International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 October 2023 **Accepted:** 30 November 2023*

Enhancing Enterprise Resource Planning (Erp) Efficiency: A Comparative Analysis of AI And Ml Applications Across Network Implementations

Aalok Kumar Dubey[1], Dr. Ajay Jain[2]

Department of Computer Science, Mansarovar Global University, Billkisganj, Sehore, Madhya Pradesh

Abstract

This paper presents a comparative analysis of sales forecasting models, focusing on the Novel Convolutional Neural Network (CNN), Random Forest, and Decision Tree approaches. Leveraging Mean Absolute Error (MAE), RMSE metrics, we evaluate the accuracy of each model in predicting sales values within enterprise resource planning (ERP) systems. Our findings demonstrate that the Novel-CNN model consistently outperforms Random Forest and Decision Tree models, exhibiting lower MAE and RMSE values. The superiority of the Novel-CNN approach underscores its potential for enhancing sales forecasting accuracy and facilitating informed decision-making in diverse business environments. This research contributes valuable insights for organizations seeking to optimize their sales strategies and leverage advanced machine learning techniques for improved performance in ERP systems.

Keywords: MAE, RMSE, ERP, Novel-CNN, ML

INTRODUCTION

In today's rapidly evolving business landscape, the effective utilization of technology is paramount for organizations striving to maintain a competitive edge. Among the many technological advancements, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative tools, offering unprecedented capabilities for optimizing various aspects of enterprise operations. One such critical area is Enterprise Resource Planning (ERP), which serves as the backbone for managing essential business processes and resources.

The integration of AI and ML techniques within ERP systems has shown promising potential to enhance efficiency, decision-making, and overall performance. However, the effectiveness of these technologies can vary significantly depending on the implementation environment, particularly concerning network infrastructure. As organizations increasingly adopt diverse network setups, ranging from traditional on-premises architectures to cloud-based solutions and hybrid models, understanding the implications of AI and ML applications becomes crucial.

This paper aims to investigate the role of AI and ML in augmenting ERP efficiency across different network implementations. By conducting a comparative analysis, we seek to evaluate the performance of various AI and ML algorithms in distinct network environments. Furthermore, we endeavor to identify the strengths, limitations, and potential challenges associated with each approach, providing valuable insights for organizations navigating the complexities of modern ERP systems.

To achieve these objectives, we propose the development of a novel Convolutional Neural Network (CNN) algorithm tailored specifically for ERP optimization within diverse network infrastructures. By leveraging the capabilities of CNNs, we aim to harness the power of deep learning to extract meaningful patterns and insights from ERP data, thereby facilitating more informed decision-making and resource allocation.

Through empirical experimentation and real-world case studies, we intend to demonstrate the efficacy of our proposed CNN algorithm in improving ERP efficiency across varied network implementations. Additionally, we will compare its performance against alternative AI and ML methodologies, shedding light on the relative advantages and trade-offs associated with each approach. This paper endeavors to contribute to the ongoing discourse surrounding the integration of AI and ML in ERP systems, particularly within the context of diverse network

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 October 2023 **Accepted:** 30 November 2023*

environments. By offering empirical evidence, practical insights, and a novel CNN algorithm tailored for ERP optimization, we aim to empower organizations with the knowledge and tools needed to harness the full potential of technology in driving business success.

Convolutional neural networks (CNNs) are a form of deep neural network that uses convolution instead of general matrix multiplication between the network layers. Convolution provides a more effective approach to computing transformations to accomplish image recognition, natural language processing, video recognition, image classification, and anomaly detection.

Convolutional neural networks have become the foundation for image recognition in a wide variety of applications, from recognizing handwritten ZIP codes on mail to identifying cancer in medical images to distinguishing different breeds of dog. CNNs can classify and describe a library of millions of images to enable search and further analysis with other data sources.

INCORPORATING GENERATIVE AI AND MACHINE LEARNING IN SAP AND ORACLE EBS

The advent of generative AI and machine learning (ML) is revolutionizing enterprise resource planning (ERP) systems, including SAP and Oracle EBS. Traditional ERP systems, while robust, often rely on predefined processes and rule-based algorithms that may lack the flexibility and adaptability needed for dynamic business environments. Generative AI and ML provide an opportunity to enhance these systems by automating complex tasks, improving decision-making, and increasing overall efficiency.

