Detecting Manipulation in Ukrainian Telegram: A Transformer-Based Approach to Technique Classification and Span Identification

Md. Abdur Rahman, Md Ashiqur Rahman
Department of Computer Science and Engineering
Southeast University, Bangladesh
{2021200000025, ashiqur.rahman}@seu.edu.bd

Abstract

The Russia-Ukraine war has transformed social media into a critical battleground for information warfare, making the detection of manipulation techniques in online content an urgent security concern. This work presents our system developed for the UNLP 2025 Shared Tasks, which addresses both manipulation technique classification and span identification in Ukrainian Telegram posts. In this paper, we have explored several machine learning approaches (LR, SVC, GB, NB) , deep learning architectures (CNN, LSTM, BiLSTM, GRU hybrid) and state-of-the-art multilingual transformers (mDeBERTa, InfoXLM, mBERT, XLM-RoBERTa). Our experiments showed that fine-tuning transformer models for the specific tasks significantly improved their performance, with XLM-RoBERTa large delivering the best results by securing 3rd place in technique classification task with a Macro F1 score of 0.4551 and 2nd place in span identification task with a span F1 score of 0.6045. These results demonstrate that large pre-trained multilingual models effectively detect subtle manipulation tactics in Slavic languages, advancing the development of tools to combat online manipulation in political contexts.

1 Introduction

The war between Russia and Ukraine highlights the critical importance of developing reliable mechanisms to identify misinformation on social media platforms. Among these platforms, Telegram stands out as particularly significant, becoming a breeding ground for channels that spread misleading information, Russian-favorable perspectives, and complete falsehoods targeting Ukrainian users. Contemporary Russian information warfare strategies deliberately foster confusion, fracture public consensus, undermine institutional credibility, and construct distorted perceptions of reality (Paul and Matthews, 2016). AI applications continue their expansion across various fields, gaining particular traction in information literacy—specifically addressing the detection and counteraction of disinformation phenomena that thrive within social media environments (Shu et al., 2020). The nuanced variety of manipulation techniques employed, spanning from emotion-laden rhetoric to intricate logical fallacies, creates substantial obstacles for natural language processing (NLP) systems.

With the urgent need to counter online manipulation, the Fourth Ukrainian NLP Workshop (UNLP 2025)[1] convened a shared task devoted to this very issue. Drawing on a Ukrainian and Russian Telegram corpus supplied by Texty.org.ua, participating teams developed and evaluated AI approaches with direct applications in both cybersecurity and disinformation research. The competition was structured around two complementary objectives: first, assigning each text to one of ten manipulation techniques, and second, precisely marking the character spans where manipulative tactics appeared.

Meeting these objectives requires models capable of detecting both overt cues and the more nuanced, context-dependent signals of manipulation. Although earlier work on propaganda and related detection tasks has laid important groundwork (Da San Martino et al., 2019; Yoosuf and Yang, 2019; Firoj et al., 2022; Solopova et al., 2024), our task's focus on Ukrainian and Russian social media and its insistence on joint span identification and fine-grained technique classification offers a novel contribution that pushes the frontier of disinformation analysis.

This paper presents our approach for the UNLP 2025 shared tasks. We test and evaluate several methods, ranging from conventional machine learning techniques to advanced deep learning and transformer models. Our key contributions include:

[1] https://github.com/unlp-workshop/unlp-2025-shared-task

- Developed transformer-based models to classify manipulation techniques and identify manipulative text spans in the dataset.
- Investigated thorough experiments with various machine learning approaches, deep learning architectures, and pre-trained transformer-based models, followed by extensive performance analysis and error examination.

2 Related Works

Despite the growing importance of defending messaging platforms against information-based attacks, most security and disinformation research remains concentrated on Twitter (Gilani et al., 2017) and Reddit (Saeed et al., 2022), while encrypted and semi-encrypted services such as Telegram, Signal, and WhatsApp have seen far less scrutiny. In sentiment analysis, Aljedaani et al. (2022) proposed an ensemble architecture that stacks LSTM and GRU layers sequentially, achieving 0.97 accuracy and a 0.96 Macro F1 score on TextBlob-labeled airline reviews. Similarly, Gandhi et al. (2021) compared CNN and LSTM models—both using word2vec embeddings—on the IMDB movie review dataset, finding that the LSTM outperformed the CNN with 88.02% accuracy. Beyond sentiment tasks, Inamdar et al. (2023) addressed mental-health detection on Reddit by combining ELMo embeddings with logistic regression and SVM classifiers, yielding a 0.76 Macro F1 score when identifying stress-related content. To tackle offensive content in code-mixed text, Ravikiran and Annamalai (2021) introduced the DOSA dataset for Tamil–English span identification; multilingual DistilBERT topped their benchmarks with a 0.405 Macro F1. In academic writing, Eguchi and Kyle (2023) presented a Dual-RoBERTa model that locates epistemic-stance spans, achieving a 0.7209 Macro F1. Finally, Papay et al. (2020) conducted a broad evaluation of span-identification methods on the CoNLL'00 chunking task, showing that their hybrid BERT+Feat+LSTM+CRF model reaches a micro-averaged F1 of 96.6%.

