applied sciences

MDPI

Article

Comparison of Modern Multilingual Text Embedding Techniques for Hate Speech Detection Task

Evaldas Vaičiukynas [1,*], Paulius Danėnas [2], Linas Ablonskis [1], Algirdas Šukys [1], Edgaras Dambrauskas [2], Voldemaras Žitkus [1], Rita Butkienė [1] and Rimantas Butleris [1,2]

[1] Department of Information Systems, Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; linas.ablonskis@ktu.lt (L.A.); algirdas.sukys@ktu.lt (A.Š.); voldemaras.zitkus@ktu.lt (V.Ž.); rita.butkiene@ktu.lt (R.B.); rimantas.butleris@ktu.lt (R.B.)

[2] Centre of Information Systems Design Technologies, Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; paulius.danenas@ktu.lt (P.D.); edgaras.dambrauskas@ktu.lt (E.D.)

* Correspondence: evaldas.vaiciukynas@ktu.lt

Abstract

Online hate speech and abusive language pose a growing challenge for content moderation, especially in multilingual settings and for low-resource languages such as Lithuanian. This paper investigates to what extent modern multilingual sentence embedding models can support accurate hate speech detection in Lithuanian, Russian, and English, and how their performance depends on downstream modeling choices and feature dimensionality. We introduce LtHate, a new Lithuanian hate speech corpus derived from news portals and social networks, and benchmark six modern multilingual encoders (gemma, qwen, bge, snow, jina, and e5) on LtHate, RuToxic, and EnSuperset using a unified Python pipeline. For each embedding type, we train both a one-class histogram-based anomaly detector (HBOS) and a two-class gradient-boosted tree ensemble (CatBoost), with and without Principal Component Analysis (PCA) compression to 32-dimensional feature vectors. Across all datasets, two-class supervised models consistently and substantially outperform one-class anomaly detection, with the best configurations achieving up to 78.8% accuracy (Kappa 0.58, AUC ROC 0.87) in Lithuanian (jina), 92.2% accuracy (Kappa 0.77, AUC ROC 0.97) in Russian (e5), and 76.9% accuracy (Kappa 0.48, AUC ROC 0.86) in English (e5). PCA compression deteriorates the discriminative power of CatBoost only slightly, with much more negative impact for the HBOS model. These results demonstrate how modern multilingual sentence embeddings combined with gradient-boosted decision trees provides robust machine learning solutions for multilingual hate speech detection applications.

Keywords: hate speech detection; anomaly detection; machine learning; sentence embeddings; dimensionality reduction

Academic Editors: Yolanda Blanco Fernández, José Carlos López Ardao and Alberto Gil Solla

Received: 18 April 2026
Revised: 12 May 2026
Accepted: 16 May 2026
Published: 20 May 2026

1. Introduction

The rapid proliferation of digital communication platforms has fundamentally transformed the way societies interact, debate, and share information. While social media, online news portals, and public forums have democratized access to public discourse, they have simultaneously become vectors for the dissemination of hate speech—language that attacks, demeans, or incites violence against individuals or groups on the basis of protected characteristics such as race, ethnicity, religion, gender, sexual orientation, or national origin [1,2]. The scale and urgency of this phenomenon are staggering: a 2023 global survey conducted by UNESCO and Ipsos across 16 countries found that 67% of Internet

users have personally encountered hate speech online, with the prevalence rising to 74% among users under the age of 35 [3]. Similarly, a European Union survey has reported that approximately 80% of respondents in the EU have encountered hate speech in online environments [4]. These figures underscore that online hate speech is not a marginal or isolated problem but a pervasive feature of the contemporary digital landscape that affects billions of users worldwide.

The societal consequences of unchecked online hate speech extend far beyond individual psychological harm. At the individual level, exposure to hateful content has been associated with increased anxiety, depression, social withdrawal, and, in extreme cases, suicidal ideation [4,5]. At the community level, persistent hate speech fosters social polarization, normalizes bigotry, and erodes the quality of public discourse [4]. More alarmingly, a growing body of empirical research has established direct links between the spread of online hate speech and real-world violence. Müller and Schwarz [6] demonstrated that anti-refugee hate speech on Facebook causally predicted violent crimes against refugees in German municipalities, with the effect disappearing during major platform outages. The role of social media in facilitating mass atrocities has been tragically illustrated in Myanmar, where Facebook was identified by a United Nations fact-finding mission as an instrument for those seeking to spread hate against the Rohingya Muslim minority, contributing to a campaign of ethnic cleansing that displaced over 700,000 people [7]. In light of such evidence, the need for scalable and reliable automated hate speech detection systems has become a pressing concern for governments, platform operators, and civil society alike.

The regulatory environment has evolved accordingly. The European Union's Digital Services Act (DSA), which became fully applicable in February 2024, imposes explicit obligations on large online platforms to address illegal content—including hate speech—through a combination of automated detection, human review, and transparent reporting mechanisms [8]. In January 2025, the European Commission integrated the revised Code of Conduct on countering illegal hate speech online into the DSA framework, further strengthening the requirements for proactive content moderation. These regulatory developments have intensified the demand for automated content moderation tools that are accurate across linguistic and cultural contexts, scalable to the volume of user-generated content (which exceeds millions of posts per minute on major platforms), and robust against adversarial evasion strategies [9].

From a natural language processing (NLP) perspective, hate speech detection is typically framed as a supervised text classification problem. The field has evolved rapidly from early approaches based on handcrafted features and classical machine learning algorithms—such as bag-of-words representations paired with support vector machines or logistic regression [10,11]—to deep learning architectures leveraging word embeddings [12,13] and, more recently, pre-trained transformer-based language models [14–16]. Fine-tuned variants of BERT [17] and its multilingual counterpart mBERT, as well as cross-lingual models such as XLM-RoBERTa [18], have achieved state-of-the-art performance on several English-language hate speech benchmarks and have demonstrated the ability to transfer detection capabilities across languages via shared multilingual representation spaces [19–21].

Despite these advances, several critical challenges remain. First, the overwhelming majority of hate speech detection research has concentrated on English-language data, leaving most of the world's approximately 7000 languages without adequate detection tools or annotated resources [1,2,9]. Recent surveys have catalogued over 60 publicly available hate speech training datasets, yet the vast majority are English-centric, and only a handful cover languages from Central and Eastern Europe, the Baltics, or other underrepresented linguistic communities [9,22]. For low-resource languages, the scarcity

of annotated corpora, the absence of language-specific NLP pre-processing tools, and the limited coverage of pre-trained models compound the difficulty of building effective detection systems [23,24]. Lithuanian is a prototypical example of such a low-resource scenario: despite the existence of active online communities and documented instances of online hate speech in the Lithuanian digital sphere, systematic studies on automated hate speech detection in Lithuanian have only recently begun to appear in the literature [25].

Second, modern multilingual sentence embedding models have emerged as a promising paradigm for cross-lingual and multilingual text classification tasks. Models such as Multilingual E5 [26], Jina Embeddings [27], Snowflake Arctic [28], and BGE-M3 [29] encode texts from dozens or hundreds of languages into a shared vector space, enabling downstream classifiers to operate on fixed-dimensional representations without requiring language-specific fine-tuning. However, systematic comparisons of these modern off-the-shelf embedding models for hate speech detection—particularly in multilingual settings that include low-resource languages—remain scarce. Most existing studies either focus on a single encoder, employ task-specific fine-tuning that obscures the contribution of the base representation, or evaluate only on well-resourced languages. This gap motivates a rigorous, controlled comparison of modern embedding techniques across diverse linguistic settings under a unified experimental protocol.

Third, practical deployment of hate speech detection systems in real-world content moderation pipelines imposes stringent constraints on computational efficiency, memory footprint, and inference latency. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), offer a straightforward mechanism for compressing high-dimensional embeddings while potentially preserving discriminative information. Understanding the trade-off between compression and detection performance is critical for designing systems that can operate at scale without prohibitive computational costs, yet this aspect has received limited systematic attention in the hate speech detection literature.

In this paper, we address these gaps by investigating the effectiveness of several recent multilingual text embedding models for hate speech detection in three languages: Lithuanian, Russian, and English. We focus on sentence-level vector representations produced by six modern multilingual encoders—Gemma, Qwen3, BGE-M3, Snowflake Arctic, Jina Embeddings (v3), and Multilingual E5 (large-instruct)—and evaluate them in combination with a two-class CatBoost supervised classifier and a one-class HBOS anomaly detection approach across three binary hate speech datasets, with and without PCA-based dimensionality reduction.

Our main scientific contributions are as follows:

- We devise a unified experimental framework for benchmarking multilingual sentence embeddings on multiple hate speech datasets.
- We prepare and release LtHate, a 14,687-comment Lithuanian hate speech corpus with subject category and severity annotations.
- We report a systematic comparison of six recent multilingual embedding models across Lithuanian, Russian and English hate speech corpora, with and without PCA-based compression.
- We provide practical recommendations for model and embedding selection under computational constraints in multilingual moderation systems.

2. Related Work

This section provides an overview of attempts at using multilingual text embeddings for hate speech detection in various languages.

