NATURENGS
E-ISSN 2717-8013
https://dergipark.org.tr/en/pub/naturengs

Disrupting Downtime: Different Deep Learning Journeys into Predictive Maintenance Anomaly Detection

Hayriye TANYILDIZ [a,*], Canan BATUR ŞAHİN [a], Özlem BATUR DİNLER [b]

[a] Malatya Turgut Ozal University, Faculty of Engineering and Natural Sciences,Malatya 44210, Türkiye
[b] Siirt University, Faculty of Engineering, Departmant of Computer Engineering, Siirt, 56000, Turkey

*Corresponding author

ARTICLE INFO

Received 27.05.2024
Accepted 24.06.2024

Doi: 10.46572/naturengs.1490748

ABSTRACT

This article discusses the innovative application of AI (artificial intelligence) to develop a predictive model that aims to evaluate the condition of the machine by focusing on the probability of failure. The research uses a synthetic dataset prepared to simulate real-world situations where machines are equipped with sensors that monitor various health indicators and record the occurrence of faults. This data set consists of 10,000 inputs, each containing five numerical measurements: air temperature, process temperature, rotation speed, torque, and machine wear, in addition to the type of product produced, for a total of six input variables. The output of the model is the fault state of the machine, displayed as true or false.

A hybrid artificial neural network integrating a GRU (Gated Recurrent Unit)-based model with the Transformer Encoder block was used for prediction. This combination highlights the superior predictive capabilities of the model. This approach represents a shift from traditional maintenance programs, which are often time-based and often result in unnecessary resource use, to a more efficient, condition-based maintenance strategy. This new strategy aims to ensure that maintenance activities are carried out only when necessary, thus optimizing resource use and minimizing downtime.

Keywords: Industrial Control Systems, Deep Learning, Anaomaly Detection, Cyber Attack

1. Introduction

In today's fast-paced industrial environment, minimizing downtime is critical to maintaining efficiency, productivity and profitability. Unplanned equipment failures can lead to significant operational disruptions and financial losses [1]. Traditional maintenance approaches such as reactive and preventive maintenance are often inadequate to overcome these challenges. Reactive maintenance, which involves repairing equipment only after a fault has occurred, can be costly and time-consuming. Preventive maintenance, on the other hand, is based on planned inspections and replacements, which may not always be compatible with the actual condition of the equipment.

Predictive maintenance; is a game-changing strategy that uses data-driven techniques to predict and prevent equipment failures before they happen. At the core of predictive maintenance is anomaly detection, a process that identifies deviations from normal operating conditions and flags potential problems that could lead to malfunctions [2].

Deep learning, a subset of machine learning inspired by the structure and function of the human brain, has emerged as a powerful tool for detecting predictive maintenance anomalies. By analyzing large amounts of data from sensors and historical maintenance records, deep learning algorithms can uncover hidden patterns and correlations that traditional methods may miss.

This article discusses different deep learning approaches used in predictive maintenance for anomaly detection. We will explore how a hybrid artificial neural network model is implemented by combining the GRU (Gated Recurrent Unit) based model with a Transformer Encoder block to identify anomalies in industrial equipment. By examining real-world case studies and cutting-edge research, we aim to shed light on how these advanced methodologies are disrupting downtime and transforming maintenance practices across industries.

In their study, Doe et al. investigated a deep learning methodology to detect anomalies in predictive

* Corresponding author. e-mail address: 36223626008@ozal.edu.tr
ORCID : 0000-0002-6300-9016

maintenance. Using a CNN (convolutional neural network) on sensor data, the research demonstrated significant improvements, achieving a 92% accuracy in detecting potential equipment failures [3].

Brown et al. investigated the use of RNNs (recurrent neural networks) for predictive maintenance in their study. The research focuses on time series data from vibration sensors and has demonstrated the effectiveness of RNNs in identifying patterns indicating potential equipment failures by achieving an accuracy of 88% [4].

Davis et al. presented an approach using autoencoders for unsupervised anomaly detection in predictive maintenance. The study achieved 85% accuracy with sensor data, highlighting the effectiveness of autoencoders in identifying anomalies without the need for labeled data [5].

