Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Kevin A. Barhaugen

Unsupervised Anomaly Detection

Master's thesis in MIS4900
Supervisor: Patrick Bours
June 2023

NTNU
Norwegian University of
Science and Technology

Kevin A. Barhaugen

Unsupervised Anomaly Detection

Master's thesis in MIS4900
Supervisor: Patrick Bours
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Abstract

This research aims to explore if unsupervised anomaly detection can be used to detect anomalies in conversations used in a highly biased dataset. A web chat based dataset from the Børns Vilkår company was received in order to preprocess the text messages, cluster them together and find potential anomalies of any kind in the dataset. The results managed to highlight conversations based on different languages as anomalous, but did not manage to highlight differences in the conversations' content. Based on these results, the conclusion to detecting anomalies in conversation used in highly biased datasets is therefore inconclusive. Recommended future work is to implement a multilingual model that are able to handle multiple languages in a dataset, to find more meaningful anomalies, based on the content of the conversations in the dataset.

Sammendrag

Denne forskningen tar sikte på å undersøke om uovervåket unormal deteksjon kan brukes til å oppdage unormaliteter i samtaler brukt i et svært partisk datasett. Et nettchatbasert datasett fra selskapet Børns Vilkår ble mottatt for å prosessere tekstmeldingene, samle de sammen og finne potensielle unormaliteter av alle slag i datasettet. Resultatene klarte å fremheve samtaler basert på ulike språk som unormale, men klarte ikke å fremheve forskjeller i samtalenes innhold. Basert på disse resultatene, er konklusjonen om å oppdage unormaliteter i samtaler brukt i svært partiske datasett derfor resultatløs. Anbefalt fremtidig arbeid er å implementere en flerspråklig modell som er i stand til å håndtere flere språk i et datasett, for å finne mer meningsfulle unormaliteter, som er basert på innholdet i samtalene i datasettet.

Contents

Figures

Chapter 1

Introduction

1.1 Topic covered by the project

The main topic of this thesis is anomaly detection, which revolves around trying to define what is "normal" and being able to detect the "abnormal" or outliers (anomalies). Anomaly detection has found it's use in a wide variety of domains like fraud detection for credit cards, health care, insurance and intrusion detection in security [1]. In the fraud detection domain the data consists of records such as user ID, amount of money spent and time between each use of the credit card. Anomaly detection can raise an alarm when a credit card is used for unusually high payments, purchasing items that the user never have purchased before and high frequency usage of the card. Other applications involve intrusion detection in the cyber security domain in order to catch anomalies in network traffic. Although anomaly detection is not the main detection method used in this field, it is being extensively researched. Lastly the most relevant application of anomaly detection is anomaly detection in text data [1]. Anomaly detection in this domain is mainly used for the collection of newspaper articles and documents. This thesis specifically will be using anomaly detection for chat conversations to find anomalies that can be considered predatory in nature.

1.2 Keywords

Anomaly detection, Machine learning, Cybergrooming, Unsupervised learning

1.3 Problem description

Some datasets have a large bias in their data where a lot of normal (negative) data exist while there is very little abnormal data. An example of such a bias lies in cybergrooming domain where there exist a lot of normal conversations, but what we actually want to detect are the very few conversations where a possible sexual predator is grooming a child online. Another problem is that we sometimes

have a lot of data in the real world that are unlabeled, meaning we do not know exactly what the data is or contains.

1.4 Justification, motivation and benefits

Cyber grooming is the process of approaching, persuading and engaging with a minor in sexual activity through the use of conversations in the form of chats on the Internet (i.e. in social media and forums) [2]. Being able to process and analyse biased and unlabeled data would be important in domains like cybergrooming and social media platforms. This way social media platforms could automatically analyse conversations through chat in real-time in order to detect not only cybergrooming but also cyberbullying and otherwise what is considered to be toxic or harmful behaviour. It could also benefit the police in several scenarios when looking for certain criminals online.

1.5 Research questions

The main research question for this thesis is as follows; "Can unsupervised anomaly detection be used to detect anomalies in conversations used in a highly biased dataset?". The particular dataset (explained in more detail in Chapter 3) is unclassified and the contents of the messages and conversations in the dataset has been monitored. This makes it very difficult to find anomalies if they even exist in the dataset. This is also the reason why the dataset is considered to be highly biased. We expect the dataset to be mostly normal conversations with little to none anomalies and therefore largely biased on the normal side.

1.6 Planned contributions

The goal of this thesis is to be able to provide some answers to the research questions in section 1.5. This thesis will try to get one step closer to process biased and unlabeled datasets. This thesis will also contribute to the Aiba company [3], which will later proceed with the work done in this thesis, and further build upon it. The thesis will contribute to Aiba in the form of scripts which will process the dataset, along with any results the scripts provide.

Chapter 2

Related work

2.1 Cybergrooming

In cybergrooming detection it is more popular to use supervised learning algorithms in order to find predatory conversations [4]. Semi-supervised learning is also used, but in a lesser degree, while unsupervised learning algorithms are not common. The most popular datasets among researchers for training and testing in cybergrooming detection is the Perverted Justice (PJ) dataset [5]. PJ is a non-profit organization that trained volunteers and police officers to act as minors online to bait in and capture sexual predators. PJ collected the chat logs of previous predators and made them available to the public. The second most popular dataset is PAN12 which is a collection of different datasets; PJ, irclogs and omegle. The dataset was created to imitate realistic scenarios which is why the PAN12 dataset collection is heavily biased towards normal data compared to predatory data [6]. Other mentions of different datasets used are Literotica which is a website for posting amateur porn stories legally, and where adults have conversations where they can express their passion [7].
The most popular and prominent algorithm for cybergrooming detection appears to be the Support Vector Machines (SVM), however several SVM algorithms exist with minor modifications for improvement. In [4] the SVM with the semi-supervised and binary version, had the best performing accuracy while the rest of the algorithms had an accuracy below the 90th percentile.

A framework has been proposed called Behavioural Feature-Profile Specific Representation (BF-PSR) which revolves around utilizing seven behavioural features [6]. Three of these features have not been used before in cybergrooming detection, while the remaining four have been previously used in forensics. The performance of the proposed framework achieves state-of-the-art results, with MultiLayer Perceptron (MLP) being the best performing classifier with all seven behavioral features combined.

The research in [7] aims to detect online grooming of minors in chat logs through

the comparison of two Term Weighting Schemes in classifying the conversation logs. The researchers use Fuzzy-rough Feature Selection (FRFS) which is a method for reducing discrete or real-valued noisy data without user input. It can be used on data with continuous or nominal decision attributes, for regression or classification. It combines the ideas of fuzziness and indiscernibility to represent uncertainty in information. In a study on online grooming detection, FRFS was used with TFIDF (Term Frequency-Inverse Document Frequency) or Bag of Words (BoW) to identify uncertain terms in natural language conversations. The study proposed a 4-step process for grooming detection: test pre-processing, text feature extraction, text feature selection, text feature normalization, and classification. A new approach, called Term Frequency-Inverse Document Frequency-Inverse Class Space Density Frequency (TF.IDF.ICSF), has been suggested to improve text classification by combining document-based and class-based approaches and giving positive discrimination to both rare and frequent terms. It is effective in high-dimensional and low-dimensional vector spaces and generates more informative terms based on a category through use of ICF and ICSF functions [7].The research framework is divided into three phases: data collection, pre-processing and text representation (Phase 1); feature selection and classification (Phase 2); and evaluation of the performance of term weighting schemes (Phase 3). The datasets used are from Perverted Justice (PJ) and Literotica. The pre-processing phase includes tokenization, transformation, stopping, and stemming to convert the text data into a data-mining ready structure for feature selection. The algorithms used for classification are TF.IDF.ICSF, Naive Bayes, and SVM. In Phase 2 of this research, data is selected and classified using two term weighting schemes: TF.IDF.ICSF and Fuzzy-rough Feature Selection [7]. Feature selection and classification processes are executed to eliminate redundant features and extract relevant information. Text classification is performed by transforming the dataset into feature vectors and building a model using Support Vector Machine. The final phase evaluates the accuracy of the classification results using performance evaluation measures such as accuracy, precision, recall, and F-score. The objective is to determine the accuracy of detecting online grooming conversation.

