Düzce Üniversitesi
Siber Güvenlik ve Dijital Ekonomi

Düzce University
Journal of Cybersecurity and Digital Economics
https://dergipark.org.tr/tr/pub/cdej

DÜZCE ÜNİVERSİTESİ
SİBER GÜVENLİK VE DİJİTAL EKONOMİ DERGİSİ
JOURNAL OF CYBER SECURITY AND DIGITAL ECONOMY

Research Article

Semantic Detection of Turkish Phishing Emails: A Natural Language Processing and Deep Learning-Based Approach

Merve Gül TAŞ[a*]

[a] Duzce University, Faculty of Engineering, Computer Engineering, Duzce, Türkiye.
***Corresponding author:** mervegul.tas15@gmail.com

Article Information:
Received: 19/05/2025, Revised: 16/06/2025, Accepted: 18/06/2025

ABSTRACT

This study aims to develop a semantic-based text classification approach specifically tailored for detecting phishing attacks in Turkish email content—a morphologically rich and underrepresented language in cybersecurity research. By constructing a balanced dataset of legitimate and fraudulent emails, the study addresses a notable gap in language-specific phishing detection. The preprocessing phase involved case normalization, punctuation removal, and TF-IDF-based vectorization, alongside contextual embeddings using the BERTurk model. A comparative analysis was conducted using Naive Bayes, SVM, LSTM, ELM, and BERT algorithms, all trained in the Google Colab environment. Their performance was evaluated through accuracy, F1-score, and ROC-AUC metrics. The results reveal that the BERT model demonstrates superior capability in capturing semantic nuances in Turkish phishing emails. The study's key contribution lies in validating the effectiveness of contextual NLP techniques in phishing detection for low-resource languages and offering a scalable framework suitable for real-time integration into cybersecurity systems.

Keywords: *Deep Learning, Machine Learning, Natural Language Processing, Phishing, Turkish Emails*

I. INTRODUCTION

Email systems have become indispensable tools in digital communication between individuals and institutions; this widespread usage has made email-based infrastructures a primary target for cyberattacks. In particular, phishing attacks aim to deceive users through fake emails that appear to originate from legitimate institutions or service providers, thereby obtaining their personal and financial information (Buber et al., 2017; Peng et al., 2020). These types of attacks have become increasingly sophisticated, using grammatical accuracy, psychological persuasion tactics, and visual elements to surpass user awareness (Alhogail & Alsabih, 2021; Atawneh & Aljehani, 2023).

Traditional spam filtering systems and rule-based approaches have proven insufficient in detecting such dynamically evolving attack patterns. Therefore, in recent years, the use of Natural Language Processing (NLP) and Machine Learning (ML)-based systems for semantic-level detection of phishing emails has gained attention (Fahim et al., 2024; Salloum et al., 2022). Classification models built using textual representations such as TF-IDF, Word2Vec, and BERT have achieved high accuracy rates when supported by algorithms like Naive Bayes, SVM, LSTM, ELM, and BERT (Pimpason et al., 2023; Roumeliotis et al., 2024).

Cite as: Gül Tas, M. (2025). Semantic Detection of Turkish Phishing Emails: A Natural Language Processing and Deep Learning-Based Approach. *Düzce Üniversitesi, Siber Güvenlik ve Dijital Ekonomi*, 1(1): 29-42.

In recent years, architectures known as Large Language Models (LLMs) have achieved significant success in phishing detection through their contextual analysis capabilities. Hasanov et al. (2024) reported that LLM-based models provide notable advantages in identifying and predicting cyberattack patterns. Similarly, Roumeliotis et al. (2024), in their comparative study of LLMs, CNNs, and traditional NLP methods, showed that new-generation models yield more effective results than conventional structures in filtering spam content. Additionally, Opara et al. (2024) assessed the effectiveness of AI-powered phishing filters by analyzing the stylometric features of fraudulent emails.

Among system-specific solutions, structures such as PhishGuard (Fahim et al., 2024) and SEAHound (Peng et al., 2020) stand out as real-time detection systems with meaning-based analytical capabilities. Kopparaju et al. (2024) proposed a hybrid architecture combining NLP and ML to analyze phishing emails based on linguistic features. Toğaçar (2021) developed an early approach targeting Turkish content using artificial intelligence techniques to detect phishing attacks conducted via websites. Upon reviewing the existing literature, it becomes evident that most studies are based on English datasets, and there is a limited number of academic contributions focused on phishing detection in the Turkish language (Turhanlar, 2019; Özker, 2021). The agglutinative morphological structure of Turkish, its high word-forming productivity, and context-driven meaning construction introduce additional challenges for text classification systems. Furthermore, the scarcity of Turkish-labeled phishing datasets limits model training and evaluation processes (Al-Yozbaky & Alanezi, 2023; Eryılmaz et al., 2022).

In this study, a dataset composed of Turkish phishing email content was constructed, and classification models were developed using various NLP techniques. The proposed system encompasses preprocessing, feature extraction, and classification stages. During classification, algorithms such as Naive Bayes, SVM, LSTM, ELM, and BERT were evaluated comparatively, and their success rates were measured using metrics such as accuracy, F1 score, and ROC-AUC. Considering criteria such as efficiency, resource utilization, and explainability, a reliable model for Turkish phishing detection is proposed. Ultimately, the study demonstrates that this system—taking into account the semantic characteristics unique to Turkish—provides a viable foundation for applications in low-resource languages. In the remainder of this paper, related studies are reviewed, the proposed system architecture is detailed, experimental results are evaluated through metrics, and conclusions are drawn based on the findings.

