Title of the manuscript

First Author Name1[orcid id ile ilgili görsel sonucu](https://orcid.org/0000-0002-9435-xxxx), Second Author Name \*2[orcid id ile ilgili görsel sonucu](https://orcid.org/0000-0003-2086-xxxx)

1Department of NLP, Trendyol

2Department of Computer Engineering, Çukurova University

|  |
| --- |
| **ABSTRACT**  E-commerce has experienced rapid growth in recent years and continues to expand dynamically. In this sector, maximizing customer satisfaction and enhancing the shopping experience are recognized as important strategic initiatives. To maximize customer satisfaction, it is essential to accurately determine customer needs and provide appropriate solutions to meet demand. In this context, feedback obtained from customers holds significant importance. However, customer comments often contain spelling errors, complicating the analysis of these comments. This study aims to automatically correct spelling errors in user comments regarding products sold on the e-commerce site Trendyol.com. For this purpose, a system based on transformer architecture has been created. Various spelling error detection and correction models were subsequently developed based on this architecture. Prediction models have been developed using two separate datasets consisting of Trendyol user comments and two additional datasets, including the Turkish Spelling Check Dataset taken from the Hunspell library, and the effects of these four datasets on prediction performance have been examined. The success of the models has been evaluated using the Accuracy metric. The performance of the developed models was also compared with that of the model in the Zemberek library. As a result of the study, it has been observed that the utilization of the Turkish Spelling Check Dataset positively influences prediction performance. The developed system enhanced customer experience by correcting spelling errors in comments. |
| **Keywords:** Spelling error system, Product reviews, Transformer architecture, Turkish spelling check dataset, Zemberek |

1. INTRODUCTION

The e-commerce sector has shown significant growth in recent years, driven by rapid technological advancements and digitalization [1]. Companies in this sector continuously expand their consumer base by leveraging the broad access provided by digital platforms, thus diversifying competitive dynamics. This rise in the popularity of e-commerce can be attributed to a global shift in consumer behavior toward digital channels, alongside the convenient, fast, and user-friendly shopping experiences offered by online platforms. Consequently, a steady increase is observed in the user base of e-commerce companies, which, in turn, drives significant growth in transaction volumes. This surge in transactions creates a pressing need for companies to manage high volumes efficiently and underscores the necessity for streamlined process management.

E-commerce platforms revolutionize customer interactions with products and services, enabling real-time feedback that significantly influences purchasing decisions. Customer reviews serve as critical feedback, where users share their experiences with products or services, express satisfaction levels, or raise complaints. These comments not only impact the purchasing decisions of potential customers but also provide businesses with valuable insights for improving product or service quality. Customer reviews may touch on aspects such as product features, quality, customer service, or delivery processes, and are essential for assessing brand reliability and customer satisfaction in the realms of digital marketing and e-commerce. Customer feedback and reviews are vital indicators of product reliability and quality, playing a key role in the decision-making processes of prospective buyers. This dynamic empowers consumers to make more informed decisions and enhances their overall shopping experience. However, the prevalence of spelling and grammar errors in reviews poses a significant issue. As long as comments do not contain harmful content like insults, profanity, or spam, they are visible to all users. Such errors can negatively affect other customers' perceptions and damage brand reputation. Correcting these errors is essential to prevent confusion, improve readability, and ensure that the intended message is clearly understood. Proper spelling enhances communication quality, lends a professional appearance to the text, and is therefore of great importance. [2]

Automating spelling and grammar error correction is a strategic necessity for e-commerce platforms. By facilitating decision-making based on clear, well-expressed information, platforms contribute to more favorable outcomes. Moreover, enabling users to articulate their thoughts clearly and concisely improves information flow, enhances customer satisfaction, boosts sales, and fosters brand loyalty.

This study aims to automatically correct spelling errors in user comments regarding products sold on the e-commerce site Trendyol.com. For this purpose, various spelling error detection and correction models have been developed, and a system based on transformer architecture has been created.

This study is organized as follows: Section 2 includes relevant literature. Dataset is given in Section 3. Section 5 presents methodology. Details of the system is presented in Section 5. Section 6 presents the results of the study. Section 7 concludes the paper.