There also exist algorithms which use the results from different simple classifiers like the Adaboost algorithm [8]. The researchers mention that related work in this area can be found in the fields of author identification, age detection, and detection of online grooming. They state that stylometric methods, or statistical analysis of writing style, are commonly used for author identification and that many different algorithms for author recognition have been proposed. They also mention that research in this area has focused on problems with a small number of potential authors, and less research has been done on problems with a large number of potential authors and smaller quantities of text material. The researchers also mention that experiments on a large scale dataset is done where author identification can can be accomplished with reasonable accuracy also on large-scale datasets. The researchers propose to use a number of different features to

help identify potential groomers. The features used in this study include stylometric features, such as stop and function words, letters of the alphabet, punctuation, and numbers. They also include emoticons, URL, and image counts, as well as a list of 191 grooming and sexual words [8]. Lastly, they use grammatical tags in the form of part of speech (POS) assigned to each word using the Stanford NLP library. These POS tags are used to help identify patterns in the text that can indicate whether the writer is an adult or a child. The researchers used the Adaboost algorithm, which is a popular instance of boosting where a sequence of simple classifiers are learned. The classifiers used were classification trees, and the R ada package was used for implementation. The researchers mention that prior to these experiments, alternative statistical models were evaluated and Adaboost was selected as the best-performing. The results of the classifier are reported using confusion matrices and the measures used to evaluate the classifier are accuracy, precision, and recall [8]. The researchers also mention that 95% confidence intervals are used to specify the interval within which they are 95% sure the true population accuracy for the classifier is found. They found that their results reproduce the achievements of previous work, showing that age discrimination on chat text performs poorly, but works well on formal text such as book reviews. The models were trained on a dataset of 10000 instances, 5000 from each class, and were then tested on a separate dataset of 2000 instances, 1000 from each class. The researchers then used the same model to distinguish between police pretending to be children and actual children using two datasets. The results were excellent with 99% accuracy and led the researchers to analyze which variables were important in performing this classification and attempt to exclude those that could be based on aspects of the data other than the writing style used by police officers. They found that it may be possible to detect adults pretending to be children in chat rooms even though it is not possible to distinguish between pretense-free adults and children [8]. However, they were concerned that the results may reflect data bias rather than a genuine ability to distinguish the writing styles. They found that the models were able to distinguish adults pretending to be children from real children in almost 100% of the cases. However, they were concerned that this may reflect data bias rather than a genuine ability to distinguish the writing styles. The researchers repeated the task without using grooming features to see if police could be distinguished from children using topic-neutral features only and found that accuracy was still excellent at almost 100%. They question why this is the case and consider three potential answers: 1) The topic concentration within police chats biased the data, 2) The models are capturing artifacts of the task the police set themselves, 3) The models are distinguishing the police from children on the basis of the way police are attempting to pretend to be children [8]. The researchers find that while the use of certain function words (like "cant" and "sometime") and parts of speech tags (like "plural nouns") may suggest that the conversation is related to the "grooming" topic, these features are not present in the top 10 most important variables used by the algorithm. Instead, the researchers suggests that the most important features used by the algorithm

are more stylistic in nature and may indicate that police officers, who are trying to identify groomers, are using different language than real children. They also suggests that the use of foreign words, misspellings, and informal slang by police officers pretending to be children may be a way for tech-savvy groomers to identify undercover police. The adaboost model were used to classify conversations on a chat platform as being either between police officers pretending to be children or genuine children [8]. The results of the model are very good, with a precision of 0.8 and a recall of 1.0. The study also performed an experiment to distinguish police pretending to be children from both child and adult chat users and the results are statistically significant. However, the dataset size was small and that the police officers involved were not randomly selected, so the results may not be generalized to the police population as a whole. The study recommends obtaining a larger set of examples of adults pretending to be children.

The authors in [4] compiled a desktop survey based on mitigation of online grooming using machine learning techniques developed by several different authors and papers. The work by Pendar mentioned in [9] aims to identify online pedophiles by using machine learning classifiers to analyze chat interactions from a collection of online chat logs. The Perverted Justice (PJ) dataset, which contains chat logs of previously convicted sexual predators, is used. Pendar uses two different classifiers, Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), to distinguish predators from victims and finds that k-NN models based on trigrams have a high accuracy rate of 94% in detecting pedophiles. Recent works, such as what Ebrahimi among others mentioned in [10] , also use lexical features and propose an anomaly detection methodology, using a semi-supervised Support Vector Machine (One-class classification) and deep learning techniques such as Convolutional Neural Network (CNN) to experiment with auto detection of sexual predator identification using the same dataset. Their experimental results show good performance with an accuracy of 98% and F1 measure of approximately 80%.

Kontostathis et al. have developed a system called ChatCoder, which uses keyword-matching and phrase-matching methodologies to generate features and categorize corresponding grooming stages for online sexual predators [4]. They use the Perverted Justice (PJ) dataset and extract frequently used terms from online conversations to create a dictionary of keywords. They use k-Means clustering to group only pedophile conversations and report 93% accuracy results that distinguish predator vs victim. They also use machine learning techniques such as J8 tool with different machine learning libraries, guided by the Luring Communication Theory (LCT) model [11], to classify communicative strategies used by sexual predators. Their results indicate that the system can accurately classify non-predatory sentences with a score of approximately 75%. Similarly, Gupta et al. also use LCT model to do an empirical analysis on various pedophile messages, propose to use linguistic models such as LICW, and use SVM for classification [12]. However, their work does not provide any automated classification system with regards to the identified stages.

Morris [13] uses behavioral and lexical features to detect suspicious predatory chat in online conversations. He uses the PAN-12 dataset and extracts information such as the number of messages sent by an author and the number of conversations they were involved in as behavioral features to increase lexical features and capture patterns of information flow in a chat conversation. He reports a good measure of F1 of approximately 83% on lexical features and a reasonable F0.5 of 56% on behavioral features. Similarly, Yun-Gyuna Cheang et al. also detect suspicious predatory chat using real data from game data from an entity called MovieStarPlannet and PAN12 dataset [14]. They use bag of words for feature representation, sentiment and rule breaking features from the dataset. Their findings show that rule breaking features are relatively useful and their experimental results have an accuracy score of 92% using MLP.

In their work to detect pedophile conversations in chat logs, Bogdanova proposes using fixated discourse, a sequence of related words, bag of words, and sentiment as content-based features [15]. A Fixated discourse is a pattern in which a predator is unwilling to change the topic and will not allow the victim to divert from or interrupt the subject of the matter, mainly sexual-related conversations. They use three different datasets, Perverted Justice (PJ) dataset as positive dataset and two other datasets as benign. They use SVM binary classification on this task. Their experimental results indicate that high-level features accurately classify pedophiles versus cybersex conversations with a score of 97%, compared to low-level features classifying pedophiles from NPS with a score of 94%, with both models showing good results.

The authors of this work have reviewed several articles on the use of machine learning techniques to detect online sexual grooming [4]. They find that most of the methods used are based on supervised learning, with little attention given to other methods such as unsupervised, semi-supervised, and reinforcement learning. They note that the available sources of data are already labeled and the performance of these models is constrained by the kind of labeling or features used. They anticipate that the use of other models such as unsupervised or semi-supervised techniques could contribute to this area, especially those that are capable of auto-labeling data.

Based on previously related work on cybergrooming detection in general, it appears that anomaly detection methods are not a popular research topic in the cybergrooming detection domain. Another topic that is not commonly explored in the cybergrooming domain are unsupervised algorithms that can process chat logs in real-time in order to prevent cybergrooming events from happening, and most researchers appear to mainly use supervised learning for their detection algorithms.

2.2 Text Preprocessing

Preprocessing text is important in order to have an accurate analysis of the text itself when it is sent to an algorithm. Each instance of the collected text data should have a unique identifier, known as a document, which makes up a document collection or corpus [16]. The focus of the analysis is on the words, or terms, in the text, which are combined to form a vocabulary. Text preprocessing involves multiple steps, including unitization and tokenization, standardization, stop word removal, and stemming or lemmatization, to standardize the data and remove unnecessary information. It is emphasized that far more time is spent on preprocessing the text data than the analysis itself, and proper preprocessing is key to the success of the analysis. The first step is to choose the unit of text to analyze, which could be a word, grouping of words, or phrase. The text is then separated based on the unit of analysis, and most text mining software contains functions to split the text into tokens. It is usually a bag-of-words (BOW) approach, meaning that the grammar and ordering of the text is not considered. The process of tokenization separates the text into more usable form, known as tokens, which can be individual words or consecutive word sequences (N-grams) with a specified length. N-grams retain information about the co-occurrence of words and can be used to create tokens for higher values of N. The process of standardizing and cleaning the tokens involves converting the terms in the text to lower case and removing numbers, punctuation, special characters, and stop words [16]. It will help make every term in each document comparable and eliminates any irrelevant information that adds no value to the analysis. An example is the SMART dictionary, which is used to identify and remove stop words from the documents [17]. The SMART dictionary contains a list of 571 stop words. An alternative to using existing stop word dictionaries is to create a custom dictionary. A custom dictionary can eliminate specific words or tokens that are not informative or add little value to a given project or topic. To create a custom dictionary, the frequency of terms in a collection of documents can be analyzed and words with high frequencies but low informational content can be considered for removal. The process should be repeated several times to check for multiple uses of the term. Custom dictionaries can also be used for keyword in context (KWIC) to find keywords rather than words for removal [18]. The article also introduces the concepts of syntax and semantics in text analysis. Syntax concerns sentence structure, including grammar and parts of speech, while semantics refers to meaning. Part-of-speech tagging is beneficial in identifying the most likely meaning of a token by identifying its part of speech. Two semantic concepts related to part-of-speech tagging are synonymy and polysemy. Synonymy refers to two different words having the same meaning, while polysemy refers to a single word having multiple meanings. Stemming and lemmatization are methods of breaking down words to their root word to reduce the number of unique tokens in a document collection [16]. Stemming involves the removal of a word's suffix to reduce the size of the vocabulary while lemmatization incorporates information about the term's part of speech to group words