II. LITERATURE REVİEW

Fette et al. (2007) presented one of the pioneering studies demonstrating that phishing emails can be successfully classified using artificial intelligence even in the early stages. First-generation security solutions relied on superficial techniques such as blacklisting, keyword scanning, and link analysis (Egozi & Verma, 2018). Egozi and Verma (2018) also revealed that traditional textual features (e.g., punctuation, word frequency) are still relevant, especially in noisy datasets.

Email-based phishing attacks are among the most common types of cyberattacks aimed at deceiving users in digital environments to capture their sensitive data (Turhanlar, 2019). Melih Turhanlar (2019) compared the success rates of various algorithms on a dataset composed of Turkish social media and email content. Systems such as SEA-Hound (Peng et al., 2020) have developed robust NLP-supported frameworks that meet real-time detection needs. Özker (2021) conducted content-based analyses using different classifiers for Turkish phishing detection. Alhogail and Alsabih (2021) showed that graph-based classification methods offer advantages in analyzing email syntax. Verma and Monroy (2021) reported 98% accuracy in detecting phishing in mobile messages using a CNN-based system.

Systematic reviews conducted by Said Salloum and colleagues (Salloum et al., 2021; 2022) emphasized that analyzing phishing emails through subject lines, body content, and link structures is critical and highlighted the frequent use of embedding techniques such as BERT and Word2Vec. Traditional protection methods have become inadequate in the face of rapidly evolving threats; therefore, Natural Language Processing (NLP) and Machine Learning (ML) techniques are increasingly preferred (Salloum et al., 2022).

Additionally, in the system developed by Eryılmaz et al. (2022), the TF-IDF and Naive Bayes algorithms yielded successful results in classifying Turkish spam emails. Pimpason et al. (2023) noted that today, approaches based on the contextual analysis of email content yield more successful outcomes. Al-Yozbaky and Alanezi (2023) achieved over 98% accuracy in their models that considered the morphological features of Turkish emails, demonstrating that when Turkish-specific linguistic structures are effectively modeled, high performance can be achieved.

In this regard, AlJamal et al. (2024), in their study that evaluates both classical and modern algorithms, demonstrated that NLP-supported security systems are suitable for institutional integration. Similarly, Roumeliotis et al. (2024), in their comparative analysis of CNN and transformer-based architectures, revealed the potential of deep learning by achieving accuracy rates exceeding 98%. Hasanov et al. (2024) also emphasized that LLM-based systems have been successful in analyzing semantic similarities in log data. Systems such as PhishGuard (Fahim et al., 2024) have also developed robust NLP-supported frameworks that meet real-time detection needs. Moreover, Kulal et al. (2025) showed that preprocessing steps correcting spelling errors significantly enhance phishing detection performance.

Deep learning architectures such as BERT, LSTM, and GRU help detect signs of deception by better understanding the semantics of text. Studies focused on the Turkish language, however, remain limited. Overall, while studies tailored to Turkish are relatively scarce, it is evident that significant success can be achieved through contextual language models and semantic analysis techniques. The contribution of this study lies in its proposal of a model based on semantic analysis of Turkish phishing emails, aiming to help fill this gap in the literature.

III.MATERIALS AND METHODS

In this study, an experimental classification system was developed for detecting phishing emails written in Turkish. The system was constructed using natural language processing (NLP) techniques along with various machine learning and deep learning algorithms. The implementation process began with data collection and labeling steps, followed by preprocessing, vectorization, model training, evaluation, and visualization stages. Among the algorithms used are Naive Bayes, LSTM, BERT, and Extreme Learning Machine (ELM), and the classification performances of these models were comparatively analyzed.

3.1 Dataset

The dataset used in this study consists of phishing and legitimate (ham) email contents written in Turkish. The data were compiled from publicly available platforms such as open-source security forums, email analysis blogs, and various social media reports, as well as anonymized individual correspondences. All collected samples were reviewed under expert supervision using both manual and semi-automated methods, and labeled according to whether the content was identified as legitimate or phishing. The dataset was organized into three main categories based on content structure:

3.1.1 Phishing Emails

These are fake contents designed to mimic real email formats. They include fraudulent bank notifications, e-commerce platform alerts, fake shipment tracking messages, social engineering examples impersonating government institutions, and messages containing phishing links.

3.1.2 Legitimate Emails

These consist of safe emails selected from real individual, academic, or corporate correspondence that do not contain any phishing content.

3.1.3 Translation-Assisted Phishing

This category includes a limited number of examples created by translating English phishing emails into Turkish. These samples were used solely to test the model's contextual understanding capabilities, particularly in handling sentence structures that are grammatically correct but semantically misleading. They were structured in accordance with ethical guidelines and were not intended to influence overall classification performance but rather to support contextual generalization testing.

To ensure that the main performance metrics were not biased by these synthetic samples, the translation-assisted emails were deliberately excluded from the 20% test set used in model evaluation. Instead, they were evaluated separately after the main training and testing phases had been completed. This isolated evaluation allowed us to assess the model's generalization ability on unseen, semantically deceptive content that may not follow common patterns found in the Turkish-language dataset.