In war-related content analysis, Park et al. (2022) examined subtle manipulation tactics in Russian media coverage of the Ukraine war using their VoynaSlov dataset. Their XLM-R frame classifier achieved 67.5% Macro-F1 on in-domain MFC data but dropped to 33.5% on VoynaSlov, revealing challenges in real-world applications. Solopova et al. (2023) compared a Transformer (BERT) and an SVM with handcrafted features for multilingual pro-Kremlin propaganda detection on newspaper and Telegram corpora, achieving F1 scores of 0.92 and 0.88 respectively; Bezliudnyi et al. (2023) trained a BERT-based classifier on a custom Twitter and Telegram database to distinguish pro-Ukrainian, pro-Russian, and neutral texts, yielding 95% training and 83% test accuracy as part of a real-time analytics tool. Ustyianovych and Barbosa (2024) released the TRWU Telegram news dataset and applied an XGBoost classifier for multi-task attitude, sentiment, and discrimination detection, reaching an AUC of 0.9065; Burovova and Romanyshyn (2024) evaluated transformer-based models for binary dehumanization detection in Russian Telegram posts, with SpERT achieving an F1 of 0.85. In related span detection work, Thanh et al. (2021) created the UIT-ViSD4SA dataset and developed a BiLSTM-CRF model with fused embeddings that reached 62.76% Macro F1 score for Vietnamese sentiment analysis spans. Despite these advances, none of these studies combine fine-grained manipulation technique classification with precise span identification in Ukrainian or Russian Telegram content—the exact gap our UNLP 2025 shared task aims to address.

3 Task and Dataset Description

Participation in the UNLP 2025 Shared Task on Detecting Social Media Manipulation involved identifying manipulative techniques and manipulative Spans within Ukrainian Telegram posts using a dataset from Texty.org.ua (train.parquet, 3822 instances; test.csv, 5,735 instances), with the original training split further partitioned into 85 % training (3,248) and 15 % validation (574) subsets for development. Table 1 summarizes the data splits and overall Dataset statistics.

Split	Instances
Train	3,248
Validation	574
Test	5,735
Total Words	805,730
Unique Words	146,410

Table 1: Instance distribution across data splits and dataset word counts.

The shared task comprised two subtasks: Subtask 1 (Technique Classification), a multi-label classification over ten predefined manipulation tech-

niques (e.g., Loaded Language, Whataboutism) evaluated via Macro F1-score; and Subtask 2 (Span Identification), which required pinpointing character-level start/end indices of manipulative text segments irrespective of technique and was assessed using span-level F1-score. The implementation details and datasets for both tasks are available in the GitHub repository[2].

4 Methodology

This section describes the methodologies employed for the Technique Classification and Span Identification tasks. The research evaluated multiple machine learning (ML), deep learning (DL), and transformer-based approaches, with hyperparameter optimization conducted to maximize performance. The architectural frameworks utilized for Technique Classification and Span Identification tasks are illustrated in Figure 1 and Figure 2.

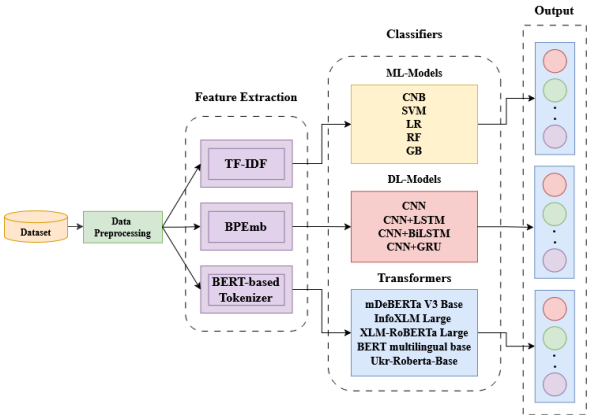

Figure 1: Schematic process for Manipulation Technique Classification

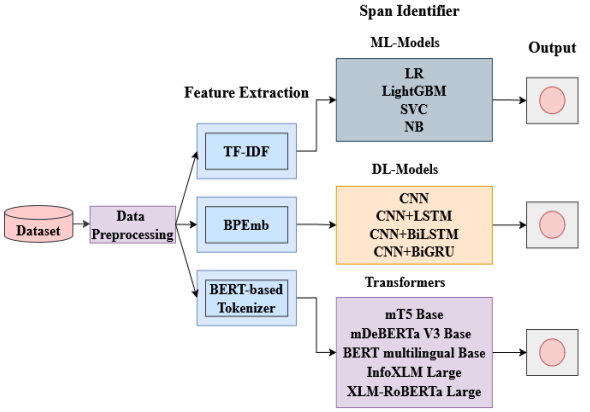

Figure 2: Schematic process for Manipulative Span Identification

[2] https://github.com/borhanitrash/Detecting-Manipulation-in-Ukrainian-Telegram/

4.1 Data Preprocessing

A single, flexible pipeline processed the provided datasets, which included 3,822 training and 5,735 test samples in Parquet and CSV formats. It begins by splitting the original training set into 85% training and 15% validation subsets, stratified by manipulation labels and seeded with 42 for reproducibility across both tasks. A uniform text-normalization routine then replaced URLs with "[URL]," normalized whitespace, imputed missing values, and detected language (Ukrainian vs. Russian). From that common foundation, task-specific steps followed. In technique classification, missing entries in the *techniques* column were filled, its string representations parsed into lists, and binary indicators generated for each technique plus a global manipulative flag, with targeted augmentation (e.g., word shuffling or deletion) applied to manipulative examples. In span identification, character-level *trigger_words* annotations were parsed into (start, end) tuples and converted into token-level BIO tags, with precise offset mapping used to align spans to the model's tokenizer.

