FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Detection of Websites with Extremist Content
Student:	Markéta Minářová
Supervisor:	Ing. Mgr. Ladislava Smítková Janků, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2022/2023

Instructions

Design and implement a hybrid knowledge system for classifying websites that contain extremist topics. The aim of the classification is to distinguish websites whose content is created by extremists from other types of websites.

1) Study the issue of radicalization detection.
2) Study SVM (Support Vector Machine) classifiers and their applications in text classification, implement the selected classifier. Use existing extremist speech datasets for SVM training.
3) Study the shell for creating knowledge systems called Pyke.
4) Design the rule base for the hybrid knowledge system and implement it in Pyke. Use the output of the SVM classifier and human answers as inputs.
5) Perform experiments and evaluate them.

Electronically approved by Ing. Karel Klouda, Ph.D. on 8 November 2021 in Prague.

Bachelor's thesis

DETECTION OF WEBSITES WITH EXTREMIST CONTENT

Markéta Minářová

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Ing. Mgr. Ladislava Smítková Janků Ph.D.
May 10, 2022

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Minářová Markéta. *Detection of Websites with Extremist Content.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Contents

List of Figures

List of Tables

I would like to express my gratitude to my supervisor, Ing. Mgr. Ladislava Smítková Janků, Ph.D., for providing guidance and feedback throughout this thesis. Furthermore, I would like to thank my family and friends for their endless support, patience and encouragement.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 9, 2022 ....

Abstrakt

Tato bakalářská práce se zabývá detekcí extremismu v online prostředí, konkrétně detekcí neonacistických a saláfistických webových stránek. Cílem je vytvořit hybridní znalostní systém, který bude pomocí metody Support Vector Machines a klasifikačních pravidel znalostního systému klasifikovat dané stránky. Řešení tohoto problému staví na základě důkladné rešerše již existujících řešení detekce extremistických textů, obrázků a videí a následně jejich analýzy s pomocí odborné literatury. V rámci této práce byl vytvořen dataset extremistických a neextremistických webových stránek pomocí existujících textových dokumentů a obrázků. Metoda SVM byla použita pro klasifikaci textu a její výstup byl použit jako vstup do znalostního systému. Ten byl vytvořen pomocí shellu PyKe a celý program byl naprogramován v jazyce Python. Jednotlivé SVM modely dávaly velmi dobré výsledky s klasifikační přesností okolo 99 %. Celková klasifikační přesnost systému byla 80 %.

Klíčová slova online extremismus, support vector machines, strojové učení, přirozené zpracování jazyka, znalostní systém, analýza sentimentu

Abstract

This Bachelor's Thesis deals with the detection of extremism in the online environment, specifically the detection of Neo-Nazi and Salafi websites. The aim is to create a hybrid knowledge system that will classify the given websites using a Support Vector Machines method and the classification rules of the knowledge system. The solution to this problem is based on thorough research of existing solutions for detecting extremist texts, images and videos and then their analysis with the help of expert literature. In this work, a dataset of extremist and non-extremist websites was created using existing text documents and images. The SVM method was used for text classification and its output served as input to the knowledge system. This was created using the PyKe shell and the whole program was programmed in Python. The SVM models gave outstanding results with around 99 % classification accuracy. The overall classification accuracy of the system was 80 %.

Keywords online extremism, support vector machines, machine learning, natural language processing, knowledge system, sentiment analysis

List of abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BOW	Bag Of Words
CNN	Convolutional Neural Network
FBI	Federal Bureau of Investigation
FN	False Negative
FP	False Positive
GUI	Graphical User Interface
HSV	Hue Saturation Value
HTML	HyperText Markup Language
ISIS	Islamic State in Iraque and Syria
KNN	K-Nearest Neighbours
LBB	Link Based Bootstrapping
LSTM	Long Short Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
OCR	Optical Character Recognition
POS	Part Of Speech
RBF	Radial Basis Function
RGB	Red Green Blue
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
VSM	Vector Space Model
WWW	World Wide Web

Chapter 1
Introduction

With the rise of the internet, expressing opinions, ideologies and beliefs has become easier than ever before. Due to the accessibility of the World Wide Web, it has become viral in our society, and it got to the point where we could not imagine our lives without it.

Indeed, it would not be such a huge trend if it were not for social media's great invention, where users can freely interact in various, liberal ways. Such platforms might be photo sharing, discussion forums, blogs, gaming platforms, chat apps or social networks. For example, Facebook is the most popular choice among social media, with almost 2.91 billion monthly active users in the fourth quarter of 2021 [1].

Free speech is an aspect of one of the elementary human rights we have – freedom – and this also applies to the online world. Nevertheless, there is a border between free speech and expressing oneself in a way that poses a threat to other people's lives and their freedom. Such 'expressing' can be loosely described as hate or extremist speech against individuals, groups of minorities or a government. This speech promotes hate, violence, crimes and even mass attacks on ordinary people.

Spreading radical ideas and extreme views has become a significant concern for governments and society. Extremists use massive platforms to radicalise and recruit new members, plan terrorist attacks and intimidate people by posting explicit content that self-propagates them. Furthermore, it is the free access to the Internet that extremists and terrorists benefit from and its massive outreach. Anyone can read radical tweets on Twitter, anyone can visit online forums and read extremist ideas in the comments. Unlike in the past, they now possess the power they did not have, and with it, they manipulate vulnerable young people from all over the world into their activities and plans.

It has also become practically impossible to filter all violent content manually in the flood of BIG data that comes in multiple media formats every second. Therefore, the automated detection of online extremism has become a significant research area desperately needed for exploration. Studying how users interact on online platforms, blogs and forums can give away lots of information about which groups tend to stand out or even predict a real-life attack.

Researchers from all over the world have tried to put their fingers on this issue and many have been successful. Their hard work has most definitely contributed to helping government agencies or counter-terrorism organizations combat extremists, yet, there are still new ways extremists get around with the algorithms. Thus, it is still an open area to explore.

The main goal of this work is to create a knowledge system as a helpful tool for an expert when classifying websites as either extremist or not.

My motivation for this work is the sense of contributing to an important matter that is relevant worldwide and a very severe one. It is my view that content like this should not be exposed to the public. Therefore, the task of detecting radical websites is crucial in order to make them not accessible. Furthermore, it felt like a big challenge that I wanted to take upon myself.

Hopefully, the outcome of this bachelor's thesis will contribute generously to the presented issue of online extremism detection.

1.1 Goals

The main goal of this Bachelor's thesis is to create a hybrid knowledge system for classifying websites that contain extremist content from other types of websites. The main extremist areas focused on in this work are Salafism and Neo-Nazism. Other types of extremist content will not be considered. I will also focus on websites in general; therefore, the system should be broad enough to classify blogs, forums and any content on a website. The proposed solution should be achieved using the Support Vector Machines classifier and its output will be an input to an interactive knowledge system. Many subtasks need to be implemented in order to achieve this thesis' goal, specifically:

- research the existing related literature,
- study SVM classifiers and implement the selected one,
- perform analysis of extremist websites and create classification rules,
- create a dataset from existing speech datasets,
- study the shell for creating knowledge systems called Pyke,
- design a rule base for the knowledge system, implement it in Pyke and create an interactive programme,
- perform experiments and evaluate them.

1.2 Approach

The approach to the solution is proposed as follows. The classification of websites consists of two parts – classification of text using AI methods and classification with the help of a user. It is proposed in such a way because image recognition is a complex task and could be a Bachelor's thesis on its own. Incorporating both text and image recognition would be far beyond the scope of this thesis assignment. The text classification is conducted using a machine learning method called Support Vector Machines. The output of this part serves as an input to the second part, the knowledge system. The knowledge system interacts with the user and processes the user's answers to the posed questions. The inference engine concludes whether the given website is extremist or not.

1.3 Bachelor Thesis Structure

This thesis is structured as follows.

Chapter 2 overviews related academic literature in the area of online extremism detection, which considers text, image, video and website classification. It also offers definitions of basic terms used in this work. *Chapter 3* presents the theoretical background of models and methods to understand the concepts this thesis is built upon. Support Vector Machines, NLP and text representation techniques and knowledge system concepts are presented in this part. *Chapter 4* analyses extremist websites and their attributes and also establishes classification rules for the knowledge system. Subsequently, it explains the proposed method as a solution to this work.

Chapter 5 describes used datasets for training the SVM models and explains the creation of datasets representing websites. *Chapter 6* outlines the technologies used in the practical part of this work. Implementation details are also explained in this part. *Chapter 7* presents proposed experiments and discusses their evaluation.

Chapter 2
Related Work

This chapter presents academic literature regarding online extremism detection methods. Firstly, definitions of topic-related terms are presented to establish a common understanding of them in this work. Secondly, the current state of research on extremism detection in cyberspace is explored. These works consider general approaches and detection of individual features, such as text documents, images, videos and websites.

2.1 Online Extremism: Definition of Basic Terms

Let us first define some of the frequent terms used in this thesis since there is no globally accepted or established definition among researchers, which may cause confusion.

Extremism is "political or religious ideas or actions that are extreme and not normal, reasonable or acceptable to most people", according to the Oxford Learner's Dictionary [2]. It is essential to realize that it is context-dependent and influenced by our age. Also, it is a subjective matter, and thus, it is a complex task to generalize something as extremist and non-extremist.

Online Extremism is therefore spreading such religious or political ideas in an online environment. It can be an interaction between members of extremist organizations and groups online, meaning that they use social media platforms, online forums and blogs, where they can radicalise people and promote their beliefs.

Terrorism, according to the FBI [3], is divided into two groups – domestic and international terrorism. However, both refer to violent crimes from people somehow associated with terrorist organizations motivated by political or religious views. Their acts pose a threat to human lives and their intent is to intimidate society, influence the policy of governments, or affect the conduct of a government.

Radicalisation is a set of multilevel processes in which an individual's attitude, beliefs and behavior are being transformed in a way that can lead to supporting, organizing, or committing violence carried out under the auspices of the ideological or religious organization. There are many theoretical models of the process of radicalisation. They mostly have in common the initial phases – the individual's frustration with socio-political contexts and the search for one's identity and success [4].

Jihadism is a contemporary western term describing the ideology of radical Islamic organizations, whose intentions are to recreate society into an Islamic theocracy. It is mainly associated with militant terrorists who advocate violent approaches and even legitimize them [4].

Salafism is often considered to have an identical meaning to Jihadism, but there is a notable difference between these two. At its core, it does not necessarily have to be violent. It aims to return to the old Islamic roots and attempts to clear present Islam. It consists of non-violent propagation, political debates and jihadists' approach [4].

Neo Nazism is a general term for a post-world war II part of a white supremacist movement whose intentions are to create a new order by the example of Nazi Germany. Members adore and celebrate Adolf Hitler, Nordic Warrior myths and Judeo-Christian Bible, from which they take their extremist views [5].

It is challenging to define the above terms precisely because there is no absolute consensus on their meanings, even in the scientific community. However, that is not among the objectives of this work. These terms were defined for the purposes of this thesis and will be used in the above-proposed meanings.

2.2 Current State of Research on the Application of AI in Extremism Detection in Cyberspace

This part reviews the existing academic literature regarding online extremism detection.

In their survey, Correa et al. [6] divided the task of online extremism detection into two most common categories: Link Based Bootstrapping (LBB) and Text classification. First mentioned, LBB is a technique in which various Internet platforms are exploited to retrieve information from it. Several URLs are given at the beginning of the algorithm, and recursively, these links are exploited, so that related URLs are collected until there are no more left. There are several techniques in this field of focused crawling. The idea is to create a net of users, webpages and links that form a whole community of extremists. To demonstrate this technique, a work by Sureka et al. [7] is presented. They developed a framework for detecting extremist videos on Youtube, identifying a list of videos, then extracting more users and videos (bootstrapping) by crawling information such as likes, comments and users' friends and subscriptions. The outcome of their work was successful with a precision of 88 %.

The second most common technique is text classification, a technique in which the documents are sorted into categories or classes. Machine Learning has been associated most with this approach as well as Natural Language Processing techniques. A standard procedure is: collecting data for a dataset, manually labelling the data (even though some semi-automated approaches have been used, but as Sharif et al. [8] have shown, they were not very reliable), extracting features, using NLP techniques for document representation, training a ML classifier and its evaluation on testing data [6].

While target organizations or selected ideologies may differ, it is helpful to look at the ways of text representation, feature extraction and their computational solutions when focusing on extremism detection. The challenges of the text classification technique contain the dilemma of how to represent the given text and what information should be extracted to a feature vector. The most common representations are Bag-of-Words approaches, such as one-hot encoding, TF-IDF, bi-grams and POS tags [6, 9]. Many authors also include sentiment and affect analysis, to distinguish between radical and non-radical texts, as emotions play a significant role in the whole process. Such analysis is a text mining task that contains opinions, emotions and attitudes. Sentiment analysis aims to analyse the views of an individual expressed in the text and measures the overall polarity of the opinions as positive, neutral, or negative. Affect analysis focuses on the emotions expressed in the text and measures their intensity [10].