with the same root into a single token. Stemming can result in errors due to the removal of word endings, while lemmatization can handle words with multiple meanings by including the part of speech in the rules for grouping word roots. Part-of-speech tagging is the process of labeling words by their part of speech, such as nouns, verbs, adjectives, etc. There are several popular tag sets used in English, including the Brown Corpus [19], the Lancaster-Oslo-Bergen (LOB) Corpus [20], and the Penn Treebank [21]. This labeling can be done using various software programs that process the documents and annotate the parts of speech. The process can be rule-based, Markov model-based, or maximum entropy-based, and machine learning techniques can also be used to identify the parts of speech. Grouping words together, such as synonyms, can help improve the accuracy of the analysis [16].

2.3 Anomaly Detection

Anomaly detection is about defining what normal is and being able to find anomalies that stray away from what is defined as normal. Most research on anomaly detection are usually unsupervised with little to no labeled data [22–25]. Different datasets have been used like the already mentioned PAN12 dataset [26], along with 1999 DARPA [24], UCI macine learning repository [22] and NSL-KDD [25]. The 1999 DARPA dataset consists of data from TCP/IP dumps for a Local Area Network (LAN) which was simulating an U.S Air Force LAN, that Lincoln Lab of MIT was in charge of [24]. The UCI machine learning repository is a repository of datasets that are mostly dedicated for classification tasks [22]. The NSL-KDD dataset is a modified version of the KDD99 dataset which consists of network attacks [25]. The KDD99 dataset has a variety of attacks which was simulated in a military network environment. However issues in the KDD99 dataset affected the performance of anomaly detection methods which is why NSL-KDD was proposed. Compared to KDD99, the NSL-KDD is more reasonable with the number of entries along with no redundancy entries in the dataset.

The Most widely used algorithms for anomaly detection appears to be SVM or One-Class SVM (OCSVM) with either improvements or modifications [22, 24–26]. A method was presented for improving the OCSVM algorithm to be used for anomaly detection, where the main focus was for intrusion detection [24]. OCSVM is a method adapted from SVM for one-class classification problems, where the origin is treated as the only member of the second class and separation parameters are introduced to separate the image of one class from the origin. The improved one-class SVM regards all data points close to the origin as outliers or anomalies. The proposed improved method involves utilizing that the OCSVM creates a origin point in the second class for anomalies, and the improvement being that all data points which are close enough to this origin point should also be considered anomalies, and belongs in the anomaly class. Therefore the main improvement is defining how far from the origin a data point can be before it can be classified

as anomalous. The data used in this study was obtained from the 1999 DARPA Intrusion Detection Evaluation Program conducted by MIT's Lincoln Lab [24]. It involved a simulation of a typical US Air Force LAN network and resulted in the collection of 4GB of compressed TCP/IP dump data, which was processed into 5 million connection records with 18 features each. 14,292 connection records were selected for the study and 4,758 were used as training data. The one-class SVM method was applied to differentiate intrusions from normal activities and received 96% accuracy, compared to 91% from the standard SVM.

One-class SVMs aim to learn a decision boundary that separates the majority of data points from the origin while considering a small fraction of data points as outliers [22]. It uses a Gaussian kernel to project data into a higher dimensional space and learns the decision boundary (hyperplane) that separates the majority of the data. The decision function returns a positive value for normal points and negative for outliers. One-class SVMs are originally used in semi-supervised settings and provide binary labels to identify normal or anomalous points. Two approaches are proposed to handle the impact of outliers on the decision boundary in one-class SVMs [22]. Both methods are based on the work of Song et al. to make supervised SVMs more robust to noise in the training data [27]. One approach minimizes Mean Square Error (MSE) using the center of the class as an average information to tackle outliers. The other approach modifies the slack variables of the one-class SVM, making them proportional to the distance from the centroid, resulting in the decision boundary being shifted towards normal points. The latter approach loses some interpretability as there is no restriction on the number of points that can appear on the other side of the decision boundary. The eta one-class SVM is a different approach to handle outliers in the decision boundary compared to robust one-class SVMs [22]. It uses an explicit outlier suppression mechanism introduced by Xu et al. [28]. The mechanism involves an estimate variable n that represents whether a point is normal. Outliers have a low value of n, so they do not contribute to the decision boundary that is learned only by the normal points. The researchers compared the performance of one-class SVM based algorithms against standard nearest-neighbor, clustering and statistical-based unsupervised anomaly detection algorithms. The SVM based algorithms used the Gaussian kernel and their performance was evaluated using the area under the ROC curve (AUC). The datasets used were from the UCI machine learning repository and preprocessed using RapidMiner. The robust and eta one-class SVM algorithms were compared against the standard semi-supervised one-class SVM. The number of support vectors and average CPU execution time were also considered in the evaluation. The eta one-class SVM performed best on two of the four datasets and outperformed all clustering-based algorithms [22]. The robust one-class SVM produced a sparser solution but performed similarly to the standard one-class SVM. The SVM based algorithms had in general a lower time complexity but the parameter tuning for the Gaussian kernel increased complexity. The researchers introduced a method for calculating an outlier score based on the distance to the decision boundary.

In conclusion, SVM based algorithms can perform well for unsupervised anomaly detection and the eta one-class SVM is a suitable candidate for investigation in practice.

Another paper presents an overview of anomaly detection in network intrusion detection systems (A-NID) [29]. It discusses the basic architecture of these systems, which generally consist of parameterization, training, and detection stages. Machine learning techniques can be used to build the required model automatically based on training data. The labels associated with the data instances are usually binary (normal/anomalous), but some researchers have used various attack types instead. Unsupervised anomaly detection algorithms operate under two basic assumptions: 1) most network connections are normal and only a small percentage is abnormal, and 2) malicious traffic is statistically different from normal traffic. Clustering techniques, such as K-Means, Self-organizing maps (SOM), and Adaptive Resonance Theory (ART), are commonly used unsupervised algorithms. These techniques work by grouping observed data into clusters according to a similarity or distance measure. There are two approaches to clustering-based anomaly detection: one is trained using labeled data and the other is trained using only normal data. SOMs are popular neural networks for anomaly detection and are trained by an unsupervised competitive learning algorithm. ART is a series of neural network models that perform unsupervised or supervised learning, pattern recognition, and prediction. Different studies have compared the performance of ART-1 and ART-2 and have deployed Fuzzy ART in dynamic environments. K-means algorithm is a traditional clustering method that divides data into k clusters based on similarity [29]. It can be sensitive to outliers and may result in empty clusters. An improved K-means algorithm was proposed for anomaly detection by reducing noise and isolated points and using a density radius to calculate cluster centroids. Fuzzy C-means is another clustering method where data can belong to multiple clusters, and it was improved by several researchers for applications where hard classification is difficult. Another approach, FC-ANN, combines fuzzy clustering and artificial neural networks for higher detection rate and stability. Another approach integrates several soft computing techniques with fuzzy C-means clustering and principal component analysis neural network. Shah et al. used Fuzzy C-Medoids to index cluster streams for intrusion detection [30]. UNC (Unsupervised Niching Clustering) is an evolutionary algorithm based clustering method with a niching strategy that handles noise and automatically determines the number of clusters. It uses a robust density fitness function for clustering and maintains niches for candidate clusters. Elizabeth et al. combined UNC with fuzzy set theory for anomaly detection in network intrusion detection [31]. EM (Expectation-Maximization) is another soft clustering method based on a meta-algorithm for finding maximum probability estimates in probabilistic models. The algorithm alternates between computing likelihood and maximum probability estimates of model parameters. Gaussian mixture models were used by Hajji to characterize utilization measurements and detect anomalies [32]. Animesh and Jung proposed

SCAN, an anomaly detection scheme that samples incoming network traffic, computes missing elements using an enhanced EM-based clustering algorithm, and enhances convergence speed through Bloom filters and data summaries [33]. The OCSVM allows for a small predefined percentage of outliers and outputs a score based on the distance of the data point being tested to the optimal hyper plane. Negative values represent abnormal behavior and positive values represent normal behavior. The SVM has been modified for use in unsupervised learning, with the enhanced SVM approach merging soft-margin SVM and OCSVM for better unsupervised learning and low false alarm capability [29]. The method for network anomaly detection based on OCSVM includes two steps: detector training and detecting anomalies. Unsupervised anomaly detection algorithms have been applied to intrusion detection to enhance its performance. The experiments show that supervised learning methods, such as non-linear SVM, multi-layer perceptron, and rule-based methods, outperform unsupervised methods if the test data does not contain unknown attacks. Unsupervised methods such as K-Means, SOM, and OCSVM have better performance than other techniques, but differ in their ability to detect all attack classes efficiently.