2.1. Hate Speech and Offensive Language Detection

Hate speech detection has gained a lot of attention from the research community since core NLP techniques were established and applied. Schmidt & Wiegand [10] provided one of the first structured overviews of early automatic hate speech detection methods using NLP, covering classical machine learning approaches and highlighting early challenges and features (e.g., bag-of-words, lexicons). One of the most important milestones in the area of hate speech research was the introduction of a benchmark dataset and taxonomy by Davidson et al. in 2017 [11]. Initial results were obtained by training a suite of traditional classifiers, such as logistic regression, naïve Bayes, decision trees, random forests, and support vector machines, resulting in the best overall F1-score of 0.90, achieved by logistic regression, while revealing a persistent confusion between hate speech and offensive content. Twitter tweets were another source for the hate speech corpus [30], which led to introducing the first deep learning systems based on convolutional neural networks (CNNs) and multiple representations [12]. The best model used word2vec semantic embeddings and achieved an F1-score of 78.3% on a four-class problem, demonstrating that learned distributed representations substantially outperform surface-form features for hate speech classification. This was further confirmed in [13], by comparing CNN, LSTM, and FastText architectures against traditional TF-IDF and bag-of-words vector baselines on a benchmark of 16,000 annotated tweets. Their results showed that LSTM models, whose embeddings were subsequently used to train gradient-boosted classifiers, significantly outperformed state-of-the-art character and word n-gram methods. The introduction of transformer architecture [14] led to substantial improvements over prior sequence models as it relies exclusively on self-attention mechanisms to model dependencies between arbitrary positions in a sequence in constant time. The proposed architecture also enabled creating fine-tuned versions of the original model, which usually outperformed the original. Mozafari et al. [15] were among the first to systematically investigate BERT fine-tuning for hate speech classification in a peer-reviewed open-access journal using a BERT-Base with both Davidson [11] and Waseem and Hovy [30] datasets. By proposing a regularization-based reweighting mechanism applied during fine-tuning, their model substantially outperformed prior deep learning baselines and helped to mitigate systematic racial biases. Further, ref. [16] followed this direction with HateBERT, pretrained on a large-scale corpus of Reddit comments containing offensive, abusive, or hateful content. Comparative experiments across three English benchmarks for abusive language detection (OffensEval, AbusEval, and HatEval) showed that HateBERT consistently outperformed the corresponding general BERT model on each of them. The OffensEval dataset [31] has since been adopted as one of the de facto standards for evaluating offensive language, and was later extended to multilingual settings in OffensEval-2020 [19], which introduced parallel datasets in Arabic, Danish, Greek, and Turkish. This led to the evaluation of multilingual transformer architectures such as mBERT and XLM-RoBERTa. Across all languages, the dominant strategy among top-performing systems was fine-tuning XLM-R on the target-language training data, either alone or in combination with language-specific pre-trained models, confirming the model's cross-lingual generalization capability.

Beyond shared tasks, a growing body of work has explored the application of multilingual transformer models to heterogeneous, real-world multilingual detection settings. Ref. [21] addressed the HASOC 2020 challenge on hate speech and offensive content identification in English, German, and Hindi using a two-stage hierarchical classification architecture built on mBERT and XLM-R. Their system first identified if content is hateful or offensive versus non-offensive, then classified detected toxic content into hate speech, offensive language, or profanity, exploiting the shared multilingual representation space of the underlying encoder across all three languages simultaneously. The results demon-

strated that multilingual transformer encoders could successfully transfer information across typologically distinct languages within a single fine-tuned model. The authors additionally found that incorporating hashtag and emoji-aware tokenization improved performance on Twitter data, where non-verbal signals frequently modify or intensify the meaning of hateful text. The performance of multilingual transformer models also inspired research in cross-lingual transfer as a substitute for in-language annotation in settings where labeled hate speech data is scarce or entirely absent. Bigoulaeva et al. [23] evaluated cross-lingual transfer from high-resource source-language English to German, for which only limited labeled examples were available. Using bilingual word embeddings that aligned the representation spaces of the two languages, they demonstrated that zero-shot transfer from English to German was feasible and competitive with in-language supervised baselines when training data was minimal. They also showed that using the transferred model's predictions to generate pseudo-labels for unlabeled German data and then training on those labels was a substantial improvement over the zero-shot baseline, establishing a practical and low-cost pathway to extending detection to new languages without full annotation campaigns. However, a subsequent and more comprehensive study [24] identified structural and cultural divergence between source and target language hate speech conventions—including differences in target group definitions, slur conventions, and expression of implicit versus explicit hatred—as the primary bottleneck for effective zero-shot transfer, as well as proposing targeted data selection strategies to partially mitigate these effects. Nevertheless, further research also confirmed the capability of projecting knowledge from English to other languages with cross-lingual contextual embeddings, reducing dependence on language-specific labeled data and enabling transfer to low-resource settings [20], as well as outperforming general-purpose embeddings in cross-lingual classification scenarios using domain-specific multilingual hate speech embeddings [32]. Recurrent neural architectures on FastText embeddings were also adapted to multilingual hate speech detection tasks across the English, Italian, and German languages in [33]. Awal et al. [34] proposed HateMAML, a meta-learning framework to improve cross-lingual transfer of hate speech classifiers in low-resource languages by adapting pretrained language models to new languages with limited data. The adoption of cross-lingual transfer in low-resource scenarios has been extensively researched in the context of multiple languages, including the Arabic [35], Turkish [35], and Indian languages [36–38]. The Lithuanian language has only recently gained attention, with the latest study exploring transfer learning and transformer-based architectures on the newly created annotated corpora of 27358 user-generated comments [25]. Research results indicated that multilingual transformer models, like Multilingual BERT, LitLat BERT or Electra, can reach competitive accuracy and F1-scores.

2.2. Multilingual Transformer Models

Multilingual embeddings encode semantic information from words, sentences, paragraphs or documents across multiple languages in a shared vector space, allowing models to compare and transfer linguistic knowledge between languages. This is foundational to multilingual NLP tasks such as cross-lingual retrieval, semantic similarity, or machine translation. Early embedding techniques like word2vec [39] demonstrated the power of dense vector representations for capturing semantic relationships within a single language, as well as revealing the capability of learning semantic relationships and vector arithmetic analogies from the distributional statistics of text corpora. Cross-lingual and multilingual embeddings extend this approach by representing words from multiple languages in a common space where semantically equivalent words are close together regardless of the language. Early cross-lingual methods often used linear mappings trained with bilingual

dictionaries or parallel corpora to align monolingual embeddings. These mapping-based models were simple and computationally efficient, and they enabled cross-lingual lexical tasks such as bilingual lexicon induction and document classification. An early survey by Ruder et al. [40] provided a comprehensive taxonomy of cross-lingual word-embedding models, classifying techniques for word-, sentence- and document-level alignments, as well as optimization objectives. These techniques were later extended to multilingual settings, handling many languages in a unified space rather than just pairs. Other researchers moved beyond mapping approaches by developing unsupervised neural language models that jointly train on raw multilingual corpora and exploit structural similarities across languages to align representations in a common space [41,42]. This idea was also extended to capture semantics across a wide variety of languages simultaneously, thus learning rich multilingual representations.

The rise of pre-trained multilingual transformer models completely changed the landscape of the existing multilingual embedding techniques and shifted this domain toward large, open-source transformer-based models with broad language coverage and flexible training objectives. Models like multilingual BERT (mBERT) [17] and XLM-R [18] leverage large amounts of text data from dozens or hundreds of languages to learn contextual embeddings that generalize across languages. These models are trained with self-supervised objectives (e.g., masked language modeling) on multilingual corpora and often yield strong zero-shot transfer performance. LaBSE (Language-agnostic BERT Sentence Embedding) [43] extends multilingual transformer pre-training to sentence-level representations trained on parallel translation data, enabling semantically comparable sentence embeddings across more than 100 languages. Multilingual E5 models [44] extend the original E5 recipe by contrastive pre-training on billions of multilingual sentence pairs followed by supervised fine-tuning and instruction tuning, yielding strong retrieval and similarity performance across many languages; the core architecture remains encoder-focused with contrastive objectives, which makes them efficient and robust but sometimes less adaptable in long-context scenarios. The Qwen3-Embedding series [45] builds on the Qwen3 LLM backbone with a dense transformer architecture and multi-stage training, including synthetic weak supervision, supervised fine-tuning, and model merging. These models support 100+ languages, benefiting from instruction-aware embedding generation and strong cross-lingual capabilities. Jina-Embeddings (v3/v4) use a transformer foundation with task-specific LoRA adapters and Matryoshka Representation Learning to produce flexible, high-quality embeddings for retrieval, clustering, and long-context tasks.

Approaches combining or fusing representations from multiple pre-trained models indicate that embedding choice and combination strategy significantly affect hate speech detection performance, although fusion can yield only modest gains relative to its computational cost. Other studies consider multilingual and multimodal settings, where text is combined with images and cultural context to improve detection of hateful memes and visually grounded content.

In contrast to prior work that either fine-tunes a single multilingual encoder or constructs custom domain-specific embeddings, our study compares several modern off-the-shelf sentence embedding models (Gemma, Qwen3, BGE, Snowflake, Jina, and Multilingual-E5) across multiple languages, using a fixed downstream machine learning model and a standardized evaluation protocol.

3. Hate Speech Datasets

The diversity of hate speech datasets enables us to assess whether the same embedding methods and downstream machine learning models are effective across various languages and dataset sizes.

3.1. Lithuanian Corpora—LtHate

LtHate [46] is a new hate speech corpus for the Lithuanian language. It consists of public media comments taken from the Litis [47] corpus and other public media sources. Comments from the Litis corpus were sourced from two of the biggest Lithuanian online news portals from the years 2010 to 2014. Comments from other media sources spanning the years 2021 to 2024 were sourced from various social media platforms and Lithuanian news portals in the Lithuanian language. Full details of the design and annotation process are provided beside the corpus in [46].

The topical composition of the corpus was inspired by the methodology described in [48]. We have chosen six subjects of hate speech: (a) ethnicity, nationality and race; (b) gender and sexual orientation; (c) country and state; (d) political views; (e) religion; and (f) institutions. For each subject, category sets of neutral and loaded samples were collected. For each loaded sample we have noted a target of the hate and the level of the hate. The levels were 1 to 4, with 1 corresponding to expressions of contempt, 4 corresponding to outright incitement to violence against the target and 2–3 being in between. Since other corpora used in this research do not have the same structure, for the experiment described in this paper, we reduced the LtHate corpus to binary labels of neutral or hate speech only. The resulting corpus contains 7465 neutral and 7222 loaded comments in six categories of subjects (see Table 1) with a total of 14,687 comments, where the target class (hate speech) corresponds to 49.17% of the Lithuanian corpus.

Table 1. Distribution of LtHate corpus with respect to subject categories.