With these features or their combinations, the following machine learning techniques have been used the most widely: SVM, Naive Bayes, KNN, Logistic Regression, Decision Trees, ensemble methods and Neural Networks.

Classifying extremist websites is very complex. Images and videos appear very often on extremists' websites, but these features were not considered in website classification as the researchers considered only websites' text. That is to the best of my knowledge. Multimedia consists of text, images, videos, audio and animations. For this work, audio and animations were excluded from the focus list. This part also overviews existing researches for image, video and website classification.

2.2.1 AI for extremist text recognition

Concentrating on the area of Afghanistan, Sharif et al. [8] used ML techniques like KNN, SVM, Naive Bayes and ensemble methods to classify tweets as pro-Afghanistan / pro-Taliban or neutral. They discovered that semi-automatic labelling of the tweets using tools like SentiWordNet or SentiStrength proved to have a high error rate. Having demonstrated this problem, they chose to annotate the tweets manually. The tweets were represented as TF-IDF features, using single words and n-grams. They tested all the ML models and concluded that Naive Bayes and SVM had over 80 % accuracy.

Chen et al. [10] research Dark Web forums in order to automatically analyse the sentiment and affect of radical Jihadists who actively use these forums. Domain experts manually labelled extracted sentences from the posts, assigning them to affect and sentiment values. Their feature representation consisted of character, word, root and collocation n-grams. Due to the multilingual character of their work, to obtain the roots of Arabic words, a clustering algorithm was used. They also used a recursive feature elimination algorithm to identify relevant features for each category. They implemented Support Vector Regression to assign an affect value to a specific topic: violence, anger, hate, racism with the value from 0 to 1, and a sentiment value, which went from -1 to 1. The worst accuracy was when predicting the sentiment value with 88 % and the best was about 96 % accuracy when predicting the affect of hate or racism.

In their work, Johnston et al. [11] employed neural networks for text classification to determine whether the web given is extremist or not, specifically Sunni extremist. The objective of their research was to propose a solution that would consider multiple languages and various text documents, short and long. Because their approach considered multiple languages, they could not simply use word embedding methods like Bag-of-Words or its variations to create a feature vector, as it does not contain the context of the sentences. Accordingly, they used word2vec as the word embedding technique, doc2vec for the whole document, as it does not remove semantics and word order. They implemented fully connected neural networks with a parametric ReLU nonlinear function. The prediction output was either 1 – meaning extremist, or 0 – non-extremist. The model accuracy was 93.2 %.

Araque et al. [12] concentrate on affect signs in cyberspace and on creating an embedding-based similarity model for radicalisation detection. This approach was quite innovative as most works focus solely on sentiment analysis, but in this work, they focus on emotions that play a massive role in detecting radicalisation. They exploited emotion features and semantic similarity for radicalisation detection by designing a model consisting of two submodels, one for emotion features and the other for embedding word similarity features. These submodels produce a feature vector, which is then concatenated and serves as input into a ML classifier. They experiment with Linear SVM and Logistic Regression models. The so-called SIMON (Similarity-based sentiment projection) method was developed to extract text features for radicalisation detection using a radical lexicon. When given a text document, it is compared to the radical lexicon from SIMON and the similarity vector is computed between these two. It is brilliant as the lexicon-based approach alone has many performances withdraws. The proposed model gave results of around 99 % F-score.

In their work, Ahmad et al. [13] focused on classifying tweets from Twitter into extremist and non-extremist classes using deep learning and sentiment analysis techniques. They compared standard features, such as BOW, n-grams and TF-IDF, with a word embedding technique, which preserves the context of the text. They proposed their own method LSTM-CNN for classification, which gave the best results compared to other techniques like SVM, Random Forest, Naive Bayes, KNN, CNN, LSTM, FastText and GRU. The LSTM-CNN with word embedding gave a precision of 90 %.

In their study, Bermingham et al. [14] crawled Youtube data to acquire an insight into interactions between radical Jihadists and vulnerable people whom they try to radicalise. They used sentiment, lexical and social network analysis to do this. The crawled data consisted of users' comments and account information. They assign positivity and negativity scores to Youtube accounts and comments to explore the interactions and relationships between users and identify users and groups by their feelings towards some stimulating concepts to jihadists. They could not perform sentiment analysis well. It was challenging to identify the level of subjectivity in the comments. Users expressed opinions by making a statement and not adding some subjective statements, such as "I think" or "I believe." Thus, only the polarity module assigned positivity and negativity scores to words and their cognitive synonyms in WordNet. Lexical analysis was implemented using TF-IDF features to determine the topic of discussion. They compared the most frequent words and the polarity towards these between men and women. They found out that most positive feelings from men were towards topics like Mubarak, Islam, or Judaism, whereas women felt very negative feelings towards Judaism and Christianity. They concluded that women might not be as tolerant to other religions as men are.

2.2.2 Image classification

In their work, Hashemi et al. [15] detected and classified visual propaganda of extremists. They mainly focused on content from ISIS, but their program was able to recognize extremist images from other types of extremist groups. CNN was used to achieve this goal, trained with a large dataset of 120 000 images. They divided the detected images into these classes: hard propaganda, soft propaganda, symbolic propaganda, general organizational communications with ISIS, landscape by ISIS, ISIS-related and other violent extremist organizations. These radical pictures contained themes like guns and weapons, militant training, rockets and missiles, masked men in black, punishments and the disturbing aftermath of attacks. They also included less extreme images yet radicalising, such as symbols, flags, religious images, etc. They used CNN, specifically AlexNet with eight layers, a NN specially developed for image classification, obtaining an accuracy of 86 % for the 8-classes classification and about 97 % for binary classification.

López-Sánchez et al. [16] used image classification to uncover radical accounts on Twitter. They implemented a case-based reasoning framework, which was supposed to identify new extremists and users targeted for the radicalisation process. With help from an external expert, they identified radical profiles. With the assumption that new radical profiles have similar profile pictures as the already uncovered ones, they scraped and classified accounts interacting with them. First, they looked at the followers, tweets and mentions of the monitored radical users then produced a score reflecting their engagement and sympathies with the users. The users with the highest score were chosen for inspection. The similarity was based on icons and symbols in their profile images. They used several image descriptors, out of which the SIFT proved to be the best. Using this, they detected iconography in extremist images and the account was then passed to the experts for monitoring.

2.2.3 Video classification

In their work, Batra et al. [17] explored 24 online extremist videos, which they segregated into single frames using a VLC media player. These frames were then used as training data for the CNN model, which was fitted and then used for prediction. The prediction accuracy was around 97 % when predicting extreme videos but was somewhat lower –56 % –when predicting non-extreme videos.

A thorough survey on automatic video classification summarizing existing techniques and approaches was conducted by Brezeale et al. [18]. Extracting features from video for the classification task can vary, as the features can represent a particular aspect of a video: text, audio or images. Generally, the most commonly used ML classification techniques are Naive Bayes, SVM, Neural Networks, GMMs (Gaussian Mixture Models) and HMMs (Hidden Markov models).

The text-based approach is not very common, but it creates features of two types. First, text in the video can be text on buildings, clothes or anywhere. After identifying this 'viewable text,' the basic text features are extracted from the video using OCR (Optical Character Recognition), responsible for the conversion. Another possibility for text features is to extract the dialog from the video using speech recognition methods.

Audio-based approach extracts the features from an audio signal, which is sampled at a certain rate. These can then be put together into so-called frames. They presented the features as two types: time-domain and frequency-domain features.

The visual-based approach is the most common one and the features are extracted per frame or shot. The frames are the images of which the video consists. Features representing video within this approach can be colour-based, MPEG, shot-based, object-based or motion-based. Eventually, colour-based features are just pixels whose values come from a specified colour space. That could be either RGB, HSV or some other. The colours of the frames can tell a lot about a video when trying to compare it to other videos. Therefore, a colour histogram, the number of pixels for each colour, is used in some frame regions for classification. MPEG, Motion Picture Expert Group, is a popular video representation in many versions. In this approach, pixels are being transformed into another colour space. The hybrid approach mixes all the possible features – textual, audio and visual. We cannot say which features give the best results as there has not been unity in the existing approaches in terms of dataset and metrics.

2.2.4 Website classification

In their study, Ester et al. [19] discussed the problem of complete websites classification. They claimed that most approaches treat websites as a single HTML page, which is inadequate as the website consists of several web pages that are not considered. They refused the opinion that a website is a set of terms or single pages, so they represented the website as a directed graph. With the main web page being a root, links to other web pages are the connectors and the web pages are nodes, so eventually, it is more of a tree representation. They proposed three approaches for web representation. First was a "single super page" approach. This means that a feature vector is created by counting word frequencies in all the web pages. They argued, however, that this approach is not efficient, for example, when classifying websites in other languages. Furthermore, the text context in a document "loses" its original location and this information is not embedded in the feature vector. Therefore, another approach was proposed – the "topic vectors", in which the single web pages of a website are represented by a single topic (which is determined by text classification methods) and the website is represented by the topic frequencies of these web pages. The third proposed approach is "website trees". With this approach, a website is a tree with nodes representing single web pages and topics.

2.3 Chapter Summary

This chapter presented definitions of basic terms for this thesis. Then, online extremism detection methods were divided into two main approaches: LBB and text classification. The LBB technique exploits links on websites and crawls them to uncover more extremist websites. The text classification technique utilizes machine learning models to detect extremist content. This approach typically needs many training data to perform well, whereas the former does not require any training data. The most standard ML methods in text classification were briefly outlined.

Then, the AI methods in extremism detection in cyberspace were examined. Firstly, papers regarding text classification were explored. These were highly successful with the models' predictions, with the SVMs being the most commonly used ones. The papers also concentrated on various text representations, from the BOW approaches to more sophisticated ones such as the word2vec. Papers regarding image, video and website classification were also examined. Furthermore, the lack of publicly available datasets for online extremism was found to be a significant drawback in this research area.

Chapter 3

Theoretical Background

This chapter presents the theoretical basis to better understand this thesis's concepts. The discussed topics are methods for text processing, such as Natural Language Processing and text representation methods. The machine learning method called Support Vector Machines and the Knowledge System concepts are briefly introduced.

3.1 Natural Language Processing

This section will introduce the most common NLP techniques for text processing.

Many types of research are active in the area of NLP. Allen [20] proposed its definition by stating that it is a "theoretically motivated range of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis to achieve human-like language processing for a range of tasks or applications".

Tokenization is the initial step of NLP. Its essence is to identify words (or/and punctuation) from an input text which will be processed later. It aims to chop off the sentences into pieces – tokens. Sometimes, punctuation and other characters might be thrown away in the process.

Stop words are terms that are immensely common in the specified language and do not add much value or information to the text as a whole. Therefore, we often need to drop these as we want to filter the most relevant words.

Normalization or perhaps equivalence classing of terms is a process of modifying terms to match similar terms. For example, "Extremism" means the same as "extremism," but the computer would not match these two terms. Therefore, in this case, Normalization will be lowering all characters. Some other modifications might be removing diacritics (valid in the English language, yet not in all languages) and case-folding, which means we transform all characters to lower case.

Stemming is a process of transforming the word to its base form to reduce the number of words and also match ones whose meaning is the same. This is done by chopping off the end of the word in the hope of finding the base form, but this is not effective all the time.

Lemmatization deals with the problem of cutting too many letters by focusing on using vocabulary for helping decide what to do and analysing the word given [21].

3.2 Text representation

One of the major problems when creating a computer program that understands natural language is representing the linguistic input. Text representation is the most fundamental task to represent text documents numerically. The most common approach is the VSM, Vector Space Model. In this model, each text document is represented by a numerical vector. The similarity between these vectors can be computed by various kernels, for example, the Cosine similarity:

$$d_{cos}(X,Y) = \frac{X \cdot Y}{\|X\| \cdot \|Y\|}. \tag{3.1}$$

Other than VSM, there are also possibilities to represent text such as neural networks or graphs [22].

The feature vector primarily used in this field of expertise can be divided into lexicon-based and word embedding. Lexicon-based, or rather keyword-based, representation is an approach that relies on word frequency and feature selection of a pre-defined lexicon. It is primarily suitable for topic detection, as the word occurrences and frequencies function well for predicting the main topic of the text. At the same time, it removes context and semantics from the documents, so it might not be able to recognize the radical text from the neutral one containing the exact words. Some incorporated sentiment analysis as a part of lexicon-based approaches and while it may add up the emotion and tone of the text, it still might not reflect the true essence of its purpose as the human language is more than word frequencies, it also has semantics and syntax [8].

Affect computing and sentiment analysis are techniques of text mining problems necessary for understanding human emotions, opinions and subjectivity of the text. Sentiment analysis, alias Opinion making, is a process of finding opinions, identifying sentiments they express and classifying their polarity [23].

Bag-Of-Words is a simple VSM representation of a document that uses the words in a document as the index of the feature vector. This vector can contain different weighting schemas, among which can be: binary values (1 if a term is present in the document, 0 otherwise, known as one-hot encoding as well), term occurrences (how many times a term occurs in a document), term frequencies (term occurrence divided by the number of words in a document) or TF-IDF, which indicates how a term given is semantically relevant to the document. The disadvantage of this approach is that it creates sparse high-dimensional vectors and demands lots of memory space. Another disadvantage to add up is, as mentioned in the previous chapter, that it does not preserve the semantics or the syntax [24].