4.2 Feature Extraction

Feature extraction was tailored to each architecture and task objective. Traditional machine learning models employed Scikit-learn's[3] TF-IDF vectorization to convert text into sparse matrices—unigrams and bigrams (limited to 10,000 features) for technique classification, and trigrams (up to 20,000 features) for span identification. Deep learning approaches utilized BPEmb (Heinzerling and Strube, 2018) subword embeddings (50,000 vocabulary size), with 300-dimensional vectors and sequences of 512 tokens for technique classification, and 100-dimensional vectors with 384-token sequences for span detection; embeddings were fine-tuned in all but one CNN-based classifier, where they remained frozen. Transformer-based systems relied on model-specific tokenization via HuggingFace AutoTokenizers (padding or truncating to 512 or 384 tokens), with classification drawing on the [CLS] token's final hidden state through a linear layer and span identification predicting BIO tag logits from the final hidden states of every token.

4.3 Machine Learning Models

Several traditional machine learning methods were applied to both Technique Classification and Span

[3] https://scikit-learn.org/stable/

Identification tasks to establish robust baseline performances. For the Technique Classification task, cast as a multi-label text classification problem, models assessed including Complement Naive Bayes ($\alpha = 1.0$ to mitigate class imbalance), Linear SVC ($C = 1.0$, `max_iter`=2000 for robust convergence on sparse features), logistic regression ($C = 1.0$, `solver`=saga, `max_iter`=1000 to balance speed and accuracy), random forest (100 trees with 'sqrt' feature splits for variance reduction) and gradient boosting (100 estimators, `learning_rate`=0.1, `max_depth`=3 to prevent overfitting). These classifiers were adapted for multi-label classification using Scikit-learn's `MultiOutputClassifier`. In the Span Identification task, framed as a word-level sequence labeling challenge under the BIO tagging scheme, involved models such as Linear SVC ($C = 0.5$, `class_weight`=balanced, `max_iter`=2000 to address token imbalance), logistic regression ($C = 1.0$, `solver`=liblinear, `multi_class`=ovr for efficient multiclass separation), multinomial Naive Bayes ($\alpha = 1.0$ smoothing for robust probability estimates) and LightGBM (300 trees, `learning_rate`=0.1 for rapid gradient-based optimization). Both tasks employed TF-IDF vectorization techniques. The classification task extracted unigrams and bigrams into a 10,000-dimensional feature space to capture local collocations. The span identification task focused on trigram contexts (target token $\pm$ one word) with up to 20,000 features to encode immediate surroundings. Table 2 provides all model configurations and complete hyperparameter settings.

Classifier	Parameter	Value
Technique Classification		
CNB	alpha	1.0
SVC	C	1.0
	max_iter	2000
LR	C	1.0
	solver	saga
	max_iter	1000
RF	n_estimators	100
	max_depth	None
	min_samples_split	2
GB	n_estimators	100
	learning_rate	0.1
	max_depth	3
Span Identification		
SVC	C	0.5
	max_iter	2000
LR	C	1.0
	solver	liblinear
	max_iter	500
MNB	alpha	1.0
LightGBM	n_estimators	300
	learning_rate	0.1
	num_leaves	31

Table 2: Hyperparameters used for Technique Classification and Span Identification tasks.

4.4 Deep Learning Models

This proposed work also employed several deep learning architectures to tackle the both Technique Classification and Span Identification tasks. For Technique Classification, models performed multi-label classification over 11 categories (one 'manipulative' label and ten manipulation techniques). Each input sequence was represented by 300-dimensional BPEmb subword embeddings. A baseline Convolutional Neural Network (CNN) featured three parallel Conv1D layers with kernel sizes of 3, 4 and 5 with 64 filters each. Each convolution used a ReLU activation. GlobalMaxPooling1D aggregated features before a dropout layer (rate 0.3) and a dense output layer of 11 units with sigmoid activations enabled multi-label prediction. To capture both local patterns and longer-range dependencies, hybrid CNN–RNN architectures were developed. The CNN frontend resembled the baseline but used 100 filters per kernel size and max-pooling. Its pooled outputs concatenated into a fixed-size feature vector. That vector merged with the final hidden state(s) of a stacked recurrent pathway. Three RNN variants were tested: two LSTM layers, two Bidirectional LSTM (BiLSTM) layers, and two GRU layers. Each recurrent layer had a hidden dimension of 256 (resulting in an effective 512 for BiLSTM). A dropout rate of 0.2 was applied between recurrent layers. After concatenation, a further dropout of 0.4 preceded the final 11-unit sigmoid layer. All classification models trained with Binary Cross-Entropy loss and class weights to address imbalance. The AdamW optimizer guided training, and gradient clipping (max norm 1.0) ensured stable updates.