2. LITERATURE REVIEW

Çınar [3] presented a two-step, deep learning-based model for detecting misspelled words in Turkish. This model incorporates character-based, syllable-based, and Byte Pair Encoding (BPE) approaches alongside Long Short-Term Memory (LSTM) and Bi-Directional LSTM (Bi-LSTM) networks. Additionally, a false positive reduction model was integrated to minimize false positives caused by the use of foreign words and abbreviations frequently found on online platforms. The results indicated that the proposed Bi-LSTM model with the BPE tokenizer performed effectively. Mehta et al. [4] proposed an LSTM-based classification model that utilizes pre-trained language models to correct erroneous query inputs and spelling mistakes. Model pruning techniques, such as No-Teacher Distillation, were employed to address latency issues. The results demonstrated at least a 16% increase in accuracy. Karimi et al. [5] aimed to improve the robustness of predictions by modifying traditional autoregressive encoder-decoder models to address spelling correction challenges. Various methods were tested, and a comparative analysis was conducted. Sun [6] proposed a solution to enhance spelling error correction in search queries within e-commerce. Various types of spelling errors were systematically analyzed and categorized. A Transformer model was employed along with synthetic data generation techniques for contextual spelling correction, showing significant improvements over multiple models on both human-labeled data and online A/B experiments. Yeomans [7] introduced a new method for spelling correction in search queries or single words using a procedure for generating artificial spelling errors. This technique was used to train a generative spelling correction model based on a Transformer architecture, outperforming the conventional noise addition approach. Yeomans [8] proposed a hybrid model using the Bidirectional Encoder Representations from Transformers (BERT) masked language model combined with Levenshtein Distance (LD) for identifying and correcting various spelling errors. Spelling errors were categorized, and a comprehensive dataset was created. The results showed that the system effectively identified and corrected spelling mistakes. Yıldız et al. [9] developed a modern spell checker model designed for integration into search engines on e-commerce platforms, specifically for Turkish. The results indicated that the typo corrector performed successfully, even beyond search engine contexts. Gupta et al. [10] introduced a two-stage framework using pre-trained language models for correcting Thai spelling errors. Character Edit Distance was applied as a post-processing step to improve corrections. Experiments with two standard datasets showed that the model corrected misspelled words with a 60% success rate. Smith [11] proposed two Seq2Seq deep learning-based automatic spelling error detectors and correctors for social media comments, analyzing input words at the phonogram level. The hybrid system demonstrated effective performance. Ahmed et al. [12] developed a spelling correction system for Indonesian using the LD algorithm, addressing complex language variations and providing more contextually appropriate corrections. The system achieved a Precision rate of 92% and an F1 Score of 90%. [13] presented a multilingual spell checker model tailored to correct user queries for specific product needs. The model was implemented for auto-completion in Adobe product searches and various applications, outperforming general-purpose spell checkers on in-domain datasets. [14] proposed a model that combines spelling correction with query expansion to enhance search engine performance. Document titles were preprocessed, with Term Frequency-Inverse Document Frequency weighting applied to terms. The LD algorithm corrected typos, and query expansion improved search results. The system was tested on a dataset of 2,045 entries, achieving an average Recall of 95.91%, Precision of 63.82%, and Non-Interpolated Average Precision of 86.29%. [15] used a Masked Language Model and Edit Distance to correct spelling errors, selecting candidates based on Recall (correction rate) metrics. Results showed significant improvement over previous studies. [16] introduced QazSpell, a Kazakh spell-check and auto-correction system. A large web crawl of noisy data was used to understand misspellings, with a substring alignment model applied to detect symmetric and asymmetric patterns in word-error pairs. The model performed well in generating correction suggestions. [17] introduced a new algorithm to improve spelling error detection in Indonesian. The algorithm has been begun by gathering and combining correct and incorrect sentences, employing Bi-LSTM networks and Multi-Head Attention mechanisms to capture sequence complexity. The model achieved an Accuracy rate of 92%. [18] described the development of a model for detecting misspelled Assamese words in digital content. This model considered the context of the expression when determining word spelling. It highlighted that certain Assamese words may have multiple meanings, and even if spelled correctly according to the dictionary, they may be unsuitable for the intended context of a particular expression. Two machine learning techniques, LSTM and BiLSTM, were employed, with the BiLSTM model achieving the highest accuracy rate of 89.52%. The results indicated that the proposed approach significantly enhanced spelling error detection, outperforming previous research on the Assamese language. [19] presented a spelling checker model for detecting misspelled Urdu words. The model utilized dictionary search and edit distance techniques to generate correct spelling suggestions. A hybrid approach incorporating ranking methods such as Soundex, Shapex, LCS, and N-gram was developed to identify the best candidate word. The system demonstrated high accuracy, achieving an F1 score of 94.02%. [20] proposed a model combining a neural network-based language model with an N-gram model to detect and correct specific Vietnamese spelling errors. In experiments, this model achieved an F1 score 1% to 14% higher than other neural network-based language models and was further compared with various N-gram models.

