Can app changelogs improve requirements classification from app reviews? an exploratory study. In: **Proceedings of the 2018 International Symposium on Empirical Software Engineering and Measurement**. [S.l.]: ACM/IEEE, 2018. p. 1–4. Citations on pages 68 and 69.

WANG, H.; BAH, M. J.; HAMMAD, M. Progress in outlier detection techniques: A survey. **IEEE Access**, IEEE, v. 7, p. 107964–108000, 2019. Citations on pages 50 and 54.

WANG, S.; CAI, J.; LIN, Q.; GUO, W. An overview of unsupervised deep feature representation for text categorization. **IEEE Transactions on Computational Social Systems**, IEEE, v. 6, n. 3, p. 504–517, 2019. Citations on pages 27, 36, 38, and 55.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T. **Fundamentals of predictive text mining**. [S.l.]: Springer, 2015. Citations on pages 25 and 42.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data mining: practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2011. Citation on page 42.

WU, H.; DENG, W.; NIU, X.; NIE, C. Identifying key features from app user reviews. In: **Proceedings of the 2021 International Conference on Software Engineering**. [S.l.]: IEEE/ACM, 2021. p. 922–932. Citations on pages 68 and 69.

XU, J.; DURRETT, G. Spherical latent spaces for stable variational autoencoders. In: **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**. Belgium: Association for Computational Linguistics, 2018. p. 4503–4513. Citations on pages 27, 38, 39, 40, 50, 55, 56, 73, 74, and 93.

XU, W.; TAN, Y. Semisupervised text classification by variational autoencoder. **IEEE transactions on neural networks and learning systems**, IEEE, v. 31, n. 1, p. 295–308, 2019. Citations on pages 27 and 40.

YANG, Z.; DAI, Z.; YANG, Y.; CARBONELL, J.; SALAKHUTDINOV, R. R.; LE, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. **Advances in Neural Information Processing Systems**, v. 32, p. 5753–5763, 2019. Citations on pages 35 and 71.

ZEPPELZAUER, M.; SCHOPFHAUSER, D. Multimodal classification of events in social media. **Image and Vision Computing**, Elsevier, v. 53, p. 45–56, 2016. Citation on page 89.

ZHAI, J.; ZHANG, S.; CHEN, J.; HE, Q. Autoencoder and its various variants. In: **Proceedings of the 2018 International Conference on Systems, Man, and Cybernetics**. Japan: IEEE, 2018. p. 415–419. Citations on pages 37, 38, and 55.

ZHANG, J.; WANG, Y.; XIE, T. Software feature refinement prioritization based on online user review mining. **Information and Software Technology**, v. 108, p. 30–34, 2019. ISSN 0950-5849. Citations on pages 67, 68, and 69.

ZHANG, X.; GHORBANI, A. A. An overview of online fake news: Characterization, detection, and discussion. **Information Processing & Management**, Elsevier, v. 57, n. 2, p. 1–26, 2020. Citations on pages 49, 50, and 51.

ZHANG, Y.; SHEN, D.; WANG, G.; GAN, Z.; HENAO, R.; CARIN, L. Deconvolutional paragraph representation learning. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. p. 4169–4179. Citation on page 38.

ZHAO, L. Event prediction in the big data era: A systematic survey. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 54, n. 5, p. 1–37, 2021. Citation on page 87.

ZHOU, H.; YIN, H.; ZHENG, H.; LI, Y. A survey on multi-modal social event detection. **Knowledge-Based Systems**, Elsevier, v. 195, p. 105695, 2020. Citations on pages 26, 87, 88, 89, and 90.

ZHUANG, L.; DAI, H. Parameter estimation of one-class svm on imbalance text classification. In: SPRINGER. **Proceedings of the 2006 Conference of the Canadian Society for Computational Studies of Intelligence**. [S.l.], 2006. p. 538–549. Citations on pages 26, 40, and 43.

ICMC USP

SÃO CARLOS