Subject Category	Neutral	Hate Speech	Total Comments
Ethnicity, nationality and race	2143	1796	3939
Gender and sexual orientation	177	865	1042
Country and state	433	463	896
Political views	1858	2516	4374
Religion	966	837	1803
Institutions	1888	745	2633

3.2. Russian Corpora—RuToxic

RuToxic is a publicly available Russian language dataset of 163,187 user comments annotated for toxicity, which has been introduced in offensive or toxic language detection research [49]. Target class (toxic/hateful comments) here comprise 19.25% of the dataset, and therefore some weak class imbalances exist. RuToxic provides a complementary perspective on Slavic-language hate speech and toxicity, allowing us to test whether multilingual embeddings can capture similar patterns across Lithuanian and Russian data.

3.3. English Corpora—EnSuperset

The English dataset EnSuperset aggregates many public hate speech and offensive language corpora into a unified binary classification benchmark [22], containing 360,493 comments. Texts originate from English social media and online platforms, with annotations indicating the presence or absence of hateful or offensive content. After harmonization of label schemes, we use a binary label 0/1 and retain only relevant textual fields (“text”, “labels”). The target class encompass 27.04% of the English dataset. EnSuperset is substantially larger than LtHate and RuToxic, providing a high-resource English setting against which multilingual embeddings can be evaluated.

4. Methodology

A machine learning pipeline in Python 3.13.5 was devised with text pre-processing, vectorization and training of models using a 10-fold stratified cross-validation (CV) strategy on various hate speech datasets, and is available as open-source code at https://github.com/evavaic/KTU-Misijos-HIPSTer (accessed on 16 February 2026). Optionally, dimensionality reduction with PCA is applied independently for each cross-validation training split.

4.1. Text Pre-Processing

All datasets investigated for the hate speech detection task are processed using an identical and shared pipeline implemented in Python. Each text comment is first passed through a *fix_punctuation* function that removes exclamation marks and

- Normalizes encoding using *ftfy* package [50];
- Removes hyperlinks;
- Collapses repeated punctuation marks while preserving limited emphasis;
- Standardizes spacing around punctuation and numbers;
- Replaces emojis with shortcode text using the *emoji* package.

Examples of hateful comments from LT, RU and EN datasets are provided in Table 2 for illustrative purposes of text pre-processing. The resulting cleaned texts are then processed with a text vectorization technique to obtain feature vectors suitable for machine learning model training and testing.

Table 2. Examples of the text pre-processing step with tone-down (removal of question marks and exclamation marks), fixing of punctuation and emojis (not shown in original comment) replacement with shortcode.

Corpus	Hate Speech Comment Example	Result of Text Pre-Processing
LtHate	asile tu vel lyti pakeitei? Kiek moki uz lyties pakeitima? Esi baisus transvestitas!... Kas minute prisistatai vis kitos lyties atstovu „„ar viskas gerai tau su galva? Amerikoje tokiu reikia.	asile tu vel lyti pakeitei. Kiek moki uz lyties pakeitima. Esi baisus transvestitas... Kas minute prisistatai vis kitos lyties atstovu, ar viskas gerai tau su galva. Amerikoje tokiu reikia. ::grinning_face_with_smiling_eyes::
RuToxic	жеребцы диванные!!! только те кто тяжелее газеты ничего не поднимал могут так вякать! поэтому ипохожи стали на «оно»а не на мужиков	::flushed_face:: жеребцы диванные. только те кто тяжелее газеты ничего не поднимал могут так вякать. поэтому ипохожи стали на «оно»а не на мужиков
EnSuperset	its like spoon feeding retarded LIBS sometimes with some of you ... just saying, be warned!! #MAGA #TRUMP2020	::rolling_on_the_floor_laughing:: its like spoon feeding retarded LIBS sometimes with some of you.. just saying, be warned. ::beer_mug:: ::smiling_face_with_smiling_eyes::

4.2. Sentence Embeddings

We compare six multilingual sentence embedding techniques for text vectorization (see Table 3). Embedding models are loaded via the *SentenceTransformers* package, with Jina embeddings requiring *trust_remote_code=True* setting. Texts are encoded in batches with configurable batch size, and embeddings are concatenated into a matrix with 768 dimensions for gemma and 1024 for the remaining vectorization techniques. Compact embedding minishlab/potion-multilingual-128M (https://huggingface.co/minishlab/potion-multilingual-128M, accessed on 16 January 2026) [51,52] of 256 dimensions was also considered in initial experiments, but due to having the lowest accuracy was later discarded.

Model google/embeddinggemma-300m (https://huggingface.co/google/embeddinggemma-300m, accessed on 16 January 2026) [53,54] is a 0.3 B parameter multilingual text-embedding

model. The composition of the dataset that the model is trained on is not disclosed; however, the model does well on the MMTEB [55] benchmark, which includes the Lithuanian language. The model uses RoPE positional encodings [56] and supports inputs of up to 2048 tokens. It has an output dimensionality of 768 with matryoshka [57] points at 512, 256 and 128 dimensions.

Table 3. Description of selected multilingual embedding models. Notes: Dims correspond to dimensionality of embedding; MRL corresponds to Matryoshka Representation Learning, which allows truncation to lower dimensionality for resulting original feature vectors.

Embedding	Dims	MRL	Model Size	Model Path (in huggingface.co Platform)
gemma	768	Yes	308 M	google/embeddinggemma-300m
qwen	1024	Yes	595 M	Qwen/Qwen3-Embedding-0.6B
bge	1024	No	569 M	BAAI/bge-m3
snow	1024	Yes	568 M	Snowflake/snowflake-arctic-embed-l-v2.0
jina	1024	Yes	572 M	jinaai/jina-embeddings-v3
e5	1024	No	560 M	intfloat/multilingual-e5-large-instruct

Model Qwen/Qwen3-Embedding-0.6B (https://huggingface.co/Qwen/Qwen3-Embedding-0.6B, accessed on 16 January 2026) [45] is a 0.6 B parameter multilingual text-embedding model. The model uses decoder-only architecture built on the Qwen3 dense backbone with 28 Transformer layers, pretrained on large, diverse multilingual corpora (web pages, code, and other text) to give broad language coverage. The Qwen3 model family covers 119 languages and dialects worldwide, including Lithuanian, among other Baltic and low-resource languages. The model outputs a 1024-dimensional embedding vector, but it can be reduced down to 32 using the matryoshka [57] technique.

Model BAAI/bge-m3 (https://huggingface.co/BAAI/bge-m3, accessed on 16 January 2026) [29,58] is a 0.56 B parameter text-embedding model specifically trained for retrieval tasks. It is trained and refined on datasets [59,60] that include most widespread European languages. The model supports inputs of up to 8194 tokens. It has an output dimensionality of 1024.

Model Snowflake/snowflake-arctic-embed-l-v2.0 (https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0, accessed on 16 January 2026) [28,61] is a 0.6 B parameter multilingual text-embedding model based on transformer encoder architecture and trained on the MIRACL dataset [62], which contains 18 languages. The model uses RoPE positional encodings [56] and supports inputs of up to 8194 tokens. It has an output dimensionality of 1024, with a single matryoshka [57] point at 256 dimensions.

Model jinaai/jina-embeddings-v3 (https://huggingface.co/jinaai/jina-embeddings-v3, accessed on 16 January 2026) [27,63] is a 0.6 B parameter multilingual text-embedding model internally combining a transformer-based encoder with 5 task-specific LoRA [64] adapters. The model is trained on 100 languages and fine-tuned on 30 languages. Interestingly, one of those 30 languages is Latvian, which is highly similar to Lithuanian. The model uses RoPE positional encodings [56] and supports inputs of up to 8194 tokens. It has an output dimensionality of 1024 with matryoshka [57] points at 32, 64, 128, 256, 512 and 768 dimensions.

Model intfloat/multilingual-e5-large-instruct (https://huggingface.co/intfloat/multilingual-e5-large-instruct, accessed on 16 January 2026) [26,65] is a 0.6 B parameter multilingual text-embedding model based on transformer encoder architecture with weights initialized from XLM-RoBERTa large [18,66], which is trained on 100 languages, including Lithuanian. The model was additionally trained on datasets from the public media and fine-tuned on curated high-quality datasets, including one used in [44]. The model supports inputs of up to 514 tokens. It has an output dimensionality of 1024.

4.3. Dimensionality Reduction

To analyze the trade-off between performance and compactness, we train models on both original embeddings and their compressed variants. Principal Component Analysis (PCA) [67] is a linear dimensionality reduction technique that finds a new set of orthogonal axes (principal components) that capture as much of the variance in the original feature space as possible in descending order of importance, thus providing a compressed projection of the original data.

By retaining only the first 32 principal components after PCA, we obtain a compact feature vector representation that concentrates most of the variance of the original embeddings. This reduces storage and computational costs for downstream machine learning models while also acting as a form of linear noise filtering: directions with very low variance, which often correspond to noise or redundant information, are discarded. However, PCA is linear transformation that maximizes the variance of computed original vectors, so its effect on semantic similarity and discriminative power needs to be experimentally measured.

PCA in experiments here is fitted only to the training data of each cross-validation split to avoid information leakage, and consequently the learned transformation is applied to compress both training and test embeddings within that CV split.

4.4. Detection Models

In experiments we consider two types of downstream models for the detection task:

1. One-class (1c) anomaly detection. The histogram-based outlier score (HBOS) [68] model from the *PyOD (Python Outlier Detection)* package, used as a one-class approach trained on the target class (hate speech) examples only. The HBOS method models each feature independently using a histogram, and due to its linear complexity is suitable for very large datasets, being significantly faster than many other multivariate outlier detection methods. The outlier score is estimated based on density corresponding to histogram bin each feature falls into, with lower density values indicating anomalous instances. Default parameters (n_bins = 10, alpha = 0.1) were used and the contamination parameter was not important since it has no influence on training and raw output scores do not depend on it. Instead, the output from the model was rescaled by dividing from 10,000 (for original feature vectors) or from 100 (for PCA-transformed feature vectors) to derive a score resembling class probability prediction.
2. Two-class (2c) supervised classification. A gradient boosting CatBoost classifier [69] with 500 maximum iterations, a learning rate of 0.05, depth 8, *LogLoss* loss function, and *scale_pos_weight* set to the ratio of negatives to positives in the training data (to address class imbalance). Within each cross-validation iteration, 80% of the training set is used to fit the model and 20% is held out for internal validation and early stopping, with a patience of 30 iterations where the best CatBoost model with respect to AUC ROC metric is retained.