The Span Identification task framed sequence labeling under the BIO scheme. Input texts used 100-dimensional BPEmb embeddings over a 50,000-token vocabulary that were fine-tuned during training. Sequences of up to 384 subwords were obtained by padding or truncation. A shared CNN feature extractor served as the frontend for all span models. It began with dropout at rate 0.25 then applied three parallel 1D convolutional layers (kernel sizes 3, 5, 7; 128 filters each) with ReLU activations and same padding to preserve length. The convolutional outputs concatenated and passed through

another dropout of 0.25. From that point, different architectures produced final BIO tags per subword. A pure CNN model applied a linear layer directly to the CNN outputs. Hybrid variants appended a single recurrent layer: unidirectional LSTM with 256 units, BiLSTM with 128 units per direction, or BiGRU with 128 units per direction. Sequence packing optimized the bidirectional models. The output sequence from the RNN (or the CNN frontend) underwent a final dropout of 0.25 before a linear layer predicted three BIO tags at each position. All span identification models used the AdamW optimizer with Cross-Entropy loss and class weights to counter label imbalance and clipping gradients at a norm of 1.0 helped keep training stable. A ReduceLROnPlateau scheduler watched the validation Span F1 score and lowered the learning rate when it stopped improving. Table 3 provides all hyperparameters for CNN, CNN+LSTM, CNN+BiLSTM, CNN+GRU, and CNN+BiGRU models used in technique classification and span identification.

Model	RNN Layers	LR	Epochs	BS
Technique Classification				
CNN	–	3e-4	50	64
CNN+LSTM	2×LSTM(256)	1.2e-4	39	32
CNN+BiLSTM	2×BiLSTM(256)	2.0e-4	28	32
CNN+GRU	2×GRU(256)	2.5e-4	25	32
Span Identification				
CNN	–	1.0e-4	20	32
CNN+LSTM	1×LSTM(256)	2.0e-4	20	32
CNN+BiLSTM	1×BiLSTM(128)	1.5e-4	20	32
CNN+BiGRU	1×BiGRU(128)	1.8e-4	20	32

Table 3: Hyperparameters of deep learning models for both Technique Classification and Span Identification, where LR and BS denote as learning rate and batch size).

4.5 Transformer-Based Models

Our approach to both the Technique Classification and Span Identification tasks rely on pre-trained multilingual Transformer models. These deep architectures use self-attention to relate every token to all others in a sequence. Such connections enable the capture of long-range and subtle contextual cues (Vaswani et al., 2017). This ability proves valuable for many natural language challenges. In this case both classification and sequence labeling require attention to fine details in text. A curated set of powerful multilingual models was selected from the Hugging Face Transformers library[4]. Each model underwent fine-tuning to adapt its learned representations to the nuances of propaganda technique detection and span identification. Multilingual pre-training ensures robust performance across languages with varying resource levels. This feature is crucial for the Ukrainian and Russian data in this shared task.

The core models evaluated for both tasks included mDeBERTa v3 base (He et al., 2021), InfoXLM large (Chi et al., 2021), XLM-RoBERTa large (Conneau et al., 2019) and BERT base multilingual cased (Devlin et al., 2018). For Technique Classification, to assess a language-specific yet relatively compact encoder, the Ukr-Roberta-Base model (Radchenko, 2020) was evaluated. This model, pre-trained extensively on a large corpus of Ukrainian texts including Wikipedia, OSCAR, and social media data, offers specialized understanding for the primary language of the dataset. For Span Identification the mT5 base model (Xue et al., 2020) was adapted from a sequence-to-sequence design. Each architecture offers a unique blend of training objectives and structure. mDeBERTa employs disentangled attention to refine token interactions. InfoXLM integrates a cross-lingual alignment objective to bridge languages. XLM-RoBERTa extends RoBERTa's robust pre-training to cover over 100 languages. mBERT provides broad multilingual coverage even without explicit alignment objectives. mT5 frames text as a generation task which can aid in decoding spans. This diversity in design helps model adaptation to varied data distributions.

Fine-tuning for the classification task began by attaching a specialized output head to each Transformer encoder. This head included one or more linear layers with GELU activation and multi-sample dropout in five parallel samples at a rate of 0.3. A consistent text preprocessing pipeline was applied. First, URLs were removed and extra whitespace collapsed. Then SentencePiece tokenization encoded the text. All sequences were padded or truncated to a maximum length of 512 tokens. To increase robustness, random word deletion at a rate of 0.3 was applied during training. Class imbalance posed a significant challenge. This was addressed using Focal Loss (Lin et al., 2017) with a gamma value of 2.0 in all setups except XLM-RoBERTa-large in which Binary Cross Entropy with inverse frequency class weights was used and it was capped at ten ensured stable gradients. Label smoothing at 0.05 reduced overconfidence. After training, optimal thresholds for each technique were tuned based

[4] https://huggingface.co/transformers

on macro F1 performance on a validation split.

Token-level span identification treated each token as an individual prediction. A token classification head was added on top of the Transformer encoder output. Most models used a three-label BIO scheme to mark span beginnings, span continuations and non-span tokens. The InfoXLM large setup was first tested with a simpler two-class approach. The sparse distribution of span labels required loss functions that focus on harder examples. Both Weighted Cross Entropy and variants of Focal Loss were evaluated. Weighted Cross Entropy was used by InfoXLM-Large and Focal Loss was used by all other models. Dropout rates within Transformer layers were increased to 0.2 for hidden modules and attention modules in InfoXLM. An optional Conditional Random Field (Lafferty et al., 2001) layer was evaluated with mDeBERTa to enforce valid tag transitions. For XLM-RoBERTa, Layerwise Learning Rate Decay (Howard and Ruder, 2018) applied smaller rates at deeper layers than at the top. Post-processing merged predicted spans within a small character distance threshold to reduce fragmentation.