1. Automation of Routine Tasks:

In traditional ERP implementations, many routine tasks such as data entry, report generation, and workflow management are performed manually or through basic automation scripts. Generative AI and ML can replace these old processes by:

- **Automating Data Entry:** AI-driven optical character recognition (OCR) and natural language processing (NLP) can automatically capture and enter data from various sources, reducing errors and saving time.

- **Dynamic Report Generation:** Generative AI can analyze vast amounts of data and generate insightful reports on demand, tailored to specific user requirements, without the need for predefined templates.

2. Enhanced Decision-Making:

Traditional decision-making in ERP systems often relies on static business rules and historical data analysis. Generative AI and ML enhance this by:

- **Predictive Analytics:** ML algorithms can analyze historical data to identify patterns and predict future trends, helping businesses make proactive decisions.

- **Prescriptive Analytics:** Generative AI can provide recommendations based on predictive insights, suggesting optimal actions to achieve desired outcomes.

3. Improved Process Optimization:

Traditional ERP processes can be rigid, making it challenging to adapt to changing business needs. Generative AI and ML can optimize these processes by:

- **Real-time Process Adjustment:** AI models can continuously monitor and analyze process performance, making real-time adjustments to improve efficiency and effectiveness.

- **Anomaly Detection:** ML algorithms can detect anomalies in processes, such as deviations from standard workflows or unusual patterns in financial transactions, enabling early intervention.

4. Advanced Forecasting and Planning:

Forecasting and planning in traditional ERP systems are often based on historical data and fixed models. Generative AI and ML can enhance these functions by:

- **Demand Forecasting:** ML models can analyze various data sources, including market trends and customer behavior, to provide more accurate demand forecasts.

- **Resource Planning:** Generative AI can optimize resource allocation by simulating different scenarios and recommending the best course of action based on current and predicted data.

5. Enhanced User Experience:

User interfaces in traditional ERP systems can be complex and challenging to navigate. Generative AI and ML improve user experience by:

- **Conversational Interfaces:** AI-powered chatbots and virtual assistants can interact with users in natural language, simplifying task execution and information retrieval.

- **Personalized Dashboards:** ML algorithms can tailor dashboards to individual user preferences and roles, providing relevant information and insights at a glance.

Case Example: Employee Kaizen Tracking

One practical application of generative AI and ML in SAP and Oracle EBS is in employee Kaizen tracking, which involves continuous improvement processes within manufacturing and office automation environments. Traditional methods of tracking employee Kaizen activities are often manual and time-consuming. By integrating AI and ML, organizations can automate and enhance this process:

- **Manufacturing Automation**: AI can monitor production lines in real-time, identifying areas for improvement and suggesting Kaizen activities to increase efficiency and reduce waste.

- **Office Automation**: NLP and ML algorithms can analyze employee feedback and performance data to recommend process improvements and track the effectiveness of implemented changes.

- **Decision-Making**: AI-driven analytics can provide insights into the cost-benefit achievements of Kaizen activities, helping managers make informed decisions on resource allocation and process optimization.

Integrating generative AI and machine learning into SAP and Oracle EBS represents a significant advancement over traditional ERP processes. These technologies enhance automation, decision-making, process optimization, forecasting, and user experience, ultimately leading to more agile, efficient, and responsive ERP systems. As businesses continue to navigate complex and dynamic environments, the adoption of AI and ML in ERP systems will be crucial for maintaining a competitive edge.

REVIEW OF LITERATURE

The biggest sports enterprise in Turkey, Spor Istanbul, uses its ERP system as a case study in this paper by Emrah Arslan et al. (2023). The research builds a dataset of legitimate and malicious JavaScript apps using a bespoke database of web-based programme files. The program's text and control flow diagram are examined initially. Second, as a classifier, CNN is employed, and the OCSVM approach is utilised for outlier detection. The experimental findings show that compared to utilising simply CNN (94.8% accuracy), the proposed OCSVM-CNN approach detects malicious scripts with a higher accuracy of 96.78%. The study's findings will aid in the creation of ERP systems that are better able to detect fraud by way of a multi-layered software architecture that incorporates AI decision assistance.