Although the datasets had binary annotation of both target class (hate speech) and non-target class (neutral speech) examples, which is required to evaluate detection success on test folds, the difference between model variants used was in training steps where construction of the model used data from both classes (2c case) or only data from a target class (1c case). Such selection of methods allows a direct comparison of text-embedding quality under a strong two-class (2c) supervised and a weaker one-class (1c) pseudo-unsupervised setting. In practice this would correspond to the scope of annotation efforts where for the one-class case collection of only hate speech examples should be sufficient to create a detector. We intentionally train HBOS in a one-class configuration on hate speech samples only. Although in realistic moderation scenarios hate speech is the minority and is typically modeled as the anomalous class, here our goal is different: we probe whether

modern embeddings form a compact, coherent region for hate speech examples. This setup concentrates on the structure of hate speech representations and decouples evaluation from the highly heterogeneous and domain-dependent distribution of non-hate content.

These models were selected due to better detection accuracy results after comparing HBOS to isolation forest and CatBoost to random forest alternatives. Researchers [70] comparing various machine learning models on 165 publicly available classification problems have also found that gradient tree boosting tends to outperform bagging ensemble (random forest). Both selected models here help to compare and benchmark various multilingual embeddings under identical machine learning setups, where early stopping used in CatBoost helps to avoid overfitting for a supervised two-class case.

4.5. Evaluation Metrics

Machine learning experiment success was assessed using k-fold-stratified CV using 10 folds (k = 10) and stratified by target attribute, where the machine learning model is trained on all except one fold, with that one fold left out to test model inference. After pooling model outputs for all test folds and comparing them to ground-truth class labels, various accuracy metrics were calculated in a micro-average fashion.

To summarize detection performance, the following metrics were used:

- Area under the receiver operating characteristic curve (AUC ROC), which corresponds to the probability that a randomly chosen non-target class instance will have a smaller estimated target class probability than a randomly chosen target class instance [71]. In short, AUC summarizes the probability of correctly ranking a (neutral, hate speech) pair of text examples based on the detector's output and is directly related to the Wilcoxon Mann–Whitney U statistic.
- The precision-Recall curve (PRC) also allows calculating the area under the curve (AUC PRC) and in the case of large class imbalances the PRC is recommended over ROC [72] when choosing a better performing detector.
- Overall accuracy (Accuracy)—the best known evaluation metric, calculating the proportion of correct predictions to the ground-truth classes:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

  where TP is true positives (correct target class predictions), TN is true negatives (correct non-target class predictions), FP is false positives (incorrect predictions of target class examples), and FN is false negatives (incorrect predictions of non-target class examples). All these counts correspond to absolute frequences from the confusion matrix, obtained after applying the threshold to the model's prediction (output probability of target class).
- Kappa [73,74]—accuracy, corrected for class imbalance:

$$Kappa = \frac{p_0 - p_e}{1 - p_e} \quad (2)$$

  where p_e—the sum of the probabilities of the predictions agreeing with the ground truth by chance; p_0—overall accuracy of the model. According to the academic literature, the Kappa value of 0.21–0.40 indicates fair agreement and 0.41–0.60 moderate agreement. Higher values correspond to a substantial and almost perfect agreement result.

Plot-based AUC ROC and AUC PRC metrics are calculated using the model's raw outputs before thresholding. To calculate the remaining metrics, one needs to obtain a confusion matrix by using a threshold on the model's raw outputs to convert soft decision (class probability) to hard decision (class prediction). Since the ad hoc choice of 0.5 is usually

suboptimal, we have selected a more effective—equal error rate—operating point where the ROC curve intersects with the diagonal and class recall metrics become equal; i.e., specificity becomes approximately equal to sensitivity and, consequently, to overall accuracy.

Additionally, besides pooling inference results on all test folds to get final metrics, we have also recorded the AUC ROC result for each test fold so that with the help of statistical analysis [75] we could investigate if there are significant differences in detection performance between embeddings compared. Since splitting into folds was identical for all approaches tried on the same dataset, which corresponds to repeated measures setups, and 10 folds provide a rather small sample, the nonparametric Friedman's test with Nemenyi's post hoc comparison (significance level alpha = 0.05) was used in the Python package *Autorank* [76].

5. Experimental Results

In this section we outline the main findings organized by dataset. Detailed ROC and PRC curves and accuracy metrics tables—for original feature vectors and for feature vectors after PCA transformation—are presented here with an overview of the results.

5.1. Lithuanian Dataset Results

Machine learning for Lithuanian language hate speech dataset results are in Table 4 and Figures 1 and 2. One-class classification (see top part of Table 4) resulted in 53.07–63.22% accuracy for original and 53.55–59.55% accuracy for PCA-compressed embeddings. Two-class classification (see bottom part of Table 4) resulted in 73.34–78.80% accuracy for original and 70.39–76.65% accuracy for PCA-compressed embeddings. Two-class supervised classification clearly outperformed one-class anomaly detection, with slightly worse performance for PCA-compressed embeddings (drop range from e5 1.39% to qwen 1.68%).

Table 4. Summary of hate speech detection success for Lithuanian language dataset.

Method	Accuracy (%)		Kappa		AUC ROC		AUC PRC	
	Orig.	PCA	Orig.	PCA	Orig.	PCA	Orig.	PCA
1c gemma	54.34	53.55	0.087	0.071	0.561	0.551	0.544	0.528
1c qwen	53.07	53.38	0.061	0.068	0.543	0.547	0.527	0.527
1c bge	56.00	55.48	0.120	0.110	0.581	0.579	0.573	0.555
1c snow	57.90	56.79	0.158	0.136	0.611	0.595	0.587	0.565
1c jina	63.22	59.55	0.264	0.191	0.684	0.633	0.662	0.606
1c e5	59.73	56.64	0.194	0.133	0.639	0.594	0.622	0.571
2c gemma	73.34	71.00	0.467	0.420	0.811	0.786	0.805	0.778
2c qwen	73.07	70.39	0.461	0.408	0.818	0.778	0.816	0.770
2c bge	77.65	75.73	0.553	0.515	0.859	0.837	0.853	0.830
2c snow	77.12	75.50	0.542	0.510	0.854	0.835	0.847	0.831
2c jina	78.80	76.65	0.576	0.533	0.868	0.846	0.862	0.840
2c e5	77.40	76.01	0.548	0.520	0.858	0.837	0.854	0.832

Detection effectiveness for Lithuanian hate speech, as measured by ROC/PRC curves and complementary accuracy metrics, was highest for jina embeddings, resulting in AUC ROC of 0.868 and AUC PRC of 0.862 for two-class classification (see Figure 1), and PCA transformation reduced those areas under detection curves slightly (see Figure 2). Other very competitive embeddings in two-class cases were bge, snow and e5. However, both gemma and qwen embeddings demonstrated noticeably inferior results.

After comparing detection performances for two-class non-PCA setups we reject the null hypothesis (p-value = 0.000) of the Friedman's test and conclude that there is a

statistically significant difference between median AUC ROC values from 10 CV folds. Based on the post hoc Nemenyi test (see Figure 3) we assume that there are no significant differences within the following groups: gemma and qwen; qwen and snow; snow, e5, bge, and jina (the best group). All other differences are significant.

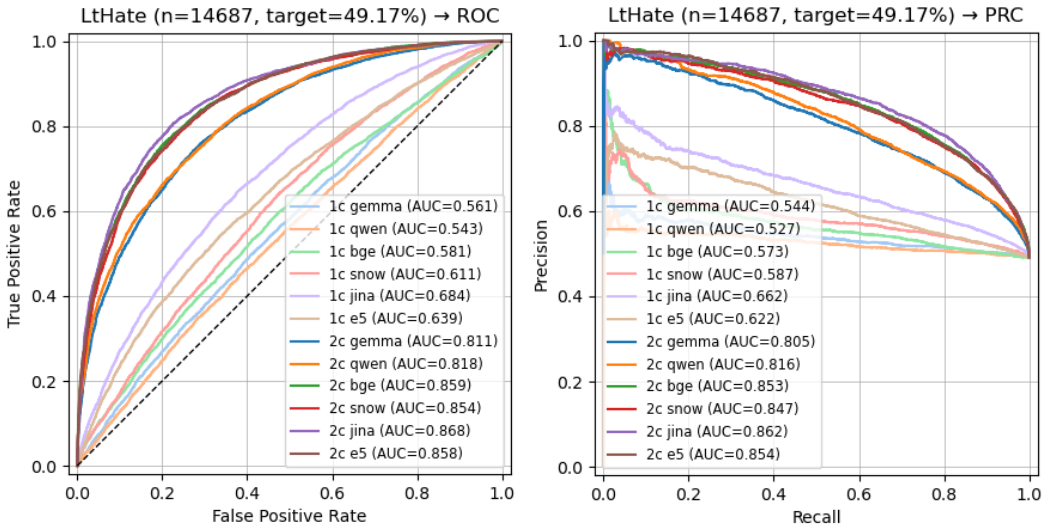

Figure 1. Lithuanian language hate speech detection curves using original embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 49.17% of the whole (n = 14,687) LT dataset. Best detection performance in one-class (1c) and two-class (2c) scenarios was achieved with jina.