3.3 Support Vector Machines

Support Vector Machines first came to light in 1995, when Corted and Vapnik [25] published their research on statistical learning theory, where they introduced Support Vector Networks, now known as Support Vector Machines.

Support Vector Machines is a supervised machine learning technique widely used for classification and regression. In this work, I will solely focus on classification, specifically binary classification. The main idea is to discriminate data into two categories by constructing a linear decision surface (a hyperplane) that separates the data into two half-spaces. An optimal hyperplane would minimize the classification error of the data, so the goal is to separate them maximally. Such hyperplane is obtained by maximizing the margin – summation of the shortest distances from the hyperplane to the closest data points from both sides. These closest data points are called the “support vectors”.

Suppose we have a set of m labelled training data:

$$D = \{(y_1, x_1), ..., (y_m, x_m)\}, \tag{3.2}$$

where $y_i \in \{-1, 1\}$, $x_i \in R^n$ and $m \in N$.

The classification function $F(x)$ is in form:

$$F(x) = w \cdot x + b, \tag{3.3}$$

where w is the weight vector, b is the bias and $x \in D$.

Let us first assume, that all our data can be classified with no errors and with a linear function. Such approach is called "The Hard-Margin". D is linearly separable if there exists a vector w and a scalar b such that the following inequalities apply to all elements of D:

$$\begin{aligned} w \cdot x_i + b \geq 1,\ if\ y_i = 1, \\ w \cdot x_i + b \leq 1,\ if\ y_i = -1. \end{aligned} \tag{3.4}$$

These inequalities can be also modified to:

$$y_i(w \cdot x_i - b) \geq 1,\ \ \forall (x_i, y_i) \in D. \tag{3.5}$$

The optimal hyperplane in this example is:

$$w_0 \cdot x + b_0 = 0, \tag{3.6}$$

where distance from a hyperplane to the closest data points on one side is:

$$arg\ min_{x \in D} d(x) = \frac{|w \cdot x + b|}{\|w^2\|} \tag{3.7}$$

and the maximum margin is:

$$margin_{max} = \frac{2}{\|w^2\|}. \tag{3.8}$$

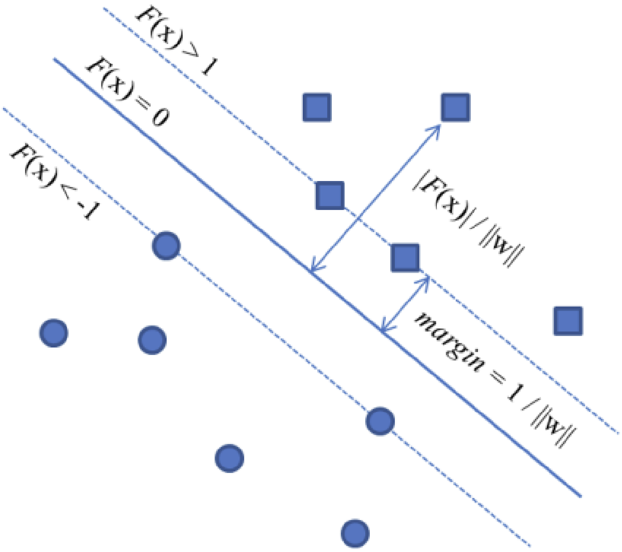


Figure 3.1 Hyperplane maximizing the margin in a 2-dimensional space. Image was taken from [26].

Constructing such hyperplane is a quadratic optimization problem, in which $\|w\|$ is minimized in order to maximize the margin. The solution to this consists of formulating a dual problem and makes use of Lagrange multipliers, however, the details will not be discussed in this work.

In reality, the majority of any training data cannot be separated without errors. Solution to this is to minimize the number of errors. To achieve this, a "Soft-Margin" approach is used, as it allows misclassified data points while still maximizing the margin. To measure the misclassification, non-negative variables (the so-called slack variables) are introduced: $\xi_i \geq 0, i = 1, ..., l$

The new optimization problem then becomes minimizing also the number of errors:

$$Q(w, b, \xi_i) = \frac{1}{2}\|w\|^2 + C \cdot \sum_{i=1}^{l} \xi_i, \tag{3.9}$$

where C is a parameter determining the tradeoff between the margin and the amount of misclassification in the training data. The presented approaches constructed hyperplanes in the input space. That is good for linearly separable data, but in the other case, there is no linear hyperplane that could separate the data.
The classification function is now in different form:

$$F(x) = w \cdot \phi(x) + b, \tag{3.10}$$

where $\phi(x)$ is some basis function. These functions are then replaced by kernel functions in the form $k(x, y) = \phi(x)^T.\phi(y),\ \forall x, y \in R^p$. This can be done thanks to a dual representation of a dot product of transformed features from the original input space. This substitution is called the *kernel trick*. In this process, the input vectors are mapped into high-dimensional feature space, where the data are linearly separable. In such a new feature space, it is possible to construct a linear hyperplane to separate the data. The mapping can be achieved using various non-linear functions, such as follows:

$$Polynomial : K(x, y) = (x \cdot y + 1)^d,\ \forall x, y \in D, \tag{3.11}$$

where d is the polynomial degree.

$$Radial\ Basis\ Function : K(x, y) = \exp(-\gamma\|x - y\|^2),\ \forall x, y \in D, \tag{3.12}$$

where $\gamma = \frac{1}{2 \cdot \sigma^2}$ and σ is the variance of input data.

$$Sigmoid : K(x, y) = \tanh(\kappa x \cdot y + c),\ \forall x, y \in D, \tag{3.13}$$

where $\kappa \in R/\{0\}$ and $c \in R$.

In the new feature space, SVMs are then used to find a maximum margin hyperplane [25, 26, 27].

3.4 Knowledge Systems

Knowledge systems are one of the areas of AI. They are computer programs used to solve problems requiring human intelligence in specific problem domains. They are designed to help experts in decision-making or provide expert knowledge to less experienced users.

A knowledge system primarily consists of the following components: a knowledge base, inference engine, explanation module, fact base and a user interface. The knowledge base contains the knowledge from experts of a specific problem domain. These can be rules, frames, mathematical logic or semantic nets. The inference engine analyses and processes the rules from a knowledge base. That way, it creates new facts and fires new rules and, in the end, comes to a conclusion when there are no more rules it could match. It operates in two approaches: forward-chaining and backward-chaining. An explanation module is a part of the knowledge system that explains the system's actions. It serves to explain why the system came to a particular decision. A fact base contains knowledge of a specific case that is input into the knowledge system or facts created during the inference engine. A user interface is a module that communicates with the user. It should be comprehensible and clear to acquire information from the user.

Tools for developing knowledge systems are called “shells”, which are environments for creating knowledge-based applications [28].

3.5 Chapter Summary

This chapter introduced models and techniques to explain the theoretical concepts of this work. Firstly, Natural Language Processing techniques for text processing were presented, followed by text representation techniques. Then, the Support Vector Machines were explained for linearly separable and non-separable data. Furthermore, the knowledge system and its main components were presented.

Chapter 4

Proposed method

This chapter presents the analysis of extremist content: text, images and videos. Firstly, it explores their characteristic traits to establish classification rules for the knowledge system. The classification rules are then presented individually for images, videos and the website. An overall classification method is summarised and proposed at the end of this chapter.

4.1 Extremist websites

A detailed analysis of individual attributes appearing on extremist websites is a necessity for the task of their classification. The challenge is to extract the right features to maximize the model's accuracy. With the help of academic literature, the below-listed attributes are presented based on my observations.

4.1.1 Analysis

Let us first start with text analysis. Some keywords associated with each ideology – Neo Nazism and Salafism – are frequently used among the communities. The Salafi community often uses a specific linguistic style – jargon – containing Arabic words, references or quotes from religious texts. Some of the most common expressions are presented in the table 4.1. Religious words, leaders' names, references to specific attacks and years are also typical in the extremist community [29, 30].

Apart from religious texts and citations from the Quran, Salafi's text is often full of powerful mixed emotions. That is, after all, what extremists try to achieve – to evoke powerful emotions and manipulate them towards their cause or beliefs. Some members identify with Islam, but

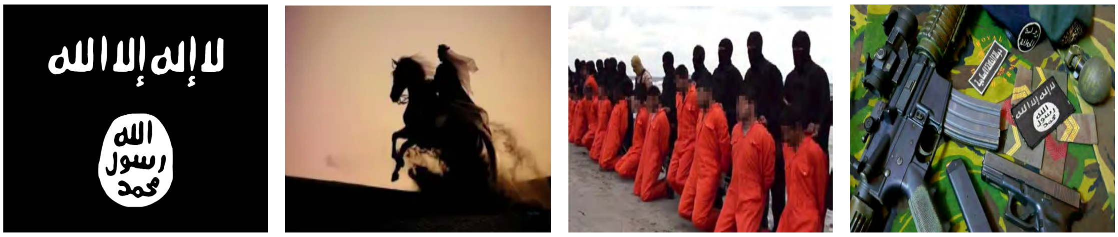
Arabic Term	Extremist meaning	Non-extremist meaning
umma	community of "true" believers – Salafists	community of believers
takfir	labelling the enemies	labelling someone as non-believer
murtad	Muslims who abandoned their loyalty	apostates
kuffar	enemies of Muslims	non-Muslim, non-believer
munafiq	Muslims, but not Salafists	hypocrite
hijra	migration to a more pro-Salafi area	emigration
sharia	holistic rules for all aspects of Muslim's life	Islamic values and norms

Table 4.1 Arabic terms in English texts and their meanings.

they do not know its religious traditions. Emotional propaganda appeals the most as they accept what sounds good and is illustrated with good imagery. Often, the pursued goal is to make the reader feel stressed because when people are stressed, it is more difficult to remain rational when presented with factual information. Instead, they incline to faster, more absolute solutions favouring totalitarian organizations [29].

In terms of Neo-Nazism, lots of German phrases are used by the members, often phrases from Nazi Germany like “Blut und Ehre” for “Blood and Honor”, “Weiss Macht” for “White Power”, “Meine Ehre Heisst Treue” for “My Honor is Loyalty” or “Sieg Hail” for “Hail Victory” [31].

Secondly, images play a supportive and very influential role as well. The human eye perceives visual information most quickly; therefore, disturbing images are often implemented to stir one’s feelings and make one more focused on the message presented. There are also many symbols among extremist communities in which they recognize themselves and are essential for their self-declaration and devotion to their cause. The Salafists successfully make use of solid imagery with catchy slogans as well as symbols or memes for their propaganda. One of the symbols representing them is the Black Banner, an official flag used by ISIS. There are two texts, the top one saying “There is no god but God” and the one in a white circle, “Muhammad is the messenger of God”. Such a banner is forbidden to wear in some countries. Another portrayal is somewhat connected to martyrdom, emphasizing the great afterlife of the martyr. For example, smiling dying soldiers. Weapons and armoury are significant aspects of Salafi images as they show strength and add up to their intimidating image. Some more extreme groups reveal very explicit content, like imagery from a beheading scene or any brutal punishment. These typically feature captives in orange jumpsuits tended by men in all black. Animals play an essential role too. The most iconic image is a Salafi riding an Arabian horse. It makes an impression of a great warrior and it is very romanticized. Cats were Muhammad’s beloved animals and held a special position in the communities. They are also often shown in pictures, often with guns and other armouries. Another animal portrayed with a slogan is a lion, evoking strength and courage. It might promote martyrdom and motivate people to become one. That is also true for green birds, who are told to represent the souls of the martyrs [29].



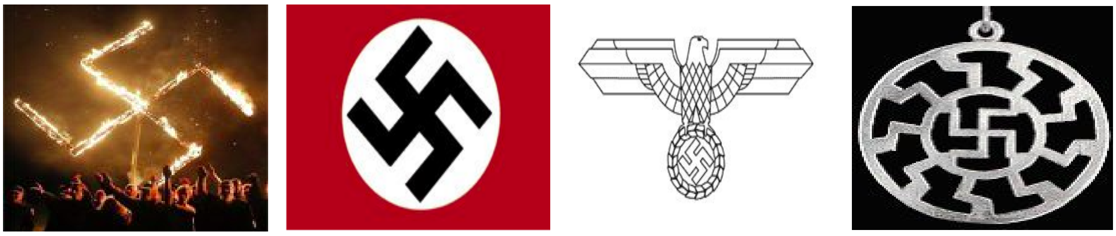
(a) ISIS flag **(b)** Rider on a horse **(c)** Execution crime scene **(d)** Weaponry

Figure 4.1 Salafi promoting imagery. Images a, c and d were taken from [32]. Image b was taken from `https://skylandtourism.com/wp-content/uploads/2018/02/horse.jpg`.