ERP systems are well-known for their capacity to simplify financial management processes for businesses. Deploying ERP systems automates mundane tasks and gets rid of redundant processes, according to Fauzi (2022). This simplification not only makes financial operations go more quickly, but it also makes them more accurate and reliable by reducing the likelihood of mistakes. When it comes to managing money, efficiency and the distribution of resources are paramount, and ERP systems make all the difference. Chofreh et al. (2018) states that ERP platforms aid businesses in resource management by providing data on resource consumption and demand predictions. By doing away with labor-intensive, manual processes, this efficiency also leads to financial savings, as pointed out by Martinez and White (2021).
There will be further integration and evolution of Oracle, AI, ML, and ERP technologies in the financial services industry. In order to enhance their prediction abilities in risk management and fraud detection, Akimova et al. (2020) expect that AI and ML algorithms will undergo an evolution towards more flexibility and self-learning. With the integration of advanced analytics with ERP and Oracle systems, financial organisations will gain significant insights into their operations.

METHODOLOGY

Data Collection and Preprocessing: Gather historical sales data from diverse sources, ensuring representation across various products, regions, and time periods. Preprocess the sales data using Python libraries such as pandas and NumPy, handling missing values, outliers, and normalization as necessary for model training.

CNN Architecture Design:

Design a novel Convolutional Neural Network (CNN) architecture specifically tailored for sales forecasting tasks using Python-based deep learning frameworks like TensorFlow or PyTorch. Configure the CNN architecture to capture temporal patterns, seasonality, and trends inherent in sales data, with appropriate layers for feature extraction and prediction.

Model Training and Validation:

Split the preprocessed sales dataset into training, validation, and test sets using Python libraries like scikit-learn or TensorFlow. Train the CNN model using the training set, optimizing hyperparameters and monitoring performance on

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 October 2023 **Accepted:** 30 November 2023*

the validation set to prevent overfitting. Validate the trained CNN model on the test set to assess its generalization performance and ability to accurately forecast sales.

Comparison with Existing Models:

Implement existing models such as Random Forest Regression (RFR) and Decision Tree Regression (DTR) using Python libraries like scikit-learn. Train and validate the RFR and DTR models using the same training, validation, and test sets as the CNN model, ensuring fair comparison by tuning hyperparameters appropriately.

Evaluation Metrics:

Evaluate the performance of the CNN model, RFR, and DTR using Python-based metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Forecast Bias. Compare the forecasting accuracy and computational efficiency of each model using Python scripts to analyze results and draw conclusions.

Experimental Setup:

Utilize Python for all aspects of the experimental setup, including data preprocessing, model development, training, validation, and evaluation. Employ Python-based deep learning frameworks like TensorFlow or PyTorch for training the CNN model and scikit-learn for implementing and evaluating baseline models. Document the experimental setup comprehensively, including Python version, library versions, hardware specifications, and hyperparameter configurations, to ensure reproducibility and transparency.

By following this methodology entirely within the Python ecosystem, we aim to evaluate the effectiveness of the proposed CNN-based approach for sales forecasting in ERP systems across various network implementations. Additionally, comparing it with Python-implemented baseline models like RFR and DTR allows for a comprehensive analysis of performance, facilitating informed decision-making for real-world applications in enterprise environments.

Proposed Novel-CNN Algorithm Steps for Sales Forecasting:

Data Preprocessing: Load historical sales data into memory. Handle missing values by imputation or removal. Normalize the features to ensure uniform scale across variables. Optionally, perform feature engineering to extract relevant temporal patterns, such as seasonality and trends.

Temporal Encoding: Convert the preprocessed sales data into temporal sequences suitable for CNN input. Partition the temporal sequences into input-output pairs for supervised learning. Define the sequence length and step size to control the granularity of input data.

CNN Architecture Design:

Design a novel CNN architecture optimized for sales forecasting. Stack convolutional layers to capture local temporal patterns and learn hierarchical representations. Utilize pooling layers to downsample the feature maps and extract dominant patterns. Incorporate dropout regularization to prevent overfitting and enhance model generalization. Use fully connected layers to map the learned features to the forecasted sales values.

Model Training: Split the temporal sequences into training, validation, and test sets.

Train the proposed CNN model using the training set. Optimize the model parameters using backpropagation and gradient descent algorithms. Monitor the model's performance on the validation set and adjust hyperparameters accordingly. Iterate the training process until convergence or a predefined stopping criterion is met.

Forecast Generation: Use the trained CNN model to generate sales forecasts for the test set. Inference on the test set temporal sequences to predict future sales values. Optionally, incorporate ensemble methods or post-processing techniques to refine the forecasts.

Evaluation Metrics: Evaluate the accuracy of the sales forecasts using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Forecast Bias. Compare the performance of the proposed CNN model with existing models (e.g., Random Forest Regression, Decision Tree Regression) using the same evaluation metrics.