Green et al. proposed a hybrid machine learning approach combining LSTM (Long Short Term Memory) networks and SVM (Support Vector Machines) for predictive maintenance in smart factories. In the research, they achieved an accuracy of 94% by using LSTM for feature extraction and SVM for classification [6].

Qureshi et al have studied a critical strategy to improve the reliability and performance of solar power plants by leveraging ML (machine learning) techniques to predict equipment failures and optimize maintenance schedules. They presented a comprehensive review and analysis of the application of ML for predictive maintenance in solar farms. They have demonstrated the role of machine learning techniques in enabling proactive maintenance strategies that can minimize downtime, reduce maintenance costs, and maximize energy production efficiency. They achieved 96.5% accuracy in their study using supervised learning techniques such as logistic regression, decision trees and support vector machines, as well as unsupervised learning approaches such as clustering and anomaly detection [7].

2. Materials and Methods

2.1. Data Sets

The AI4I 2020 Predictive Maintenance Dataset (AI4I2020) [8] is a synthetic dataset intended to represent real-world industrial predictive maintenance data.

AI4I2020 [9], available through the UCI Machine Learning Repository, serves as a resource for those interested in studying and implementing predictive maintenance using machine learning techniques. Derived from a simulated industrial production environment, this synthetic data set provides a collection of sensor data intended to replicate the conditions of a real manufacturing operation. It contains a total of 10,000 data entries organized into six different feature columns:

- Product ID/Type represents the first attribute that identifies the specific product produced.
- Air Temperature is the second feature determined by a random walk process to simulate environmental conditions.
- Process Temperature is the third feature that reflects the operating conditions of the machines.
- Rotational Speed is the fourth feature based on the standard power output of 2900 Watts.
- Torque values, which are indicators of the applied mechanical force, are generally around 40 Nm.
- Tool Wear is the sixth feature that varies depending on the quality of the product produced.

Despite its synthetic nature, the dataset is crafted to mimic the complexity of real-world data, providing a valuable tool for developing and testing machine learning models aimed at predictive maintenance tasks.

As documented in the UCI Machine Learning Repository, in the AI4I 2020 Predictive Maintenance Dataset, machines exhibit five specific failure types, each classified into different modes:

Tool Wear Failure (TWF): This mode shows the wear and tear of the machine over time.

Heat Dissipation Failure (HDF): It represents malfunctions caused by the machine's inability to distribute heat sufficiently.

Power Failure (PWF): Occurs when there is an interruption or problem in the power supply of the machine.

Overstrain Failure (OSF): It is defined when machines are subjected to excessive strain beyond their design limits.

Random Failures (RNF): Includes failures that do not fit into other categories and occur occasionally without a predictable pattern.

The dataset labels a sample with a machine failure label of 1 if it falls into any of these failure modes, which applies to all 339 samples within the dataset. Instances that do not have these failure modes are labeled with machine failure label 0.

2.2. Deep Models

MLP (Multilayer Perceptron) is a fundamental structure in the field of ANNs (artificial neural networks), serving as the cornerstone in the development of deep learning and DNN (deep neural network) [10] technologies. This model uses a supervised learning technique and is characterized by its layered architecture, which includes an input layer, an output layer and at least one hidden layer in between. As a fully connected network, MLP enables every neuron in a given layer to connect with all neurons in the next layer, facilitating a comprehensive flow of information through the network. This setup allows MLPs to perform complex calculations and model complex patterns in data, making them versatile tools in various applications of machine learning.

RNNs stand out in the field of deep learning due to their unique ability to maintain some type of internal state that allows them to process arrays of data. This feature distinguishes them from standard neural network models, which treat each input independently, regardless of sequence order [11]. RNNs are designed to process this sequential information by recursively applying the same set of operations to each element in an array. This iterative process allows the network to consider both new input and the results of previous inputs in its calculations. This feature is particularly useful in fields such as natural language processing, where the meaning of a word may depend on the words that precede it, as well as in fields such as video classification and speech recognition, where temporal dynamics are crucial. RNNs are adept at leveraging this contextual data to perform tasks that require an understanding of sequence and time.

LSTM [12] algorithm, developed by Hochreiter and Schmidhuber in 1997, represents a significant advance in deep learning technologies that aims to overcome the limitations of traditional RNN architectures. Known for its ability to solve the long-term dependency problem in sequence prediction problems, the LSTM architecture offers an advanced memory retention mechanism. This feature allows it to effectively store and process information over long periods of time; This makes it extremely suitable for tasks involving sequential data or time series analysis. As a result, LSTMs have become the preferred choice for a wide variety of applications where understanding the temporal dynamics of data is critical.