Symbols associated with the Neo-Nazi groups often originate in the 1930s when Nazi Germany was on its rise. However, there are some that are quite newly adopted, too. For starters, an “88” symbol is a code for the phrase “Heil Hitler” as the letter “H” is eighth in the alphabet. It is used not only by the Neo Nazis but in the whole white supremacy movement. It can be sometimes combined with the number “14”, which refers to the “14 words slogan” that goes like this: “We must secure the existence of our people and a future for white children”. The *swastika* is one of the ancient symbols the Nazis had adopted and it is the most recognizable symbol of hate. It appears mainly in the Nazi Party flag, Nazi Eagle, or Sonnenrad, an ancient sun wheel.

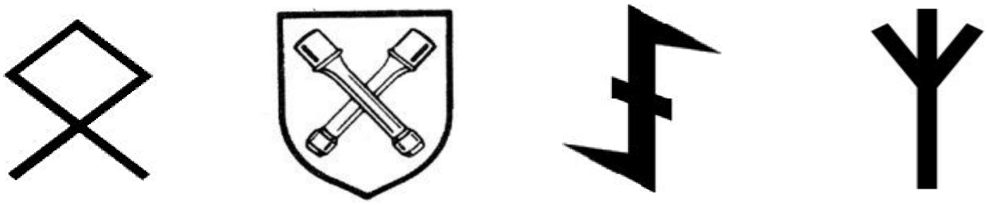
Another ancient symbol adopted by the Nazis is the Life Rune, the Othala Rune, Wolfsangel, a symbol to ward off wolves. “Zyklon B,” the deadly gas Nazis used in concentration camps, can be found in tattoos, clothes, flags, or Nazi images [31].

The Celtic Cross was originally symbol for the Celts, but is used by white supremacists (including the Neo Nazi) these days. Other common symbols might be a thunderbolt, symbolizing Hitler's military arm Schutzstaffel, crossed hammers, or W.A.R. abbreviation, standing for "White Aryan Resistance". A sample of these hate symbols is shown in figures 4.2 and 4.3.



(a) Burning swastika **(b)** The Nazi Party Flag **(c)** The Nazi Eagle **(d)** Sonnenrad

Figure 4.2 Common Neo-Nazi hate symbols with Swastika. Images were taken from [31].



(a) Othala **(b)** Crossed Grenades **(c)** The Wolfsangel **(d)** Life rune

Figure 4.3 Other Neo-Nazi hate symbols. Images were taken from [31].

As for the videos, there are explicit ones with various executions. These can be shooting, beheading, burning and more ways to execute prisoners. Extreme content like this almost always shows up on an extreme website. Again, it is a way of spreading violence, which is sometimes successful among young people, to copy these actions in real life. This propaganda-style is typical not only for Salafists but also for other groups, among which we can classify the Neo-Nazis, who are mimicking them [32].

There can also be non-explicit videos of individuals talking about their beliefs, stating facts, or explaining their actions or motives. With these kinds of videos, it is crucial to look for extremist signs and listen to what they say.

4.2 Classification methods

This section presents an overview of how the text, images and videos will be classified and how they will contribute to the whole website's classification.

4.2.1 Text classification

The task of text classification must first undergo various pre-processing subtasks to convert the text and symbols into numbers. Text documents will be pre-processeded with the NLP techniques tokenization, stop words removal, normalization, stemming and lemmatization. They will be represented as a BOW feature vector with term frequencies. Sentiment analysis will be implemented with *Subjectivity* and *Polarity* values.

4.2.2 Image classification

Classification rules for the images are presented in this part. The user will be asked to examine all images to look for specific clues and aspects. Any show of explicit violence increases the probability of an extremist website. The user will be shown several symbols or scenes that can appear somewhere in the website's images.

If images are present, four categories can represent them: explicit images, militant images, hate signs images and soft images. These categories are not exclusive. Explicit images are those most extreme; these can be execution scenes, their aftermath, all sorts of punishments (whipping, stoning), images with blood, injured people and other disturbing scenes. For the most part, these images are always considered extremist.

Militant images show weapons, grenades, knives and other sharp weaponry displayed or used. Militant vehicles, the destruction caused by military attacks, masked soldiers and soldiers doing the Nazi salute are also considered.

Hate sign images contain various signs and symbols, as shown in the previous analysis part. This category is valid if one or both of the Neo-Nazi and Salafi group signs are present.

The soft images category includes scenes that portray the group as an idyllic choice for life, showing images featuring children, people praying, family activities, religious themes, farming and more. This is more common for Salafi images than Neo-Nazi but can be found in both. These images alone are not necessarily extremist, however, in combination with other features, they can be considered extremist.

Facts created from these rules will have a “True/False” form. That is two possible values for each category. To consider all possible situations, sixteen combinations of the facts need to have a rule.

The rules were established in the following way: any combinations with explicit images fall into the very extremist images category as it is content likely produced by extremists. Apart from these rules, there is also a possibility of combining militant images, hate sign images and soft images without explicit ones to be considered very extremist. Medium extremist images are without explicit images but have militant images with any of the remaining and hate signs images with any of the remaining. Non-extremist images include a variation when none of the four categories is met and the one with only soft images being present.

These are the classification rules for imagery. To obtain the four facts' categories, questions are posed to a user along with pictures to identify correct features. Individual categories are set to “True” if at least one image contains given features.

4.2.3 Video classification

Classification rules for videos are principally analogous to the ones for images. The user is requested to examine the videos on the website and look for specific elements. If videos are present on the website, four categories can represent them: explicit videos, militant videos, hate sign videos and soft videos.

Explicit videos show executions, attacks, bombing, torturing, the aftermath of previously mentioned scenes and other disturbing, explicit content.

Militant videos can be considered driving vehicles, scenes with masked men training, making speeches, skinheads gatherings and protests, shots with guns and other weaponry.

Hate sign videos are videos in which hate symbols or texts are identified from the previous analysis. The user is once again shown the pictures of signs.

The soft video category includes Muslim people praying, a day-in-life video, family traditions, discussion about religion, speech about race, white people gatherings, traditions and more. This category is again the most complex one and would require external expertise. It is here because although these scenes may not seem extremist individually, they can reasonably make an extremist website in combination with other aspects.

Sixteen combinations of these categories need to have rules. If an explicit video is present, it is a red flag of an extremist website, so it is therefore in the very extremist videos category. It is also the same one when there is not an explicit video but a militant and hate sign. If either one of the militant and hate signs videos is present, it is in the medium extremist videos category.

Furthermore, non-extremist videos are the same as images – either there are only soft videos or none of the categories.

These are the classification rules for videos.

4.2.4 Website classification

Previously constructed categories and facts are assembled in the second part of the knowledge system after the user provides enough information about the images and videos. In this part, rules for the whole website classification are presented. As mentioned earlier, there are three categories for both images and videos: very extremist, medium extremist and non-extremist. There is also another one about their presence. Furthermore, there is also a text, which was classifier in the first part by an SVM model. So there are three aspects: text, images and videos. The text has two categories and the other two have four. That makes the number of combinations equal to thirty-two.

When one of the features is extremist, the website is regarded as extremist. So, if the text alone is extremist, the other features do not matter. That makes half of the possible combinations and narrows the rules. A website can be classified as extremist with non-extremist text but very extremist videos or images. One other possibility is non-extremist text and medium extremist both images and videos.

A non-extremist website is classified with non-extremist text with medium extremist images and not present or non-extremist videos. The same is in the situation when there are medium extremist videos. If images and videos are either not present or non-extremist, then a website is classified as non-extremist.

4.3 Proposed approach

The entire classification consists of two parts.

In the first one, the SVM classifier takes the website as input and after text pre-processing, it will predict the value of 1 or 0, a value indicating whether a text given is extremist or not. This information will be saved as a fact in a fact base of the knowledge system and then used in the following procedures. In the second part, the knowledge system interacts with the user and uses the inference mechanism to prove that the website is extremist. The user will be asked to examine the images on a website and answer several "yes/no" questions. The questions will be supported by example images so that the user knows what to look for. This question-answer process produces new facts: very extremist images, medium extremist images and non-extremist images. They indicate how extreme the images are and, according to further classification, will be used to decide on the website.

Then, the user will be requested to examine videos on the website. Based on the user's answers, the system will produce three types of facts: very extremist videos, medium extremist videos and non-extremist videos. These facts will also play a crucial role in deciding on the website's extremism. The acquiring of new facts is done by using forward-chaining rules. Proving a goal utilizes backwards-chaining rules. These rules consider all the possible combinations of the images and video facts, making it a total of 32 combinations since images or videos can be missing. The outcome of the system is information on whether the website is extremist or not.

4.4 Chapter Summary

This chapter presents a detailed analysis of extremist websites and their features. Studying simultaneously two extremist groups of Salafists and Neo-Nazis, text, image and video features were covered for both groups. It was observed that these groups have a broad symbolic basis and also other forms of expressing themselves, widely in a militant and explicit sense.

Rules for image classification were established in the following section. They were divided into four distinct groups: explicit, militant, hate signs and soft images. If present, they could be divided into other higher-level groups in many combinations of either very extremist, medium extremist or non-extremist images.

Identical rules were established for the videos, forming a sense of unity within the rules. They were also separated into four distinct groups and from combinations of these into three main groups.

The website classification rules consisted of rules for text, images and videos. It was proposed that if any of the features is extremist, the website is automatically classified as extremist. More complex cases were also considered.

Finally, the whole classification process was proposed to summarise all the rules and individual features.

Chapter 5
Dataset

This chapter describes existing speech datasets that were used to train and test the SVM models. Furthermore, it also explains the creation of a dataset that represents extremist and non-extremist websites, which had to be created manually due to the lack of publicly accessible extremist datasets.

Datasets are one of the most significant drawbacks surrounding this research area due to their lack. It is understandable as the researchers do not wish to spread extremist material; however, it is an obstacle in this area. Many papers that were analysed in chapter 2 used datasets with messages from Twitter because they are one of the few publicly available ones.

5.1 Datasets for classification

One of the crucial steps in this thesis was to train a machine learning model. Thus, a proper dataset was needed. Kaggle[1] was an excellent data source, so all the datasets used for training of the SVM model were downloaded from this site.

It was desired that the dataset is balanced with the same amount of extremist and non-extremist data. The first introduced datasets represented Salafi and Neo-Nazi extremist and non-extremist texts.

The first dataset used for this work is called "Terrorism and Jihadist Speech Detection" [33]. These were, in fact, two separate datasets, but for the data pre-processing part, they were merged into one. Also, they were filtered as they contained messages in Arabic and French, but this work focuses on websites written in English. Eventually, this dataset was reduced to 217 data records and had two columns: "text" and "label", in which label value was either 1 or 0, extremist, respectively non-extremist.

The second dataset is called "Religious Texts Used By ISIS" [34], containing 2,685 religious texts from ISIS English-based magazines Dabiq and Rumiyah. It contained information about the texts' source, purpose, or dates, but only the "Quote" column was used. All of these were classified as extremist, with a value of 1.

The third dataset also contained Salafi related text messages, called "How ISIS Uses Twitter" [35]. Having 1 7410 messages from Twitter pro-ISIS accounts, only the text column was used. It also contained account name, username, description, number of followers and other data. These were assigned value of 1, as extremist.

The next dataset focused on Neo Nazi tweets, the "Nazi Tweets" [36], with 114 240 tweets scraped from extremist Twitter accounts. It also contains user information, hashtags and dates;

[1] https://www.kaggle.com

however, the only used columns were the actual content. Then, the column “label” was created to assign extremist/non-extremist values to the documents. However, only 20,000 of these were taken so that the data would be balanced as there was not as much Salafi textual data. The rest of the data was used to create the website text dataset.

Following datasets needed to be non-extremist, ideally reports, news or articles. The only non-extremist dataset used for text classification was “All the news” [37], containing over 143 000 articles from several American publications. Approximately 20 000 of these were used to represent the non-extremist category. This amount was chosen since the content of the articles is much longer than those of Twitter messages. It also contained other columns like heading, author or publication dates; however, only the “content” was used for the classification. Furthermore, all the data were labelled as non-extremist with zero value.

Overall, a dataset of 20 312 Salafi, 20 312 Neo Nazi and 20 312 news articles was concatenated to form a dataset of 60 936 data records.

5.2 Dataset of websites

Dataset for the knowledge system did not have to be as large as the one for the SVM training and testing, since it was to be tested by a user. Therefore, the created dataset contains records of 40 supposed websites as a representative sample. Half of them are Salafi-related and the other half are Neo-Nazi-related. These two halves are then half extremist and half non-extremist.

Individual websites were not created classically as HTML pages, as it could be considered to be spreading extremist content. They are represented as supposedly extracted text files from the imaginary websites and a directory of images downloaded from the websites. Parts of described dataset in section 5.1 the “Nazi tweets” were used to represent the extremist text within the Neo-Nazi websites and the “All the news” datasets was used for non-extremist text. There was a lack of Salafi text documents, so the text documents were created as randomly generated list of words. These words were selected from table 4.1 and also some frequent words found in the datasets used for the SVM classification.

Then, various images of symbols, hate signs, militant themes and such were downloaded and used as illustrative pictures and by combinations of them represented the images of websites. Videos were not inserted as the content is rather disturbing. Furthermore, the very extremist features classification is always straightforward, so the attention turned to the ones less extreme – symbols, militant themes and soft images. Explicit images were not used.