**3. DATASET**

The dataset has been created by analyzing spelling errors in at least 1 million comments on the Trendyol platform. The Turkish Spelling Dataset taken from the Hunspell library [21] has also been incorporated into the main dataset. Table 1 presents the main dataset, while Table 2 shows the dataset with the added Turkish Spelling Check Dataset.

|  |  |
| --- | --- |
| **Table 1.** The main dataset | |
| **Dataset Name** | **Trendyol Reviews** |
| **Train Dataset** | 45000 |
| **Test Dataset** | 2500 |
| **Validation Dataset** | 2500 |

|  |  |
| --- | --- |
| **Table 2.** The dataset with the added Turkish spelling check dataset | |
| **Dataset Name** | **Trendyol Reviews & Turkish Spelling Check Dataset** |
| **Train Dataset** | 75000 |
| **Test Dataset** | 3000 |
| **Validation Dataset** | 3000 |

To expand the spelling error dataset, neighboring letters on the Turkish Q keyboard have been identified and replaced with their respective main letters. The sample representation of turkish q keyboard characters and neighbors is given in Figure 1.

metin, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Fig 1.** The sample representation of Turkish q keyboard characters and neighbors.

During dataset preparation, various error types, including omission, insertion, combination, substitution, transposition, splitting, and identity errors, have been considered. The types of errors considered are described below.

1. Omission: A letter is randomly removed from the word.
2. Addition: A letter is randomly added to the word.
3. Combination: A new word is created by combining two existing words.
4. Substitution: A randomly selected letter is replaced with one of its neighboring letters.
5. Transposition: The positions of two letters within the word are swapped.
6. Division: The word is randomly split into two parts.
7. Identity: The word is left unchanged.

As a result of the operations performed, four different datasets have become ready. The high-quality and diverse spelling error dataset is presented in Table 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 3.** The high quality and diverse spelling error dataset | | | | |
| **Error Text** | **Dataset** | **Error Distribution** | **Dataset Size** | **Output Size** |
| A1 | trendyol\_reviews.csv | “identity”: 0.10,  “insertion”: 0.16,  “omission”: 0.16,  “substitution”: 0.08,  “transposition”: 0.16,  “combination”: 0.16,  “split”: 0.13,  “english”: 0.02,  “shuffle”: 0.0,  “minedit”: 0.03,  “phonetic”: 0.0 | 45000 | 135000 |
| A2 | trendyol\_reviews.csv | “identity”: 0.10,  “insertion”: 0.10,  “omission”: 0.10,  “substitution”: 0.14,  “transposition”: 0.19,  “combination”: 0.19,  “split”: 0.13,  “english”: 0.02,  “shuffle”: 0.0,  “minedit”: 0.03,  “phonetic”: 0.0 | 45000 | 135000 |
| A3 | trendyol\_reviews\_data\_and\_hunspell.csv | “identity”: 0.10,  “insertion”: 0.16,  “omission”: 0.16,  “substitution”: 0.08,  “transposition”: 0.16,  “combination”: 0.16,  “split”: 0.13,  “english”: 0.02,  “shuffle”: 0.0,  “minedit”: 0.03,  “phonetic”: 0.0 | 75000 | 300000 |
| A4 | trendyol\_reviews\_data\_and\_hunspell.csv | “identity”: 0.10,  “insertion”: 0.10,  “omission”: 0.10,  “substitution”: 0.14,  “transposition”: 0.19,  “combination”: 0.19,  “split”: 0.13,  “english”: 0.02,  “shuffle”: 0.0,  “minedit”: 0.03,  “phonetic”: 0.0 | 75000 | 300000 |