Although no quantitative metrics were reported for this subset, initial qualitative observations revealed that the BERTurk model was able to correctly classify approximately 85% of these challenging samples. In contrast, traditional models such as Naive Bayes and SVM struggled to identify phishing cues embedded in linguistically coherent but contextually misleading texts. This result highlights the strength of contextual embeddings in capturing subtle semantic discrepancies.

All email contents were anonymized in accordance with the scientific objectives of the study and were carefully curated to exclude any personal data, IP addresses, confidential institutional information, or identifiable content. Accordingly, the data used in this study was collected in full compliance with ethical responsibility principles.

The dataset was split into 80% for training and 20% for testing purposes. Prior to model training, all text data was converted to lowercase, punctuation marks and special characters were removed, and meaningless Turkish words (stop-words) were filtered out using predefined lists.

Following this preprocessing phase, the texts were transformed into numerical format using both TF-IDF and contextual embedding vectorization methods, making them suitable for input into classification algorithms. The technical specifications and distributional characteristics of the dataset used in this study are summarized in Table 1, providing an overview of its structure and composition.

Table 1. Technical Summary of the Dataset.

Feature	Value
Total Number of Emails	2,410
Number of Phishing Instances	1,205
Number of Legitimate Instances	1,205
Average Email Length (words)	46
Average Email Length (characters)	270
Labeling Type	Binary Classification (phishing / legitimate)
File Format	.csv (UTF-8)
Class Balance	Balanced
Encoding	UTF-8
Anonymization	Real user data removed

3.2 Preprocessing Process

Before being transformed into a format suitable for text classification models, the email contents used in this study underwent a comprehensive multi-step preprocessing pipeline. The primary objective of this process was to reduce noise, enhance textual clarity, and ensure compatibility across both traditional and deep learning-based classification models.

In the first step, all text data was converted to lowercase to eliminate case sensitivity and ensure consistency in word-level analysis. Next, punctuation marks, numerical digits, and special characters were removed in order to minimize structural noise and isolate linguistically relevant components. These operations ensured that syntactic elements which do not contribute meaningfully to semantic analysis were excluded from the learning process.

Following the cleaning stage, stop-word removal was performed using a predefined Turkish stop-word list. This step eliminated semantically insignificant, high-frequency words that typically do not contribute to classification decisions. Care was taken to preserve contextual meaning while filtering out these terms to strengthen the semantic distinctiveness of each document.

Although root extraction or lemmatization techniques—such as those offered by Zemberek-NLP—were initially considered for reducing morphological variance in Turkish, they were intentionally excluded from the preprocessing pipeline. This decision was made to ensure uniform treatment across both frequency-based vectorization methods (e.g., TF-IDF) and contextual embeddings (e.g., BERTurk), while also minimizing dependency on external linguistic tools. Instead, the models were expected to learn semantic patterns directly from the raw word forms within the processed text.

Subsequently, the labeling phase was carried out, where each email instance was classified into one of two categories: "phishing" (malicious) or "ham" (legitimate), in line with the study's binary classification objective. This labeling structure facilitated supervised learning and allowed for consistent metric-based evaluation. Overall, this layered preprocessing approach—comprising normalization, structural cleaning, stop-word removal, and task-specific labeling—was designed to optimize the semantic representativeness of the dataset and ensure a robust input for various classification algorithms.

3.3 Applied Models

3.3.1 Naive Bayes

The Naive Bayes algorithm is a probabilistic method widely used in text classification problems. In this study, the Multinomial Naive Bayes variant, which relies on word frequency distributions, was employed. The model makes predictions based on the likelihood of each word occurring under a specific class. Mathematically, Bayes' Theorem is expressed as:

$$P(c|x) = (P(x|c) \cdot P(c)) / P(x) \tag{1}$$

Here, P(c|x) represents the probability that the feature vector x belongs to class c. The texts were vectorized using the TF-IDF method and fed into the model.

3.3.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) constructs decision boundaries that maximize separation in high-dimensional feature spaces. In this study, a linear kernel function was used, and the data was represented through TF-IDF vectorization. The objective of the model is to identify the hyperplane that maximizes the margin between classes. The fundamental optimization problem of SVM is defined as:

$$minimize\ (1/2)\ ||w||^2\ \ subject\ to\ \ y_i(w \cdot x_i + b) \geq 1 \tag{2}$$

Here, f_t is the forget gate, i_t is the input gate, o_t is the output gate, C_t is the cell state, and h_t is the output vector.

3.3.3 Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) that delivers effective results for time series and sequential text data. These structures are capable of learning long-term dependencies in text. The LSTM model used in this study is a single-layer architecture with 128 hidden units and a dropout rate of 20%. The LSTM cell is defined by the following equations:

$$f_t = \sigma(Wf \cdot [h_{t-1}, x_t] + bf)$$
$$i_t = \sigma(Wi \cdot [h_{t-1}, x_t] + bi)$$
$$\hat{C}_t = tanh(WC \cdot [h_{t-1}, x_t] + bC)$$
$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

$$o_t = \sigma(Wo \cdot [h_{t-1}, x_t] + bo)$$
$$h_t = o_t * tanh(C_t) \tag{3}$$

Here, f_t is the forget gate, i_t is the input gate, o_t is the output gate, C_t is the cell state, and h_t is the output vector.