Non-extremist websites were created in combination with the “All the news” article’s text and with the same images from the supposed extremist websites.

All of the websites were separated into single directories.

5.3 Chapter Summary

This chapter described used datasets and their creation for the SVM classification and also the knowledge system. All the datasets were downloaded from Kaggle.com and most of them contained text messages from Twitter. Datasets created for the knowledge system represented websites, however for legal reasons, they were not created as HTML pages, but contained supposedly scraped text and images from the websites. The text documents for Salafi extremist websites were randomly generated from a defined set of words.

Chapter 6

Analysis and Implementation

This chapter clarifies the implementation details of the thesis's practical part. Firstly, a quick introduction to the Pyke knowledge engine shell and its structure is explained. Secondly, a summarization of used libraries, frameworks and technologies is outlined. Then, the actual implementation part takes place, starting with the data pre-processing part described in detail. Then, the SVM implementation is discussed and used tools were outlined along with the tuning of the hyperparameters. Next, the knowledge system structure is explained and individual components are described. A created custom GUI is described in the following part.

6.1 Technologies

The entire application and the ML model are implemented in the *Python* programming language. This one was chosen mainly for its diverse, accessible libraries and frameworks written in Python, which are specifically helpful in working with mathematical models and data science.

6.1.1 Pyke

The knowledge system was programmed in a shell called *Pyke*, the Python Knowledge Engine. It is a form of Logic Programming inspired by Prolog, a knowledge system shell, but written purely in Python. Therefore, it can be integrated into Python programmes pretty easily.

The main object is called the *engine*, which is a core component when creating a Pyke knowledge system. When created, Pyke scans the folder for specific files and compiles them into Python source files. All of the necessary functions to work in Pyke are provided by this object.

Pyke has three types of knowledge bases, which have their own specific source files: fact base, rule base and question base. A fact base stores facts, simple statements with a name and arguments. They are of two types: universal, which form input data when the engine is launched and case-specific facts, which are added later during the inference mechanism. The latter ones are deleted once the programme finishes. These facts have to be stored in a Knowledge Fact Base (KFB) file.

A rule base consists of a collection of rules, both forward-chaining and backward-chaining. The forward-chaining rules cannot directly prove a statement's truth, but they can create new facts to the fact base with the inference mechanism. To prove a goal, backward-chaining rules have to be used. A knowledge system can have multiple rule bases, but one must be activated before its utilization. The rule bases have to be stored in Knowledge Rule Base files (KRB), which have a specific syntax.

A question base contains questions that will be provided to the system's user. These are

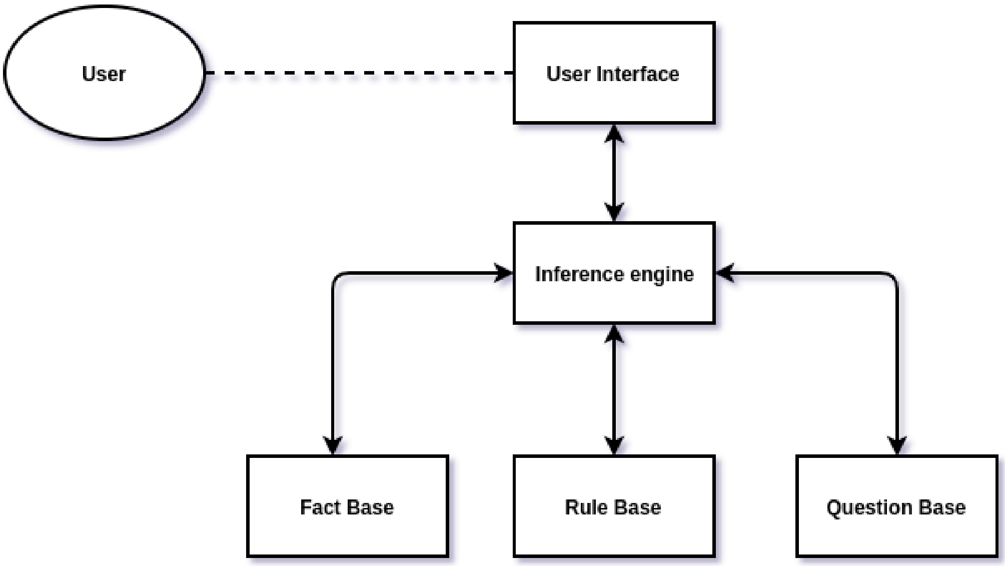


Figure 6.1 Pyke system's structure

typically used within the rule bases to get additional information from the user to create new facts. They have to be put into the Knowledge Question Base (KQB) files.

The core of a programme is typically in a file called *driver.py*, where the knowledge engine is initiated. It is also where the rule bases need to be activated and where all the programme logic lies [38].

6.1.2 Libraries

To highlight the most important used libraries, a brief overview is presented.

Scikit-learn [39], or so-called Sklearn, is a Python library for machine learning built on other popular Python libraries like NumPy, SciPy, or Matplotlib. It was used to train the Support Vector Machines model with the help of parameter tuning functions.

Pandas library was utilised for data pre-processing tasks and analysis.

TextBlob [40], a textual data pre-processing library, served well for calculating the sentiment analysis values, the subjectivity and polarity scores.

NLTK [41], the most popular library for NLP tasks like tokenization, stemming, stop words removal and generally any tasks for working with human language data, was used to complete these.

Tkinter [42] is a graphical user interface toolkit which operates on many programming languages. In this work, it was used to build a GUI for the knowledge system.

6.2 Implementation

In this part, the actual implementation of the SVM model and knowledge system is described.

6.2.1 Data pre-processing

Data pre-processing consisted of loading individual datasets, removing unnecessary columns and keeping only “text” and “label” columns. If the “label” column was missing, a new one was created and values were assigned according to the nature of the dataset.

Then, the tokenization process was performed with an NLTK’s function *word_tokenize()*, followed by removing non-letter characters and normalizing the tokens to lower case. English stop words were extracted from the list of tokens and a stemming technique was applied to the remaining ones. A set of these was then merged into one string, and thus, a vocabulary was created. For demonstration purposes, most common tokens from the extremist and non-extremist datasets were extracted into a picture as shown in figures 6.2 and 6.3 with the *WordCloud* library.

Then, all the individual datasets were concatenated together to create one extensive dataset of approaximately 60 000 data points.

TextBlob’s sentiment analysis tools were applied to the untokenized text in order to obtain a sentiment value and a polarity value.

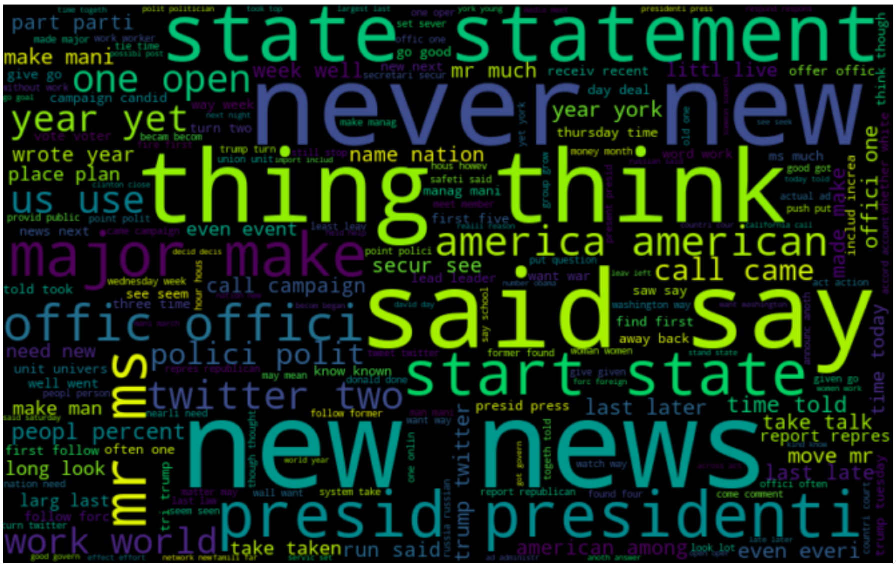


Figure 6.2 Most frequent tokens in the non-extremist data

To finish the data pre-processing part, the datasets had to be put into a computer-friendly form – the numbers. *TfidfVectorizer* from *scikit-learn* was used to convert a collection of text documents into a matrix of TF-IDF features. This implementation outputs a sparse matrix to save some space and the following functions from scikit-learn work well with such sparse representation.

6.2.2 Implementing the SVM model

Data were divided into training and testing groups, with the former being used to train the models and the latter used to evaluate the models’ performance. The division ratio was 7:3. Validation data did not need to be created as the model was put into cross-validation to find the optimal hyperparameters of the model; therefore, the validation data were created during the process. The model was fitted with these hyperparameters on the whole training data when they were found.

To divide the data and perform the cross validation, *train_test_split()* and *GridSeachCV()* from sklearn.model_selection was used. GridSeachCV needs a machine learning model and a dic-

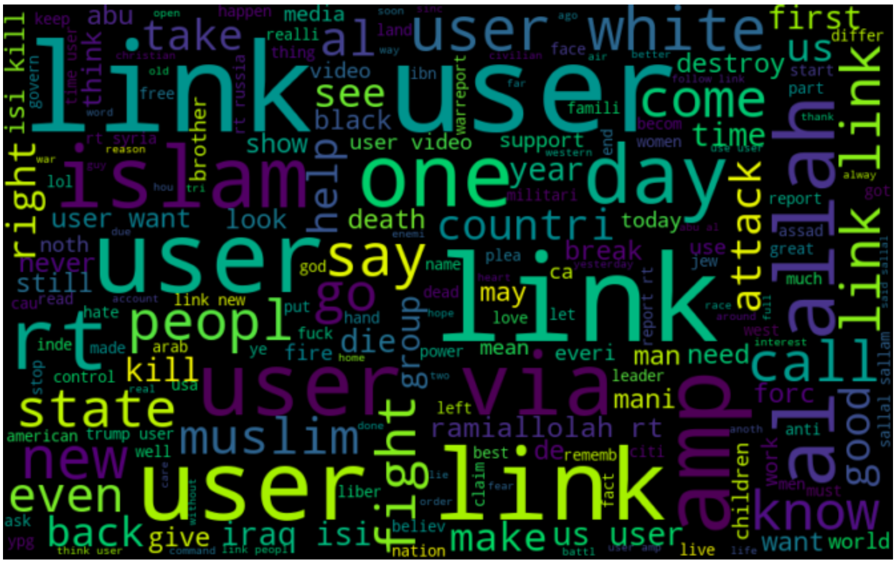


Figure 6.3 Most frequent tokens in the extremist data

tionary with parameter names and their values that will be explored within the cross-validation. After calling its *fit()* function with training data and a target variable, the GridSearchCV performs the cross-validation for each of the parameters and evaluates each model's average score when it finishes all folds for specific parameters. At the end, it evaluates the model with the best parameters and saves that model, which can then be used for the prediction.

6.2.2.1 Tuning the hyperparameters

Scikit-learn's implementation of the SVM, *svm.SVC()*, allows using different kernels for the classification. In this work, three were used: linear, RBF and sigmoid kernel. The polynomial kernel was left out due to the computational demands. All of these kernels have a C regularization parameter, which represents the amount of the allowed misclassification of the data during the classification. It controls the tradeoff between the decision boundary and misclassified data points. The models were trained with four C values for all the kernels: $\{0.1, 1, 10, 100\}$.

Another parameter only for the RBF and sigmoid kernels, was γ. This parameter defines how far the influence of individual data points reaches, that effects the decision boundary. In this work, it was set to "scale", which in *Scikit-learn*'s implementation sets the value of gamma to $1/(n\_features * X.var()$, where *n_features* is a number of features seen during the training of the model and X.var() is the data's variance.

The trained models were saved into files with a *pickle* library so that the knowledge system does not have to train them with every execution of the programme.

6.2.3 Knowledge system structure

The system's structure corresponds to the Pyke's system structure, meaning that it has one driver.py file for the central commands and then three types of files representing the knowledge bases. There are two rule bases, one for forward-chaining rules and the other for backward-chaining rules. The system's structure is as follows:

driver.py is the main file and represents the programme's logic.

compiled_krb is a directory containing compiled .kfb, .krb and .kqb files in a form of .fcb and .qbc pickle files and Python source files.

images directory contains images used by the system's GUI.

models is a directory containing the trained model chosen for SVM classification and the trained *TfidfVectorizer* needed for the pre-processing.

websites directory contains the websites datasets for the knowledge system evaluation.

ask_gui.py is a custom module for GUI containing implementation of a *ask_yn()* function, which is needed by the Pyke's engine.

facts.kfb file is a Knowledge Fact Base containing facts as input to the knowledge system.

fc_rule_base.krb is a Knowledge Rule Base containing forward-chaining rules for the inference engine.

bc_rule_base.krb is a second Knowledge Rule Base containing backward-chaining rules for the inference engine.

question_base.kqb is a Knowledge Question Base containing defined questions for the user.

svm_module.py is a file representing the first part of the programme, the SVM classification.