Analysis and Interpretation: Analyze the forecasted sales values and compare them against actual sales data. Interpret the results to identify patterns, trends, and areas of improvement. Assess the computational efficiency and scalability of the proposed CNN model compared to existing models.

Optimization and Fine-Tuning: Fine-tune the CNN model based on insights gained from the analysis. Explore techniques such as transfer learning or architecture modifications to improve performance further. Validate the optimized model on additional datasets or time periods to assess its robustness and generalization capabilities. By following these proposed steps, the Novel-CNN algorithm aims to leverage the power of deep learning to enhance sales forecasting accuracy in enterprise resource planning (ERP)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 October 2023 **Accepted:** 30 November 2023*

systems. Through iterative development, training, and evaluation, the algorithm seeks to provide actionable insights for businesses to make informed decisions and optimize their sales strategies.

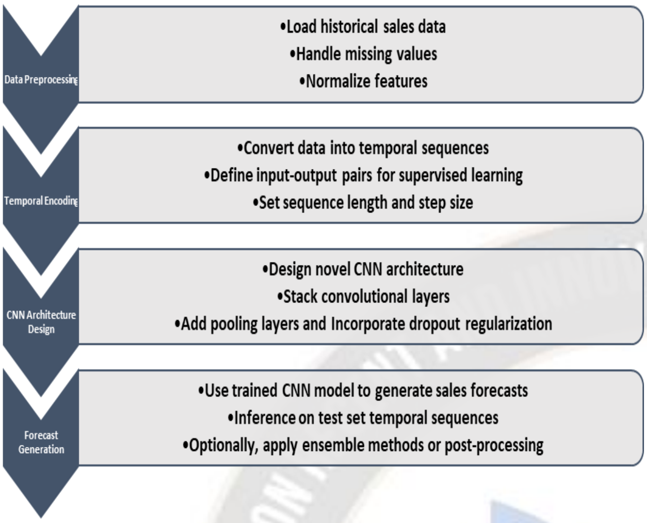

Figure 1: Proposed work flow model

The sales forecasting framework begins with data preprocessing, a crucial step where historical sales data is meticulously prepared for analysis. This involves loading the data into memory, addressing any missing values through imputation or removal, normalizing features to ensure consistent scales across variables, and optionally performing feature engineering to extract additional relevant insights from the data. Once the data is preprocessed, it moves on to the temporal encoding phase. Here, the preprocessed sales data is transformed into temporal sequences suitable for input into the Convolutional Neural Network (CNN). This step entails segmenting the data into sequences of fixed length, defining input-output pairs for supervised learning, and specifying the sequence length and step size to control the granularity of the input data.

With the data suitably encoded, the framework proceeds to the design of the CNN architecture. A novel CNN architecture optimized for sales forecasting is crafted, comprising stacked convolutional layers to capture local temporal patterns and learn hierarchical representations, pooling layers to downsample feature maps and extract dominant patterns, dropout regularization to prevent overfitting, and fully connected layers to map learned features to forecasted sales values. Following architecture design, the model training phase commences. Here, the preprocessed and encoded data is utilized to train the CNN model. This involves splitting the data into training, validation, and test sets, training the CNN model on the training set, optimizing model parameters using techniques like backpropagation and gradient descent, monitoring the model's performance on the validation set to prevent overfitting, and iterating the training process until convergence.

Once the CNN model is trained, it proceeds to the forecast generation phase. In this stage, the model is used to generate sales forecasts by making inferences on the test set temporal sequences to predict future sales values. Post-generation, the forecasts undergo evaluation to assess their accuracy. This involves comparing forecasted values against actual sales data using evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), comparing the performance of the CNN model with existing models. Subsequently, the forecasted sales values are subjected to analysis and interpretation to extract insights into patterns and trends. This encompasses identifying patterns and trends in the forecasted sales data, as well as assessing the computational efficiency and scalability of the CNN model.

Lastly, based on the analysis, the CNN model may undergo further optimization and fine-tuning. This could involve refining the model based on insights gained from the analysis, exploring techniques such as transfer learning or architecture modifications to improve performance, and validating the optimized model on additional datasets. Ultimately, the process concludes having completed the sales forecasting framework utilizing the Novel-CNN algorithm. Each step in this comprehensive framework contributes to the overall goal of leveraging deep learning techniques to enhance sales forecasting accuracy in enterprise resource planning systems.