All experiments used the AdamW optimizer. A cosine scheduling approach adjusted the learning rate while a linear warmup phase consumed ten percent of the total steps. Learning rates ranged from 1×10^{-5} to 2×10^{-5}. Gradient accumulation allowed large effective batch sizes despite GPU memory limits. Many runs used four accumulation steps to reach an effective batch size of thirty-two. Training proceeded with varying epochs for different models. Detailed hyperparameters such as batch sizes, andweight decay values appear in Table 4. This uniform setup ensured reproducibility and fair comparison across models. It also provided clear insight into which pre-training objectives and fine-tuning strategies work best for multilingual propaganda detection and span identification.

Model	LR	WD	BS	GA	EP
Technique Classification					
mDeBERTa-B	1e-5	0.01	8	1	10
InfoXLM-L	1.2e-5	0.01	8	1	10
XLM-R-L	1.8e-5	0.01	8	4	8
mBERT-base	1.5e-5	0.01	16	1	8
Ukr-Roberta-B	2e-5	0.01	32	1	10
Span Identification					
InfoXLM-L	1.2e-5	0.01	8	1	5
mDeBERTa-B	2e-5	0.01	4	4	5
XLM-R-L	2e-5	0.01	2	4	8
mBERT-base	2.2e-5	0.01	4	4	5
mT5-B	1.5e-5	0.01	4	4	5

Table 4: Hyperparameters used for Technique Classification and Span Identification, where LR: Learning Rate, WD: Weight Decay, BS: Batch Size, GA: Gradient Accumulation, EP: Epochs.

Classifier	Precision	Recall	F1 Score
Technique Classification			
ML Models			
LinearSVC	0.3543	0.2878	0.3102
CNB	0.2680	0.2818	0.2553
LR	0.2807	0.5433	0.3291
RF	**0.5688**	0.1060	0.1309
GB	0.3926	0.1423	0.1846
DL Models			
CNN	0.2991	0.3287	0.2816
CNN+LSTM	0.3125	0.3388	0.3077
CNN+BiLSTM	0.3403	0.3443	0.3252
CNN+GRU	0.3649	0.3087	0.3179
Transformers			
mDeBERTa V3 Base	0.3453	0.5055	0.3901
InfoXLM Large	0.3855	0.5477	0.4451
XLM-RoBERTa-large	0.3917	**0.5667**	**0.4498**
BERT multilingual base	0.3710	0.3930	0.3772
Ukr-Roberta-Base	0.3687	0.4366	0.3660
Span Identification			
ML Models			
LinearSVC	0.4020	0.3921	0.3970
LR	0.4169	0.3578	0.3851
MNB	0.4169	0.3578	0.3851
lightGBM	0.3599	0.4794	0.4112
DL Models			
CNN	0.2596	0.8715	0.4001
CNN+LSTM	0.2566	**0.9187**	0.4012
CNN+BiLSTM	0.2878	0.8126	0.4251
CNN+BiGRU	0.2949	0.8023	0.4313
Transformers			
infoXLM-large	0.5646	0.5510	0.5577
mDeBERTa-v3-base	**0.6367**	0.4644	0.5371
XLM-RoBERTa-large	0.5616	0.6500	**0.6026**
BERT-base-multilingual	0.5188	0.5697	0.5431
mt5-base	0.3930	0.6645	0.4939

Table 5: Performance Comparison of ML, DL, and Transformer Models for both tasks

5 Result Analysis

This analysis covers three model families, machine learning, deep learning and transformer based systems on both technique classification and span identification tasks using Ukrainian and Russian Telegram content. Performance was measured by macro precision, recall and F1 score as shown in Table 5.

Machine learning baselines defined the starting point. For technique classification Logistic Regression achieved the highest F1 of 0.3291, driven by strong recall of 0.5433 but lower precision. Random Forest reached precision of 0.5688 yet suffered recall of 0.1060, yielding an F1 of 0.1309. In span identification lightGBM led ML methods with an F1 of 0.4111 thanks to recall of 0.4794 and

moderate precision. Logistic Regression and Multinomial Naive Bayes tied at F1 0.3851, trading recall for higher precision. These classic approaches struggled to balance both metrics on complex multilingual data.

Deep learning variants showed mixed strengths. In technique classification the CNN+BiLSTM model reached an F1 of 0.3252 by processing context in both directions. Other CNN with GRU or LSTM followed, all outperforming the standalone CNN at F1 0.2816. On span identification models such as CNN+BiGRU scored an F1 of 0.4313 but combined recall above 0.80 with precision below 0.30. This suggests strong token detection yet imprecise boundary placement.

Transformer based systems outperformed both other groups. XLM RoBERTa Large achieved F1 of 0.4498 for technique classification (precision 0.3917, recall 0.5667) and F1 of 0.6026 for span identification (precision 0.5616, recall 0.6500). InfoXLM Large followed closely (classification F1 0.4451; span identification F1 0.5577). Models like mDeBERTa v3 base and multilingual BERT also surpassed ML and DL methods. Their pretrained multilingual embeddings and deep attention mechanisms enable a nuanced grasp of subtle cues.

Overall transformer pretrained models deliver the most reliable performance for detecting propaganda techniques and marking their exact spans in bilingual social media text. Their ability to learn rich contextual patterns clearly outstrips earlier paradigms.

6 Error Analysis

Quantitative and qualitative error analyses of the technique classification and span identification tasks employed confusion matrices and focused examination of example predictions to reveal model strengths and limitations.