This paper [25] proposes a new unsupervised anomaly intrusion detection algorithm called SSC-OCSVM that combines two methods, Sub-Space Clustering (SSC) and OCSVM, to detect attacks. SSC is an extension of traditional clustering techniques that produces clusters from small subspaces of the original dataset. OCSVM is an extension of SVM, a supervised learning model, suitable for unlabeled data. The SSC-OCSVM algorithm has four steps: initialization, clustering and learning, evidence accumulation, and anomaly detection. In the evidence accumulation step, a dissimilarity vector is updated based on the partitions produced by OCSVM in each subspace. In the anomaly detection step, the dissimilarity vector is ranked and samples with a dissimilarity value greater than a threshold are considered as anomalies. The KDD99 dataset is widely used for network attack research but has been found to have significant issues that affect the performance of anomaly detection methods. The NSL-KDD dataset was proposed to address these issues, with reasonable number of records and no redundant records [25]. Each record in the NSL-KDD dataset has 41 features, three of which are non-numeric and were transformed into numerical features through one-hot encoding, increasing the number of features to 132. A feature selection process was necessary, using the F-test, and each feature was normalized. The NSL-KDD dataset was split into four single attack subsets and a mixed subset, to evaluate the proposed algorithm in detecting different types of attacks. The training subset was used for parameter tuning and the performance of the algorithm was evaluated on the test subset. The researchers conducted an experiment to evaluate the performance of their proposed SSC-OCSVM algorithm for intrusion detection in a computer network. They compared it with three other algorithms (K-means, DBSCAN, and SSC-EA) using the NSL-KDD dataset. The results showed that the SSC-OCSVM algorithm had the largest area under the ROC curve, detecting a large fraction of attacks

with low false alarm rates. It also achieved better DR and FMR values compared to the other methods, especially in detecting low-frequency attack classes [25]. The computation time of SSC-OCSVM was higher than the other algorithms, but it was noted that each sub-space could be executed in parallel to reduce the time.

2.3.1 Anomaly detection in text

Research has been done on unsupervised anomaly detection on text in documents [23]. It was done by characterizing segments of text like percentage of words present, average word and sentence length and capturing the tone or attitude of the written text. The researchers were artificially inserting segments of text into documents to create an anomalous element within those documents. Any genuine anomalies that already existed in the documents were removed for the purpose of their experiments. Examples of anomalies in text are topic specific advertisement or spam in a bulletin board in a off-topic discussion. Another example is plagiarism where a segment has been written by a different author within a document. Plagiarism could be difficult to detect since the plagiarized segments are on the same topic as the rest of the document. However options like change of tone or attitude in the writing can help detect anomalous segments. The researchers goal was to develop a technique to detect anomalous segments in text without knowing what anomalies already exists (unsupervised). Since the researchers were working with a limited amount of data or text, like segments of a document, they had to characterize the language using techniques that were less dependent on the distribution of words, and therefore less affected by the amount of text in a document [23]. The focus is then on the techniques regarding the characterization of style, tone, and classes of lexical items. The researchers represent each segment of text as a vector, in which the components of that vector are based on a variety of features. This way the researchers could rank every segment based on the amount of differences in the segment compared to the rest of the document. In other words, if given a document with n segments, then each of the segments are ranked from one to n based on their degree of anomaly in the document. The components of the vector consists of features like average word and sentence length, average number of syllables per word, along with several readability features, and the percentage of words which happens only once [23]. All the segments were passed through the Robust and Accurate Statistical Parser (RASP) part-of-speech tagger. CLAWS 2 tagset were also used to tag words, symbols and punctuation. Percentages of words that are articles, prepositions, pronouns, conjunction, punctuation, adjectives, and adverbs are only a few examples of the representations of a segment. The researchers also keep track of the author's preferences for certain linguistic constructions by creating lists of the most frequent words, tri-grams, bi-grams, etc. The analysis also incorporates the tone, attitude, and perspective of the text by using the General Inquirer Dictionary to determine the percentage of words in each segment that fall into specific content-analysis categories [23]. The method involves creating 4 vectors for each segment: a feature vector characteriz-

ing the segment, a feature vector characterizing the complement of the segment, a vector of lists for rank features for the segment, and a vector of lists for rank features for the complement of the segment. The process then calculates a rank feature difference score by computing the Spearman rank correlation coefficient for each pair of lists and summing all the values. The difference between two segments is calculated as the average difference in their feature vectors plus the Rank Feature Difference Score. To handle different scales of the features, the variables can also be standardized to values between zero and one. The method returns a list of all segments ranked by their anomalousness [23]. The experiment aims to determine if the truly anomalous segment can be found in the top 5, 10 or 20 segments. A baseline probability of finding the anomalous segment by chance is established. Test sets are created by inserting a segment from one author into a document written by another author and performing anomaly detection. The experiment is performed with different segment sizes and the results show that the average accuracy of detecting the anomalous segment increases as the segment size increases. For 1000 word segments, the anomalous segment was found in the top 20 ranked segments 95% of the time. The average accuracy for 500 word segments ranges from 76% to 47% and for 100 word segments it ranges from 65% to 27%. The authors performed experiments to detect opinion in a factual story and to detect English translations of Chinese newspaper segments in a collection of English newswire. The experiments used text segments of various sizes, with the results being better for larger segment sizes. The experiments showed high accuracy for detecting opinion and translations, with the best results being for detecting Chinese translations in English newswire. Standardizing scores on a scale of 0 to 1 improved results for some tasks but not for all, with the worst results being for cases where the genre distinction was great (such as anarchist's cookbook segments in newswire) [23].
The researchers' goals was to find anomalies in text documents which is very similar to this thesis, however this thesis revolves more around chat logs and conversations in the cybergrooming domain, rather than general text-based documents by different authors. In this way, this thesis will be somewhat different since we will be focusing more on anomaly detection algorithms and their performance on cybergrooming datasets.

A PhD thesis was done regarding automatic identification of online predators in chat logs by anomaly detection and deep learning by Ebrahimi [26]. Data preprocessing for chat log analysis involves parsing the raw textual log files, removing noise, and reducing dimensionality. Essential elements of the log include authors, message text, and time stamps. Noise removal procedures in Online Predator Identification (OPI) analysis include removing noisy conversations and noisy features. Removing noisy conversations involves eliminating useless samples such as non-textual samples, single-participant conversations, and extremely short messages. Removing noisy features involves eliminating noise from features obtained during feature extraction, such as terms not in proper encoding, small images, and un-