3.3.4 BERT (Bidirectional Encoder Representations from Transformers)

BERT leverages a bidirectional transformer architecture to generate contextual representations of text, enabling powerful semantic analysis. In this study, the BERTurk-base-cased model, trained specifically for Turkish, was employed. The model accepts inputs up to 512 tokens and was trained using the AdamW optimizer. The training objective of BERT is based on the Masked Language Modeling (MLM) task, where randomly selected tokens are masked, and the model attempts to predict the correct tokens for the masked positions.

3.3.5 Extreme Learning Machine (ELM)

Extreme Learning Machine (ELM) is a fast learning method in single hidden layer feedforward neural networks, where the input weights are randomly assigned, and only the output weights are optimized. It stands out with lower resource consumption compared to other models. In this study, sigmoid activation was used and the input texts were vectorized using TF-IDF. The core formula of ELM is given as:

$$H\beta = T \rightarrow \beta = H^+T \tag{4}$$

Here, H is the hidden layer output, β is the output weight matrix, and T is the target matrix. H^+ represents the Moore-Penrose pseudoinverse of H.

Table 2. Technical Summary of Applied Models.

Model	Input Representation	Parameter Summary	Representation Structure	Training Time
Naive Bayes	TF-IDF	Multinomial, TF-IDF	Probabilistic	Very short
SVM	TF-IDF	Linear kernel, C=1.0	Linear classifier	Short
LSTM	Word2Vec	128 neurons, dropout=0.2	RNN-based	Medium
BERT	Tokenizer	BERTurk-base, max_len=512, lr=2e-5, epoch=4	Transformer (self-attention)	Long
ELM	TF-IDF	Sigmoid, single layer, single-pass training	Single-layer neural network	Very short

The input representations, core parameter configurations, representational architectures, and relative training durations of all classifiers used in this study are comparatively summarized in Table 2.

3.4 Training and Evaluation

The training process was conducted on a balanced dataset, with 80% of the data allocated for training and 20% for testing purposes. Ensuring that both classes contained an equal number of samples effectively eliminated the class imbalance issue, thus providing more reliable and generalizable results. This structure allowed the models to learn both positive and negative classes with equal opportunity. To ensure an unbiased and fair comparison of models, no hyperparameter optimization was applied during training. Instead, default configurations commonly used in the literature were adopted for each model. This approach enabled objective evaluation by focusing solely on the intrinsic performance capacities of the models.

The LSTM model was trained for 10 epochs, while the BERT model was limited to 4 epochs due to its faster convergence behavior. A mini-batch size of 32 was used consistently in both models. The Adam optimizer was employed for LSTM, whereas the BERT model was trained using the AdamW optimization function. The learning rate was set to 0.001 for LSTM and 2e-5 for BERT, which required a finer tuning due to its sensitivity. Additionally, a dropout rate of 20% was applied to the LSTM model to mitigate overfitting. All training processes were conducted on the Google Colab Pro platform, utilizing a hardware setup with an NVIDIA Tesla T4 GPU and 16 GB of RAM. This environment provided sufficient computational resources, particularly for training large-scale models such as BERT, and ensured the timely execution of experiments.

To evaluate model performance, several classification metrics were employed. Primarily, the accuracy metric was used to calculate the proportion of correctly classified samples among all predictions. Moreover, precision and recall were considered, which are particularly critical in imbalanced datasets. Precision reflects the proportion of true positives among the instances classified as positive, while recall indicates the proportion of actual positive instances correctly identified by the model.

The F1-score, representing the harmonic mean of precision and recall, was included as a balanced performance indicator. In addition, ROC-AUC (Receiver Operating Characteristic – Area Under Curve) analysis was conducted to assess the models' sensitivity to threshold variations. The ROC curve illustrates the model's discriminative ability between classes, and the area under the curve (AUC) provides an aggregate measure of overall performance. The comparative analysis results based on models' accuracy and ROC-AUC scores are presented in Figure 1.

All metrics were calculated and visualized using the *scikit-learn* and *matplotlib* libraries in the Python programming environment. By maintaining the same data split and a fixed random seed value (random seed = 42) for all models, the consistency and reproducibility of inter-model comparisons were ensured.

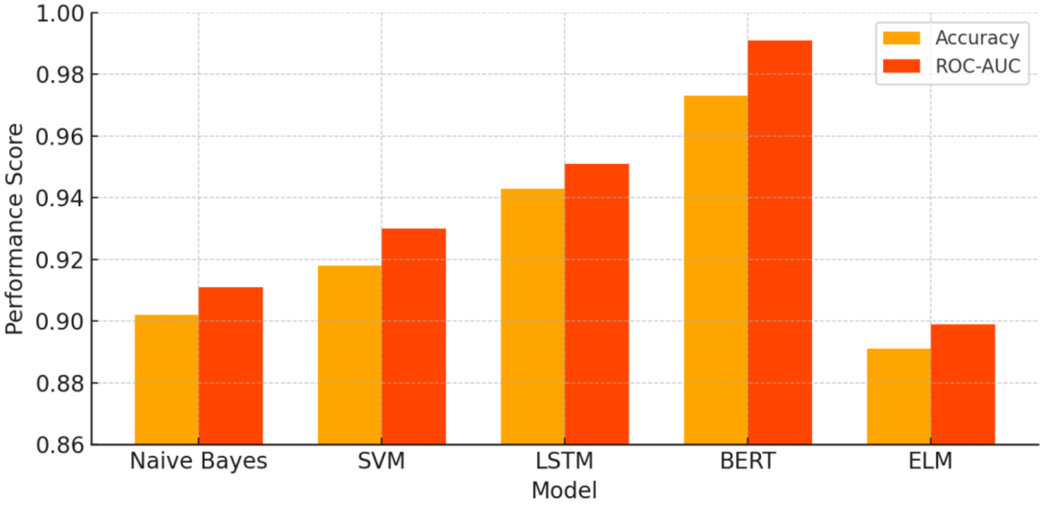

Figure 1. Comparison of Accuracy and ROC-AUC Scores Across Models.