6.1 Quantitative Analysis

The confusion matrix for technique classification shown in Figure 3 reveals clear strengths and weaknesses. The model excelled at common tactics. Loaded_language was identified correctly 2 079 times. Cherry_picking (619), glittering_generalities (516) and fud (410) also scored well. Rare or subtle techniques proved tougher. Straw_man (83), bandwagon (67) and whataboutism (101) each had low diagonal counts. Off-diagonal entries highlight both misclassifications and genuine multi-technique usage, a known challenge when applying standard confusion matrices to multi-label tasks for which specialized approaches have been developed (Heydarian et al., 2022). For example loaded_language co-occurred with fud (840), appeal_to_fear (743), cherry_picking (736) and cliche (620). The 275 instances where fud co-occurred with appeal_to_fear reflect their conceptual link. Such overlaps suggest the model struggles when persuasive strategies share emotional or thematic features.

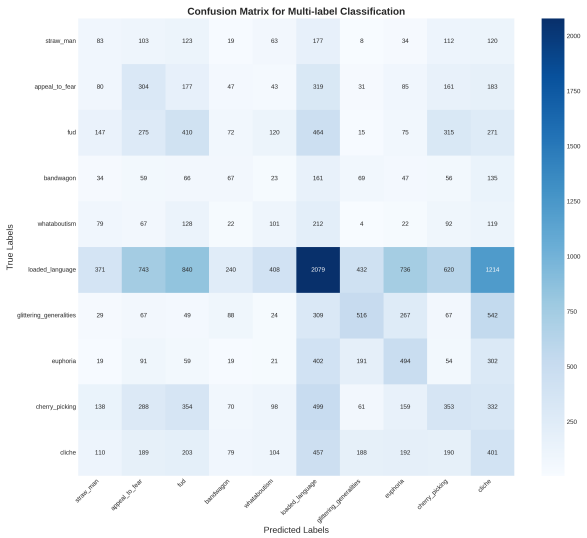

Figure 3: Confusion matrix of the proposed model (fine-tuned XLM-RoBERTa large) for technique classification

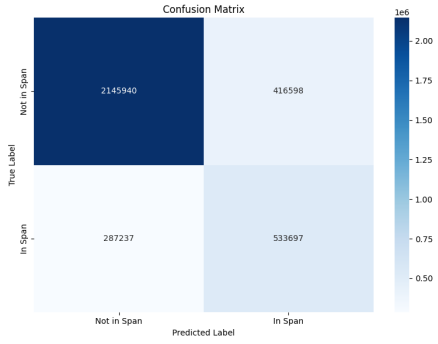

Figure 4: Confusion matrix of the proposed model (fine-tuned XLM-RoBERTa large) for span identification

At the token level, span identification shows similar patterns shown in Figure 4. True negatives (2,145,940) far outnumber false positives (416,598) and false negatives (287,237). True positives reached 533,697. The high false positive rate indicates a tendency to over-predict span boundaries. The model often tags neutral words next to

manipulative text as part of the span. This behavior lowers token-level precision more than recall and drags down the span-level F1 score. The root cause appears to be the blurred line between neutral phrasing and subtle manipulation.

6.2 Qualitative Analysis

Examination of specific cases shown in Figure 5 sheds light on these quantitative trends. In classification tasks the main technique is usually correct but extra labels slip in. For instance a post marked appeal_to_fear and loaded_language might also pick up fud in prediction. This mirrors the confusion seen in off-diagonal counts. Sometimes three techniques blur into one another when the text uses layered emotional appeals.

Content	Actual Label	Predicted Label
Соловйов, стервятник пропаганди | Реконструкція правди | Віталій Портников https://youtu.be/kB4Kq3yqiXY	Loaded Language	Loaded Language
В Черновцах укроживотные -могилизаторы похитили велосипедиста ... очередной доброволец уехал на фронт...	Appeal_to_fear, loaded_language	Appeal_to_fear, fud, loaded_language
Депутаты Рады, кажется, саму малость без интереса слушают первое выступление нового министра обороны 😁	Loaded_language, cherry_picking	Fud, Whataboutism, Loaded_language, cherry_picking

Figure 5: Few examples of predictions produced by the proposed XLM-R Large model on the technique classification task

Content	Actual Span	Predicted Span
Юзернейм. Если ты радуешься пожару на Новочеокасской ГРЭС - ты расчеловечиваешь электричество. Помни!	[(0, 101)]	[(1, 4), (10, 101)]
Русская весна плавно перейдёт в русское лето и весь Донбасс вернётся домой. Этого мы ждём всей душой.	[(0, 74), (76, 100)]	[(0, 101)]
Сподіваюсь усі зрозуміли хто така русня, а то до цього часу Ізраїль намагався на двох стільцях всидіти.	[(0, 103)]	[(0, 103)]
Соловйов, стервятник пропаганди | Реконструкція правди | Віталій Портников	[(0, 31)]	[(0, 31)]

Figure 6: Few examples of predictions produced by the proposed XLM-R Large model on the span identification task

In span identification, boundary errors are the most prevalent as shown in Figure 6. A manipulative segment may be predicted to start one token too late or end early. In other cases two distinct ground-truth spans merge into one predicted span and skip a short neutral segment. For example, the model may fragment what should be a single manipulative span [(0,101)] into smaller segments [(1,4), (10,101)], thereby omitting important introductory cues. In another case, two distinct spans [(0,74) and (76,100)] are merged into one [(0,101)], inadvertently swallowing a neutral segment. Yet when manipulative language is sharply defined—say a direct threat or an unmistakable claim—the model nails both start and end points perfectly.