intentional misspelled words. However, intentional misspelled words can play an important role in this domain and differentiation between intentional and unintentional spelling errors is a challenge. Regarding feature selection; Stemming, a text-specific dimensionality reduction technique, is widely used in text mining but may not provide the best results for the OPI problem due to its potential to distort information about predator writing styles. Other researchers have reported better results by avoiding stemming [26]. The feature extraction phase involves the lexical features which are extracted from the sentences using the bag-of-words approach, where each word in the chat log is considered as a candidate feature. The feature set is filtered using stop words, and the frequency of each word is used to weight the features using the Term Frequency-Inverse Document Frequency approach. Bigrams (pairs of consecutive words) are often used to improve classification performance but can also increase the size of the feature set. The behavioral features include initiative, attentiveness, and conversation dominance, which are used to distinguish predators from victims by capturing their typical actions within a conversation. The extraction of psycho-linguistic features, including "fixated discourse," refers to a predator's unwillingness to change the topic and often a gradual shift to sexual conversation [26]. Chat logs can include implicit/explicit sexual content, and predators may understand their actions are not moral and try to shift responsibility to the victim while acting like children in their linguistic style. Linguistic features include the number of words in a line, personal pronouns, personal information nouns, approach verbs, and emoticons. Emoticons can reveal a predator's sentiment and tendency for dominance. Sentiment analysis of chat logs can also provide markers for predator identification, as predatory conversations tend to have more positive words and less negative words. Semi-supervised techniques were chosen due to their superior performance compared to unsupervised methods [26]. The study used the PAN-2012 dataset, the largest publicly available dataset, and a smaller SQ dataset gathered from real chat logs. The study used Naive Bayes and Support Vector Machines as the main binary text classifiers and evaluated the models using k-fold cross-validation with accuracy, precision, recall, and F-measure as performance criteria. The raw textual data was extracted, parsed, and represented using bag-of-words models and unigram and bigram features. The TFIDF weighting scheme was used for normalizing the data representation. The researcher obtained unigram and bigram features and selected the best feature set based on the performance on the training set. The performance was measured by accuracy, precision, recall, and F1-measure. The results showed that One-class SVM outperforms Naive Bayes and is comparable to binary SVM after adding a noise removal module for the PAN12 dataset. The study also found that Naive Bayes has a high recall while SVM has a high precision. For the SQ (Sûreté du Québec) dataset, Ebrahimi had limitations in accessing French conversations due to privacy concerns, resulting in a small sample size [26]. The results of experiments on this small dataset cannot be considered meaningful, but they can be used as proof of concept or to test the validity of the hypothesis. The system was trained and evaluated using 2-fold cross validation, with predatory

instances considered as anomalies and non-predatory instances as normal. The semi-supervised approach performed better than other algorithms on the small dataset, due to the one-class SVM's ability to capture the minimum enclosing hyperplane around a small set of either positive or negative instances.
Ebrahimi's thesis focuses on comparing the performance between only one class label and where both class labels are used (binary classification). Deep learning architectures is also a topic of interest for this researcher, whereas we will look more into how viable anomaly detection is in the cybergrooming domain and how anomaly detection algorithms perform with datasets from this domain.

The researchers in [34] present a new one-class classification method called Context Vector Data Description (CVDD) that uses pre-trained word embedding models for anomaly detection on text data. The method maps variable-length sequences of word embeddings to fixed-length text representations using a multi-head self-attention mechanism. The representations and context vectors are trained together to capture multiple modes of normalcy and enable contextual anomaly detection with sample-based explanations and improved interpretability. The authors evaluate CVDD on Reuters-21578 and 20 Newsgroups datasets and qualitatively on IMDB Movie Reviews. The pre-trained word embeddings used in the experiments include GloVe, BERT and fastText, with GloVe achieving better results than fastText and BERT. The researchers compare three methods for aggregating word vector embeddings to fixed-length sentence representations: mean, tf-idf weighted mean, and max-pooling. They then evaluate the performance of these sentence embeddings in one-class classification using the OCSVM with cosine kernel. The text data is pre-processed by lowercasing, stripping punctuation, numbers, and redundant whitespaces, removing stopwords, and only considering words with a minimum length of 3 characters. In every one-class classification setup, one class is considered normal and the others are considered anomalous. The models are trained only with the training data from the normal class and the testing is performed on all classes with normal samples labeled as "normal" (y=0) and anomalous samples labeled as "anomalous" (y=1) [34]. The results show that CVDD performs well and is robust to different parameters. They also examine the top words for each context in CVDD to understand the themes captured by each context. They found that the contexts indeed reflect the characteristics of the classes. Additionally, the authors noted that the tf-idf weighted embeddings perform well on larger datasets, while the CVDD method has the advantage of strong interpretability and potential for contextual anomaly detection. CVDD was also used to detect anomalous movie reviews on IMDB by training a model with 10 context vectors on 25,000 movie reviews from the IMDB train set. The themes present in the movie reviews were captured well by the different contexts of the CVDD model [34].
They have presented a new one-class classification method for anomaly detection on text. Even though this is relevant and similar to this thesis, we will be focusing more on potentially different types of algorithms and datasets. Also this thesis

is a bit more focused on the cybergrooming domain and how anomaly detection methods can work there rather than looking at text in anomaly detection in general, which is the case in [34].

Anomaly detection are known to have high false alarms and being able to differentiate which attack that activated those alarms is difficult in intrusion detection systems [29]. These disadvantages also holds true for anomaly detection in other domains, and might cause anomaly detection methods to not be the best performing algorithmic method to exist. Another weak point in research exist in the cybergrooming domain, where anomaly detection methods are not a popular method of choice for researchers when trying to detect cybergrooming occurrences online.

Chapter 3

Data

The name of the dataset is called Børns Vilkår (BV), and is named after the danish Børns Vilkår company that the dataset comes from. The BV company revolves around helping kids, youth and parents with all kinds of problems that the kids cannot speak to others about. They fight for helping and ensuring the kids' safety and well being in their childhood. On the child phone line they receive thousands of calls and chat messages from kids asking for help about a variety of subjects, and the BV dataset is comprised of the web chat messages only which are manually monitored. The BV dataset has already been anonymized meaning no real persons, locations or organizations are able to be identified based on the information in the dataset alone. The BV dataset consists of 313,127 different entries of messages between kids and BV employees, and the dataset is multilingual containing several other languages. The most prominent languages being mostly Danish, a little English and a Cyrillic language believed to be Ukrainian. In total there are 10,822 conversations along with 15 columns of different information;

ConversationType:
The first column in the BV dataset is the conversationType which classifies what type of conversation the message is a part of. The conversatonType value is always 'Dialogue'.

SenderId:
The senderId is a unique value for each person in the dataset. This value stays the same for the BV employee and the caller throughout the entire conversation. The senderId also stays the same for BV employees across conversations. In other words, one is able to identify if a certain BV employee has participated in other conversations in combination with the sender column in the dataset.

Sender:
The sender column will say something about what type of person the sender is. The sender field explains if the sender of the message is a BV employee (Bornsvilkar), a child (Barn) or a young adult (Ung). The sender value together with the

senderId value mentioned above can tell if the senderId that is listed belongs to a BV employee or not.

ReceiverId:
ReceiverId is the id of the person receiving the message. If the receiverId belongs to the caller then the receiverId and the senderId will have the same value. However if the receiver is a BV employee then the receiverId will be listed as 'not available' (<NA>).

Receiver:
Similarly to the sender column, the receiver can be listed as a child or a young adult. If the receiver is a BV employee then this field is also listed as 'not available' in the same way as receiverId.

ConversationCode:
The conversationCode is simply a unique code for the entire conversation between a caller and a BV employee. All messages belonging to a certain conversation will have the same conversationCode. In the dataset, the messages with the same codes are not sorted together.

MessageId:
MessageId is a unique id for each message in the conversation. This id is also unique throughout the dataset as well. The messageId simply exists to uniquely identify each message.

Message:
The message column contains the actual message that is sent between the sender and the receiver.

Category:
Each conversation are categorized based on the content and topic of the conversation. Love is an example of a category in this dataset.

CreatedOn:
The createdOn field is a timestamp of each message. The format of the timestamp is year, month and day followed by hour, minute and second (yy-mm-dd hh:mm:ss).

ChannelId:
ChannelId is a unique identifier for what type of medium the person uses to contact BV.

Channel:
Channel names the medium the person is using to contact BV. Examples of this are sms or web chat.

isIncoming:
The isIncoming has a boolean value with either true or false.

chatName:
ChatName contains the official name on the channels people use. Examples are child phone (børnetelefonen) and heard chat (Hørt chat).

subChatId:
The last column; subChatId is similar to conversationCode but for the web chat medium specifically. If the channel field is listed as sms, then this field has the value 'not available' (<NA>).