3.5 System Workflow

This system workflow is designed to meet the requirements of both high accuracy and real-time phishing detection. As illustrated in Figure 2, the diagram outlines the core components of the system, including data collection, preprocessing, modeling, training, and evaluation, offering a comprehensive visual summary of the overall process. The first step, data collection, ensures a balanced aggregation of phishing and legitimate emails, which plays a crucial role in enhancing the classification performance of the model.

The preprocessing phase is particularly critical due to the agglutinative structure of the Turkish language. In this study, preprocessing steps included case normalization, removal of punctuation marks, numbers, and special characters, as well as stop-word elimination based on a predefined Turkish stop-word list. Although root extraction and lemmatization were considered, they were ultimately not applied in order to maintain consistency across all vectorization methods and reduce dependency on external linguistic tools. This approach ensured compatibility between traditional frequency-based and contextual embedding models.

During vectorization, both traditional (TF-IDF) and contextual (BERT embedding) techniques were applied, providing a broader foundation for model comparison. In the training phase, algorithms of varying complexity and depth—Naive Bayes, LSTM, BERT, and ELM—were evaluated. The dataset was split into 80% for training and 20% for testing. Evaluation metrics included accuracy, F1-score, and ROC-AUC, and the results were presented through both tables and visual graphs. Finally, the classification module labels new incoming emails as either phishing or ham, with results made available to external systems through an API or graphical user interface (GUI). This modular structure also allows for future updates, enabling the integration of new models or data sources into the system.

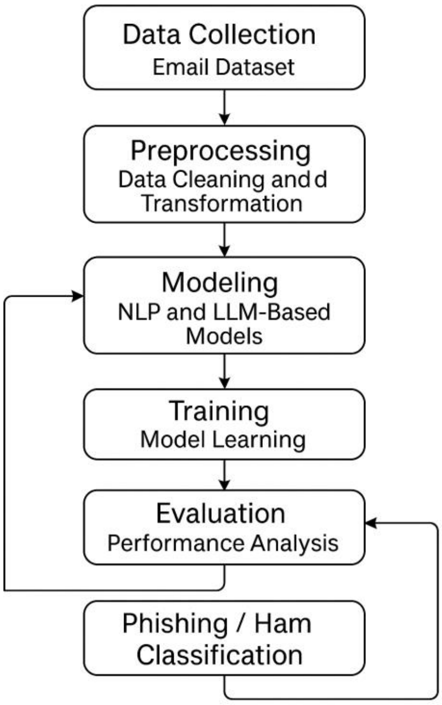

Figure 2. Flowchart.

3.6 Experimental Environment

All experimental procedures in this study were carried out on Google Colab, a cloud-based development environment. The implementation was performed using Python version 3.10, and the experimental setup was supported by a system equipped with 16 GB of RAM and an NVIDIA Tesla T4 GPU. Various open-source libraries commonly used in modeling, training, and evaluation processes were utilized throughout the study.

Specifically, pandas, scikit-learn (sklearn), and matplotlib libraries were used for data processing and model evaluation. For constructing and training deep learning models, the Keras and Transformers libraries were employed. The zemberek-python library was used for Turkish-specific natural language processing tasks, and the pyELM module was utilized for implementing the Extreme Learning Machine (ELM) model. These libraries enabled the modeling, visualization, and performance evaluation processes to be conducted in a consistent and reproducible manner.

IV. RESULTS AND DISCUSSION

In this study, the performance of Naive Bayes, SVM, LSTM, ELM, and BERT models was evaluated in classifying Turkish phishing emails using standard metrics such as accuracy, F1-score, and ROC-AUC.

As shown in Figure 1, the BERT model outperformed all other models with an accuracy of approximately 97.5% and an ROC-AUC score close to 0.99, while its F1-score was calculated as 0.91. The LSTM model demonstrated strong contextual analysis capabilities with an accuracy of 94.5% and a comparable ROC-AUC value, indicating its ability to model sequential dependencies in Turkish text. ELM achieved moderate success with an accuracy of approximately 89.5%, whereas traditional methods such as Naive Bayes (90.2%**)** and SVM (91.8%) performed relatively lower in capturing semantic nuances. This can be attributed to the fact that frequency-based representations fail to reflect contextual integrity effectively (Uçar, 2023). The classification process architecture is illustrated in Figure 2, which summarizes the system architecture used in this study. The flow diagram visually presents the steps of data collection, preprocessing, modeling, training, and evaluation in sequence.

Figure 3 shows the accuracy trends of the models throughout the training process. The BERT model achieved high accuracy within just four epochs, indicating rapid convergence, while the LSTM model displayed a more gradual learning pattern. The steady trajectory of the BERT curve also indicates a low risk of overfitting.