4. METHODOLOGY

4.1 Transformers Architecture

Before the development of transformers, machine translation and sequential data processing tasks commonly used encoder-decoder architectures, typically employing Recurrent Neural Network (RNN), LSTM, or Gated Recurrent Unit cells. In these architectures, often referred to as Seq2Seq models, input tokens are encoded into hidden states through multiple RNN cells. These encoded hidden states are then combined before being passed to the decoder. Decoders can leverage all encoded data to predict output tokens using these hidden states. However, although the last hidden state from the encoder contains substantial meaning and context, this information may not be fully captured by the Seq2Seq decoder. This issue, known as the ‘bottleneck problem,’ implies that as the input sequence grows longer and more complex, the hidden state becomes less effective. The attention mechanism addresses this limitation by allowing simultaneous processing of multiple inputs. During this process, it generates weight matrices for each hidden state and computes a weighted sum of all previous encoder states, enabling the decoder to identify which hidden states to “pay more attention” to. Since Seq2Seq architecture processes input sequentially, the information from the previous hidden state (t-1) is required to calculate the token at the current time (t). This sequential processing may hinder the correction of final tokens, particularly in tasks with numerous erroneous tokens, such as spelling correction. The limitations of these earlier systems are overcome by the transformer architecture and its attention mechanism. The transformer, based on the encoder-decoder structure, incorporates fully connected layers and multi-head self-attention mechanisms. This structure minimizes performance degradation from long dependencies and enables parallel computation. Positional embeddings and multi-head self-attention encode additional information about token positions and inter-token relationships [22].

4.2 Zemberek Library

Zemberek's current goal is to offer a general (Natural Language Processing – NLP) framework for other, largely ignored Turkic Languages in addition to Turkish. At the moment, the framework offers fundamental NLP functions like spell checking, morphological parsing, stemming, word construction, word suggestion, word conversion from words written exclusively in American Standard Code for Information Interchange characters (also known as "deasciifier"), and syllable extraction. To facilitate the work of language developers, Zemberek externalizes some language data to text-based configuration files. However, externalization typically has a negative impact on performance and flexibility. As a result, certain details are retained in the code, such as special cases and the suffix production mechanism [23].

5. DETAILS OF THE SYSTEM

The Seq2Seq architecture based on the encoder-decoder structure has been designed for training the obtained dataset. The variations of the created dataset have been diversified using different options. The constructed architecture has been trained in various configurations with different hyperparameters. Figure 2 illustrates the transformer architecture.

metin, diyagram, ekran görüntüsü, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Fig 2.** Transformer architecture

Models for spelling error detection and correction have been developed. Models A1, A2, A3, and A4 have been developed for four distinct datasets. The parameters such as hidden dimension, encoder/decoder layers, number of epochs and number of errors have been evaluated in different combinations. In addition, the influence of these parameter values on model performance has been systematically examined. In this context, eight different models have been developed. The parameter values of the developed models are presented in Table 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 4.** The parameter values of the developed models | | | | | |
| **Model Name** | **Dataset** | **Hid Dim.** | **Enc/Dec. Layers** | **Epoch** | **Num of Errors** |
| A1-1 | A1 | 128 | 4/2 | 8 | 2 |
| A1-2 | A1 | 256 | 6/4 | 10 | 2 |
| A2-1 | A2 | 128 | 4/2 | 8 | 2 |
| A2-2 | A2 | 256 | 6/4 | 10 | 2 |
| A3-1 | A3 | 128 | 4/2 | 8 | 2 |
| A3-2 | A3 | 256 | 6/4 | 10 | 2 |
| A4-1 | A4 | 128 | 4/2 | 8 | 2 |
| A4-2 | A4 | 256 | 6/4 | 10 | 2 |

Additionally, the performance of the model in the Zemberek library has been tested as an alternative spelling corrector for the developed solution. The validation dataset has been provided as input to this module, and the results obtained have been compared.

6. RESULTS AND DISCUSSION

The prediction performance of the developed models has been evaluated with the Accuracy metric. Accuracy has been chosen because the problem at hand is a classification problem. It is widely recognized in the literature as a frequently used and reliable metric for such problems. The Accuracy values of the models obtained using different datasets are presented in Table 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 5.** The Accuracy values of the models | | | | |
| **Method/Achievement (Accuracy)** | **Trendyol Reviews-1** | **Trendyol Reviews-2** | **Trendyol Reviews & Turkish Spelling Check Dataset-1** | **Trendyol Reviews & Turkish Spelling Check Dataset-2** |
| A1-1 | %78.40 | %75.15 | %69.50 | %68.37 |
| A1-2 | %75.75 | %71.45 | %69.17 | %68.98 |
| A2-1 | %77.40 | %79.10 | %70.16 | %67.17 |
| A2-2 | %73.21 | %75.15 | %69.35 | %70.15 |
| A3-1 | %79.45 | %79.63 | %85.72 | %84.37 |
| A3-2 | %77.16 | %77.72 | %80.64 | %84.02 |
| A4-1 | %79.08 | %78.98 | %83.27 | %81.29 |
| A4-2 | %77.02 | %76.17 | %82.27 | %82.79 |
| Model in the Zemberek | %54.48 | %55.31 | %61.27 | %61.74 |