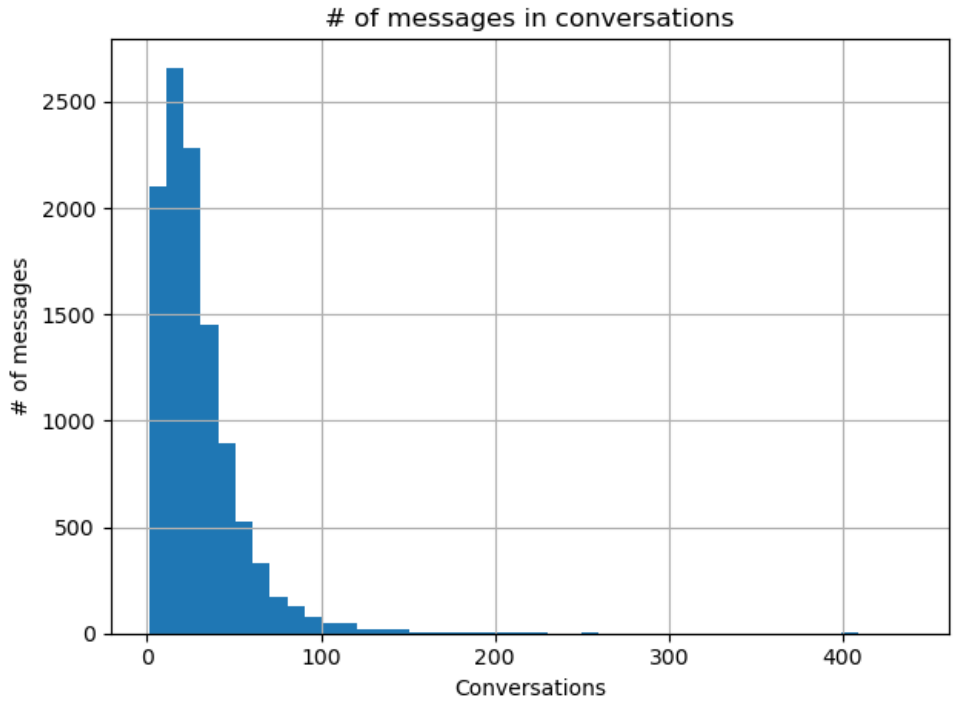


Figure 3.1: Histogram of the number of messages in conversations

The histogram in Figure 3.1 shows how many messages there are in all conversations in the dataset. From this figure most conversations have less than a hundred messages and there even are a few conversations with more than 400 messages in total.

This dataset only consists of the chat messages and not the calls from the BV company. This makes anomaly detection more difficult since the BV company can monitor the chat messages and not the calls. This means that the chances for

finding relevant anomalies in the dataset is severely decreased.

Chapter 4

Analysis

4.1 Text Preprocessing

In order to apply algorithm methods to the text it is first required to standardize the input text. We do this to increase accuracy of the results for the applied algorithms along with increasing reproducibility and reliability of the results. The first preprocessing method is to tokenize the BV dataset [16]. Every sentence is split into tokens which consists of a single word per token. After tokenization stop word removal is also applied to the BV dataset. Stop words in languages only provide a grammatical meaning to sentences and doesn't provide any additional meaning in terms of content. Examples of stop words are; a, an, the, is, has, was, my, he, she. Stemming and lemmatization is also performed on the dataset. Stemming breaks down words to their root word by removing the word's suffix to reduce the size of the vocabulary. Lemmatization further enhances the stemming process by adding part of speech to the stemmatized words. This helps in cases where a word can have multiple meanings based on the context of a sentence. The word "meeting" will turn to "meet" and shows an example of the lemmatization's ability to extract the root word. Lemmatization can also convert tense words to present tense like the word "was" converts to "be". Finally lemmatization can turn plural words into the singular root word; "mice" converts to "mouse".
Punctuations of all kinds are counted as words by themselves and is therefore necessary to remove since they do not provide any contextual meaning. Examples of these are dots, commas, exclamation marks, questions and emojis.
Lastly the final processing step was to group the BV dataset messages together into full conversations. Doing anomaly detection clustering on per message basis is achievable but provides little to no meaning in a conversation setting, especially if the conversations as a whole are long. It will also be difficult to find anomalies in the content of each conversation on a per message basis without the context of the rest of the messages in a conversation.

4.1.1 Bag of words

To preprocess the BV dataset, Bag-of-Words (BoW) model is used to extract the features from the text corpus before using it as an input for clustering algorithms. The BoW model represents the text by the occurrence of words in a document, and involves a vocabulary of known words and being able to measure each word that are present [35]. It is referred to as a "bag" of words because any information about the structure of the document is discarded. The idea is that documents are considered similar to each other if they contain similar content, and therefore learn something about the meaning of the content in the documents.

4.1.2 Doc2vec

To create the feature vector to be used as input for the clustering algorithms, doc2vec was used. The doc2vec model is a continuous BoW model, and instead of only looking at each word separately, doc2vec takes into account the surrounding words and therefore the context of what a sentence or a longer document is trying to convey [36]. The default implementation of doc2vec introduces some randomness into the results which removes reproducibility. In order to completely remove this randomness and achieve reproducible results, a few parameters had to be set for doc2vec; "seed","worker" and "pythonhashseed" [37]. The "seed" parameter affects the concatenation which enforces a random hash onto every word in the vector. The "worker" parameter is an integer about the amount of worker threads to use for the operating system. This has to be set to one to prevent ordering jitter from the operating system thread scheduling. Lastly the environment variable "pythonhashseed" involves is set to zero to remove python's own hash randomization. Only when these three parameters are set will the results from doc2vec be reproducible after every run.

4.2 Natural Language Processing Model

DaCy is an end-to-end framework for Danish Natural Language Processing (NLP) and is the model that is used for processing the BV dataset. DaCy offers state-of-the-art performance on tasks like Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and dependency parsing [38]. It is built on SpaCy v.3 which is a popular Python library, and contains linguistical data, algorithms, optimization, user-friendliness, and documentationfor NLP [39]. DaCy includes three fine-tuned language models: DaCy small, DaCy medium, and DaCy large, each with different parameter sizes. In addition to the pre-existing models, DaCy allows convenient integration of other SpaCy models into the pipeline. It provides wrappers for adding Danish models for polarity, emotion, and subjectivity classification. The aim of DaCy is to serve as a unified framework for Danish NLP, offering well-documented functionality and tutorials.

4.3 Clustering Algorithms

4.3.1 K-Means

The k-means clustering algorithm is a popular method used in various fields such as information retrieval, computer vision, and pattern recognition [40]. Its purpose is to group a given set of data points into k clusters, with the goal of grouping together similar data points. The algorithm iteratively assigns each data point to the cluster whose centroid (representative point) is closest to it. It then recalculates the centroids of these clusters by taking the average of the data points assigned to each cluster. This process continues until the algorithm converges and the clusters stabilize. The K-means algorithm was chosen due to it's popularity and ease of use [41].

4.3.2 DBSCAN

DBSCAN, or Density-Based Spatial Clustering of Applications with Noise [42]. It is a pioneering clustering method that can effectively group data points of various shapes, while also handling noise, in both spatial and high-dimensional databases. The fundamental concept behind DBSCAN is that, for each object within a cluster, its neighborhood within a specified radius (Eps) must include a minimum number of objects (MinPts). This ensures that the size of the neighborhood surpasses a certain threshold, indicating its significance in defining a cluster. DBSCAN algorithm works by examining the neighborhood of each object in the dataset. If the neighborhood of an object contains more data points than the specified MinPts value, a new cluster is formed with that object as a core point [42]. The algorithm then iteratively gathers directly density-reachable objects from these core points, potentially merging them into a new cluster. The process continues until no additional objects can be added to any cluster, signifying the termination of the algorithm. The reason for choosing DBSCAN as one of the clustering algorithms was because of it's ability to divide the data points into clusters by itself without a human specifying the amount of clusters through input parameters. DBSCAN can then determine how many clusters are ideal for the total given data points.

Chapter 5

Results & Discussion

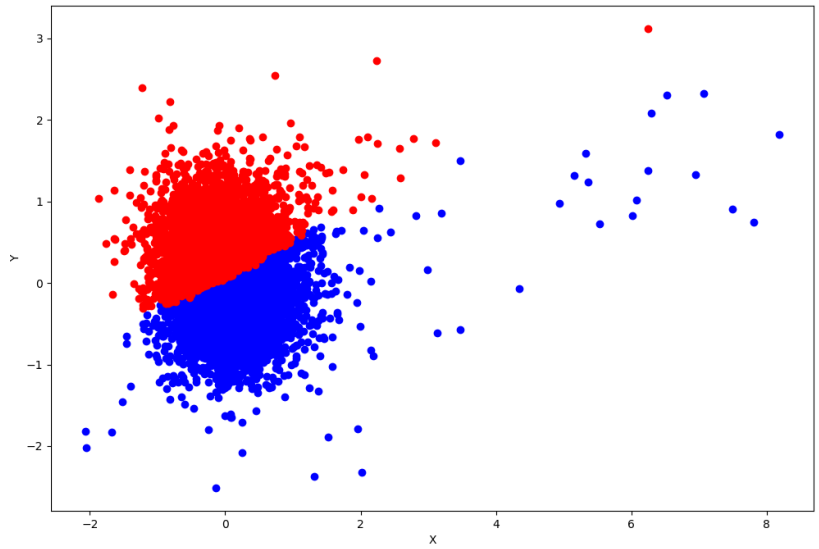


Figure 5.1: K-means clustering results of conversations in the BV dataset with two clusters

Figure 5.1 and Figure 5.2 show the results of each clustering algorithm after processing the feature vector that was produced. Both of the figures highlights the same feature vector, one with K-means and one with DBSCAN. Focusing on Figure 5.1 for simplicity, the small group of blue data points in the top right corner are the English conversations in the BV dataset. The other blue data points below the main cluster contains a few Cyrillic, but also Danish conversations. The other data point anomaly groups or points not aforementioned are mostly Danish conversations, while the main cluster of conversations are all in the Danish language. The results mentioned here at least dictates that different languages

are detected as anomalies. However our main goal is to be able to differentiate conversations based on the actual content and not on the language. After having manually looked through many of these conversations, it is difficult to determine any significant difference between a normal conversation (conversation from the main cluster) and an anomaly conversation. The reason for why the K-means algorithm is having two clusters is to showcase the difficulty the clustering algorithm is having when deciding which conversation belongs in which cluster. Increasing the amount of clusters, and trying different amounts of clusters have also been tried, but gave no notable results to highlight. Figure 5.2 shows the same data points but with the DBSCAN algorithm. Unlike K-means, DBSCAN does not require a number for defining the amount of clusters as a prerequisite. Based on the data points as input, DBSCAN determines by itself that one cluster fits the best. Since the data points are the same in both figures, the same explanation applies to DBSCAN as well in regards to what the data points mean.