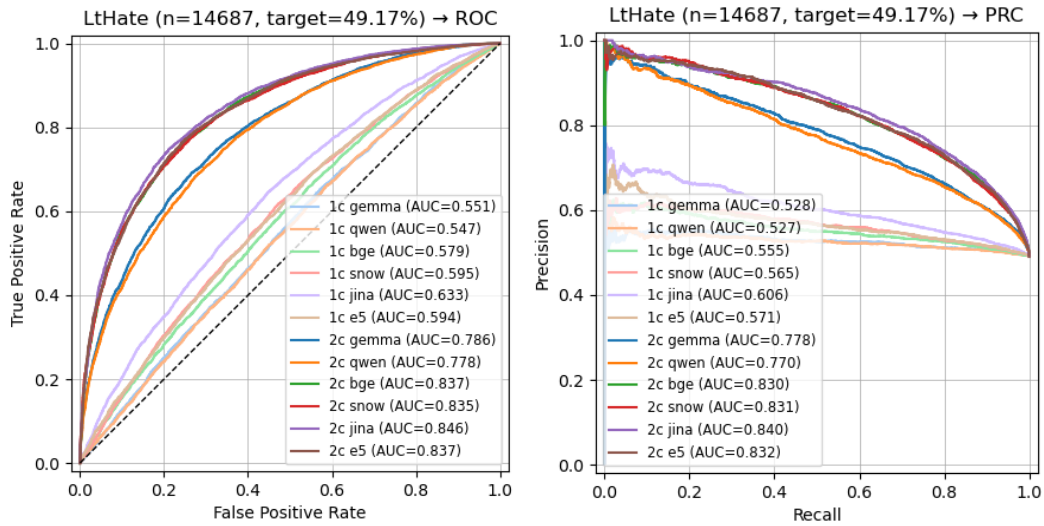

Figure 2. Lithuanian hate speech detection curves using compressed embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 49.17% of the whole (n = 14,687) LT dataset. Best detection performance in one-class (1c) and two-class (2c) scenarios was achieved with jina. PCA compression deteriorated one-class (1c) performance, noticeably reducing AUC values.

Overall, slightly above moderate agreement between predicted class and ground truth (best Kappa = 0.58 for jina embeddings) demonstrates average success in hate speech detection for the Lithuanian language dataset.

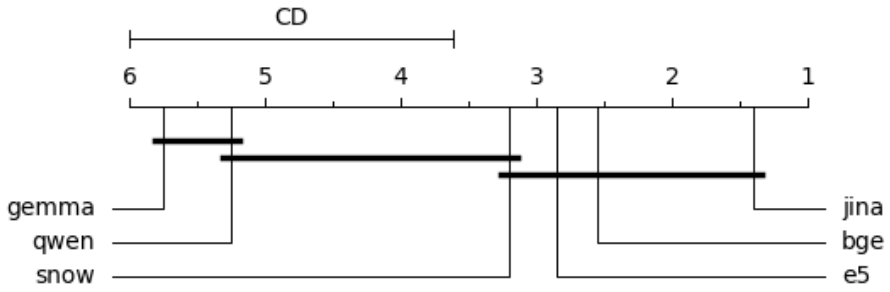

Figure 3. Lithuanian hate speech detection performance comparison for two-class non-PCA setup: critical difference (CD) diagram to visualize the ranking from the Nemenyi post hoc test, where horizontal lines indicate that differences in fold-wise AUC ROC values are not statistically significant.

5.2. Russian Dataset Results

Machine learning for Russian language hate speech dataset results are in Table 5 and Figures 4 and 5. One-class classification (see top part of Table 5) resulted in 67.09–80.85% accuracy for original and 64.09–75.71% accuracy for PCA-compressed embeddings. Two-class classification (see bottom part of Table 5) resulted in 89.34–92.20% accuracy for original and 85.22–91.09% accuracy for PCA-compressed embeddings. Two-class supervised classification clearly outperformed one-class anomaly detection with minor differences (drop range from jina 0.7% to qwen 2.37%) between original and PCA-compressed embeddings in two-class cases.

Table 5. Summary of hate speech detection success for Russian language dataset.

Method	Accuracy (%)		Kappa		AUC ROC		AUC PRC	
	Orig.	PCA	Orig.	PCA	Orig.	PCA	Orig.	PCA
1c gemma	78.99	69.60	0.462	0.286	0.869	0.764	0.666	0.444
1c qwen	67.09	64.09	0.244	0.196	0.734	0.692	0.406	0.324
1c bge	74.54	70.47	0.375	0.301	0.816	0.774	0.569	0.475
1c snow	69.13	70.56	0.278	0.303	0.767	0.773	0.468	0.469
1c jina	76.12	70.71	0.405	0.305	0.841	0.774	0.622	0.464
1c e5	80.85	75.71	0.500	0.397	0.890	0.836	0.712	0.583
2c gemma	89.34	88.21	0.696	0.668	0.961	0.954	0.885	0.864
2c qwen	87.59	85.22	0.653	0.597	0.950	0.931	0.855	0.804
2c bge	90.92	89.44	0.737	0.699	0.969	0.961	0.905	0.875
2c snow	91.07	90.08	0.741	0.715	0.971	0.966	0.912	0.893
2c jina	91.09	90.38	0.742	0.723	0.972	0.967	0.909	0.894
2c e5	92.20	91.09	0.771	0.741	0.978	0.971	0.931	0.911

Detection effectiveness for Russian hate speech, as measured by ROC/PRC curves and complementary accuracy metrics, was highest for e5 embeddings, resulting in AUC ROC of 0.978 and AUC PRC of 0.931 for two-class classification (see Figure 4), and PCA transformation did not affect it noticeably (see Figure 5). Other embeddings (jina, snow, bge) also were very competitive compared to e5 in two-class cases, with gemma embedding performing only slightly worse. The worst performance was for qwen embeddings, both for one-class and two-class cases.

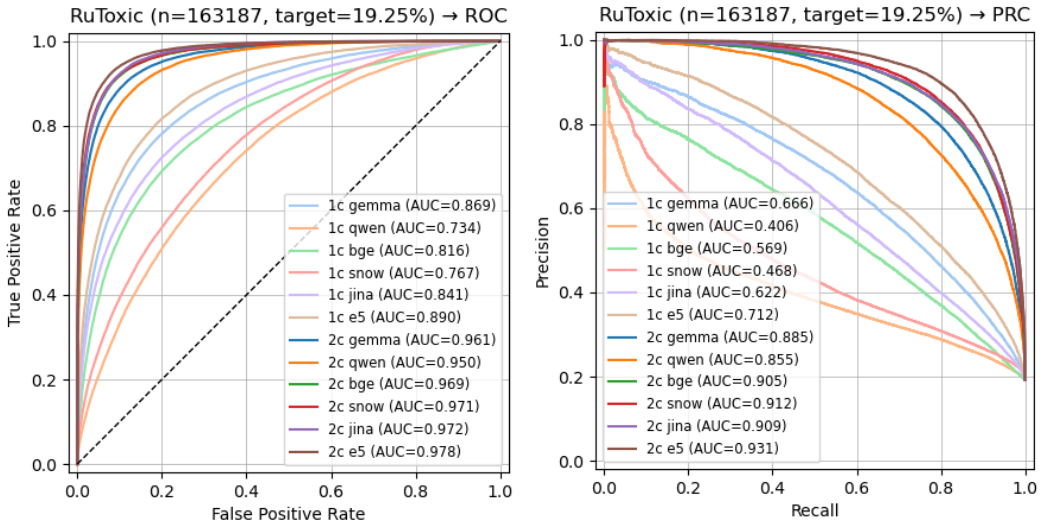

Figure 4. Russian hate speech detection curves using original embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 19.25% of the whole (n = 163,187) RU dataset. Best detection performance in one-class (1c) and two-class (2c) scenarios was achieved with e5 embeddings.

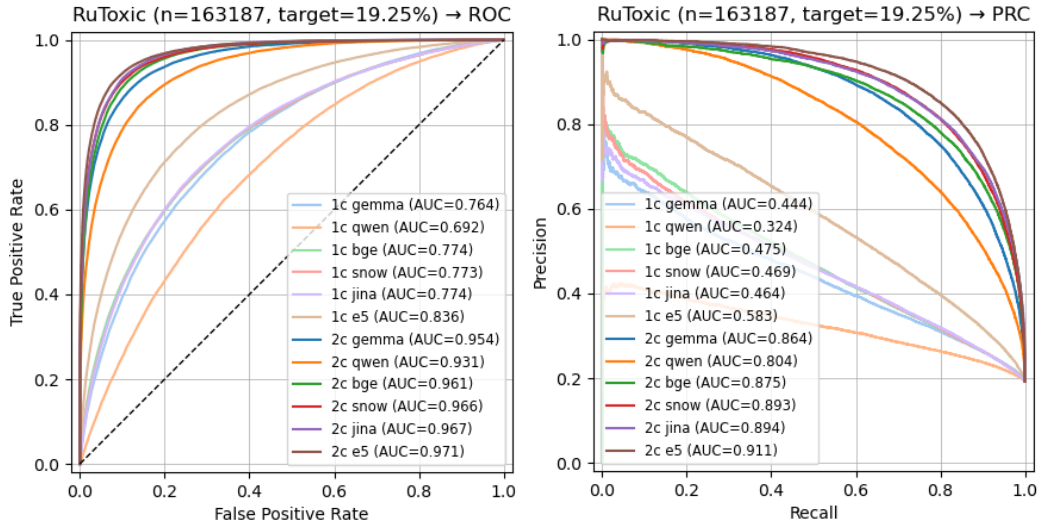

Figure 5. Russian language hate speech detection curves using compressed embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 19.25% of the whole (n = 163,187) RU dataset. Best detection performance in one-class (1c) and two-class (2c) scenarios was achieved with e5. PCA compression deteriorated one-class (1c) performance, noticeably reducing AUC values.

After comparing detection performance for the two-class non-PCA setup we reject the null hypothesis (p-value = 0.000) of the Friedman's test and conclude that there is a statistically significant difference between median AUC ROC values from 10 CV folds. Based on the post hoc Nemenyi test (see Figure 6), we assume that there are no significant differences within the following groups: qwen, gemma, and bge; gemma, bge, and snow; bge, snow, and jina; snow, jina, and e5 (the best group). All other differences are significant.

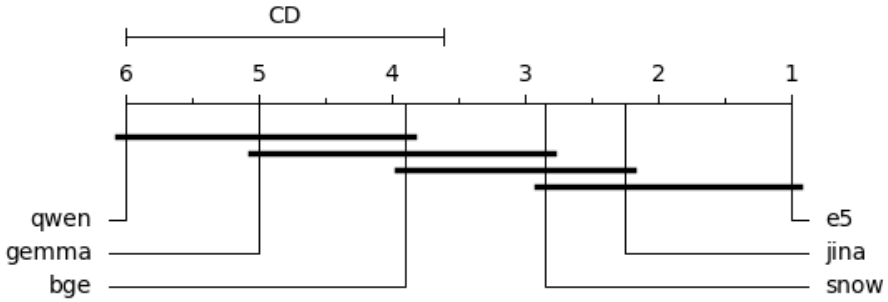

Figure 6. Russian hate speech detection performance comparison for two-class non-PCA setup: critical difference (CD) diagram to visualize the rankings from the Nemenyi post hoc test, where horizontal lines indicate that differences in fold-wise AUC ROC values are not statistically significant.