* Models within the A3 series consistently achieved higher Accuracy rates relative to their counterparts.
* The A3-1 model stood out as the most successful model by providing 85.72% Accuracy with the Trendyol Reviews & Turkish Spelling Check Dataset-1 and 84.37% Accuracy with the Trendyol Reviews & Turkish Spelling Check Dataset-2. The closest performance to this model has been shown by A3-2 with 84.02% Accuracy in the Trendyol Reviews & Turkish Spelling Check Dataset-2.
* The A1-1 and A1-2 models lag behind the leading A3 models, recording Accuracy rates of 78.40% and 75.75%, respectively.
* The Accuracy rates obtained through the model in the Zemberek library, specifically 54.48%, 55.31%, 61.27%, and 61.74%, are markedly lower than those of the other models. This finding suggested that the model in the Zemberek library is less effective for the datasets under consideration.
* The prediction models developed using the Trendyol reviews datasets exhibited similar performance.
* The models developed with the Trendyol Reviews & Turkish Spelling Check Datasets delivered more accurate forecast results than those developed with the Trendyol reviews datasets.
* It has been observed that the forecast models developed with the Trendyol Reviews & Turkish Spelling Check Dataset-1 and Trendyol Reviews & Turkish Spelling Check Dataset-2 datasets showed similar performance.

Overall, the A3 series models, particularly A3-1 and A3-2, achieved the highest performance. Incorporating the Trendyol Reviews & Turkish Spelling Check Dataset appears to enhance overall accuracy rates, while the Zemberek library model demonstrates lower performance across all models, failing to produce sufficient results for these datasets. When the results obtained with the developed prediction models have been examined, it has been observed that the transformer-based architecture exhibits satisfactory performance in detecting and correcting spelling errors.

An example of a comment processed by the developed system is provided below. The first sentence displays the raw comment received as input from the user. The second sentence shows the corrected version of this comment, with spelling errors addressed by the system.

Sample outputs have been obtained using FastAPI Swagger. The screenshot of the outputs is shown in Figure 3.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Fig 3.** FastAPI swagger example outputs

The developed system contributed to improving the customer experience by correcting typos in comments. These findings offer a significant foundation for evaluating the efficacy of the proposed methodologies and highlight opportunities for further refinement and optimization in future research endeavors.

7. CONCLUSION

E-commerce has experienced rapid growth in recent years and continues to expand dynamically. To maximize customer satisfaction in this sector, accurately identifying customer needs and developing strategies to meet demand are essential. Customer reviews are valuable for e-commerce platforms; however, they frequently contain typographical errors. In this study, a system has been developed to automatically correct typos in user comments on products sold on the e-commerce site Trendyol.com and models for spelling error detection and correction have been developed. Model performance has been evaluated using the Accuracy metric, revealing that the A3 series models demonstrated the highest performance. This study presents a more up-to-date system compared to previous research. The effect of using Turkish Spelling Check Dataset has been analyzed. In addition, the performance of the developed models has been also compared with the performance of the model in the Zemberek library. The study emphasizes the importance of spelling error detection and provides valuable insights for e-commerce businesses.

ORCID

First Author [orcid id ile ilgili görsel sonucu](https://orcid.org/0000-0002-9435-xxxx) https://orcid.org/0000-0002-9435-xxx

Second Author [orcid id ile ilgili görsel sonucu](https://orcid.org/0000-0003-2086-xxxx) https://orcid.org/0000-0003-2086-xxx

REFERENCES

[1]. Şevik, S., Aktaş, M., Doğan, H. & Koçak, S. (2013). Mushroom drying with solar assisted heat pump system. *Energy Conversion and Management*, 72, 171-178. <https://doi.org/10.1016/j.enconman.2012.09.035>

[2]. Indian Institute of Horticultural Research, Indian Council of Agricultural Research (ICAR-IIHR), Retrieved from <https://www.iihr.res.in/post-harvest-technologies>. accessed on 02.10.2018.

[3]. Çınar, İ. (2009). Osmotic dehydration, mechanism and applications. *The Journal of Food*, 34(5), 325-329.

[4]. Mehta, B. K., Jain, S. K., Sharma, G. P., Mudgal, V. D., Verma, R. C., Doshi, A., & Jain, H. K. (2012). Optimization of osmotic drying parameters for button mushroom (Agaricus bisporus). *Applied Mathematics*, 3(10), 1298.