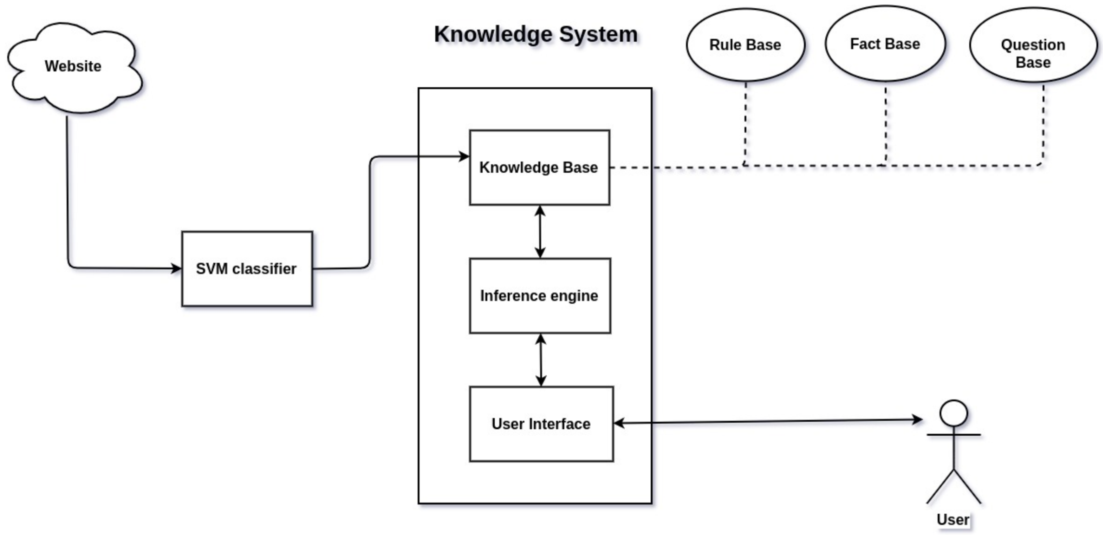


Figure 6.4 Knowledge system's structure

6.2.3.1 The main file

Driver.py is the main programme file. There is a total of three functions, *run()* being the main function of the programme. After running the main function, *run_fc_rules()* function is called, activating the rule base for forward-chaining which itself launches the inference mechanism for creating new facts.

When the forward chaining mechanism finishes, *run_bc_rules()* function is called from the main function activating rule base for the backward-chaining rule base and launching the engine's inference mechanism with a *prove_goal()* function. This function takes a statement which is desired to be proved as a parameter. It either yields an answer or throws an Exception if the goal cannot be proved.

6.2.3.2 Fact Base

There is a single .kfb file representing the knowledge fact base. The only universal fact is be "extremist_text(True)" or "extremist_text(False)", which is written after the SVM module predicts the value. Other case-specific facts are be added during the forward-chaining inference mechanism part, which are asserted with *assert* command followed by a statement that consists of the fact base name and the fact. For example, *facts.explicit_images(True)*.

6.2.3.3 Rule Base

As already mentioned, there are two rule bases.

The first one contains forwards-chaining rules. These have a simple "if-then" syntax, but instead of "if-then", keywords "foreach-assert" are used within the Pyke implementation. In the list of rules, Pyke searches the "if", respectively "foreach", clauses of already known facts and when the corresponding fact is found in the fact base, the "then", respectively "assert", clause is executed, adding a new fact or several facts into the fact base. The forward-chaining algorithm runs until there are no more new facts that can be added to the fact base.

An example of forward-chaining rules:

```
explicit_images
    foreach
        facts.images_present(True)
        question_base.are_there_explicit_images(True)
    assert
        facts.explicit_images(True)


non_explicit_images
    foreach
        facts.images_present(True)
        question_base.are_there_explicit_images(False)
    assert
        facts.explicit_images(False)
```

The second rule base contains backward-chaining rules. These utilize reversed approach of the "if-then" method – "then-if". The keywords are also different – "use-when". The algorithm first searches the clauses in the "then", respectively "use", clause, which matches the goal. Then it analyses the "if", respectively "when", clause, which may link to the "then", respectively "use", clause of another rule or directly to a fact. If all the facts are in the fact base, the goal can be proved.

An example of backward-chaining rules:

```
is_extremist_website
    use extremist_website(extremist)
    when
        facts.extremist_text(False)
        facts.very_extremist_images(True)
        facts.videos_present(False)
```

```
is_not_extremist_website
    use extremist_website(nonextremist)
    when
        facts.extremist_text(False)
        facts.non_extremist_images(True)
        facts.non_extremist_videos(True)
```

6.2.3.4 Question Base

Question base contains questions that will be posed to a user. Answers to these questions are remembered, so even if they appear in the rule base multiple times, user only answers them once. They are stored in .kqb files. For this work, only “yes/no” questions needed to be implemented. If a question looks like this:

```
are_there_images($ans)
    Are there any images on the website?
    ---
    $ans = yn
```

then the following statement can be used in the rule base to get information from a user, given that the question base is called “question_base”:

```
question_base.are_there_images(True)
```

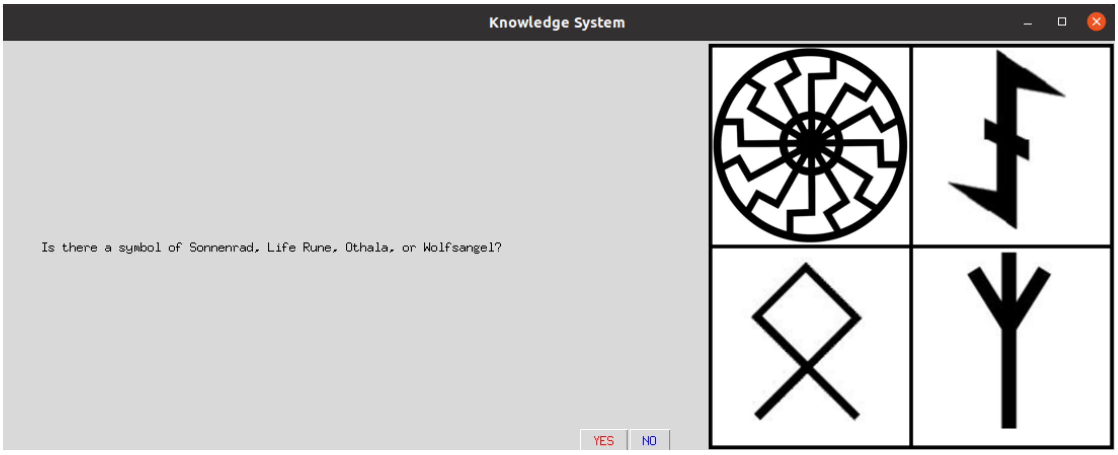
6.2.3.5 User Interface

Pyke provides two modules for a user interface, *ask_tty* and *ask_wx*. The first one is default and it writes the question on stdout and reads the user’s answer from stdin. The second one uses the *wxpython* library, which is more user-friendly with a dialogue box.

However, due to the need to display images so that the user could see the hate symbols, for example, and correctly identify attributes on the website, the provided modules’ implementation was not sufficient to meet these requirements. Therefore, I designed a new simple module with a more accessible interface, specifically *tkinter*. The challenge was to create the dialogue box and a way of connecting the questions from the question base and images corresponding to a particular question. This was accomplished by matching the question value to a long list of “if” rules, which would then trigger a function that launches up the dialogue window with the corresponding question text and image if there were any.

This module’s implementation does not cover all the types of questions and functionalities provided by the other modules as it only has an implementation of the “ask_yn” function. However, this thesis only operated with the “yes/no” questions, so for these purposes, it was sufficient.

After the programme starts, several dialogue windows pop up with each question. This is done by initializing the *tkinter.Tk()* main widget. It can be observed in figure 6.4 that there is a text of the question, which is a simple *tkinter.Label()* widget, two buttons, implemented with *tkinter.Button()* and *tkinter.Frame()* and sometimes an image on the right. The image is also implemented with *tkinter.Label()* and the image itself is loaded using the *PIL* library. The user only has to click on the buttons, after which the window closes and another one appears until there are no more questions.



Knowledge System

Is there a symbol of Sonnenrad, Life Rune, Othala, or Wolfsangel?

Figure 6.5 The knowledge system's GUI

6.3 Chapter Summary

This chapter describes the used technologies and explains the implementation details of the thesis. Firstly, a brief introduction to the Pyke shell was presented in order to explain the system's structure. Then, the most important libraries used in this work were outlined.

The implementation of data pre-processing task was then described, with all the functions used to accomplish it. The implementation and training of the SVM models was discussed. The training process included tuning the hyperparameters using cross-validation. These hyperparameters were C and γ; however, gamma was set to 'scale', so it was not actually tuned. The SVM classification runs in a separate module and its output is then written into a file for the knowledge system.

The knowledge system's structure was then presented. The programme's main file *driver.py* contains all the logic and starts up several processes. It activates the rule bases and, subsequently, the inference engine. There is also a fact base, two rule bases with forward-chaining and backward-chaining rules and a question base. The GUI was implemented in order to satisfy the system's requirements. It utilised the *Tkinter* library and its functionality was demonstrated in images.

Chapter 7

Experiments and results

This chapter presents and discusses the results of conducted experiments. Firstly, an overview of used evaluation methods is presented. Then, the experiments and their evaluation are presented. Two parts of the system were tested: the SVM models and the whole knowledge system. The SVM models were trained on data with and without sentiment analysis and evaluated for both cases.

7.1 Evaluation metrics

In this part, a brief overview of evaluation metrics that will be used to evaluate the experiments.

A *confusion matrix* is an N*N matrix, with N being the number of predicted classes. This thesis deals with binary classification; therefore, the matrix will be 2x2. It consists of four values: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). True positive and True Negative are numbers of data points classified correctly as positive and negative, respectively. False Positive is a Type I error in statistics when data points are incorrectly classified as positive. False Negative is a Type II error when data points are incorrectly classified as negative. From these values, the following metrics are computed:

Precision, or also a Positive Predictive Value, is a proportion of correct positive predictions that were correctly classified from all predicted positives.

$$P = \frac{TP}{TP + FP} \tag{7.1}$$

Recall, or also a Sensitivity, is a proportion of correct positive predictions that were correctly classified from all the actual positives.

$$R = \frac{TP}{TP + FN} \tag{7.2}$$

Specificity is a proportion of correct negative predictions that were correctly classified from all the actual negatives.

$$S = \frac{TN}{TN + FP} \tag{7.3}$$

Negative Predictive Value is a proportion of correct negative predictions that were correctly classified.

$$N = \frac{TN}{TN + FN} \tag{7.4}$$

Accuracy is a ratio of correct predictions by all the predictions.

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{7.5}$$

F1-score is a combination of precision and recall.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{7.6}$$

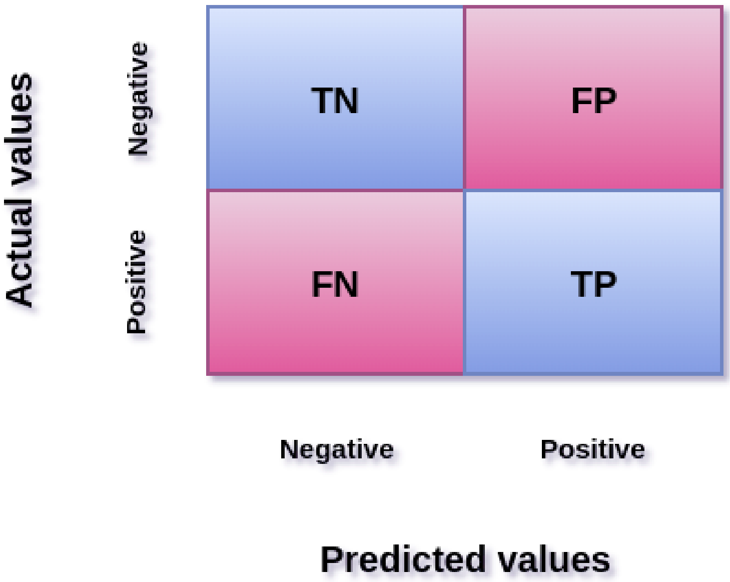


Figure 7.1 Confusion matrix for binary classification. Inspired by `https://miro.medium.com/max/1051/0*3bu6WmCk_gBdydQA.png`

7.2 Experiments

This part of the thesis offers experimental results of the applied models. The first is the evaluation of the SVM model and the second is the evaluation of the whole system and the classifier's performance with the website dataset.

7.2.1 SVM kernels

The first experiment tests the SVM models' performance and whether there is any difference between the efficiency of the models trained on data that does not consider sentiment analysis and those trained on data with sentiment analysis. Furthermore, the goal is to compare the performance of individual kernel functions. Data with sentiment analysis contain two more data columns, which are added to the training and testing data: "Subjectivity" and "Polarity". These indicate the sentiment of the text. The Support Vector Machines classifiers were trained on textual data described in section 5.1.