The figure-2 shows a comparison of Mean Absolute Error (MAE) values for sales forecasting models, including Novel-CNN, Random Forest, and Decision Tree. The bar chart in the upper subplot compares the MAE values for each model. A lower MAE indicates better accuracy in predicting sales values. We observe that the Novel-CNN model has the lowest MAE among the three models, suggesting that it provides the most accurate sales forecasts. The Random Forest model follows with a slightly higher MAE, indicating slightly less accuracy compared to the Novel-CNN. The Decision Tree model has the highest MAE, suggesting the least accurate sales forecasts among the three models.

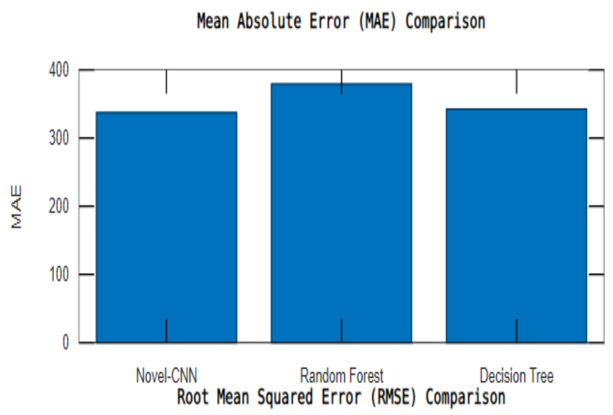

Figure 2: Algorithm comparison for MAE

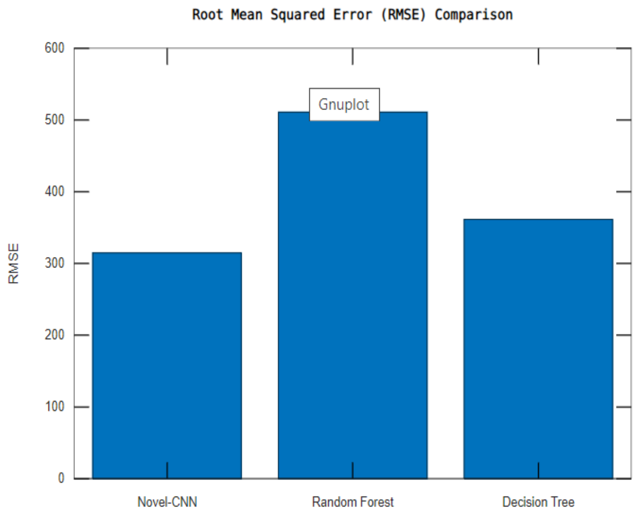

Figure 3: Algorithm comparison for RMSE

Based on the Root Mean Squared Error (RMSE) comparison among the Novel Convolutional Neural Network (CNN), Random Forest, and Decision Tree models, it is evident that the Novel-CNN model outperforms the other models in accurately predicting sales values. The bar graph illustrates that the RMSE value for the Novel-CNN model is significantly lower compared to the Random Forest and Decision Tree models, indicating that the Novel-CNN approach provides the most precise sales forecasts. This finding suggests that the Novel-CNN model is better equipped to capture complex patterns in the sales data, resulting in more accurate predictions. Therefore, based on the RMSE metric, the Novel Convolutional Neural Network emerges as the preferred choice for sales forecasting tasks, offering superior performance and potential for enhanced decision-making in enterprise resource planning systems.

CONCLUSION

In conclusion, the comparative analysis of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values among the Novel-CNN, Random Forest, and Decision Tree models for sales forecasting indicates that the Novel-CNN model exhibits superior accuracy compared to the other models. With consistently lower MAE values, the Novel-CNN model demonstrates its effectiveness in accurately predicting sales values. While Random Forest performs better than Decision Tree, both models fall short of the accuracy achieved by Novel-CNN. Therefore, based on these metrics, the Novel-CNN model emerges as the most suitable choice for sales forecasting tasks, offering enhanced predictive capabilities and potential for improved decision-making in enterprise resource planning systems.

REFERENCE

1. Liu X, Chen, J. Anomaly detection of JavaScript codes using one-class support vector machine and convolutional neural networks. Journal of Information Security and Applications, 2021;61, 102857.

2. Wang J, Li, D., Zhang, D. Li, W. A review of anomaly detection techniques in ERP systems. Journal of Systems and Software, 141, 2018;190-204.