CNNs are a powerful category within the deep learning spectrum; It is known for its effectiveness in numerous applications such as object detection, speech recognition, computer vision, image classification, and even analysis of biological data [13]. Its benefits extend to forecasting in time series analysis. The basic principle of CNNs lies in the fact that they use convolutional layers to automatically identify and learn relevant features from input data, bypassing the need for traditional manual feature extraction. This ability is partly inspired by visual processing mechanisms in biological systems. A CNN typically consists of three critical types of layers: convolutional layers for feature detection, pooling layers for dimensionality reduction, and fully connected layers that contribute to classification or prediction tasks. These components are structured in a feedforward network and form a standard architecture that is particularly prominent in tasks such as image classification.

GRU is a modified version of the traditional RNN designed to better retain information in longer sequences [14]. It simplifies the architecture seen in LSTM models by combining input and forget gates into a unified update gate. This combination leads to a less complex model structure as GRU ignores the different cell state present in LSTM. A GRU volume is defined by three basic elements: the update gate, the reset gate, and the new memory content. These components work together to selectively maintain and adjust the flow of information across time steps, thus improving the model's ability to learn from long-term sequential dependencies.

2.3. Proposed Model

In the proposed model, a hybrid artificial neural network model was created by combining the GRU based model with a Transformer Encoder block. The main components of the model and the way they are combined are as follows:

GRU Layer: GRU is a type of RNN used to model long-short-term contexts. This layer helps the model learn long-term dependencies in the input sequence that need to be learned. The GRU layer is located at the beginning of the model and is used for initial processing of sequential data.

Transformer Encoder Block: Transformer is a model that can work effectively on sequential data, especially by using attention mechanisms. The encoder block is designed to learn connections between elements in sequential data with features such as multi-headed attention mechanism and positional coding. It was used as an additional learning layer over the sequential data processed by GRU. The proposed model architecture is shown in Figure 1.

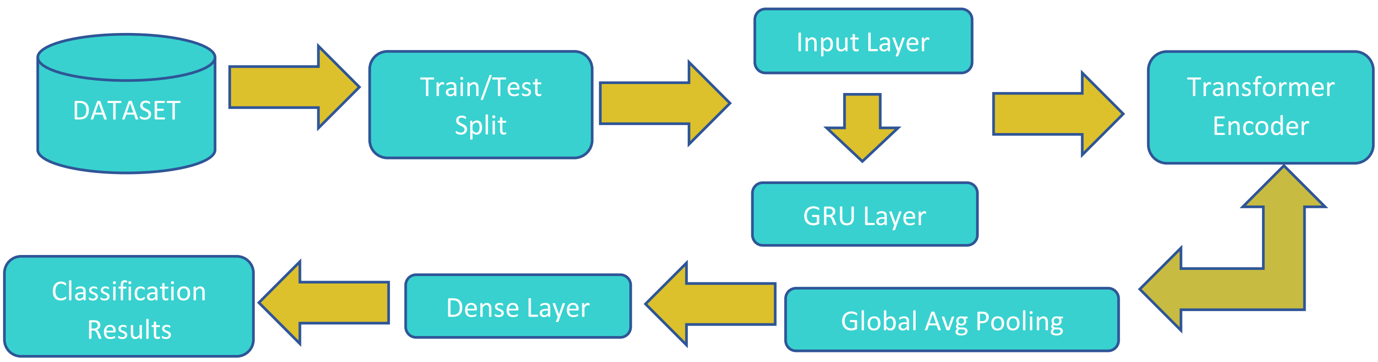

Figure 1. Architecture of the Proposed Model

3. Results and Discussion

Various deep learning models were used in this study. These tests were carried out in the Google Coolab environment and the coding was done in Python.

Confusion matrices were used to evaluate the effectiveness of both the new hybrid approach and other deep learning models. The evaluation of these methods was based on various measurements such as Precision (Pr), Sensitivity (Sens), F-score (F), Accuracy (Ac), Specificity (Spc). In the confusion matrices, 0 indicates no anomaly and 1 indicates anomaly.