Three kernels were tested and tuned with cross-validation to find the optimal hyperparameters: linear, RBF and sigmoid kernels. The optimized hyperparameters were C, which was tested on values $\{0.1, 1, 10, 100\}$. In the RBF and sigmoid kernels, the parameter γ was set to $1/(n\_features * X.var()$, where *n_features* is a number of features seen during the training of the model and X.var() is the data's variance.

The results are evaluated using the following metrics: precision, recall, specificity, NPV, Accuracy and F1-score.

7.2.2 Knowledge System

The second experiment tests the whole system's performance and also the SVM classifier's performance on the website dataset. The SVM model with the best accuracy and F1-score is chosen to operate with the knowledge system. The hybrid knowledge system is evaluated with the help of a user, whose job is to examine the images of the created website dataset and answer corresponding questions about them. The metrics used for evaluating the system are precision, recall, specificity, NPV, Accuracy and F1-score.

7.3 Evaluation

The evaluation of data consisted of building a confusion matrix and, from its values, calculating the evaluation metrics. All of these were rounded to 4 decimal places.

The SVM model with linear kernel had precision, recall, specificity, NPV, accuracy and F1-score of approximately 99 %, both with and without sentiment analysis. The results differed slightly, with the linear kernel trained on data without sentiment analysis performing better on the third or fourth decimal. In figure 7.2, it can be observed that linear kernel with sentiment analysis was able to classify more extremist text correctly; however, it had a higher error rate. The linear kernel without sentiment analysis had a lower error rate and more correctly classified non-extremist text.

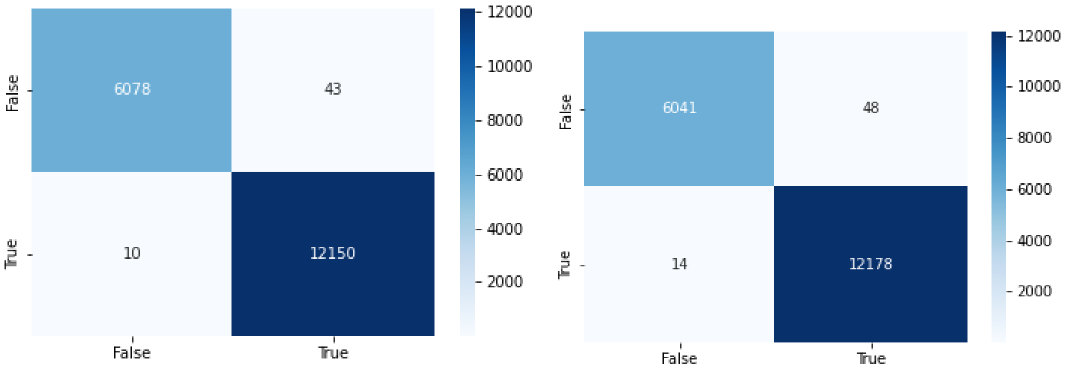
Models with the RBF kernel also had all of the metrics of value 99 % with and without sentiment analysis. It performed very similarly to the linear kernel in both cases. The RBF with sentiment analysis had a higher error rate with extremist text classified as non-extremist.

The sigmoid kernel without sentiment analysis performed well for all evaluation metrics, with 99 % each. However, the model with sentiment analysis performed noticeably more poorly, with the lowest NPV reaching 86 %, accuracy of 92 % and F1 score of 94 %. It can be seen in figure 7.4 that the model with SA significantly more misclassified extremist and non-extremist texts. It had a higher rate of false negatives than false positives, which is an exception to all the models.

The knowledge system was evaluated in the SVM module and the knowledge system classification. Based on the results from tables 7.1 and 7.2, the chosen SVM classifier was with the RBF kernel without sentiment analysis. Its precision was 100 %, meaning that the model did not classify any non-extremist text as extremist, and therefore, the specificity was also 100 %. The recall was 50 %, meaning that only half of the extremist text was classified as so. The classification accuracy was 75 % and the recall was 66 %.

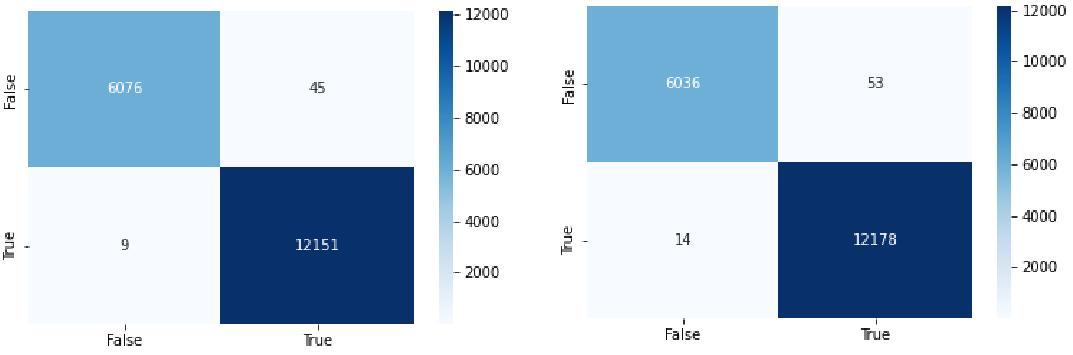
The knowledge system had similar results as the SVM module, with the precision and specificity of 100 %. The recall was slightly better with 60 %. The overall classification accuracy was 80 % and the F1 score was 75 %.

Looking at the appendix, we can see that the SVM model did not correctly classify any Neo Nazi text but did so with the Salafi text. It is also seen that although the text was misclassified, the knowledge system was able to classify it correctly by examining images and videos.



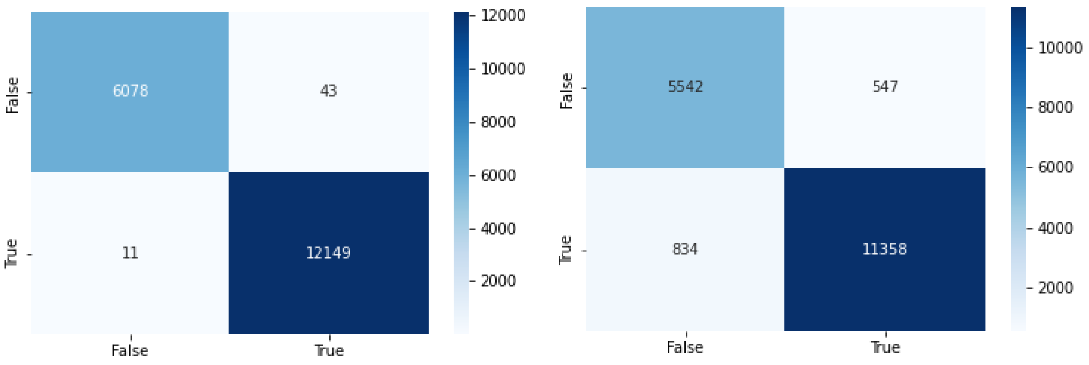
(a) Data without sentiment analysis **(b)** Data with sentiment analysis

Figure 7.2 Confusion matrices of SVM with linear kernel.



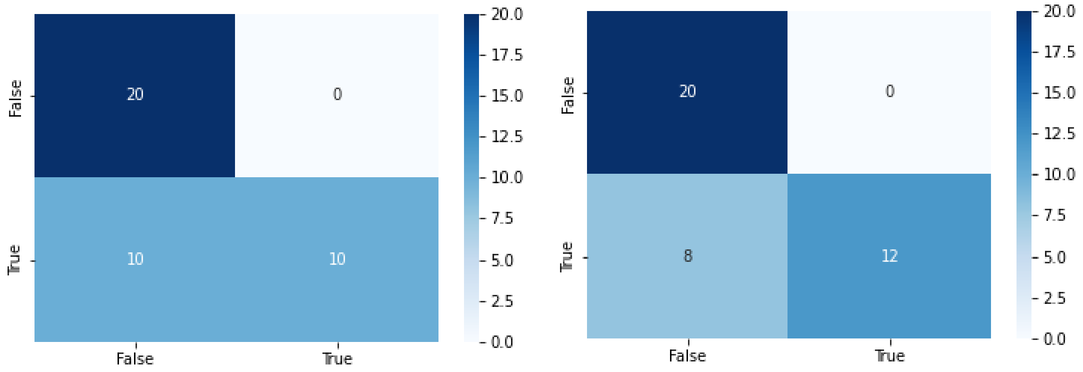
(a) Data without sentiment analysis **(b)** Data with sentiment analysis

Figure 7.3 Confusion matrices of SVM with RBF kernel.



(a) Data without sentiment analysis **(b)** Data with sentiment analysis

Figure 7.4 Confusion matrices of SVM with sigmoid kernel.



(a) SVM classifier

(b) Knowledge system

Figure 7.5 Confusion matrices of the knowledge system.

Kernel Function	C	Precision	Recall	Specifity	NPV	Accuracy	F1-score
Linear	1	0.9965	0.9992	0.9930	0.9984	0.9971	0.9978
RBF	10	0.9963	0.9993	0.9926	0.9985	0.9970	0.9978
Sigmoid	1	0.9965	0.9991	0.9930	0.9982	0.9970	0.9978

Table 7.1 Results of the SVM models without sentiment analysis.

Kernel Function	C	Precision	Recall	Specifity	NPV	Accuracy	F1-score
Linear	1	0.9961	0.9989	0.9921	0.9977	0.9966	0.9975
RBF	10	0.9957	0.9989	0.9913	0.9977	0.9963	0.9973
Sigmoid	0.1	0.9541	0.9316	0.9102	0.8692	0.9245	0.9427

Table 7.2 Results of the SVM models with sentiment analysis.

Part	Precision	Recall	Specifity	NPV	Accuracy	F1-score
SVM module	1.0	0.5	1.0	0.6667	0.75	0.6667
Knowledge system	1.0	0.6	1.0	0.7143	0.8	0.75

Table 7.3 Results of the knowledge system.

7.4 Discussion

Results presented in the previous part showed that the system's overall performance is very good, with an accuracy of 80 %. Although the SVM predicted the text on testing data well, it performed significantly worse on data from the website dataset. This could be due to the inconsistency in data or not enough representative examples of them. Data for the Nazi websites were taken from various datasets. Although labelled as extremist, they might have been of distinct nature and not contain keywords and expressions from the training and testing datasets. Neo Nazi website's text was wrongly classified for each of them. Therefore, the whole website was also often misclassified, as the knowledge system's rules give the SVM's classification a very high weight. However, it can be seen that even though some text was misclassified as non-extremist, in combinations with extremist images the website was classified as extremist. One way to improve the performance would be to collect a more representative and suitable dataset of extremist text

Website	Actual text	Predicted text	Actual website	Predicted website
nazi01	extremist	non extremist	extremist	non extremist
nazi02	extremist	non extremist	extremist	non extremist
nazi03	extremist	non extremist	extremist	non extremist
nazi04	extremist	non extremist	extremist	non extremist
nazi05	extremist	non extremist	extremist	non extremist
nazi06	extremist	non extremist	extremist	non extremist
nazi07	extremist	non extremist	extremist	extremist
nazi08	extremist	non extremist	extremist	extremist
nazi09	extremist	non extremist	extremist	non extremist
nazi10	extremist	non extremist	extremist	non extremist
nazi11	non extremist	non extremist	non extremist	non extremist
nazi12	non extremist	non extremist	non extremist	non extremist
nazi13	non extremist	non extremist	non extremist	non extremist
nazi14	non extremist	non extremist	non extremist	non extremist
nazi15	non extremist	non extremist	non extremist	non extremist
nazi16	non extremist	non extremist	non extremist	non extremist
nazi17	non extremist	non extremist	non extremist	non extremist
nazi18	non extremist	non extremist	non extremist	non extremist
nazi19	non extremist	non extremist	non extremist	non extremist
nazi20	non extremist	non extremist	non extremist	non extremist
salafi01	extremist	extremist	extremist	extremist
salafi02	extremist	extremist	extremist	extremist
salafi03	extremist	extremist	extremist	extremist
salafi04	extremist	extremist	extremist	extremist
salafi05	extremist	extremist	extremist	extremist
salafi06	extremist	extremist	extremist	extremist
salafi07	extremist	extremist	extremist	extremist
salafi08	extremist	extremist	extremist	extremist
salafi09	extremist	extremist	extremist	extremist
salafi10	extremist	extremist	extremist	extremist
salafi11	non extremist	non extremist	non extremist	non extremist
salafi12	non extremist	non extremist	non extremist	non extremist
salafi13	non extremist	non extremist	non extremist	non extremist
salafi14	non extremist	non extremist	non extremist	non extremist
salafi15	non extremist	non extremist	non extremist	non extremist
salafi16	non extremist	non extremist	non extremist	non extremist
salafi17	non extremist	non extremist	non extremist	non extremist
salafi18	non extremist	non extremist	non extremist	non extremist
salafi19	non extremist	non extremist	non extremist	non extremist
salafi20	non extremist	non extremist	non extremist	non extremist

Table 7.4 Tested websites and their results

documents. Another would be to modify the system's rules not to put so much weight on the SVM's decision.

Furthermore, the system showed a better classification of the Salafi websites; however, the text of these was generated from the directly extremist words, so it was easier for the system to detect only relevant tokens.