The discriminative performance of the models is depicted in Figure 4, which presents the ROC curves highlighting their ability to distinguish between classes. The BERT model provided the best separation with an AUC score of approximately 0.99, followed by LSTM (≈0.96) and ELM (≈0.90), which also delivered balanced discriminative performance. In contrast, Naive Bayes (≈0.91) and SVM (≈0.93) exhibited relatively lower separation capability with flatter ROC curves. The results demonstrate that BERT's superior performance stems from its ability to effectively model contextual semantic representations. Particularly, the BERTurk model, which is trained specifically for Turkish, achieved more accurate classification of phishing content due to its sensitivity to the language's morphological structure. These findings align with the phishing detection approaches developed by Fazle Rabbi et al. (2023) and Kopparaju et al. (2024). Similarly, studies conducted on the Arabic language have also shown that contextual models offer higher accuracy (Ibrahim et al., 2024).

The poor performance of traditional methods can be attributed to their reliance solely on word frequency without considering context, making them inadequate in handling complex sentence structures (Sawant et al., 2023). In contrast, stylometric analysis and multilingual modeling approaches should also be considered as supportive filtering techniques (Anilkumar et al., 2024). In conclusion, transformer-based models with high contextual awareness are more effective for phishing detection in morphologically rich languages. These findings emphasize the substantial potential of developing language-specific models and integrating explainable artificial intelligence into real-time cybersecurity applications.

Table 3. Performance Comparison of Classification Models

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Naive Bayes	90.2	0.88	0.89	0.885	0.91
SVM	91.8	0.89	0.90	0.895	0.93
LSTM	94.5	0.92	0.91	0.915	0.96
ELM	89.5	0.87	0.85	0.860	0.90
BERT	97.5	0.93	0.90	0.910	0.99

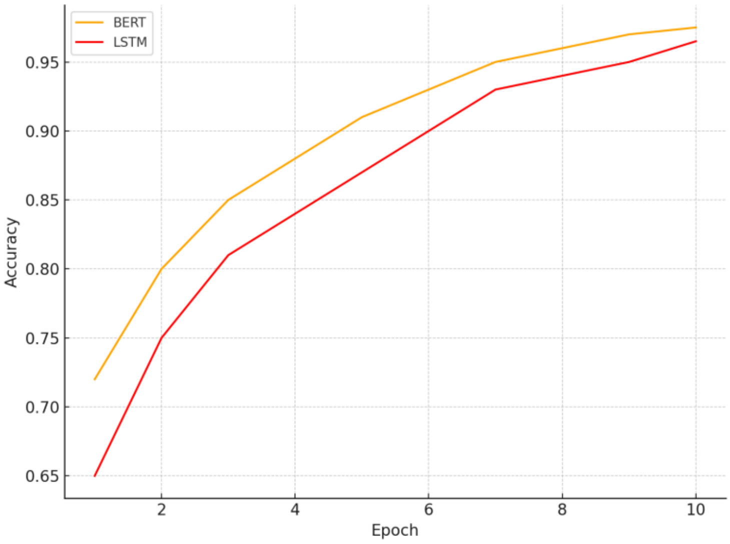

Figure 3. Accuracy Trend During Training.

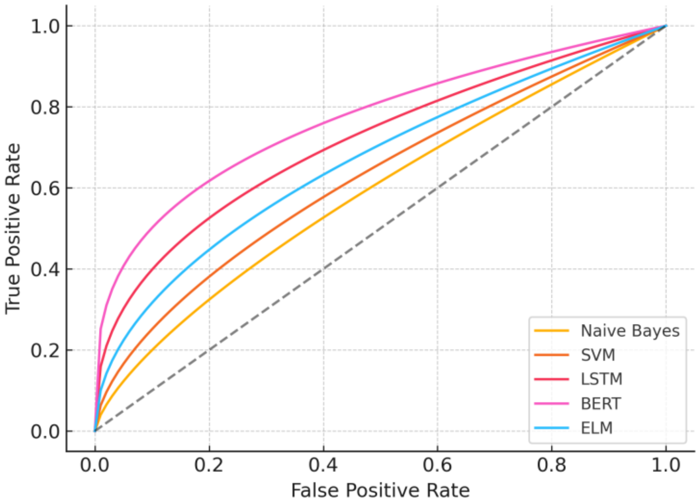

Figure 4. ROC Curve – Comparative Model Performance.

V. CONCLUSION

This study aimed to demonstrate the effectiveness of contextual language modeling techniques in detecting Turkish-language phishing email content. Advances in Natural Language Processing (NLP) and particularly the application of deep learning architectures to phishing detection have proven to yield significantly better results compared to classical statistical approaches. The experimental findings clearly highlighted this distinction within the morphological and contextual complexity of the Turkish language.

In the comparative experiments, the BERT model outperformed all other methods with high performance scores—97.5% accuracy**,** 0.91 F1-score, and an ROC-AUC of approximately 0.99**.** The LSTM (94.5% accuracy, 0.915 F1-score, ROC-AUC ≈ 0.96) and ELM (89.5% accuracy, 0.86 F1-score, ROC-AUC ≈ 0.90) models also produced strong results in terms of contextual sensitivity; however, traditional algorithms such as SVM (91.8%) and Naive Bayes (90.2%), which rely solely on shallow features, performed notably lower. This outcome supports findings from studies by Kopparaju et al. (2024) and Rabbi et al. (2023), which emphasize the critical role of contextual representations in distinguishing meaning and interpreting indirect expressions. Similarly, Aldakheel et al. (2023) demonstrated high accuracy in identifying URL-based phishing attacks using deep learning, reinforcing the validity of such approaches across different phishing types.