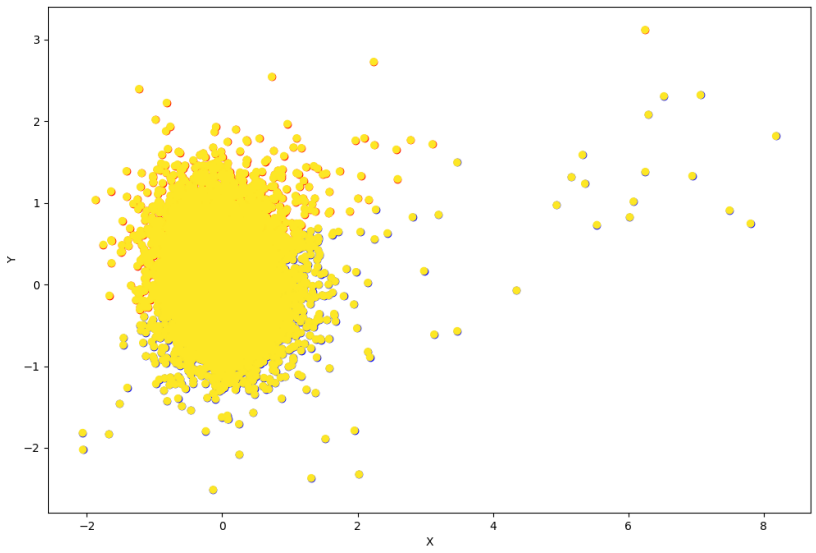


Figure 5.2: DBSCAN clustering results of conversations in the BV dataset

5.1 Discussion

From what Figure 5.1 and 5.2 show, the BV dataset i skewed towards the normal conversations. From what is explained in the results above, it appears that it is possible to detect a difference of languages that exist in the BV dataset. Even though this was not the expected outcome of the results we set out to find initially, it is at the very least a step in the right direction. Ideally, one would rather want to find

anomalies in the content or context of the conversations themselves, rather than finding differences elsewhere. Regardless, this somewhat answers the research question; "Can unsupervised anomaly detection be used to detect anomalies in conversations used in a highly biased dataset?". Language differences between conversations can indeed be found in the dataset by using anomaly detection. The biggest reason why the language was the biggest difference in the results was due to the lack of multilingual models. Implementing a multilingual model could most likely provide some better results, or at the very least provided us with anomalies in the content of conversations. However, this was not implemented in time. Another explanation for the results was briefly mentioned in Chapter 3 and in section 1.5, and revolves around the BV dataset coming from web chat messages only. These web chat conversations have been manually monitored by employees, and therefore significantly reduces the chances of finding any significant anomalies of contextual value. Since the experiments and research is mostly centered around the BV dataset itself, looking for other datasets is slightly out of scope for this thesis. However unsupervised anomaly detection on other text or chat based datasets could be a potential topic for future work.

Chapter 6

Conclusion & Future Work

This research aimed to use unsupervised anomaly detection in order to detect anomalies in conversations used in a highly biased dataset. Based on the results of the experiments, the answer to the research question is deemed to be inconclusive. More research is required in the form of future work to better answer the research question. On one side, we were able to find anomalies in conversations through different languages, however we were unable to find anomalous conversations based on context.
Based on the results being inconclusive to the research question, future work for Aiba should be to continue the existing work that has already been accomplished in this thesis. Implementing a multi-language model or utilizing a translation algorithm to be able to handle the several different languages that exist in the dataset, and be able to run the clustering algorithms afterwards. That would be able to achieve better results based upon actual context of the conversations, and move past the language barrier. Utilizing different clustering algorithms with different input parameters could also bring interesting results and easier detection of anomalies. The research has aimed to give Aiba a head start on anomaly detection in biased unlabeled datasets. It has also contributed to fill the research gap in the cybergrooming domain where unsupervised anomaly detection has not been used compared to more traditional classification methods.

Bibliography

[1] V. Chandola, A. Banerjee and V. Kumar, 'Anomaly detection: A survey,' *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–3, 13, 17, Jul. 2009, ISSN: 0360-0300. DOI: 10.1145/1541880.1541882.

[2] M. Mladenović, V. Ošmjanski and S. V. Stanković, 'Cyber-aggression, cyberbullying, and cyber-grooming: A survey and research challenges,' *ACM Comput. Surv.*, vol. 54, no. 1, p. 11, Jan. 2021, ISSN: 0360-0300. DOI: 10.1145/3424246.

[3] NTNU. 'Real-time detection of fake profiles, grooming and toxicity,' Norwegian University of Science and Technology. (Dec. 2022), [Online]. Available: https://aiba.ai/.

[4] C. Ngejane, G. Mabuza-Hocquet, J. Eloff and S. Lefophane, 'Mitigating online sexual grooming cybercrime on social media using machine learning: A desktop survey,' in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, 2018, pp. 1–6. DOI: 10.1109/ICABCD.2018.8465413.

[5] P. R. Borj, K. Raja and P. Bours, 'Online grooming detection: A comprehensive survey of child exploitation in chat logs,' *Knowledge-Based Systems*, vol. 259, p. 110 039, 2023, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2022.110039.

[6] D. F. Milon-Flores and R. L. Cordeiro, 'How to take advantage of behavioral features for the early detection of grooming in online conversations,' *Knowledge-Based Systems*, vol. 240, p. 108 017, 2022, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.108017.

[7] N. R. Sulaiman and M. Md. Siraj, 'Classification of online grooming on chat logs using two term weighting schemes,' *International Journal of Innovative Computing*, vol. 9, no. 2, pp. 3–6, Nov. 2019. DOI: 10.11113/ijic.v9n2.239.

[8] M. Ashcroft, L. Kaati and M. Meyer, 'A step towards detecting online grooming – identifying adults pretending to be children,' in *2015 European Intelligence and Security Informatics Conference*, 2015, pp. 98–104. DOI: 10.1109/EISIC.2015.41.

[9] N. Pendar, 'Toward spotting the pedophile telling victim from predator in text chats,' in *International Conference on Semantic Computing (ICSC 2007)*, IEEE, 2007, pp. 235–241.

[10] M. Ebrahimi, C. Y. Suen, O. Ormandjieva and A. Krzyzak, 'Recognizing predatory chat documents using semi-supervised anomaly detection,' *Electronic Imaging*, vol. 28, pp. 1–9, 2016.

[11] L. N. Olson, J. L. Daggs, B. L. Ellevold and T. K. Rogers, 'Entrapping the innocent: Toward a theory of child sexual predators' luring communication,' *Communication Theory*, vol. 17, no. 3, pp. 231–251, 2007.

[12] A. Gupta, P. Kumaraguru and A. Sureka, 'Characterizing pedophile conversations on the internet using online grooming,' *arXiv preprint arXiv:1208.4324*, 2012.

[13] C. Morris, 'Identifying online sexual predators by svm classification with lexical and behavioral features,' *Master of Science Thesis, University Of Toronto, Canada*, 2013.

[14] Y.-G. Cheong, A. K. Jensen, E. R. Guðnadóttir, B.-C. Bae and J. Togelius, 'Detecting predatory behavior in game chats,' *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 3, pp. 220–232, 2015.

[15] D. Bogdanova, P. Rosso and T. Solorio, 'Exploring high-level features for detecting cyberpedophilia,' *Computer speech & language*, vol. 28, no. 1, pp. 108–120, 2014.

[16] M. Anandarajan, C. Hill and T. Nolan, 'Practical text analytics,' *Maximizing the Value of Text Data.(Advances in Analytics and Data Science. Vol. 2.) Springer*, pp. 45–59, 2019.