Substantial agreement between predicted class and ground truth (best Kappa = 0.77 for e5 embeddings) demonstrates high success levels in hate speech detection for the Russian language dataset.

5.3. English Dataset Results

Machine learning for English language hate speech dataset results are in Table 6 and Figures 7 and 8. One-class classification (see top part of Table 6) resulted in 60.07–66.70% accuracy for original and 50.94–56.19% accuracy for PCA-compressed embeddings. Two-class classification (see bottom part of Table 6) resulted in 73.71–76.89% accuracy for original and 72.45–76.72% accuracy for PCA-compressed embeddings (see bottom part of Table 6). Two-class supervised classification clearly outperformed one-class anomaly detection, with negligible differences (drop range from jina 0.17% to qwen 1.3%) between original and PCA-compressed embeddings in two-class case.

Table 6. Summary of hate speech detection success for English language dataset.

Method	Accuracy (%)		Kappa		AUC ROC		AUC PRC	
	Orig.	PCA	Orig.	PCA	Orig.	PCA	Orig.	PCA
1c gemma	66.70	56.19	0.284	0.100	0.731	0.586	0.484	0.342
1c qwen	60.07	53.57	0.166	0.057	0.648	0.552	0.391	0.310
1c bge	60.28	50.94	0.170	0.015	0.647	0.512	0.380	0.281
1c snow	60.99	54.33	0.182	0.070	0.663	0.564	0.417	0.322
1c jina	61.89	53.41	0.198	0.055	0.671	0.548	0.416	0.310
1c e5	64.22	55.58	0.239	0.090	0.702	0.577	0.464	0.322
2c gemma	76.35	76.03	0.468	0.462	0.848	0.844	0.679	0.669
2c qwen	73.71	72.45	0.416	0.391	0.819	0.804	0.633	0.608
2c bge	74.46	73.16	0.430	0.405	0.828	0.814	0.650	0.625
2c snow	75.91	75.53	0.459	0.451	0.842	0.838	0.668	0.657
2c jina	75.33	74.41	0.448	0.429	0.837	0.827	0.656	0.638
2c e5	76.89	76.72	0.479	0.475	0.855	0.854	0.698	0.697

Detection effectiveness for English hate speech, as measured by ROC/PRC curves and complementary accuracy metrics, was highest for e5 embeddings, resulting in AUC ROC of 0.855 and AUC PRC of 0.698 for two-class classification (see Figure 7), and PCA transformation surprisingly had almost no effect for this result (see Figure 8). Other very competitive embeddings in two-class cases were gemma, snow and jina, with notably good results for gemma embeddings.

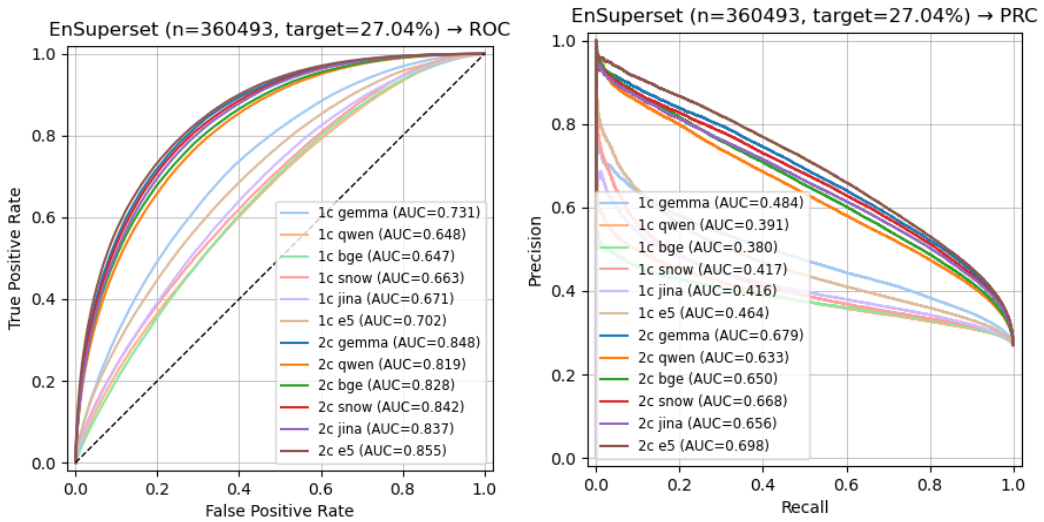

Figure 7. English language hate speech detection curves using original embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 27.04% of the whole (n = 360,493) EN dataset. Best detection performance in one-class (1c) scenarios was achieved with gemma. Best detection performance in two-class (2c) scenarios was achieved with e5.

After comparing detection performance for two-class non-PCA setups, we reject the null hypothesis (p-value = 0.000) of the Friedman's test and conclude that there is a statistically significant difference between median AUC ROC values from 10 CV folds. Based on the post hoc Nemenyi test (see Figure 9), we assume that there are no significant differences within the following groups: qwen, bge, and jina; bge, jina, and snow; jina, snow, and gemma; snow, gemma, and e5 (the best group). All other differences are significant.

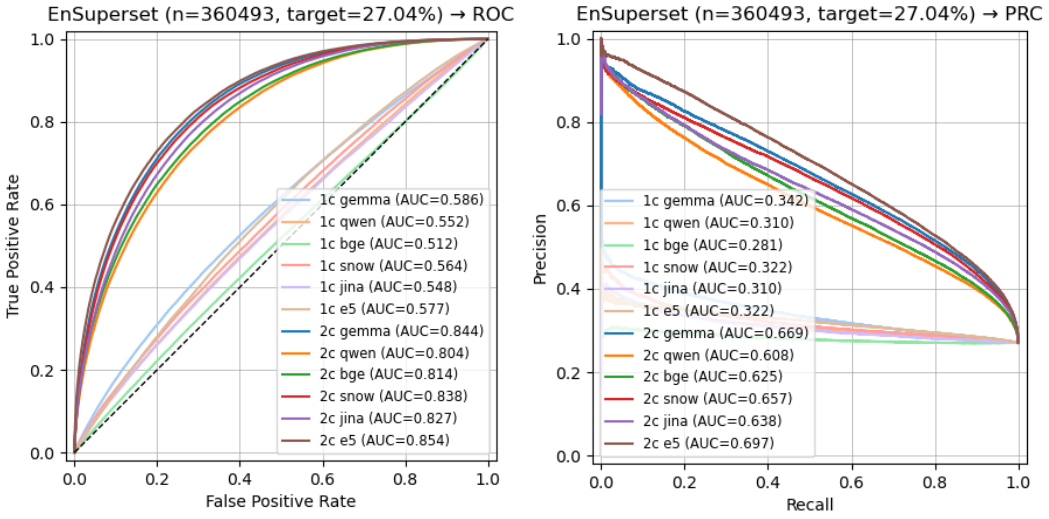

Figure 8. English language hate speech detection curves using compressed embeddings: ROC (**left**) and PRC (**right**). Hate speech comments constitute 27.04% of the whole (n = 360,493) EN dataset. Best detection performance in one-class (1c) scenarios was achieved with gemma. Best detection performance in two-class (2c) scenarios was achieved with e5. PCA compression deteriorated one-class (1c) performance, noticeably reducing AUC values.

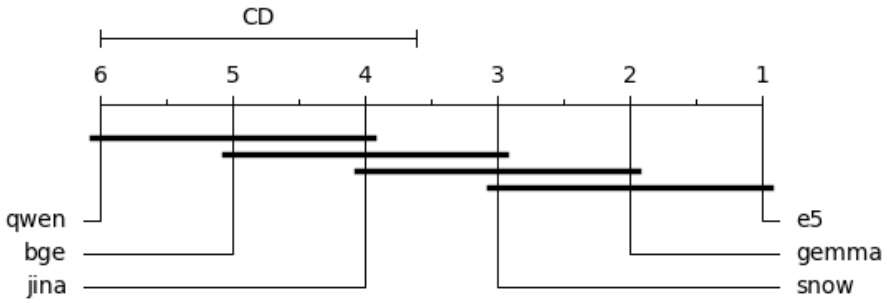

Figure 9. English hate speech detection performance comparison for two-class non-PCA setup: critical difference (CD) diagram to visualize the rankings from the Nemenyi post hoc test, where horizontal lines indicate that differences in fold-wise AUC ROC values are not statistically significant.

Moderate agreement between predicted class and ground truth (best Kappa = 0.48 for e5 embeddings) demonstrates below average success in hate speech detection for English language dataset.

5.4. Overview of All Results

Across the three hate speech datasets and six multilingual embedding models, several consistent patterns can be observed. First, two-class (2c) supervised CatBoost classifiers systematically and substantially outperform one-class (1c) HBOS anomaly detectors in terms of accuracy, Kappa, and AUC metrics for all languages and embedding families. For best results in each task on Lithuanian, LtHate accuracy for jina embeddings increased from 63.22% to 78.80%, while e5 embeddings accuracy on Russian RuToxic increased from 80.85% to 92.20% and on English EnSuperset from 66.70% (gemma) to 76.89% (e5). These results suggest that even if neutral (non-target class) examples can be time-consuming to annotate, a strong supervised approach should be preferred whenever a reasonably balanced labeled dataset can be constructed.

Second, detection effectiveness depends strongly on both the language and embeddings used. For Lithuanian LtHate, the best result was achieved with jina embeddings, but the other three embeddings (bge, e5, snow) seem to also be very competitive (with no statistically significant differences in fold-wise AUC ROC values). For Russian RuToxic and English EnSuperset the highest accuracy was achieved with e5 embeddings and other competetive embeddings were: jina and snow for Russian language; gemma and snow for English language. Overall, modern large multilingual encoder-based embeddings (jina, e5, snow, bge, gemma) consistently outperform the decoder-based qwen variant. Embedding models with excellent global multilingual benchmarks do not necessarily transfer uniformly across lower-resource target languages.