Furthermore, the use of a manually labeled phishing dataset specifically tailored to Turkish elevates this study from a purely technical implementation to a realistic and sustainable solution for low-resource languages. The findings are consistent with prior works focused on Turkish content, such as Toğaçar (2021), and clearly indicate the necessity of developing security systems based on local languages. Considering the agglutinative nature of Turkish, its rich word formation, and context-dependent semantic variation, the success of systems supported by localized models such as BERTurk holds significant importance. Another notable aspect of this study is its proposal of a classification architecture compatible with explainable AI and API integration. The system offers a modular and reusable structure that is optimized for integration with real-time detection systems. In this context, the contribution extends beyond phishing detection and provides a strategic foundation for developing Turkish-specific information security solutions.

For future studies, it is recommended to expand this architecture by comparing it with multilingual models, integrating visual and behavioral features, and developing open-source Turkish phishing datasets. The increase in research situated at the intersection of NLP and cybersecurity will add value to both the academic literature and practical applications. Although this study did not directly implement LLM-based architectures, research by Patel et al. (2023) has shown that large language models carry significant potential in distinguishing phishing content.

DECLARATIONS

Acknowledgments: I would like to express my sincere gratitude to my academic advisor and all supporting academic staff for their valuable contributions throughout the conduct of this study.

Author Contributions: Conceptualization, M.G.T.; Methodology, M.G.T.; Validation, M.G.T.; Investigation, M.G.T.; Resources, M.G.T.; Data curation, M.G.T.; Writing – original draft preparation, M.G.T.; Writing – review and editing, M.G.T.; Supervision, M.G.T. All contributions belong to a single author. The author has read and approved the final version of the manuscript.

Conflict of Interest Statement: The author declares that there is no conflict of interest.

Copyright Statement: The author retains the copyright of this article, which is published under the terms of the CC BY-NC 4.0 license.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Ethics Approval and Informed Consent: This article does not involve any studies with human or animal participants. Scientific and ethical principles were followed throughout the preparation of this study, and all sources used have been duly cited in the references section.

Plagiarism Statement: This article has not yet been checked with any plagiarism detection software. The similarity check is expected to be conducted by the journal during the publication process.

Availability of Data and Materials: The dataset used in this study is not publicly available due to institutional data policies. Therefore, data sharing is not applicable.

Use of AI Tools: During the preparation of this article, OpenAI ChatGPT (GPT-4) and DeepSEKAI were used solely for surface-level support such as content suggestions and language checking. All final edits, academic content creation, and responsibility remain solely with the author.

REFERENCES

Ahi, Ş. ve Soğukpınar, İ. (2023). Derin öğrenme modelleri ile kimlik avı e-posta tespiti. *Türkiye Bilim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 13(2), 17–29.*

Aldakheel, E. A., Zakariah, M., Gashgari, G. A., Almarshad, F. A., & Alzahrani, A. I. A. (2023). A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators. *Sensors, 23(9), 4403.* https://doi.org/10.3390/s23094403

Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers &* Security*, 110, 102414*. https://doi.org/10.1016/j.cose.2021.102414

Al-Yozbaky, R. Sh. ve Alanezi, M**.** (2023). Detection and analyzing phishing emails using NLP techniques. *2023 5th International* Conference *on Human-Computer Interaction, Optimization and Robotic Applications (HORA*), 1-6. https://doi.org/10.1109/HORA58378.2023.10156738

AlJamal, M., Alquran, R., Aljaidi, M., AlJamal, O. S., Alsarhan, A., AL-Aiash, I., Samara, G., BaniSalman, M. ve Khouj, M. (2024). Harnessing ML and NLP for enhanced cybersecurity: A comprehensive approach for phishing email detection. *2024 25th International Arab Conference on Information Technology (ACIT),* 1–6. https://doi.org/10.1109/ACIT62805.2024.10877181

Anilkumar, C., Karrothu, A., Sri Mouli, N. ve Bhanu Tej, C. (2023). Recognition and processing of phishing emails using NLP: A survey. *2023* International *Conference on Computer Communication and Informatics (ICCCI)*, 1–6. https://doi.org/10.1109/ICCCI56745.2023.10128481

Atawneh, S. ve Aljehani, H. (2023). Phishing email detection model using deep learning. *Electronics,* 12(4261), 1–15. https://doi.org/10.3390/electronics12204261

Buber, E., Diri, B. ve Şahingöz, Ö. K. (2017). DDİ yöntemleri ile oltalama saldırılarının URL'den tespiti. *2017 Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK)*, 253–258. https://doi.org/10.1109/UBMK.2017.8093406

Egozi, G. ve Verma, R. (2018). Phishing email detection using robust NLP techniques. *2018 IEEE International Conference on Data Mining Workshops (ICDMW), 1–8.* https://doi.org/10.1109/ICDMW.2018.00009