[17] G. Salton, *The SMART retrieval system—experiments in automatic document processing*. Prentice-Hall, Inc., 1971.

[18] B. Inmon, *Turning text into gold: Taxonomies and textual analytics*. Technics Publications, 2017.

[19] H. Kučera, W. Francis, W. F. Twaddell, M. L. Marckworth, L. M. Bell and J. B. Carroll, 'Computational analysis of present-day american english,' *(No Title)*, 1967.

[20] J. Stig, G. N. Leech and H. Goodluck, 'Manual of information to accompany the lancaster-oslo: Bergen corpus of british english, for use with digital computers,' 1978.

[21] A. Taylor, M. Marcus and B. Santorini, 'The penn treebank: An overview,' *Treebanks: Building and using parsed corpora*, pp. 5–22, 2003.

[22] M. Amer, M. Goldstein and S. Abdennadher, 'Enhancing one-class support vector machines for unsupervised anomaly detection,' in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ser. ODD '13, Chicago, Illinois: Association for Computing Machinery, 2013, pp. 8–15, ISBN: 9781450323352. DOI: 10.1145/2500853.2500857.

[23] D. Guthrie, L. Guthrie, B. Allison and Y. Wilks, 'Unsupervised anomaly detection.,' in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1624–1628.

[24] K.-L. Li, H.-K. Huang, S.-F. Tian and W. Xu, 'Improving one-class svm for anomaly detection,' in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, vol. 5, 2003, 3077–3081 Vol.5. DOI: 10.1109/ICMLC.2003.1260106.

[25] G. Pu, L. Wang, J. Shen and F. Dong, 'A hybrid unsupervised clustering-based anomaly detection method,' *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2021. DOI: 10.26599/TST.2019.9010051.

[26] M. Ebrahimi, 'Automatic identification of online predators in chat logs by anomaly detection and deep learning,' Ph.D. dissertation, Concordia University, 2016.

[27] Q. Song, W. Hu and W. Xie, 'Robust support vector machine with bullet hole image classification,' *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 32, no. 4, pp. 440–448, 2002.

[28] L. Xu, K. Crammer, D. Schuurmans *et al.*, 'Robust support vector machine training via convex outlier ablation,' in *AAAI*, vol. 6, 2006, pp. 536–542.

[29] S. Omar, A. Ngadi and H. H. Jebur, 'Machine learning techniques for anomaly detection: An overview,' *International Journal of Computer Applications*, vol. 79, no. 2, p. 1, 2013.

[30] H. Shah, J. Undercoffer and A. Joshi, 'Fuzzy clustering for intrusion detection,' in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, IEEE, vol. 2, 2003, pp. 1274–1278.

[31] E. Leon, O. Nasraoui and J. Gomez, 'Anomaly detection based on unsupervised niche clustering with application to network intrusion detection,' in *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, IEEE, vol. 1, 2004, pp. 502–508.

[32] H. Hajji, 'Statistical analysis of network traffic for adaptive faults detection,' *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1053–1063, 2005.

[33] A. Patcha and J.-M. Park, 'Network anomaly detection with incomplete audit data,' *Computer Networks*, vol. 51, no. 13, pp. 3935–3955, 2007.

[34] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake and M. Kloft, 'Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text,' in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4061–4071. DOI: 10.18653/v1/P19-1398.

[35] W. A. Qader, M. M. Ameen and B. I. Ahmed, 'An overview of bag of words;importance, implementation, applications, and challenges,' in *2019 International Engineering Conference (IEC)*, 2019, pp. 1–2. DOI: 10.1109/IEC47844.2019.8950616.

[36] Q. V. Le and T. Mikolov, *Distributed representations of sentences and documents*, 2014. arXiv: 1405.4053.

[37] R. Řehůřek, *Doc2vec paragraph embeddings*. [Online]. Available: https://radimrehurek.com/gensim/models/doc2vec.html.

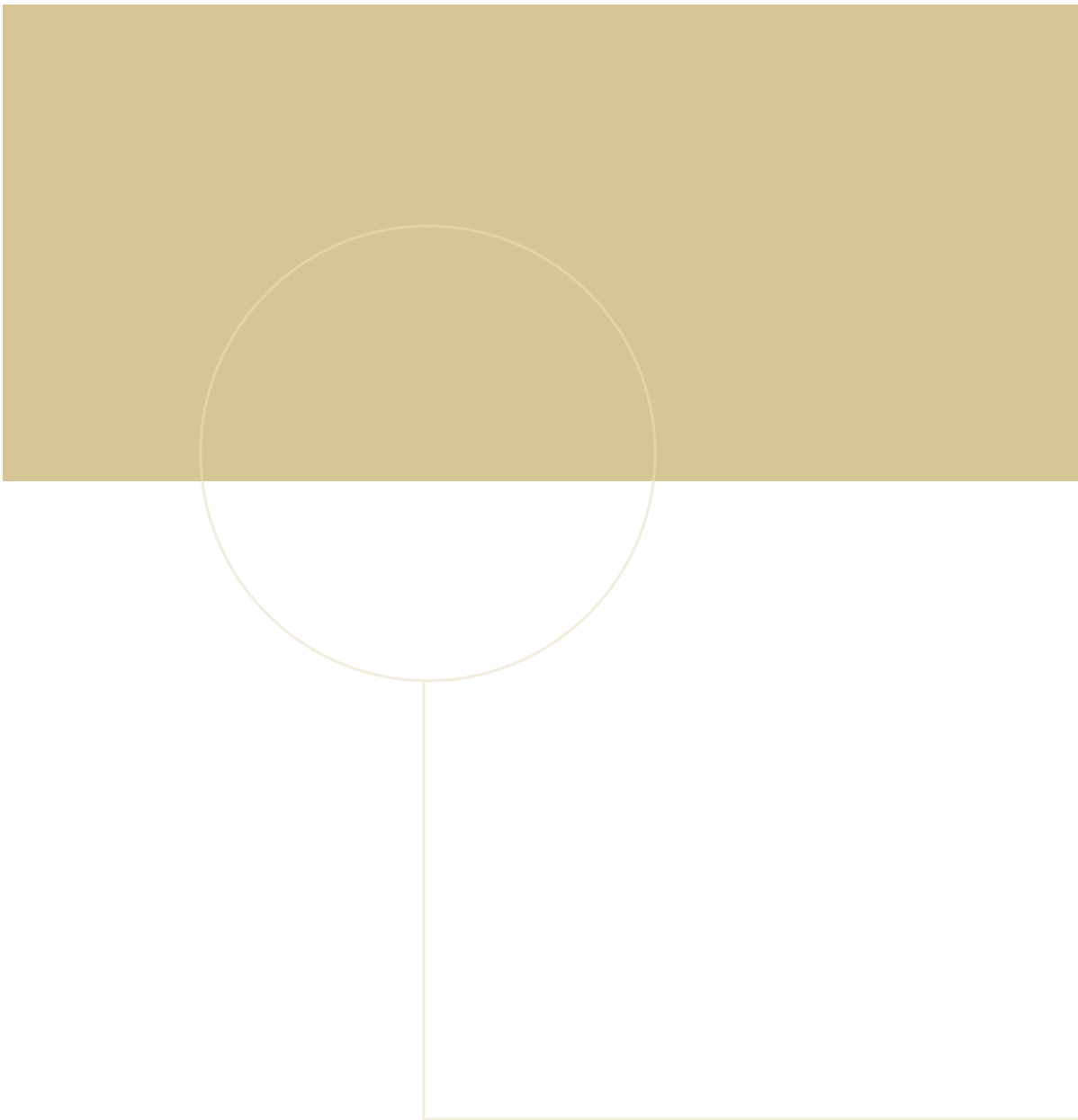
[38] K. Enevoldsen, L. Hansen and K. Nielbo, *Dacy: A unified framework for danish nlp*, 2021. arXiv: 2107.05295.

[39] Y. Vasiliev, *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press, 2020, pp. xvi–xvii.

[40] S. Shukla and S. Naganna, 'A review on k-means data clustering approach,' *International Journal of Information & Computation Technology*, vol. 4, no. 17, pp. 2–3, 2014.

[41] M. Ahmed, R. Seraj and S. M. S. Islam, 'The k-means algorithm: A comprehensive survey and performance evaluation,' *Electronics*, vol. 9, no. 8, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9081295.

[42] K. Khan, S. U. Rehman, K. Aziz, S. Fong and S. Sarasvady, 'Dbscan: Past, present and future,' in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 2014, p. 2. DOI: 10.1109/ICADIWT.2014.6814687.



NTNU

Norwegian University of
Science and Technology