Third, simple linear PCA transformation, where the goal is to preserve the highest variance of the embeddings for the training data, seems not to deteriorate inherent semantic information and allows discrimination between classes rather successfully. Across all datasets and embedding models, accuracy values for original and PCA-compressed representations differ only marginally in the two-class setting, often by approximately up to two percentage points, but Kappa and AUC metrics appear to be effected more. However, in the one-class HBOS scenario, PCA compression can noticeably degrade performance, particularly for RuToxic and EnSuperset, indicating that fine-grained density information is more important for histogram-based anomaly scoring than for gradient-boosted decision trees.

Finally, there are clear differences in achievable performance across languages and datasets. Russian RuToxic, which is relatively large and has a moderate class imbalance, yields the highest scores (best Kappa = 0.77), suggesting that current multilingual

embeddings can model Russian toxic language patterns particularly well under a supervised two-class setup. Lithuanian LtHate attains lower but still competitive results (best Kappa = 0.58), reflecting both its smaller size and the increased difficulty of modeling a newly constructed low-resource language hate speech corpus. English EnSuperset yields intermediate performance (best Kappa = 0.48), which is slightly lower than might be expected for English but this may be explained by the heterogeneity of the source corpora.

6. Discussion and Conclusions

In this paper, we presented a comparative study of six modern multilingual sentence embedding models—gemma, qwen, bge, snow, jina, and e5—for hate speech detection in Lithuanian, Russian, and English. We introduced LtHate, a new Lithuanian hate speech corpus with detailed topical and severity annotations that we reduced to a binary classification setting for experiments, and we evaluated all embedding models in both one-class (HBOS) and two-class (CatBoost) configurations with and without PCA-based dimensionality reduction. Experimental results show that contemporary multilingual encoders combined with a popular gradient-boosted classifier can achieve moderate to substantial agreement with human annotations across all three languages, with the strongest performance observed for Russian and competitive results for the newly created Lithuanian dataset.

From a practical perspective, the experiments suggest several recommendations for multilingual hate speech detection systems. Whenever it is feasible to obtain labeled examples for both hateful and non-hateful categories, a two-class supervised setup with CatBoost (or similar gradient-boosting methods) should be preferred over purely one-class anomaly detection, as the latter consistently lags behind across datasets and metrics. Among the embedding models, jina appears to be the most suitable choice for Lithuanian, whereas e5 embeddings are most suitable for the Russian and English languages; other embeddings (snow, bge, gemma) remain competitive alternatives, often with non-significant differences in AUC ROC values, but this depends on the language. Additionally, our findings indicate that applying PCA to reduce embeddings yields only a marginal loss in CatBoost classification performance, offering a straightforward way to lower memory and computation costs in real-world deployments.

At the same time, several limitations of the current study point to directions for future work. First, we focused exclusively on the off-the-shelf sentence encoders without any task-specific fine-tuning, meaning that further gains are likely achievable via supervised or contrastive adaptation on in-domain hate speech corpora. Second, our experiments considered only binary hate vs. neutral labels, whereas LtHate and many existing datasets provide richer taxonomies (e.g., fine-grained target groups and severity levels), and modeling these distinctions may be necessary for more nuanced moderation decisions. Third, the current setup is text-only and does not incorporate multi-modal information such as images, emojis beyond textual normalization, or conversation context, which are often crucial in real-world hateful or abusive content.

Therefore, an important direction for further research would be systematic evaluation of instruction-tuned large language models in zero-shot and few-shot classification regimes, as well as hybrid architectures combining frozen multilingual embedding encoders with lightweight adapters fine-tuned on hate speech and toxicity detection tasks. Also, integrating explainability techniques and bias assessments into the evaluation protocol will be essential for understanding and mitigating potential harms when deploying multilingual hate speech detectors in high-stakes, real-world moderation scenarios.

Author Contributions: Conceptualization, E.V. and P.D.; methodology, E.V.; software, E.V.; validation, L.A. and P.D.; formal analysis, L.A.; investigation, A.Š., E.D. and V.Ž.; resources, R.B. (Rimantas Butleris); data curation, A.Š., E.D., V.Ž. and R.B. (Rita Butkienė); writing—original

draft preparation, E.V., P.D. and L.A.; writing—review and editing, E.V. and P.D.; visualization, E.V.; supervision, R.B. (Rimantas Butleris) and R.B. (Rita Butkienė); project administration, R.B. (Rimantas Butleris) and R.B. (Rita Butkienė); funding acquisition, R.B. (Rita Butkienė). All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted as part of the execution of the project "Mission-driven Implementation of Science and Innovation Programs" (No. 02-002-P-0001), funded by the Economic Revitalization and Resilience Enhancement Plan "New Generation Lithuania".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All code used to implement the unified Python pipeline for preprocessing, multilingual sentence embeddings, PCA compression, and model training is publicly available at https://github.com/evavaic/KTU-Misijos-HIPSTer (accessed on 16 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area under the curve
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional neural network
CV	Cross-validation
DSA	Digital Services Act
HBOS	Histogram-based outlier score
LaBSE	Language-agnostic BERT Sentence Embedding
LLM	Large language model
LoRA	Low-rank adaptation
mBERT	Multilingual BERT
NLP	Natural language processing
PCA	Principal Component Analysis
PRC	Precision-Recall curve
PyOD	Python Outlier Detection
ROC	Receiver operating characteristic
TF-IDF	Term frequency–inverse document frequency
XLM-R	Cross-lingual language model–RoBERTa

References

1. Fortuna, P.; Nunes, S. A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.* **2018**, *51*, 85. [CrossRef]
2. Poletto, F.; Basile, V.; Sanguinetti, M.; Bosco, C.; Patti, V. Resources and benchmark corpora for hate speech detection: A systematic review. *Lang. Resour. Eval.* **2021**, *55*, 477–523. [CrossRef]
3. UNESCO; IPSOS. Survey on the Impact of Online Disinformation and Hate Speech. 2023. Available online: https://www.unesco.org/sites/default/files/medias/fichiers/2023/11/unesco_ipsos_survey.pdf (accessed on 16 January 2026).
4. Paz, M.A.; Montero-Díaz, J.; Moreno-Delgado, A. Hate Speech: A Systematized Review. *Sage Open* **2023**, *13*, 1–18. . [CrossRef]
5. Hawdon, J.; Oksanen, A.; Räsänen, P. Exposure to Online Hate in Four Nations: A Cross-National Consideration. *Deviant Behav.* **2017**, *38*, 254–266. [CrossRef]
6. Müller, K.; Schwarz, C. Fanning the Flames of Hate: Social Media and Hate Crime. *J. Eur. Econ. Assoc.* **2021**, *19*, 2131–2167. [CrossRef]
7. United Nations Human Rights Council. Report of the Independent International Fact-Finding Mission on Myanmar. Available online: https://www.ohchr.org/en/hr-bodies/hrc/myanmar-ffm/index (accessed on 16 January 2026).

8. European Parliament and Council of the European Union. Regulation (EU) 2022/2065 of the European Parliament and of the Council on a Single Market for Digital Services and amending Directive 2000/31/EC (Digital Services Act). *Off. J. Eur. Union* **2022**, *L 277*, 1–102. Available online: https://eur-lex.europa.eu/eli/reg/2022/2065/oj (accessed on 16 January 2026).
9. Vidgen, B.; Derczynski, L. Directions in Abusive Language Training Data, a Systematic Review: Garbage In, Garbage Out. *PLoS ONE* **2020**, *15*, e0243300. [CrossRef]
10. Schmidt, A.; Wiegand, M. A Survey on Hate Speech Detection using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, SocialNLP@EACL 2017, Valencia, Spain, 3 April 2017*; Ku, L., Li, C., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 1–10. [CrossRef]
11. Davidson, T.; Warmsley, D.; Macy, M.W.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, QC, Canada, 15–18 May 2017*; AAAI Press: Menlo Park, CA, USA, 2017; pp. 512–515.
12. Gambäck, B.; Sikdar, U.K. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, 4 August 2017*; Waseem, Z., Chung, W.H.K., Hovy, D., Tetreault, J.R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 85–90. [CrossRef]
13. Badjatiya, P.; Gupta, S.; Gupta, M.; Varma, V. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017*; Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E., Eds.; ACM: New York, NY, USA, 2017; pp. 759–760. [CrossRef]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. Mozafari, M.; Farahbakhsh, R.; Crespi, N. Hate speech detection and racial bias mitigation in social media based on BERT model. *PLoS ONE* **2020**, *15*, e0237861. [CrossRef]
16. Caselli, T.; Basile, V.; Mitrović, J.; Granitzer, M. HateBERT: Retraining BERT for Abusive Language Detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 17–25. [CrossRef]
17. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805. [CrossRef]
18. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv* **2019**, arXiv:1911.02116. [CrossRef]
19. Zampieri, M.; Nakov, P.; Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Mubarak, H.; Derczynski, L.; Pitenis, Z.; Çöltekin, Ç. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Online, 12–13 December 2020; pp. 1425–1447. [CrossRef]
20. Ranasinghe, T.; Zampieri, M. Multilingual Offensive Language Identification with Cross-lingual Embeddings. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5838–5844. [CrossRef]
21. Roy, S.G.; Narayan, U.; Raha, T.; Abid, Z.; Varma, V. Leveraging Multilingual Transformers for Hate Speech Detection. In Proceedings of the Working Notes of FIRE 2020—Forum for Information Retrieval Evaluation, Hyderabad, India, 16–20 December 2020; Volume 2826, pp. 128–138.
22. Tonneau, M.; Liu, D.; Fraiberger, S.; Schroeder, R.; Hale, S.A.; Röttger, P. From Languages to Geographies: Towards Evaluating Cultural Bias in Hate Speech Datasets. In Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024), Mexico City, Mexico, 20 June 2024; pp. 283–311. [CrossRef]
23. Bigoulaeva, I.; Hangya, V.; Fraser, A. Cross-Lingual Transfer Learning for Hate Speech Detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, LT-EDI@EACL 2021, Online, 19 April 2021*; Chakravarthi, B.R., McCrae, J.P., Zarrouk, M., Bali, R.K., Buitelaar, P., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 15–25.
24. Bigoulaeva, I.; Hangya, V.; Gurevych, I.; Fraser, A. Addressing the Challenges of Cross-Lingual Hate Speech Detection. *arXiv* **2022**, arXiv:2201.05922. [CrossRef]
25. Mandravickaitė, J.; Rimkienė, E.; Petkevičius, M.; Songailaitė, M.; Zaranka, E.; Krilavičius, T. Exploring Hate Speech Detection Models for Lithuanian Language. In Proceedings of the 9th Workshop on Online Abuse and Harms (WOAH), Vienna, Austria, 31 July–1 August 2025; pp. 206–218.
26. Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; Wei, F. Multilingual E5 Text Embeddings: A Technical Report. *arXiv* **2024**, arXiv:2402.05672. [CrossRef]