[5]. Karimi, F., Rafiee, S., Taheri-Garavand, A., & Karimi, M. (2012). Optimization of an air drying process for Artemisia absinthium leaves using response surface and artificial neural network models. *Journal of the Taiwan Institute of Chemical Engineers,* 43(1), 29-39.

[6]. Sun, D. W. (2014). *Emerging technologies for food processing*. Elsevier, London.

[7]. Yeomans, J. S. (2014). A parametric testing of the firefly algorithm in the determination of the optimal osmotic drying parameters of mushrooms. *Journal of Artificial Intelligence and Soft Computing Research,* 4(4), 257-266.

[8]. Yeomans, J. S. (2015). Determining optimal osmotic dehydration process parameters for papaya: a parametric testing of the Firefly Algorithm for Goal Programming optimization. *Scientia Agriculturae*, 10, 127-136.

[9]. Yıldız, A. K., Polatcı, H., & Uçun, H. (2015). Drying of the banana (Musa cavendishii) fruit and modeling the kinetics of drying with artificial neural networks under different drying conditions. *Journal of agricultural Machinery Science*, 11(2), 173-178.

[10]. Gupta, P., Bhat, A., Chauhan, H., Ahmed, N., & Malik, A. (2015). Osmotic dehydration of button mushroom. *International Journal of Food and Fermentation Technology*, 5(2), 177.

[11]. Smith, A. M. (2015). *Optimization of pulsed-vacuum osmotic dehydration of blueberries*. (Doctoral dissertation), West Virginia University.

[12]. Ahmed, I., Qazi, I. M., & Jamal, S. (2016). Developments in osmotic dehydration technique for the preservation of fruits and vegetables. *Innovative Food Science & Emerging Technologies*, 34, 29-43.

[13]. Bahmani, A., Jafari, S. M., Shahidi, S. A., & Dehnad, D. (2016). Mass transfer kinetics of eggplant during osmotic dehydration by neural networks*. Journal of food processing and preservation*, 40(5), 815-827.

[14]. Cao, T., & Yeomans, J. S. (2017). An evolutionary firefly algorithm, goal programming optimization approach for setting the osmotic dehydration parameters of papaya*. Journal of Software Engineering and Applications*, 10(2), 128.

[15]. Yeomans, J. S. (2018). Computing optimal food drying parameters using the firefly algorithm. *GSTF Journal on Computing* (JoC), 4(1).

[16]. Goula, A. M., Kokolaki, M., & Daftsiou, E. (2017). Use of ultrasound for osmotic dehydration. The case of potatoes. *Food and Bioproducts Processing*, 105, 157-170.

[17]. Ramya, V., & Jain, N. K. (2017). A Review on osmotic dehydration of fruits and vegetables: an integrated approach. *Journal of Food Process Engineering*, 40(3), e12440.

[18]. Akpınar, E. K., & Daş, M. (2018). Mushroom drying in air heated solar collector drying system and modeling of drying performance with artificial neural network*. Erzincan University Journal of Science and Technology*, 11(1), 23-30.

[19]. Das, I., & Arora, A. (2018). Alternate microwave and convective hot air application for rapid mushroom drying. *Journal of Food Engineering*, 223, 208-219.

[20]. Anshu, S., & Anju, B. (2018). Effect of osmotic dehydration on quality of oyster mushrooms. *Developmental Psychology,* 6(2), 1601-1605.

[21]. González-Pérez, J. E., López-Méndez, E. M., Luna-Guevara, J. J., Ruiz-Espinosa, H., Ochoa-Velasco, C. E., & Ruiz-López, I. I. (2019). Analysis of mass transfer and morphometric characteristics of white mushroom (Agaricus bisporus) pilei during osmotic dehydration. *Journal of Food Engineering*, 240, 120-132.

[22]. Imanirad, R., & Yeomans, J. S. (2015). Fireflies in the fruits and vegetables: combining the firefly algorithm with goal programming for setting optimal osmotic dehydration parameters of produce. *In Recent Advances in Swarm Intelligence and Evolutionary Computation*, pp. 49-69, Springer, Cham.

[23]. Pence, I., Cesmeli, M. S., Senel, F. A., & Cetisli, B. (2016). A new unconstrained global optimization method based on clustering and parabolic approximation. *Expert Systems with Applications*, 55, 493-507.

**Techno-Science** *Paper ID:* *xxx*