These findings point to key areas for future work: sharpening distinctions among similar techniques and tightening span boundaries. Targeted refinements in feature representation and boundary detection could raise both precision and recall without sacrificing one for the other.

7 Conclusion

This paper introduces a system developed for the UNLP 2025 shared tasks on manipulation technique classification and manipulative span identification in Ukrainian and Russian Telegram posts, and demonstrates its effectiveness through extensive experiments comparing traditional machine learning methods, deep learning architectures, and transformer-based models. Among these, XLM-RoBERTa-large achieved the strongest performance, with a macro-averaged F1 of 0.4498 in technique classification and a span-level F1 of 0.6026 in span identification. Detailed error analysis revealed two key challenges: distinguishing between semantically similar manipulation tactics, particularly loaded language versus appeal to fear and precisely delineating span boundaries in morphologically complex Slavic texts. These findings emphasize contextual modeling and cross-lingual pretraining for detecting persuasive cues in Slavic texts. Future works involve boundary-aware span detection, contrastive learning, architectures for low-resource conflict zones, and synthetic data augmentation against evolving encrypted-channel tactics.

Limitations

Although the transformer model delivered strong performance it faces several limitations. (i) The dataset remains imbalanced with few instances of whataboutism and straw man which reduces detection reliability. (ii) The model struggles to identify span boundaries in morphologically complex Slavic languages resulting in overextended or merged manipulative segments. (iii) Techniques with similar emotional or rhetorical characteristics such as loaded language fear appeal and FUD are frequently misclassified. (iv) Validation has been

confined to Telegram data so performance on other social media platforms and emerging propaganda methods remains unexamined. Addressing these limitations presents key opportunities for enhancing multilingual manipulation detection.

Acknowledgments

This work was supported by Southeast University, Bangladesh.

References

Wajdi Aljedaani, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. 2022. Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry. *Knowledge-Based Systems*, 255:109780.

Y Bezliudnyi, V Shymkovych, P Kravets, A Novatsky, and L Shymkovych. 2023. Pro-russian propaganda recognition and analytics system based on text classification model and statistical data processing methods.

Kateryna Burovova and Mariana Romanyshyn. 2024. Computational analysis of dehumanization of ukrainians on russian social media. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 28–39.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Masaki Eguchi and Kristopher Kyle. 2023. Span identification of epistemic stance-taking in academic written english. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

PNA Firoj, H Mubarak, Zaghouani Wajdi, and GDS Martino. 2022. Overview of the wanlp 2022 shared task on propaganda detection in arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates*, pages 7–11.

Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. 2021. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, pages 1–10.

Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. 2017. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 489–496.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Benjamin Heinzerling and Michael Strube. 2018. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mohammadreza Heydarian, Thomas E Doyle, and Reza Samavi. 2022. Mlcm: Multi-label confusion matrix. *Ieee Access*, 10:19083–19095.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 328–339.

Shaunak Inamdar, Rishikesh Chapekar, Shilpa Gite, and Biswajeet Pradhan. 2023. Machine learning driven mental stress detection on reddit posts using natural language processing. *Human-Centric Intelligent Systems*, 3(2):80–91.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895.

Chan Young Park, Julia Mendelsohn, Anjalie Field, and Yulia Tsvetkov. 2022. Challenges and opportunities in information manipulation detection: An examination of wartime russian media. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5209–5235.

Christopher Paul and Miriam Matthews. 2016. The russian "firehose of falsehood" propaganda model. *Rand Corporation*, 2(7):1–10.

Vitalii Radchenko. 2020. Ukrainian roberta: Pretrained language model for ukrainian. https://huggingface.co/youscan/ukr-roberta-base. Accessed: 2025-06-01.

Manikandan Ravikiran and Subbiah Annamalai. 2021. Dosa: Dravidian code-mixed offensive span identification dataset. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 10–17.

Mohammad Hammas Saeed, Shiza Ali, Jeremy Blackburn, Emiliano De Cristofaro, Savvas Zannettou, and Gianluca Stringhini. 2022. Trollmagnifier: Detecting state-sponsored troll accounts on reddit. In *2022 IEEE symposium on security and privacy (SP)*, pages 2161–2175. IEEE.

Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. *Disinformation, misinformation, and fake news in social media*. Springer.

Veronika Solopova, Viktoriia Herman, Christoph Benzmüller, and Tim Landgraf. 2024. Check news in one click: Nlp-empowered pro-kremlin propaganda detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 44–51.

Veronika Solopova, Oana-Iuliana Popescu, Christoph Benzmüller, and Tim Landgraf. 2023. Automated multilingual detection of pro-kremlin propaganda in newspapers and telegram posts. *Datenbank-Spektrum*, 23(1):5–14.

Kim Nguyen Thi Thanh, Sieu Huynh Khai, Phuc Pham Huynh, Luong Phan Luc, Duc-Vu Nguyen, and Kiet Nguyen Van. 2021. Span detection for aspect-based sentiment analysis in vietnamese. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 318–328.

Taras Ustyianovych and Denilson Barbosa. 2024. Instant messaging platforms news multi-task classification for stance, sentiment, and discrimination detection. In *Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)@ LREC-COLING 2024*, pages 30–40.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned bert. In *Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda*, pages 87–91.