27. Sturua, S.; Mohr, I.; Akram, M.K.; Günther, M.; Wang, B.; Krimmel, M.; Wang, F.; Mastrapas, G.; Koukounas, A.; Wang, N.; et al. jina-embeddings-v3: Multilingual Embeddings with Task LoRA. *arXiv* **2024**, arXiv:2409.10173. [CrossRef]
28. Yu, P.; Merrick, L.; Nuti, G.; Campos, D. Arctic-Embed 2.0: Multilingual Retrieval Without Compromise. *arXiv* **2024**, arXiv:2412.04506. [CrossRef]
29. Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; Liu, Z. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv* **2025**, arXiv:2402.03216. [CrossRef]
30. Waseem, Z.; Hovy, D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego CA, USA, 12–17 June 2016*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 88–93. [CrossRef]
31. Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; Kumar, R. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, 6–7 June 2019*; May, J., Shutova, E., Herbelot, A., Zhu, X., Apidianaki, M., Mohammad, S.M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 75–86. [CrossRef]
32. Monnar, A.A.; Perez Rojas, J.; Labra, B.P. Cross-lingual hate speech detection using domain-specific word embeddings. *PLoS ONE* **2024**, *19*, e0306521. [CrossRef]
33. Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A Multilingual Evaluation for Online Hate Speech Detection. *ACM Trans. Internet Technol. (TOIT)* **2020**, *20*, 10. [CrossRef]
34. Awal, M.R.; Lee, R.K.W.; Tanwar, E.; Garg, T.; Chakraborty, T. Model-Agnostic Meta-Learning for Multilingual Hate Speech Detection. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 1086–1095. [CrossRef]
35. Singhal, K.; Bedi, J. Transformers at HSD-2Lang 2024: Hate Speech Detection in Arabic and Turkish Tweets Using BERT Based Architectures. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, CASE 2024, St. Julians, Malta, 22 March 2024*; Hürriyetoglu, A., Tanev, H., Thapa, S., Uludogan, G., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 190–194.
36. Singh, A.; Thakur, R. Generalizable Multilingual Hate Speech Detection on Low Resource Indian Languages using Fair Selection in Federated Learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, 16–21 June 2024*; Duh, K., Gómez-Adorno, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 7211–7221. [CrossRef]
37. Ghosh, S.; Senapati, S.K. Hate speech detection in low-resourced Indian languages: An analysis of transformer-based monolingual and multilingual models with cross-lingual experiments. In *Natural Language Engineering*; Cambridge University Press: Cambridge, UK, 2024. . [CrossRef]
38. Chavinda, K.; Thayasivam, U. A Dual Contrastive Learning Framework for Enhanced Hate Speech Detection in Low-Resource Languages. In Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), Abu Dhabi, United Arab Emirates, 19 January 2025; pp. 115–123.
39. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *arXiv* **2013**, arXiv:1310.4546. [CrossRef]
40. Ruder, S.; Vulic, I.; Søgaard, A. A Survey of Cross-lingual Word Embedding Models. *J. Artif. Intell. Res.* **2017**, *65*, 569–630. [CrossRef]
41. Kanayama, H.; Cohn, T.; Ma, T.; Bird, S.; Duong, L. Multilingual Training of Crosslingual Word Embeddings. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017. [CrossRef]
42. Chen, X.; Cardie, C. Unsupervised Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 261–270. [CrossRef]
43. Feng, F.; Yang, Y.; Cer, D.; Arivazhagan, N.; Wang, W. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 878–891. [CrossRef]
44. Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; Wei, F. Improving Text Embeddings with Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 11897–11916. [CrossRef]
45. Zhang, Y.; Li, M.; Long, D.; Zhang, X.; Lin, H.; Yang, B.; Xie, P.; Yang, A.; Liu, D.; Lin, J.; et al. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv* **2025**, arXiv:2506.05176. [CrossRef]
46. Butkienė, R.; Edgaras, D.; Algirdas, Š.; Voldemaras, Ž. Lithuanian Hate Speech Corpus v.1. CLARIN-LT Digital Library in the Republic of Lithuania. Available online: http://hdl.handle.net/20.500.11821/69 (accessed on 6 May 2026).

47. Amilevičius, D.; Petkevičius, M. LITIS v.1. CLARIN-LT Digital Library in the Republic of Lithuania. Available online: http://hdl.handle.net/20.500.11821/11 (accessed on 16 January 2026).
48. Sanguinetti, M.; Poletto, F.; Bosco, C.; Patti, V.; Stranisci, M. An Italian Twitter Corpus of Hate Speech against Immigrants. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
49. Dementieva, D.; Moskovskiy, D.; Logacheva, V.; Dale, D.; Kozlova, O.; Semenov, N.; Panchenko, A. Methods for Detoxification of Texts for the Russian Language. *Multimodal Technol. Interact.* **2021**, *5*, 54. [CrossRef]
50. Speer, R. rspeer/python-ftfy: V6.3.1. 2024. Available online: https://zenodo.org/records/13994393 (accessed on 16 January 2026).
51. Tulkens, S.; van Dongen, T. Model2Vec: Fast State-of-the-Art Static Embeddings. Available online: https://github.com/MinishLab/model2vec (accessed on 16 January 2026).
52. minishlab/potion-multilingual-128M at Hugging Face. Available online: https://huggingface.co/minishlab/potion-multilingual-128M (accessed on 16 January 2026).
53. Vera, H.S.; Dua, S.; Zhang, B.; Salz, D.; Mullins, R.; Panyam, S.R.; Smoot, S.; Naim, I.; Zou, J.; Chen, F.; et al. EmbeddingGemma: Powerful and Lightweight Text Representations. *arXiv* **2025**, arXiv:2509.20354. [CrossRef]
54. google/embeddinggemma-300m at Hugging Face. Available online: https://huggingface.co/google/embeddinggemma-300m (accessed on 16 January 2026).
55. Enevoldsen, K.; Chung, I.; Kerboua, I.; Kardos, M.; Mathur, A.; Stap, D.; Gala, J.; Siblini, W.; Krzemiński, D.; Winata, G.I.; et al. MMTEB: Massive Multilingual Text Embedding Benchmark. *arXiv* **2025**, arXiv:2502.13595. [CrossRef]
56. Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **2024**, *568*, 127063. [CrossRef]
57. Kusupati, A.; Bhatt, G.; Rege, A.; Wallingford, M.; Sinha, A.; Ramanujan, V.; Howard-Snyder, W.; Chen, K.; Kakade, S.; Jain, P.; et al. Matryoshka Representation Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: New York, NY, USA, 2022; Volume 35, pp. 30233–30249.
58. BAAI/bge-m3 at Hugging Face. Available online: https://huggingface.co/BAAI/bge-m3 (accessed on 16 January 2026).
59. Shitao/MLDR Datasets at Hugging Face. Available online: https://huggingface.co/datasets/Shitao/MLDR (accessed on 16 January 2026).
60. Shitao/bge-m3-data Datasets at Hugging Face. Available online: https://huggingface.co/datasets/Shitao/bge-m3-data (accessed on 16 January 2026).
61. Snowflake/snowflake-arctic-embed-l-v2.0 at Hugging Face. Available online: https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0 (accessed on 16 January 2026).
62. Zhang, X.; Thakur, N.; Ogundepo, O.; Kamalloo, E.; Alfonso-Hermelo, D.; Li, X.; Liu, Q.; Rezagholizadeh, M.; Lin, J. MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 1114–1131. [CrossRef]
63. jinaai/jina-embeddings-v3 at Hugging Face. Available online: https://huggingface.co/jinaai/jina-embeddings-v3 (accessed on 16 January 2026).
64. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685. [CrossRef]
65. intfloat/multilingual-e5-large-instruct at Hugging Face. Available online: https://huggingface.co/intfloat/multilingual-e5-large-instruct (accessed on 16 January 2026).
66. FacebookAI/xlm-roberta-large at Hugging Face. Available online: https://huggingface.co/FacebookAI/xlm-roberta-large (accessed on 16 January 2026).
67. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417–441. [CrossRef]
68. Goldstein, M.; Dengel, A. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In Proceedings of the KI-2012: Poster Demo Track, Saarbrücken, Germany, 24 September 2012; Volume 1, pp. 59–63.
69. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. In *Proceedings of the Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
70. Olson, R.S.; Cava, W.L.; Mustahsan, Z.; Varik, A.; Moore, J.H. Data-driven advice for applying machine learning to bioinformatics problems. *Biocomputing* **2018**, *23*, 192–203. [CrossRef]
71. Huang, J.; Ling, C. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310. [CrossRef]
72. Beger, A. Precision-Recall Curves. *SSRN Electron. J.* **2016**. [CrossRef]
73. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]

74. McHugh, M.L. Interrater reliability: The kappa statistic. *Biochem. Med.* **2012**, *22*, 276–282. [CrossRef]
75. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
76. Herbold, S. Autorank: A Python package for automated ranking of classifiers. *J. Open Source Softw.* **2020**, *5*, 2173. [CrossRef]