7.5 Chapter Summary

This chapter discussed and evaluated conducted experiments. Firstly, an overview of the evaluations methods built upon the confusion matrix was presented. These metrics were precision, recall, specificity, negative predictive values, accuracy and F1-score. Then, conducted experiments were described. There were two of them: the first one tested the SVM classifiers with and without sentiment analysis in the data. It was found that the sentiment analysis – calculating the “Subjectivity” and “Polarity” values - slightly worsened the models’ performance. Therefore, it was decided that the models trained on data without analysis were the better option; however, both approaches gave excellent results, mostly over 99 % for each of the metrics. The most effective classifier with this data was the one with the linear kernel and the ones without sentiment analysis performed similarly well.

The second experiment tested the performance of the whole hybrid knowledge system, with the best SVM model from the first one. That was chosen to be the RBF without the sentiment analysis. The overall system’s accuracy proved to be 80 %.

Chapter 8

Conclusion

This thesis's goal was to create a hybrid knowledge system for classifying websites that contain extremist topics. Firstly, a summary of frequently used terms was presented to establish a common understanding of them for the purposes of this work. Then, thorough literature research was conducted, starting with dividing the task of online extremism detection into two main categories – Link Based Bootstrapping and Text Classification. The latter is the approach implemented in this work; therefore, relevant articles regarding this approach and the topic of online extremism detection were explored. For the most part, the literature review focused on classifying text documents, but few works also explored image, video and website classification. It was evident that the SVMs were widely used in text classification and gave excellent results. However, the lack of available data appeared to be a significant problem in this field.

In the next part, theoretical models and principles used in the practical part were described. Natural Language Processing techniques, text representation methods, Support Vector Machines and knowledge systems were briefly explained.

The proposed solution was formed in the next part and consisted of determining classification rules for three attributes on websites: text, images and videos. Furthermore, classification rules for the whole website needed to be established for the knowledge system. Therefore, a proper analysis of the features of extremist websites had to be performed to gain knowledge. The analysis discovered phrases and words frequently used within the communities of Salafists and Neo-Nazis.

Subsequently, symbols and characteristic features in images and videos were uncovered from both extremist groups that usually serve as a part of their propaganda. Based on the knowledge from the literature, they were divided into four characteristic groups: explicit, militant, symbolic and soft. The explicit category contains scenes from attacks, executions, or disturbing scenes. The militant one includes war themes, such as weapons, military vehicles, army and soldiers. The symbolic category covers symbols and hate signs, and finally, the soft category deals with motives that are not considered extremist alone, however, with specific features, they can be considered extremist. This might be religious themes, families or children.

Three higher-level categories were established from combinations of these characteristic categories: very extremist, medium extremist and non-extremist. Furthermore, the classification rules for classifying whole websites were established from these categories.

Data used for the training and testing of the SVM models were downloaded from Kaggle[1]. These were mostly datasets of messages from Twitter and a representative amount of them was taken to evenly represent the extremist groups. A news article dataset was used for the non-extremist text documents. A new dataset representing extremist and non-extremist websites was

[1] https://www.kaggle.com

created for demonstration purposes and to test the knowledge system. It consisted of text files with extremist or non-extremist text and a set of images for each supposed website.

Then, the SVM classifier was trained on training data and tested on the testing ones. The classifiers were implemented with three kernel functions: linear, RBF and sigmoid. Their hyperparameters were tuned with cross-validation. Furthermore, this training was done twice: once with data without sentiment analysis and once with it, incorporating subjectivity and polarity values into the dataset. The classifier with the best results proved to be one with a linear kernel trained on data with sentiment analysis. From the classifiers trained on data without sentiment analysis, it was with an RBF kernel.

These classifiers were then incorporated into a Python programme as a separate module. Its output served as an input to the knowledge system, which was implemented in the Pyke knowledge engine shell. A graphical user interface was developed to incorporate all the system's requirements. The evaluation showed that the knowledge system gave a classification accuracy of 80 % for both classifiers.

The thesis's goal and subgoals are considered to be successfully fulfilled. There are many possibilities for improvements, for example, a more representative dataset of realistic websites or a more user-friendly GUI. However, I intend to further work on this topic within the Výzkumné léto na FIT (VýLeT) programme, where these improvements can be applied.

Bibliography

1. STATISTA. *Number of Monthly Active Facebook Users Worldwide as of 4th quarter 2021* [online]. 2015 [visited on 2022-04-15]. Available from: `https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/`.
2. PRESS, Oxford University. *Oxford Learner's Dictionaries* [online]. 2022 [visited on 2022-05-07]. Available from: `https://www.oxfordlearnersdictionaries.com/definition/english/extremism?q=extremism`.
3. FBI. *Terrorism* [online] [visited on 2022-05-07]. Available from: `https://www.fbi.gov/investigate/terrorism`.
4. BARBORA, Vegrichtová. *Hrozba radikalizace: terorismus, varovné signály a ochrana společnosti.* Grada Publishing, as, 2019.
5. MORRIS, Travis. Networking vehement frames: neo-Nazi and violent jihadi demagoguery. *Behavioral sciences of terrorism and political aggression*. 2014, vol. 6, no. 3, pp. 163–182.
6. CORREA, Denzil; SUREKA, Ashish. Solutions to detect and analyze online radicalization: a survey. *arXiv preprint arXiv:1301.4916*. 2013.
7. SUREKA, Ashish; KUMARAGURU, Ponnurangam; GOYAL, Atul; CHHABRA, Sidharth. Mining youtube to discover extremist videos, users and hidden communities. In: *Asia information retrieval symposium*. 2010, pp. 13–24.
8. SHARIF, Waqas; MUMTAZ, Shahzad; SHAFIQ, Zubair; RIAZ, Omer; ALI, Tenvir; HUSNAIN, Mujtaba; CHOI, Gyu Sang. An empirical approach for extreme behavior identification through tweets using machine learning. *Applied Sciences*. 2019, vol. 9, no. 18, p. 3723.
9. GAIKWAD, Mayur; AHIRRAO, Swati; PHANSALKAR, Shraddha; KOTECHA, Ketan. Online extremism detection: A systematic literature review with emphasis on datasets, classification techniques, validation methods, and tools. *IEEE Access*. 2021, vol. 9, pp. 48364–48404.
10. CHEN, Hsinchun. Sentiment and affect analysis of dark web forums: Measuring radicalization on the internet. In: *2008 IEEE International Conference on Intelligence and Security Informatics*. 2008, pp. 104–109.
11. JOHNSTON, Andrew H; WEISS, Gary M. Identifying sunni extremist propaganda with deep learning. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017, pp. 1–6.
12. ARAQUE, Oscar; IGLESIAS, Carlos A. An approach for radicalization detection based on emotion signals and semantic similarity. *IEEE Access*. 2020, vol. 8, pp. 17877–17891.

13. AHMAD, Shakeel; ASGHAR, Muhammad Zubair; ALOTAIBI, Fahad M; AWAN, Irfanullah. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Human-centric Computing and Information Sciences*. 2019, vol. 9, no. 1, pp. 1–23.

14. BERMINGHAM, Adam; CONWAY, Maura; MCINERNEY, Lisa; O'HARE, Neil; SMEATON, Alan F. Combining social network analysis and sentiment analysis to explore the potential for online radicalisation. In: *2009 International Conference on Advances in Social Network Analysis and Mining*. 2009, pp. 231–236.

15. HASHEMI, Mahdi; HALL, Margeret. Detecting and classifying online dark visual propaganda. *Image and Vision Computing*. 2019, vol. 89, pp. 95–105. ISSN 0262-8856. Available from DOI: `https://doi.org/10.1016/j.imavis.2019.06.001`.

16. LÓPEZ-SÁNCHEZ, Daniel; CORCHADO, Juan M.; GONZÁLEZ ARRIETA, Angélica. Dynamic Detection of Radical Profiles in Social Networks Using Image Feature Descriptors and a Case-Based Reasoning Methodology. In: COX, Michael T.; FUNK, Peter; BEGUM, Shahina (eds.). *Case-Based Reasoning Research and Development*. Cham: Springer International Publishing, 2018, pp. 219–232. ISBN 978-3-030-01081-2.

17. BATRA, Rishab; KHARE, Prabhat; JAIN, Somya. Determine Extremist Videos on Social Media. In: *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*. 2018, pp. 26–27.

18. BREZEALE, Darin; COOK, Diane J. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2008, vol. 38, no. 3, pp. 416–430.

19. ESTER, Martin; KRIEGEL, Hans-Peter; SCHUBERT, Matthias. Web site mining: a new way to spot competitors, customers and suppliers in the world wide web. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 249–258.

20. ALLEN, James F. Natural language processing. In: *Encyclopedia of computer science*. 2003, pp. 1218–1222.

21. MANNING, Christopher; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. Introduction to information retrieval. *Natural Language Engineering*. 2010, vol. 16, no. 1, pp. 100–103.

22. CHOWDHURY, Gobinda G. Natural language processing. *Annual Review of Information Science and Technology*. 2003, vol. 37, no. 1, pp. 51–89. Available from DOI: `https://doi.org/10.1002/aris.1440370103`.

23. MEDHAT, Walaa; HASSAN, Ahmed; KORASHY, Hoda. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*. 2014, vol. 5, no. 4, pp. 1093–1113.

24. YAN, Jun. Text Representation. In: *Encyclopedia of Database Systems*. Ed. by LIU, LING; ÖZSU, M. TAMER. Boston, MA: Springer US, 2009, pp. 3069–3072. ISBN 978-0-387-39940-9. Available from DOI: `10.1007/978-0-387-39940-9_420`.

25. CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. *Machine learning*. 1995, vol. 20, no. 3, pp. 273–297.

26. YU, Hwanjo; KIM, Sungchul. SVM Tutorial-Classification, Regression and Ranking. *Handbook of Natural computing*. 2012, vol. 1, pp. 479–506.

27. JAKKULA, Vikramaditya. Tutorial on support vector machine (svm). *School of EECS, Washington State University*. 2006, vol. 37, no. 2.5, p. 3.

28. TRIPATHI, KP. A review on knowledge-based expert system: concept and architecture. *IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications*. 2011, vol. 4, pp. 19–23.

29. COHEN, Katie; KAATI, Lisa. Digital Jihad Propaganda from the Islamic State. *Swedish Defence Research Agency (FOI)*. 2018.

30. INNERES, Ministerium für; NORDRHEIN-WESTFALEN, Kommunalesdes Landes. „Extremistischer Salafismus als Jugendkultur-Sprache, Symbole und Style “. *Geldern. jva druck+ medien.* 2015.

31. LEAGUE, Anti-Defamation. *Hate on Display Hate Symbols Database* [online]. 2022 [visited on 2022-04-19]. Available from: `https://www.adl.org/hate-symbols?cat_id%5B151%5D=151`.

32. KOCH, Ariel. Jihadi beheading videos and their non-Jihadi echoes. *Perspectives on Terrorism.* 2018, vol. 12, no. 3, pp. 24–34.

33. HERMESSI, Haithem. *Terrorism And Jihadism Speech Detection* [online] [visited on 2022-05-08]. Available from: `https://www.kaggle.com/datasets/haithemhermessi/terrorism-and-jihadist-speech-detection`.

34. TRIBE, Fifth. *Religious Texts Used By ISIS* [online] [visited on 2022-05-08]. Available from: `https://www.kaggle.com/datasets/fifthtribe/isis-religious-texts`.

35. TRIBE, Fifth. *How ISIS Uses Twitter* [online] [visited on 2022-05-08]. Available from: `https://www.kaggle.com/datasets/fifthtribe/how-isis-uses-twitter`.

36. ROSENBERG, Sarai. *Nazi Tweets* [online] [visited on 2022-05-08]. Available from: `https://www.kaggle.com/datasets/saraislet/nazi-tweets`.

37. THOMPSON, Andrew. *All the newt* [online] [visited on 2022-05-08]. Available from: `https://www.kaggle.com/datasets/snapcrack/all-the-news`.

38. FREDERIKSEN, Bruce. *Welcome to Pyke* [online]. 2008 [visited on 2022-04-28]. Available from: `http://pyke.sourceforge.net/index.html`.

39. PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand; GRISEL, Olivier; BLONDEL, Mathieu; PRETTENHOFER, Peter; WEISS, Ron; DUBOURG, Vincent, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research.* 2011, vol. 12, no. Oct, pp. 2825–2830.

40. LORIA, Steven. textblob Documentation. *Release 0.15.* 2018, vol. 2.

41. BIRD, Steven; KLEIN, Ewan; LOPER, Edward. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

42. LUNDH, Fredrik. An introduction to tkinter. *URL: www. pythonware. com/library/tkinter/introduction/index. htm.* 1999.

Contents of attached medium

```
README.md .................... a brief description of the content of the medium
├── src
│   ├── svm ...................... source codes of the SVM implementation
│   └── knowledge_system ......... source codes of the knowledge system implementation
└── text
    ├── thesis.pdf ............... text of the Bachelor’s thesis in PDF format
    └── thesis.zip ............... source code of the Bachelor’s thesis in LaTeX
```