A Frequency of Manipulation Techniques Across Data Splits

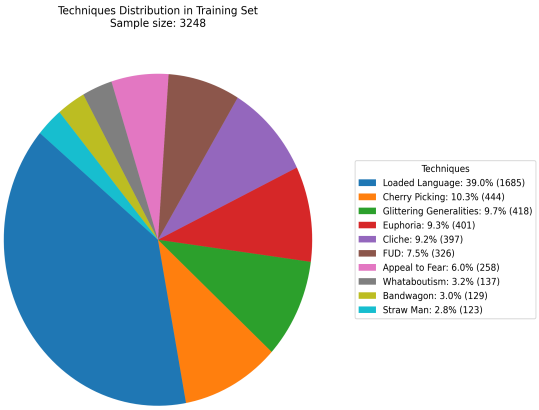

Figure 7: Manipulation techniques distribution in training set

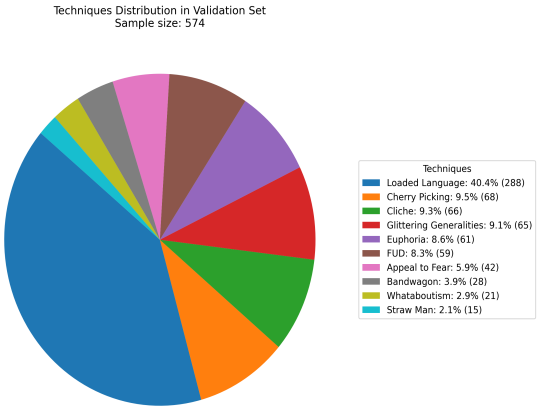

Figure 8: Manipulation techniques distribution in validation set

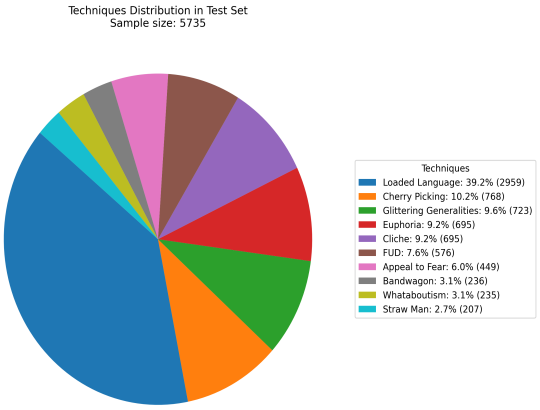

Figure 9: Manipulation techniques distribution in test set

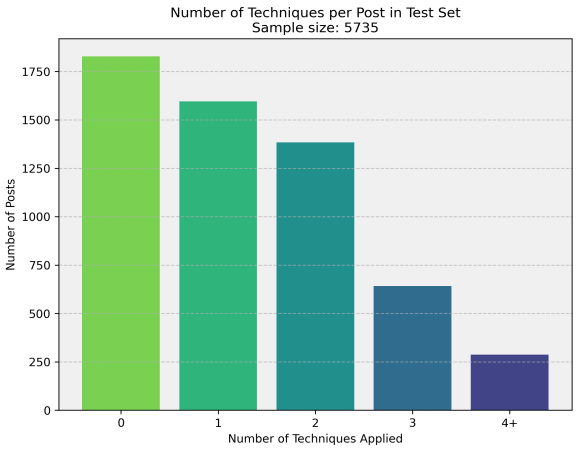

Figure 12: Number of techniques per post in test set

B Number of Techniques per Post Across Data Splits

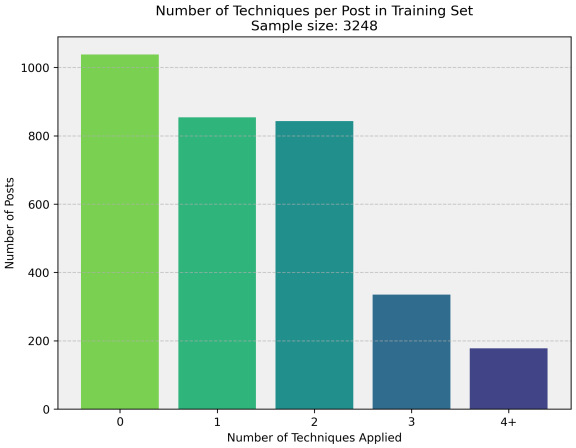

Figure 10: Number of techniques per post in training set

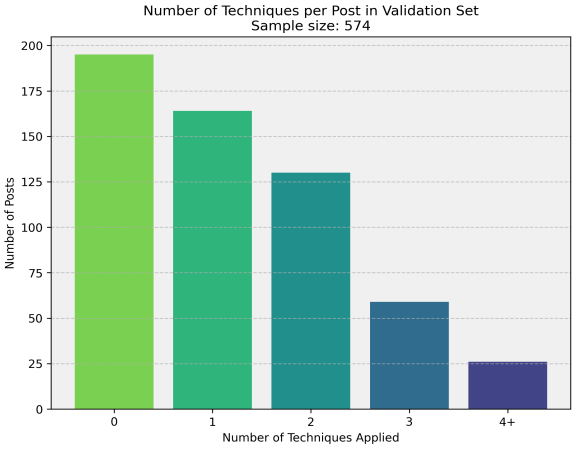

Figure 11: Number of techniques per post in validation set