TURKISH SMS SPAM MESSAGE DETECTION PROJECT REPORT

1. INTRODUCTION

These days, mobile phones are everywhere, and SMS messages have become a normal part of how we communicate daily. But unfortunately, not all messages are welcome. Many people get annoying spam texts—ads, scams, phishing attempts—that waste time and sometimes cause real harm, both financially and emotionally.

Our project focuses on building a system that can automatically tell whether a Turkish SMS is spam or not. Turkish is a complex language with its own quirks, so detecting spam here isn't straightforward. That's why we tried out both traditional machine learning methods and newer deep learning models to see which works best.

Spam detection sits right where natural language processing meets machine learning, making it a fascinating and important challenge. We hope this work adds value to research and also results in a tool that can be used practically.

2. DATASET AND PREPROCESSING

2.1 About the Dataset

We used a dataset of 4,751 Turkish SMS messages collected from various sources around Turkey. Out of these, 2,536 are spam, and 2,215 are regular messages. This mix helps us test how well our models might perform in real life.

2.2 Splitting the Data

We split the data so that 80% was used for training the models, and 20% was saved for testing how well the models work on new messages. We also applied cross-validation on the training set to fine-tune the models.

2.3 Preparing the Text

Raw text can't be fed directly into models, so we cleaned and prepared the messages by:

- Turning all letters to lowercase to avoid confusion between capitalized and small letters.
- Removing punctuation and numbers to reduce noise.
- Breaking messages down into individual words (tokens).

- Removing common Turkish words like “ve” (and), “bir” (a), and “ama” (but) that don’t add much meaning.
- Converting words into numbers using TF-IDF and word embeddings, so the models can understand them.

These steps help the models learn better and faster.

3. CLASSICAL MACHINE LEARNING METHODS

3.1 What is TF-IDF?

TF-IDF is a way to measure how important a word is in a message compared to all messages. Words that appear a lot in one message but rarely elsewhere get higher scores, helping the model focus on meaningful words.

3.2 The Models We Used

- Naive Bayes: A simple model that assumes words are independent. It’s popular because it works well and is fast.
- Logistic Regression: A straightforward model good for yes/no decisions, especially with numerical data like TF-IDF.

Support Vector Machines (SVM): A strong model that finds the best boundary to separate spam from normal messages, even in complex cases.

3.3 How We Measured Success

We looked at:

- Accuracy: How many messages were correctly classified overall.
- Precision: Of the messages predicted as spam, how many really were spam.
- Recall: Of all the spam messages, how many did the model catch.
- F1 Score: A balance between precision and recall.

Logistic Regression got 100% accuracy on the test set, but since the dataset is small, this might be too optimistic.

4. DEEP LEARNING MODEL (LSTM)

4.1 What is LSTM?

LSTM is a special kind of neural network that remembers long-term information, making it great for understanding the order and context of words in sentences.

4.2 How We Built and Trained It?

- We turned messages into sequences of numbers representing words and padded them to the same length..
- An embedding layer transformed these numbers into dense vectors.
- The LSTM layer learned patterns in the word sequences.
- The final layer decided if a message was spam or not using a sigmoid function.
- We trained the model for 5 rounds over the training data.

4.3 What We Found?

The LSTM model performed well, similar to classical methods. Its strength lies in understanding context, which might make it better for bigger, more complicated datasets.

5. CONCLUSIONS

Our experiments showed that both classical machine learning and deep learning models work well on this dataset. Logistic Regression with TF-IDF is simple and effective, while LSTM has the potential to capture deeper language nuances.

This suggests that for smaller, balanced datasets, traditional methods may be enough, but deep learning could shine with larger data.

6. FUTURE WORK

To make this better, we suggest:

- Using Turkish-specific pre-trained models like BERTurk to improve understanding.
- Building real-time spam filters for phones or servers.
- Keeping the dataset updated with new spam messages.
- Expanding to other platforms like email and social media.
- Using user feedback to keep improving the system.

7. REFERENCES

1. Çiçekli, İ., & Kılıç, H. (2020). Machine Learning Methods for Turkish SMS Spam Detection. Journal of Computer Sciences, 12(3), 45-58.
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*.
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.
4. Keras Documentation. (2024). https://keras.io

CONTRIBUTORS

Serhat ŞİMŞEK – 030721031

Ömer EDEBALI – 030721030

Mervan ŞAHİN – 030720012