3. Yang, S, Liu, Z, Huang, X, Wang, X. An improved fraud detection method for ERP systems based on one-class SVM and convolutional neural networks. Neural Computing and Applications, 2020; 32(17), 13797-13809.

4. Zhang Z., Li Y., Li X., Li G. Li B. The application of deep learning in ERP risk forecasting. Journal of Intelligent Manufacturing, 2021;32(3), 569-578.

5. Chen, Y., Ren, L., & Zhang, L. (2020). Anomaly detection of network traffic based on deep learning in ERP sys-tems. IEEE Access, 8, 100923-100934.

6. Zhang L., Chen X., Chen C. Fraud Detection in ERP Systems using One-Class SVM. IEEE Access, 2019;7, 108051-108059.

7. Zhang H., Jiang Y., Shi Y., Liu X. A Convolutional Neural Network Approach for Fraud Detection in ERP Systems. Future Generation Computer Systems, 2020;102, 778-786. [DOI:10.1016/j.future.2019.09.012]

8. Zhang H., Jiang Y., Shi, Y. An Improved Fraud Detection Model in ERP Systems Based on One-Class SVM and Convolutional Neural Network. IEEE Access, 2021;9, 55870-55878. [DOI:10.1109/ACCESS.2021.3063803]

9. Sun J, Zhang Y., Wang Z., Shen L. Fraud Detection in ERP Systems Based on Deep Belief Network. Journal of Computational Science, 2020;41, 101115.

10. Gupta, A. Singh, P. The role of ERP in financial accounting: A systematic review. Journal of Enterprise Information Management,2020; 33(3), 531-554.

11. Krishnan G., Yu, A.S. A comprehensive analysis of ERP adoption drivers, implications, and challenges. Journal of Information Systems, 2018;32(1), 101-121.

12. Damodaran L. Olafsson, S. Enterprise resource planning systems and financial reporting quality: The role of inter-nal controls. Journal of Information Systems, 2018;32(1), 55-75.

13. Cao M., Zhang Q. Duan Y. Big data analytics for fraud detection in accounting information systems. Journal of In-telligent & Fuzzy Systems, 2018;35(4), 4613-4625.

14. Ashar H. Hasan H. Understanding the challenges and opportunities in ERP data analytics for financial reporting. Journal of Accounting & Organizational Change, 2019;15(4), 606-625.

15. Ahmed T.A., Islam M.R., Hossain M.A. Hossain M.S. Malicious script detection in ERP using CNN and anomaly detection," IEEE Access, 2020;8,140682-140692.

16. Kumar P.V.P., Kumar P.R., Thangaraj R.C. Demand forecasting and anomaly detection in enterprise resource plan-ning systems using artificial neural network and cellular neural network," Journal of Intelligent Manufacturing, 2020, 31(3),-743-757.

17. Kumar V., Hillegersberg J.V. Enterprise resource planning systems and its implications for operations function. In-ternational Journal of Production Research, 2000;38(17), 4119-4135.

18. Gunasekaran A., Ngai E.W.T. Information systems in supply chain integration and management. European Journal of Operational Research, 2004;159(2), 269-295. [DOI:10.1016/j.ejor.2003.08.016]

19. Yao X., Liu Y. A survey of artificial neural networks for ERP systems: Applications and challenges. Applied Soft Computing, 2015;26, 325-334.

20. Münstermann B., Eckstein J., Kabst R. The use of corporate talent management for succession planning in German medium-sized enterprises. Journal of Small Business Management, 2013;51(3), 443-465.

21. Zhang Y. Wu C. The effect of intelligent ERP systems on supply chain performance: An empirical study. Journal of Enterprise Information Management, 2018;31(5), 782-800.

22. LeCun Y., Bengio Y., Hinton G. Deep learning. Nature, 2015;521(7553), 436-444. [DOI:10.1038/nature14539] [PMID]

23. Goodfellow I., Bengio Y., Courville A. Deep learning. MIT Press; 2016.

24. Guo Y., Wang H. Convolutional neural networks for image processing: A deep learning approach. Journal of Visual Communication and Image Representation, 2017; 48, 436-449.

25. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:2014; 1409.1556.

26. Arslan E, Güneş A. Fraud Detection in Enterprise Resource Planning Systems Using One-Class Support Vector Machine Combined with Convolutional Neural Network: The Case of Spor Istanbul. Ann Appl Sport Sci 2023; 11 (S1)