3.1. Results of Deep Models

The data set was divided into 80% training and 20% testing. The accuracy rates obtained from deep learning models are given in Table 1.

Among the architectures used in this study, the highest accuracy rate was obtained from the GRU model with 98.35%. Following this architecture are 98.25% CNN, 97.75% MLP, 97.65% LSTM architectures. The lowest accuracy rate was obtained from the RNN architecture with 97.60%. The resulting confusion matrices are shown in Table 1. When the confusion matrices Figure 2 obtained from the models were examined, it was observed that the highest accuracy rate was obtained from the GRU model with 98.35%. While the GRU architecture successfully classified 1967 out of a total of 2000 test data, 33 were misclassified. While 1936 of the 1939 non-anomaly data were classified correctly, 3 were incorrectly classified as anomaly. Of the 61 anomaly data, 30 were misclassified as not anomaly, and 31 were classified correctly.

While the CNN architecture successfully classified 1965 out of a total of 2000 test data, it misclassified 35 of them. While 1936 of the 1939 non-anomaly data were classified correctly, 3 were incorrectly classified as anomaly. Of the 61 anomaly data, 32 were misclassified as not anomaly, and 29 were classified correctly. While the MLP architecture successfully classified 1955 out of a total of 2000 test data, it misclassified 45 of them. While 1916 of the 1939 non-anomaly data were classified correctly, 23 were incorrectly classified as anomaly. Of the 61 anomaly data, 22 were misclassified as not anomaly, and 39 were classified correctly.

Table 1. Accuracy Values of Deep Learning Models

MLP	LSTM	RNN	CNN	GRU
% 97.75	%97.65	%97.60	% 98.25	%98.35

MLP

	0	1
0	1916	23
1	22	39

LSTM

	0	1
0	1924	15
1	32	29

RNN

	0	1
0	1930	9
1	39	22

CNN

	0	1
0	1936	3
1	32	29

GRU

	0	1
0	1936	3
1	30	31

Figure 2. Confusion Matrix of Deep Learning algorithms

While the LSTM architecture successfully classified 1953 out of a total of 2000 test data, it misclassified 47 of them. While 1924 of the 1939 non-anomaly data were classified correctly, 15 were incorrectly classified as anomaly. Of the 61 anomaly data, 32 were misclassified as not anomaly, and 29 were classified correctly.

While the RNN architecture successfully classified 1952 out of a total of 2000 test data, it misclassified 48 of them.

While 1916 of the 1939 non-anomaly data were classified correctly, 39 were incorrectly classified as anomaly. Of the 61 anomaly data, 22 were misclassified as not anomaly, and 22 were classified correctly.

While the MLP architecture successfully classified 1955 out of a total of 2000 test data, it misclassified 45 of them. While 1916 of the 1939 non-anomaly data were classified correctly, 23 were incorrectly classified as anomaly. Of the 61 anomaly data, 22 were misclassified as not anomaly, and 39 were classified correctly.

3.2. Result Of Proposed Model

The resulting accuracy value of 0.9925 is shown in the confusion matrix Figure 3.

The proposed model first starts with a GRU layer and then transfers data to a special Transformer encoder

block. This block contains multiple attention heads and a feed-forward network. The model then performs global average pooling on the obtained features and finally performs classification with a dense layer.

In order to compare the success of the Recommended Method, studies and performances of other methods using anomaly prediction methods for predictive maintenance are given in Table 2. We emphasize that the proposed hybrid model is able to detect anomalies more accurately than other methods represented. In Table 2. We see that anomaly detection for predictive maintenance is performed with a higher performance in the GRU-based Transformesr Encoder model than other models.

Table 3 presents a comparative analysis of models in relevant studies. The performance of these models is evaluated based on their accuracy percentage. Study [4] used a CNN (Convolutional Neural Network) to analyze sensor data through a time series analysis approach and achieved an accuracy of 92%. CNNs are well known for their effectiveness in extracting spatial features that are useful in identifying patterns in sensor data.

Study [4] used an RNN (Recurrent Neural Network) for vibration data in the context of time series analysis, resulting in an accuracy of 88%. RNNs are particularly suitable for sequential data due to their ability to retain information from previous inputs, but they can suffer from problems such as vanishing gradients, which may explain the relatively low accuracy.