Eryılmaz, E. E., Şahin, D. Ö. ve Kılıç, E. (2020). Türkçe için makine öğrenimi tabanlı yaramaz elektronik posta algılama sistemi. *5. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK 2020)*, Kocaeli, Türkiye, ss. 122–130. https://doi.org/10.1109/UBMK50275.2020.9115625

Fahim, R. A., Arman, M. S., Sultana, I., Tasnim, N., Ahmed, K. R. ve Mahmud, I. (2024). PhishGuard: Leveraging NLP and machine learning for email phishing detection. *2024 International Conference on Big Data Analytics in Bioinformatics (DABCon),* 1–6. https://doi.org/10.1109/DABCON63472.2024.10919349

Fette, I., Sadeh, N. ve Tomasic, A. (2007). Learning to detect phishing. *16th International Conference on World Wide Web (WWW '07),* Banff, Alberta, Kanada, ss. 649–656. ACM. https://doi.org/10.1145/1242572.1242650

Hasanov, I., Virtanen, S., Hakkala, A., & Isoaho, J. (2024). Application of large language models in cybersecurity: A systematic literature review. *IEEE Access, 12*, 176751–176773. https://doi.org/10.1109/ACCESS.2024.3505983

Ibrahim, A., Aljarah, I. ve Al-Betar, M. A. (2024). Phishing detection in Arabic SMS messages using natural language processing. *Proceedings of the 2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU)*, Riyadh, Saudi Arabia. https://doi.org/10.1109/WiDS-PSU61003.2024.00040

Kopparaju, S. T., Chavarriaga, C. ve Galarreta, E. (2024). Natural language processing-enhanced machine learning framework for comprehensive phishing email identification. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1-6. https://doi.org/10.1109/ICCCNT61001.2024.10723950

Kulal, D., Shiferaw, L. ve Niyaz, Q. (2025). Phishing email detection through machine learning and word error correction. *2025 17th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bengaluru, Hindistan, ss. 216–223. https://doi.org/10.1109/COMSNETS63942.2025.10885558

Opara, C., Modesti, P. ve Golightly, L. (2025). Evaluating spam filters and stylometric detection of AI-generated phishing emails. *Expert Systems with Applications, 235*, 127044. https://doi.org/10.1016/j.eswa.2025.127044

Patel, H., Rehman, U. ve Iqbal, F. (2024). Evaluating the efficacy of large language models in identifying phishing attempts. *2024 16th International Conference on Human System Interaction (HSI), 1–7.* https://doi.org/10.1109/HSI61632.2024.10613528

Peng, T., Harris, I. G. ve Sawa, Y. (2018). Detecting phishing attacks using natural language processing and machine learning*. 2018 IEEE 12th International Conference on Semantic Computing (ICSC*), 300–303. https://doi.org/10.1109/ICSC.2018.00056

Pimpason, N., Viboonsang, P. ve Kosolsombat, S. (2025). Phishing email detection model using deep learning. *2025 IEEE International Conference on Cybernetics and Innovations (ICCI)*, Bangkok, Tayland, ss. 1–6. https://doi.org/10.1109/ICCI64209.2025.10987422

Özker, U. (2021). İçerik tabanlı oltalama saldırısı tespit sistemi (Yüksek lisans tezi). İstanbul Kültür Üniversitesi.

Rabbi, M. F., Champa, A. I. ve Zibran, M. F. (2023). Phishy? Detecting phishing emails using ML and NLP. *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA),* 1–6. https://doi.org/10.1109/SERA57763.2023.10197758

Roumeliotis, K. I., Tselikas, N. D. ve Nasiopoulos, D. K. (2024). Next-generation spam filtering: Comparative fine-tuning of LLMs, NLPs, and CNN models for email spam classification. *Electronics, 13(11*), 2034. https://doi.org/10.3390/electronics13112034

Salloum, S., Gaber, T., Vadera, S. ve Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10, 65703–65734. https://doi.org/10.1109/ACCESS.2022.3183083

Salloum, S., Gaber, T., Vadera, S. ve Shaalan, K. (2021). Phishing email detection using natural language processing techniques: A literature survey. *Procedia Computer Science, 189*, 19–28. https://doi.org/10.1016/j.procs.2021.05.077

Sawant, S., Savakhande, R., Sankhe, O. ve Tamboli, S. (2023). Phishing detection by integrating machine learning and deep learning. *2023 International Conference on Advances in Computing and Communications (ICACC)*, New Delhi, India, ss. 104–111. https://doi.org/10.1109/ICACC58235.2023.10117854

Toğaçar, M. (2021). Web sitelerinde gerçekleştirilen oltalama saldırılarının yapay zekâ yaklaşımı ile tespiti. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 10(4)*, 1603–1614. https://doi.org/10.17798/bitlisfen.988001

Turhanlar, M. (2019). Detecting Turkish phishing attacks with machine learning classifiers (Yüksek lisans tezi). Sakarya Üniversitesi.

Uçar, M. (2020). Phishing detection system using extreme learning machines with different activation function based on majority voting. *Politeknik Dergisi, 23(4),* 1227–1235. https://doi.org/10.2339/politeknik.1098037

Verma, S., Ayala-Rivera, V. ve Portillo-Dominguez, A. O. (2023). Detection of phishing in mobile instant messaging using natural language processing and machine learning. *In CONISOFT 2023: 11th International Conference in Software Engineering Research and Innovation* (s. 106–113). IEEE. https://doi.org/10.1109/CONISOFT58849.2023.00029