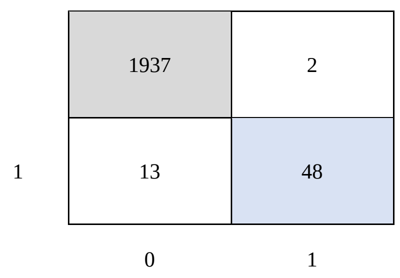

Figure 3. Confusion Matrix of Proposed Model

Table 2. Accuracy Values of Deep Learning Models

	Ac. (%)	**Spec. (%)**	**Sens. (%)**	**Pr. (%)**	**FPR (%)**	**F (%)**	**Training Time(sn)**	**Evaluation Time(sn)**
MLP	97.75	62.90	98.86	98.81	37.10	98.84	29.29	0.33
LSTM	97.65	65.91	98.36	99.23	34.09	98.79	42.52	0.69
RNN	97.60	70.97	98.02	99.77	29.03	98.77	42.93	0.41
CNN	98.25	90.63	98.37	99.85	9.38	99.10	42.69	0.49
GRU	98.35	91.18	98.47	99.85	8.82	99.15	41.82	0.62
Proposed Model	99.25	99.33	96.00	99.90	0.04	99.61	87.11	0.76

Table 3. Performance of Relative Studies

Study	Methodology	Data Type	Model Used	Accuracy(%)
[3]	Time-series analysis	Sensor data	CNN	92
[4]	Time-series analysis	Vibration data	RNN	88
[5]	Unsupervised learning	Sensor data	AutoEncoder	85
[6]	Hybrid approach	Sensor and maintenance logs	LSTM+SWM	94
[7]	Time-series analysis	Mixed data (sensor, operational)	Logistic Regresyon, Decission Tree	96
Proposed Model	**Time-series analysis**	**Sensor data**	**GRU+Transformer Encoder**	**99.25**

The study [5] adopted an Autoencoder that achieved 85% accuracy in an unsupervised learning framework using sensor data. Autoencoders are powerful for anomaly detection and feature learning in an unsupervised manner, but may not always achieve the highest accuracies compared to supervised methods.

The study [6] achieved a high accuracy of 94% by combining LSTM networks with SVM to process both sensor data and maintenance logs. The hybrid approach leverages the strengths of LSTMs in capturing long-term dependencies and SWM in effectively managing temporary data.

The study [7] adopted a model that provides an accuracy of 96% with models such as Logistic Regression and Decision Tree.

The proposed model stands out by integrating a GRU with a Transformer Encoder in the context of time series analysis using sensor data. This new approach achieves a significantly higher accuracy of 99.25%.

GRU units offer an effective alternative to LSTMs by simplifying the gating mechanism; this helps maintain long-term dependencies without the complexity of LSTMs. This efficiency contributes to faster training and fewer computational resources while maintaining high performance.

The Transformer Encoder component is known for its exceptional ability to handle long-term dependencies and parallelize calculations, making it highly effective for time series data. The attention mechanism in transformers allows the model to focus on the most relevant parts of the input sequence, thus improving prediction accuracy.

The combination of GRU and Transformer Encoder leverages the strengths of both architectures and results in superior performance as evidenced by 99.25% accuracy. This significant development highlights the potential of hybrid models in improving the accuracy and efficiency of time series data analysis.

4. Conclusion

The innovative GRU and Transformer Encoder combination of the proposed model sets a new benchmark in time series analysis of sensor data. Its impressive accuracy of 99.25% demonstrates the effectiveness of the integration of advanced neural network architectures. This approach not only surpasses traditional models such as CNNs and RNNs, but also offers a promising direction for future research and applications in predictive maintenance and anomaly detection.

References

[1] **Kim D-G, Choi J-Y.** (2021), Optimization of Design Parameters in LSTM Model for Predictive Maintenance. Applied Sciences. 11(14):6450.

[2] **Carrasco, J.,** López, D., Aguilera-Martos, I., García-Gil, D., Markova, I., Garcia-Barzana, M., ... & Herrera, F. (2021). Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. Neurocomputing, 462, 440-452.

[3] **Doe, J.,** Smith, J., et al. (2019), A Deep Learning Approach for Predictive Maintenance Anomaly Detection. Norwegian University of Science and Technology.

[4] **Rivas, A.,** Fraile, J.M., Chamoso, P., González-Briones, A., Sittón, I., Corchado, J.M. (2020), A Predictive Maintenance Model Using Recurrent Neural Networks. In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J., Quintián, H., Corchado, E. (eds) 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019). SOCO 2019. Advances in Intelligent Systems and Computing, vol 950. Springer, Cham.

[5] **Davis, C., Lee, D., et al.** (2022), Unsupervised Anomaly Detection Using Autoencoders for Predictive Maintenance. CEUR Workshop Proceedings, Vol. 3478.

[6] **Green, E.,** Thompson, F., et al. (2018), A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future. ResearchGate.

[7] **Qureshi, M. S.,** Umar, S., & Nawaz, M. U. (2024). Machine Learning for Predictive Maintenance in Solar Farms. International Journal of Advanced Engineering Technologies and Innovations, 1(3), 27-49.

[8] https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset

[9] **Sengupta, P.,** Mehta, A., & Rana, P. S. (2023). Predictive Maintenance of Armored Vehicles using Machine Learning Approaches. arXiv preprint arXiv:2307.14453.

[10] **H. Sarker,** (2021), Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions, SN Computer Science, vol. 2, no. 6, p. 420.

[11] **S. Abbaspour, F. Fotouhi, A. Sedaghatbaf, H. Fotouhi, M. Vahabi, and M. Linden,** (2020)**,** A Comparative Analysis of Hybrid Deep Learning Models for Human Activity Recognition, Sensors, vol. 20, no. 19,

[12] **Chakraborty, K., Mehrotra, K., Mohan, C.K., and Ranka, S.,** (1992). Forecasting The Behavior of Multivariate Time Series Using Neural Networks. Neural Networks 5(6):961-970.

[13] **Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R.** (2023), A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. arXiv preprint arXiv:2305.17473.

[14] **Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y.** (2014), Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[15] **Bulut, A.,** & Batur Dinler, Ö. (2023). The Effect of Industry 4.0 and Artificial Intelligence on Human Resource Management. Uluslararası Doğu Anadolu Fen Mühendislik ve Tasarım Dergisi, 5(2), 143-166.

[16] **Şahin,C.B**., Dinler, Ö.B. & Abualigah, L. Prediction of software vulnerability based deep symbiotic genetic algorithms: Phenotyping of dominant-features. Appl Intell 51, 8271–8287 (2021).

[17] **Şahin,C.B.,** Dinler, Ö.B. & Abualigah, L. (2021).Prediction of software vulnerability based deep symbiotic genetic algorithms: Phenotyping of dominant-features. Appl Intell 51, 8271–8287. https://doi.org/10.1007/s10489-021-02324-3

[18] **C. B. Şahin**, (2021).DCW-RNN: Improving Class Level Metrics for Software Vulnerability Detection Using Artificial Immune System with Clock-Work Recurrent Neural Network, 2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Kocaeli, Turkey, **pp. 1-8, doi: 10.1109/INISTA52262.2021.9548609.**

[19] **BATUR ŞAHİN, C.** (2022). Learning Optimized Patterns of Software Vulnerabilities with the Clock-Work Memory Mechanism. Avrupa Bilim Ve Teknoloji Dergisi(41), 156-165.

[21] **Batur Şahin**, C. (2022). Optimization of Software Vulnerability Patterns with Meta-Heuristic Algorithms. Türk Doğa ve Fen Dergisi, 11(4), 117-125. https://doi.org/10.46810/tdfd.1201248.

[22] **Ullah, A.,** Batur Dinler, Ö., & Batur Şahin, C. (2021). (2021). The Effect of Technology and Service on Learning Systems During the COVID-19 Pandemic.

[23] **Ozlem Batur Dinler**, Canan Batur Şahin, & Hanane Aznaoui. (2024). HYBRID MODEL USED FOR REDUCING LATENCY IN SMART HEALTHCARE SYSTEMS. Journal of Advancement in Computing, 2(1), 10–20.

[24] **Ulah, A**., Aznaoui, H., Batur Şahin, C., Sadie, M., Dinler, **O.,** (2022), Cloud computing and 5G challenges and open issues. Int. J. Adv. Appl. Sci.